The Visual Complexity of Instructional Animations in Training Simulations to Promote Learning

Iwan Kelaiah
(formerly known as Iwan Kartiko)
B. Eng. Mechatronics with Hons1

11 January 2012

Department of Computing
Faculty of Science
Macquarie University
North Ryde, NSW 2109
Australia

Thesis Supervisors:
A/Prof. Manolya Kavakli, Department of Computing
Prof. Ken Cheng, Department of Biological Sciences

This thesis is presented for the degree of Doctor of Philosophy.

Keywords: virtual reality, learning, simulation, animated-virtual actors, virtual scene, virtual environment, visual complexity, learning, motion sickness, simulator sickness, multimedia, presence, affective quality, emotion, training, education, human-computer interaction.

Contents

Contents iii

List of Figures vii

List of Tables ix

List of Abbreviations xi

Abstract xi

Statement and Contribution of the Candidate xiii

Dedication xvii

Acknowledgment xix

I Introduction 1

1 Introduction 3
- 1.1 Research Problems 5
- 1.2 Research Questions 7
- 1.3 Simulations Developed for Testing 9
- 1.4 Questionnaires Used in Testing 11
- 1.5 Structure of the thesis 14
  - 1.5.1 Overview of the Chapters 16

II Submitted and published articles 25

2 Avatar creation for believable agents 29

3 Effects of digital content on motion sickness in immersive virtual environments 37

4 Effects of depth cues on simulator sickness 51

5 RiskMan: A multi-agent system for risk management 57

6 Virtual reality police training: How much visual information is too much for knowledge transfer? 79

7 The Impacts of Animated - Virtual Actors' Visual Complexity and Simulator Sickness in Virtual Reality Applications 93

8 Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity 101

9 Correlations between Simulator Sickness and Visual Complexity of Animated-Virtual Actors and Virtual Environments 113

10 The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. 141

III Closing Remarks 157

11 Conclusion 159
11.1 Significance of the Outcomes 159
11.2 Contribution of the thesis 162
11.3 Future Directions of research 171

IV Appendix 175

A Does the visual complexity of animated-virtual actors in virtual reality applications affect learning? 177

B Testing Platforms and Instructional Animations 185
B.1 Introduction 186
B.2 ANTECATALYST 187
B.2.1 Virtual Environment 190
B.2.2 AVA 1: ANTique 193
B.2.3 AVA 2: ANThony 193
B.2.4 AVA 3: ANTlike 195
B.3 VEAR 198
B.3.1 Simple AVAs 204
B.3.2 Complex AVAs 206
B.3.3 Simple VE 208
B.3.4 Complex VE 208
B.4 Conclusion 209

C Storyboards 213
C.1 Storyboard of ANTECATALYST 213
C.2 Storyboard of VEAR 228

D Screenplays **237**
D.1 Screenplay of ANTECATALYST 237
D.2 Screenplay of VEAR 245

E Questionnaires **253**
E.1 Questionnaires of ANTECATALYST 253
E.2 Questionnaires of VEAR 284

F Python codes of ANTECATALYST **317**

G R codes **383**
G.1 R codes of ANTECATALYST 383
G.1.1 Codes for analysing Simulator Sickness 383
G.1.2 Codes for analysing Learning Outcomes 394
G.2 R codes of VEAR 410
G.2.1 Codes for analysing Simulator Sickness 410
G.2.2 Codes for analysing Learning Outcomes 421

H Final Ethics Approval **439**

References **441**

List of Figures

1.1 The Roller-coaster simulation has two experimental conditions: a simple black and white scene (top), and complex visuals (bottom). 10
1.2 Two VR simulations of differing depth cues are used to assess the effects of depth cues and SS. The Snow Simulation (top) contains less depth cues due to the smooth-looking surface of the VE. The Trenches Simulation (bottom) shows more depth cues due to the VE that contains many protrusion of hard edges. 11
1.3 The three experimental conditions of ANTECATALYST. From top to bottom: (1) flat, (2) cartoon, or (3) lifelike. 12
1.4 The four experimental conditions of VEAR of alternating visual complexity: (A) simple, (B)simple AVAs, (C) simple World and (D) lifelike. . . . 13
1.5 The structure of the chapters in the thesis. 15

B.1 The desert with a clear blue sky, test channels and a skylight pattern as shown in Blender with many wire-frame boxes, which served as camera positions and camera target points. 191
B.2 The view of the virtual world from the test channel. 191
B.3 The desert as rendered in VR lab by using Vizard. 192
B.4 The sketch of ANTique: the flat AVAs in ANTECATALYST on stumps and stilts. 194
B.5 ANTique: The flat AVAs as shown in Blender. 194
B.6 ANTique: The flat AVAs as rendered in VR lab by using Vizard. 195
B.7 The sketch of ANThony: cartoon AVA, on all experimental condition. . . 196
B.8 ANThony: The cartoon AVA as shown in Blender with a pair of Japanese sandals as the stilts. It shows the rotoscoping technique by putting a reference image at the back of the 3D model to trace or duplicate animations from an animation handbook by Williams (2002, pp.192, 193). 197
B.9 ANThony: The cartoon AVA as shown in the VR lab by using Vizard. . . 197
B.10 ANTlike: The lifelike AVA as shown in Blender with the stilts. 198
B.11 ANTlike: The lifelike AVA as shown in the VR lab by using Vizard. . . . 199

B.12 A screenshot of the camera movement in Blender. The window on the left shows the graphs of key-framed camera movements around frame number 2500. The middle window shows what the user sees in VEAR. The middle bottom window shows the logic bricks of the camera. The top logic brick plays the narration and the movement of the camera from frame 0 to frame 10650, when the space bar button is pressed. The bottom logic brick sends a stop signal to Blender game engine when Ctrl, Shift and R buttons are pressed. The rightmost window shows the environment of VEAR without textures. 203
B.13 A screenshot of composing the scene of VEAR prior to applying texture on the 3D objects. 204
B.14 The final sketch of simple AVAs in VEAR: (A) a bystander, (B) a firefighter, (C) a driver in a car, and (D) a victim. 205
B.15 Development screenshot of VEAR in Blender that shows simple AVAs and VE. The AVAs appear flat in the 3D world. 205
B.16 The creation of the AVA that represents the victim of the car accident in VEAR in Evolver. The figure shows the step in choosing skin textures from the Evolver library. 206
B.17 The AVA that represents the victim of the car accident was seamlessly integrated into Blender as Evolver generated the armature system for Blender. In the background, the firefighter with armature system was implemented and the body was also deformable in Blender as the AVAs were also imported from Evolver. 207
B.18 Development screenshot of VEAR in Blender that shows visually complex AVAs and a complex VE. Visually complex AVAs give the 3D shape and other extraneous visual details that are not found in the simple AVAs. 207
B.19 Development screenshot of VEAR in Blender that shows visually complex AVAs and a simple VE. The simple VE is made of simple black-and-white drawings on flat polygons. Therefore, the simple VE appear flat in the 3D world. 208
B.20 Development screenshot of VEAR in Blender that shows simple AVAs and a visually complex VE. The visually complex VE is visually rich as it contains greater visual details than the simple VE. 209

List of Tables

1 Contribution to publications in Chapter 3 to 7. xiv
2 Contribution to publications in Chapter 8 to 12. xv

11.1 The summary of studies conducted, experimental conditions and the correlations 161

B.1 Visual complexity between each experimental condition in ANTECATALYST. 189
B.2 Filename tags and camera settings for a skybox in Vizard. 193
B.3 Visual complexity of each experimental condition in VEAR. 202

Abstract

There is a trend in systems design in creating complex visuals for Virtual Reality (VR) simulations. Given "generous" polygon and hardware allowance in developing digital contents, we may reach the hardware limitations easily to create aesthetically pleasing visuals. With affordable and free 3D engines, and modelling and animation tools, more educational institutions are expected to use the technology for teaching and learning. One concern is that the latest VR technology often surpasses the optimum requirements for learning, especially in visual complexity. There are two problems of learning with complex visuals: Simulator Sickness (SS), due to greater optic flow from greater visual details and cognitive load due to extraneous visual information. At the same time, complex visuals may add motivational factors such as presence and perceived affective quality, to improve learning. In assessing the effects of visual complexity in VR-based instructional animation, this thesis draws findings from studies in SS, Cognitive Theory of Multimedia Learning (CTML), presence and perceived affective quality within the flow theory.

The thesis contains three main parts consisting of smaller chapters. The first part contains two chapters including the introduction and the making of the VR simulations. The second part contains the chapters of published journals and papers, a book chapter as well as manuscripts submitted to journals. The last part of this thesis contains the conclusion that maps all the research questions to the empirical results, and gives an overview for the future directions of research.

This thesis started with the completion of two research projects at the beginning of the candidature: SS in VR simulations and the development of a risk assessment simulator. The author merged the findings of these two projects while exploring visual complexity, which became the focus of this thesis. Since the existing simulations could not be used to assess SS and learning outcomes in a unified approach, the candidate developed two non-interactive VR simulations: (1) ANTECATALYST (ANimaTEd CATAgLYphiS anTs) and (2) VEAR (VEhicle Accident and fiRes). ANTECATALYST examined the

effects of Animated-Virtual Actors' (AVAs) visual complexity and learning, and showed *Cataglyphis* ants' navigational behaviour on a flat and featureless desert with three learning conditions: (1) flat, (2) cartoon, or (3) lifelike AVAs. VEAR confirmed and extended the findings by taking into accounts the visual complexity of the AVAs and the Virtual Environments (VEs). VEAR is a firefighter training simulation and shows the hazards and the best strategy to manage car accidents safely. VEAR contains four visual conditions for learning: (1) Simple – visually simple AVAs and VE, (2) Simple AVAs – simple AVAs and a visually complex VE, (3) Simple world – visually complex AVAs and a simple VE, and (4) Lifelike – visually complex AVAs and VE. In assessing SS and learning outcomes, the thesis employed standardised and custom-made questionnaires: Simulator Sickness Questionnaire (SSQ), Presence Questionnaire (PQ), Perceived Affective Quality (PAQ) ratings, Program Ratings (PR), and custom-made retention and transfer tests.

The results from two principal experiments encourage the application of Coherence principle, derived from CTML, to promote retention and transfer by removing extraneous visual information, while keeping the production cost and each project's time-frame down. Due to the slow-moving camera in both VR simulations, none of them generate sufficient level of optic flow to induce high level of SS. The results also show that motivational factors in VR simulations, presence and perceived affective quality do not correlate with retention and transfer. In closing, the thesis leads to future research in cinematography and quantifying visual complexity in VR-based instructional animations.

Statement and Contribution of the Candidate

The work presented here has not been submitted for a higher degree to any other university or institution. The author has acknowledged the sources of information in this thesis by citing and putting the references at the end of each chapter.

The format of this thesis is thesis-by-publication and this section clarifies the contribution of the author in each chapter as shown in Table 1 and 2. The author would like to thank these co-authors, supervisors and colleagues, whose contribution to the thesis has been invaluable.

Manolya Kavakli is an Associate Professor in the Department of Computing at Macquarie University (manolya.kavakli@mq.edu.au). She obtained her PhD from Istanbul Technical University, in Knowledge Based System Design. In 2003, she established Virtual Reality research group at Macquarie University. Her area of research covers visual perception and cognition, human-computer interaction, user interface design, computer games and artificial intelligence.

Ken Cheng is a Professor in the Department of Biological Sciences at Macquarie University (ken.cheng@mq.edu.au). He obtained his PhD in Psychology from the University of Pennsylvania. His research interests include comparative cognition, especially spatial cognition, ranging from humans' to desert ants'.

Nicolas Szilas completed his PhD in 1995 on cognitive science (Grenoble, France) and two post-doctoral positions in Montreal (nicolas.szilas@unige.ch). While appointed as the chief scientist at the dept. of innovation of an European company between 1999-2001, he worked on his own project called IDtension, a narartive engine. He continued to worked on IDtension in University of Paris 8 and at Macquarie University. He entered the University of Geneva in 2006 as an associate professor. His research interest include games, narrative and learning.

John Porte completed his diploma in system analysis and programming in 1999 (john.porte @mq.edu.au). He is an IT officer in the Faculty of Science and a senior research assistant in the department of Computing at Macquarie University. Currently he is the manager of the Virtual Reality Lab and actively taking part in Virtual Reality related studies. His interest is in the development of interactive computer-generated graphics and Artificial Intelligence. He has been collaborating with various digital artists and programmers in developing 3D graphics engines.

Nolwenn Bigoin was an exchange student from Universite De La Mediterranee and was involved in various studies on Simulator Sickness and interaction at Macquarie University between 2006 - 2007 (Nolwenn.bigoin@etumel.univm ed.fr). Her expertise lies in the quick prototyping of interactive simulations.

Table 1: Contribution to publications in Chapter 3 to 7.

Author	Research topic	Lit. review	Data analysis	Simulations and experiments	Reviewing of drafts	**Total**
Chapter 3						
Kavakli M.	50	50	N/A	20	50	43
Kartiko I.	50	50	N/A	70	50	**57**
Chapter 4						
Kavakli M.	35	35	15	–	30	29
Kartiko I.	35	35	35	40	20	**33**
Porte J.	15	15	15	35	15	19
Bigoin N.	15	15	35	25	30	24
Chapter 5						
Bigoin N.	35	25	35	–	25	24
Porte J.	15	25	15	35	25	23
Kartiko I.	35	25	35	65	25	**37**
Kavakli M.	15	25	15	–	25	16
Chapter 6						
Kavakli M.	25	25	N/A	25	25	25
Szilas N.	25	25	N/A	25	25	25
Porte J.	25	25	N/A	25	25	25
Kartiko I.	25	25	N/A	25	25	**25**
Chapter 7						
Kartiko I.	80	60	N/A	N/A	50	**64**
Kavakli M.	20	40	N/A	N/A	50	36

Table 2: Contribution to publications in Chapter 8 to 12.

Author	Research topic	Lit. review	Data analysis	Simulations and experiments	Reviewing of drafts	**Total**
Chapter 8						
Kartiko I.	70	80	80	100	35	**73**
Kavakli M.	15	10	10	–	30	13
Cheng K.	15	10	10	–	35	14
Chapter 9						
Kartiko I.	80	80	70	100	35	**73**
Kavakli M.	10	10	15	–	30	13
Cheng K.	10	10	15	–	35	14
Chapter 10						
Kartiko I.	80	80	70	100	35	**73**
Kavakli M.	10	10	15	–	30	13
Cheng K.	10	10	15	–	35	14
Chapter 11						
Kartiko I.	80	80	70	100	35	**73**
Kavakli M.	10	10	15	–	30	13
Cheng K.	10	10	15	–	35	14
Chapter 12						
Kartiko I.	80	80	60	100	35	**71**
Kavakli M.	10	10	10	–	30	12
Cheng K.	10	10	30	–	35	17

Dedication

First and foremost, unto Thee O Lord, Saviour and Redeemer the Lord Jesus Christ, whom has given me the privilege, grace and mercies in completing this study. I thank Thee O Lord for carrying me through all these years. I thank Thee for the daily bread, and each every single blessing, which Thou hast given. I give Thee thanks for those whom Thou has led in the way throughout the years, return the blessings upon them. Bless Thy Word also, which has gone out while I sojourn here unto the glory of Thy name. Unless the LORD had been my help, my soul had almost dwelt in silence. (Psalms 94:17). Whom have I in heaven but Thee? and there is none upon earth that I desire beside thee. My flesh and my heart faileth: but God is the strength of my heart, and my portion for ever. (Psalm 73:25, 26).

My Lord, Saviour and Redemeer, Christ Jesus.
Such grace, wonderful grace to come and seek a lost sinner such as I,
such love, boundless love to bear my sin on Calvary,
such mercies, tender mercies to redeem this wretched soul,
such victory, matchless victory over death and grave,
such freedom, glorious freedom from the chain of sin,
such life, everlasting life with my risen Saviour,
such joy, unspeakable joy full of glory,
such Shepherd, I shall not want.

Acknowledgement

My gratitude to my dear wife, Ariel Kelaiah, for her love, patience, and prayers.

My heartfelt appreciation for the prayers of the Saints (my brothers and sisters in Christ) in the Church of the Living God in Sydney, Melbourne, Rockhampton, and other places throughout the islands and continents.

I'm thankful to my parents and my sister for their love and help throughout PhD candidature.

I am also grateful for both of my supervisors who helped me throughout the years of PhD candidature. What a blessing!

Special thanks to John Porte, Matt Cabanag and Darius Taslim for their support in troubleshooting and running the VR simulations. We are also grateful to Geoff Plummer, Adam Hughes, Eric Fassbender and Meredith Taylor for making the audio recording, and to Noah Beamer and the Blender community for maintaining Blender3D and Cal3D exporter. I am also grateful to Gordon Davies and Susan Bruck for their assistance in proofreading some portions of this thesis.

The Australian Research Council Discovery Grant titled Risk Management Using Agent Based Systems (DP0558852 to Richards, Kavakli, and Dras) has sponsored this project.

Part I

Introduction

Chapter 1

Introduction

Virtual Reality (VR) is a computer-generated artificial world, which appears and feels like a real world (Gaddis, 1998; Kavakli & Lloyd, 2005). Is it possible for a VR environment to appear and feel like a real world? Kavakli & Lloyd (2005) states that it must have three principal components: immersion, visualisation and interaction, just as we experience in the real world. There are two types of interaction in VR systems, direct and indirect interactions. Ecological communities are shaped by a complex array of direct and indirect interactions. These interactions are spatially and temporally dynamic and can be challenging to disentangle (Moon, Moon, & Keagy, 2010). Interaction in nature is often indirect, through agent actions that cause persistent and observable changes that later affect other agents (Milner, 1993). Direct user interaction with the system inside VR is not an absolute "must-have" in order to have a VR system. User in a vehicle or a rollercoaster simulation may not interact directly with the vehicle. However, the vehicle in the simulation may interact with other objects (e.g., road, rocks, railings, etc) within the VR by exhibiting avoidance behaviour to create an illusion of a driving experience for the user (e.g., Lin, Duh, Parker, Abi-Rached, & Furness, 2002; Bigoin, Porte, Kartiko, & Kavakli, 2007; Kavakli, Kartiko, Porte, & Bigoin, 2007). We see interaction is a way of framing the relationship between people and objects designed for them — and thus a way of framing the activity of design. Similar to our approach, E4MAS community dis-

tinguishes interaction via message transport from interaction via the environment (Keil & Goldin, 2006). Indirect interaction in this thesis is precisely the nontrivial use of the environment in the way suggested by E4MAS. The E4MAS community accepts environments of multi-agent systems as a first-class entity, distinguishing indirect interaction via the environment from the environment's role in message transport. Keil & Goldin (2006) define classes of interaction (sequential and multi-agent, direct and indirect) and environments (physical and virtual, persistent and amnesic, dynamic and static).

VR applications offer significant advantages for learning a new skill and enhancing existing knowledge as they offer three main benefits for learning. First, they are physically safe as we can simulate high-risk or difficult-to-create real-life scenarios. Second, they are financially safe, as they cost much less than re-enacting real-life scenarios. Third, they are ethically safe, as VR simulations do not involve real people or animals. Institutions have used VR simulations for various purposes, such as risk assessment and military training (e.g., Wiederhold, 2005; Kavakli, 2006), history education (e.g., Bruno et al., 2010), science education (e.g., Dede & Ketelhut, 2003), driving education (e.g., C. J. Chen, 2006), psychological rehabilitation (e.g., Powers & Emmelkamp, 2008), and surgical training (e.g., G. Chen et al., 2010).

VR simulations are becoming more affordable and easy to develop, due to advanced 3D modelling tools and 3D engines, of which many are open-source and readily available. There is a trend of using complex visual materials for learning as the computer graphics technology advances. Complex visuals are attractive and, therefore, meet the demands of the entertainment industry (Zyda, 2005; Roberts, 2007). However, educators must examine if a complex visual has any adverse effects on learning. Being entertaining is not always enough to form a valid pedagogical model for VR simulations. Retention and the highly regarded transfer or problem-solving tests are the measures to test if a learning system *"works"* (Sweller, 1988; Mayer, 2001) [as detailed in Chapter 6, 8 and 10].

1.1 Research Problems

According to Young & Munro (1998, pg. 4), "*Visualisation should present as much information as possible without overwhelming the user. Again, there must be a trade-off between high information content and a low visual complexity. Both are desirable yet contradictory.*" There are two obvious problems associated with learning with complex visuals in VR simulations. First, a VR simulation may induce Simulator Sickness (Stanney & Zyda, 2002, SS) and may hinder learning. Second, humans' working memory is limited. Unnecessary visual information may cause extraneous cognitive load that may impair learning (as demonstrated by Sweller (1994), Baddeley (1998), and Mayer (2001) [see details in Chapter 6, 8 and 10]. One may implement various principles to facilitate learning with complex visuals (e.g., Mayer, 2001; Moreno, 2006; Ayres & Paas, 2007; Moreno & Mayer, 2007), but it benefits no-one if a VR simulation induces SS. Thus, the impact of visual complexity on SS must be examined to develop safe VR simulations that "work" for educational purposes, as well as training.

On the negative side, VR simulations may induce SS, but on the positive side, they may induce the sense of presence and affective quality. Presence is the subjective experience of being in the virtual world, while physically not in it (Witmer & Singer, 1998). It is reported that presence has the potential of enhancing learning (T. A. Mikropoulos & Strouboulis, 2004; T. Mikropoulos, 2006). Affective quality is the ability of an environment to change one's mood or feeling that influences perception, cognition and behaviour (Russell, 2003). Thus, presence in conjunction with affective quality may improve learning (Russell, 2003; T. Mikropoulos, 2006; Takatalo, Nyman, & Laaksonen, 2008) [see details in Chapter 8 and 10]. How should VR designers develop digital contents to promote learning? This thesis examines the impact of the digital contents' visual complexity on learning in VR simulations. The author categorises digital contents of a VR simulation in two groups: Animated-Virtual Actors (AVAs) and Virtual Environment (VE). These are investigated in Chapter 7, 8, 9, 10, and Appendix A.

In this thesis, "visual complexity" refers to the amount of visual information in depicting a scene (Donderi, 2006), as detailed in chapter 8. Although the notion of realism often conveys visual complexity, a scene can be realistic but simple, e.g., a drop of oil on blue-painted concrete. Additionally, a scene can also be non-realistic yet visually complex, e.g., an acrylic painting of a pack of wolves in a rain forest. This thesis deals with the amount of visual information in VR simulations, not the degree of realism compared with photographic images.

In this thesis, a greater share of research is on the application of AVAs than VEs, because the author believes that the use of AVAs is a more significant problem. The application of VEs requires less justification and is more straightforward than the application of AVAs in VR simulations. For example, we demonstrated that the training module of a risk assessment system evaluating incoming aircraft passengers requires a VE that resembles an airport (Kavakli, Kartiko, et al., 2007; Richards, Dras, Porte, & Taylor, 2009; Dras, Richards, Taylor, & Gardiner, 2010) [see details in Chapter 5]. Without the depiction of an airport, the VE will not convey the scenario properly and this may hinder learning (Tversky, Morrison, & Betrancourt, 2002; Veletsianos, 2007). Unlike the application of VEs, using AVAs for learning requires a further justification. As described in Chapter 2 and 10, the deployment of AVAs is a complex process. AVAs can be used to enhance the depiction of a scenario with relevant VEs and some VR simulations require AVAs for learning to occur. For example, Johnsen, Raij, Stevens, Lind, & Lok (2007) used a virtual patient to teach medical students to gain interpersonal skills by observing the behavioural cues of the virtual patient. On the other hand, one may replace AVAs with text or on-screen narration and still they may not affect the learning material (e.g., Moreno, 2006; Wagner, Billinghurst, & Schmalstieg, 2006).

1.2 Research Questions

The aim of this thesis is to conceptualise preliminary guidelines in a framework for the design and deployment of AVAs and VEs in VR-based instructional animations that promote learning. Using the thesis by publication format, the author investigates the following fundamental questions:

1. **What is the impact of a VE's visual complexity on SS?**
   (a) *Does a visually complex VE induce more SS than a visually simpler VE?* This question is addressed in Chapter 3, Effects of Digital Content on Motion Sickness in Immersive Virtual Environments (Kavakli, Kartiko, et al., 2007).
   (b) *Does a scene with stronger depth cues induce more SS?* This question is addressed in Chapter 4, Effects of Depth Cues on Simulator Sickness (Bigoin et al., 2007).
2. **What are the correlations and the structural order among the degree of visual complexity, the level of SS, presence, affective quality and the quality of learning?**
   (a) *Does an increase in SS follow the order of AVAs visual complexity?* This question is addressed in Chapter 7, The Impacts of Animated-Virtual Actors' Visual Complexity and Simulator Sickness in Virtual Reality Applications (Kartiko, Kavakli, & Cheng, 2009b).
   (b) *Does an increase in SS follow the order of scenes' (AVAs and VEs) visual complexity?* This question is addressed in Chapter 9, Correlations between Simulator Sickness and Visual Complexity of Animated-Virtual Actors and Virtual Environments (Kartiko, Kavakli, & Cheng, 2011b).
   (c) *Does the visual complexity of AVAs increase presence and affective quality ratings?* This question is addressed in Chapter 8, Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity

(Kartiko, Kavakli, & Cheng, 2010).

(d) *Does an increase in the AVAs' visual complexity affect learning?* This question is addressed in Chapter 8, Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity (Kartiko et al., 2010) and Appendix A, Does the Visual Complexity of Animated-Virtual Actors affect learning in Virtual Reality Applications? (Kartiko, Kavakli, & Cheng, 2009a).

(e) *Does an increase in the scenes' (AVAs and VEs) visual complexity increase presence and affective quality ratings?* This question is addressed in Chapter 10, The Impacts of Visual Complexity of Animated-Virtual Actors and the Virtual Environment in Virtual Reality-based Instructional Animations (Kartiko, Kavakli, & Cheng, 2011a).

(f) *Does an increase in the scenes' (AVAs and VEs) visual complexity affect learning?* This question is addressed in Chapter 10, The Impacts of Visual Complexity of Animated-Virtual Actors and the Virtual Environment in Virtual Reality-based Instructional Animations (Kartiko et al., 2011a).

(g) *Does presence correlate with perceived affective quality?* This question is addressed in Chapter 8, Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity (Kartiko et al., 2010) and Chapter 10, The Impacts of Visual Complexity of Animated-Virtual Actors and the Virtual Environment in Virtual Reality-based Instructional Animations (Kartiko et al., 2011a).

(h) *Do the sense of presence and perceived affective quality affect the learning outcomes?* This question is addressed in Chapter 8, Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity (Kartiko et al., 2010) and Chapter 10, The Impacts of Visual Complexity of Animated-Virtual Actors and the Virtual Environment in Virtual

Reality-based Instructional Animations (Kartiko et al., 2011a).

1.3 Simulations Developed for Testing

Five VR simulations were developed for testing the research questions outlined in section 1.2.

- **Roller-coaster Simulation**. This simulation shows a cityscape with enormous amount of visual details. The extraneous details in the scene include cracks on the wall, dirt on concrete, detailed surface appearance, specular effects on the surfaces of the buildings, and many others (Figure 1.1). The user follows the path of a rollercoaster.
- **Snow Simulation**. This simulation takes the user through a pleasant scenery of snowy mountains with smoothened polygons to minimise depth cues (Figure 1.2).
- **Trenches Simulation**. This simulation plunges the user in a futuristic cityscape with protrusions and hard-edged geometries to enhance the depth cues in the scene (Figure 1.2).
- **ANTECATALYST (ANimaTEd CATAgLYphiS anTs)**. This simulation shows the navigation behaviour of *Cataglyphis* ants on featureless desert floor, see Figure 1.3. The virtual camera captures the behaviour of the AVAs that resemble *Cataglyphis* ants as they move about in the VE to seek and to retrieve food items into the nest.
- **VEAR (VEhicle Accident and fiRes)**. This simulation shows a car accident and demonstrate the safe procedures for firefighters to overcome the possible hazards. VEAR is designed to display greater visual complexity of AVAs and VEs than ANTECATALYST, see Figure 1.4.

In evaluating the visual complexity of VEs and SS, the author employed used a ”Roller-coaster Simulation” with two kinds visual themes: (1) a realistic scene with a

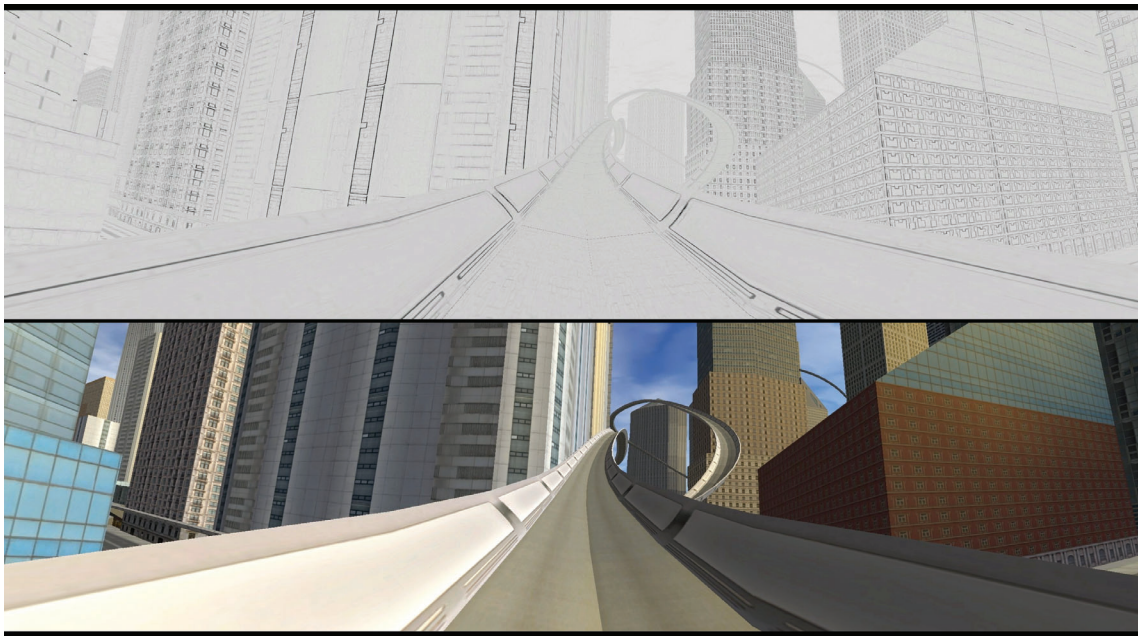


Figure 1.1: The Roller-coaster simulation has two experimental conditions: a simple black and white scene (top), and complex visuals (bottom).

plethora of extraneous details, and (2) a simple black-and-white scene [see details in Chapter 3]. In testing the effects of depth cues, or the strength of the 3D effect and SS, the author used two scenes with various depth cues: a smooth ”Snow Simulation”, and a ”Trenches simulation” with many solid protrusion objects [see details in Chapter 4].

The author also developed non-interactive VR simulations called “ANTECATALYST” (ANimaTEd CATAgLYphiS anTs) and “VEAR” (VEhicle Accident and fiRes) to investigate the impacts of visual complexity and learning. The study with ANTECATALYST only examined the effect of AVAs’ visual complexity on SS and learning [as detailed in Chapter B, 7, and 8]. ANTECATALYST taught the second-year psychology students in 2008 about *Cataglyphis* ants navigational behaviour on a salt-pan desert [as detailed in Chapter 9 and 10]. Three groups learned with (1) flat, (2) cartoon, or (3) lifelike AVAs as shown in Figure 1.3.

VEAR was deployed to confirm and extend the findings from ANTECATALYST with the participation of Macquarie University students, staff and people outside the Univer-

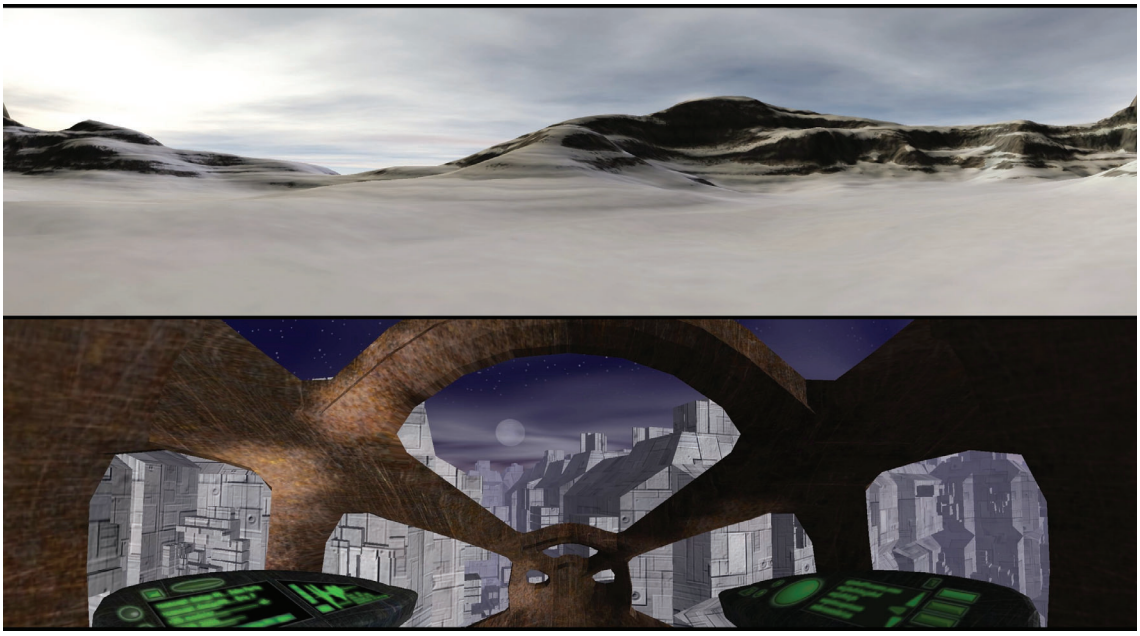


Figure 1.2: Two VR simulations of differing depth cues are used to assess the effects of depth cues and SS. The Snow Simulation (top) contains less depth cues due to the smooth-looking surface of the VE. The Trenches Simulation (bottom) shows more depth cues due to the VE that contains many protrusion of hard edges.

sity (public). VEAR showed greater visual complexity and extended the findings of the first study by taking account the VE's visual complexity. VEAR showed the concept of Village Firefighting in Fire Behaviour of a car accident. ANTECATALYST and VEAR simulations are explained in greater detail in Chapter B, 9 and 10. As shown in Figure 1.4, the participants learned with four visual conditions:

1. **Simple condition** - visually simple AVAs and background objects.
2. **Simple AVAs condition** - simple AVAs and visually complex background objects.
3. **Simple VE condition** - visually complex AVAs and simple background objects.
4. **Lifelike condition** - visually complex AVAs and background objects.

1.4 Questionnaires Used in Testing

This thesis employed a number of standardised questionnaires and custom-made retention and transfer tests to assess the effects of visual complexity on SS, presence, perceived

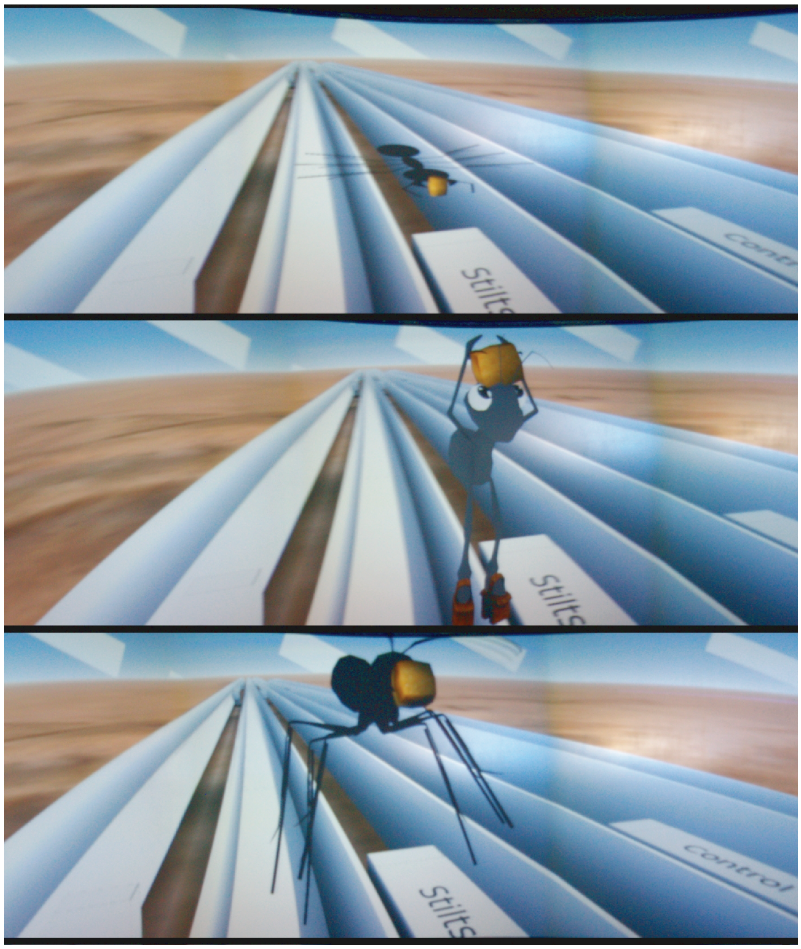


Figure 1.3: The three experimental conditions of ANTECATALYST. From top to bottom: (1) flat, (2) cartoon, or (3) lifelike.

affective quality and learning:

- **Simulator Sickness Questionnaire** (Kennedy, Lane, Berbaum, & Lilienthal, 1993, SSQ). SSQ assessed participants' subjective level of SS.
- **Presence Questionnaire** (Witmer & Singer, 1998, PQ) measured the subjective experience of being remotely present in the VR simulation. The research in the notion of "Presence" in itself, quantifying and interpreting immersive response in VR applications is still in its early stage. Presence is a complex phenomenon as there are various factors that may influence one's state being present in artificial worlds, such as, hardware factors (image quality, User Interface, display resolution, sounds, etc), and other behavioural and psychological factors of the user (past experience, per-

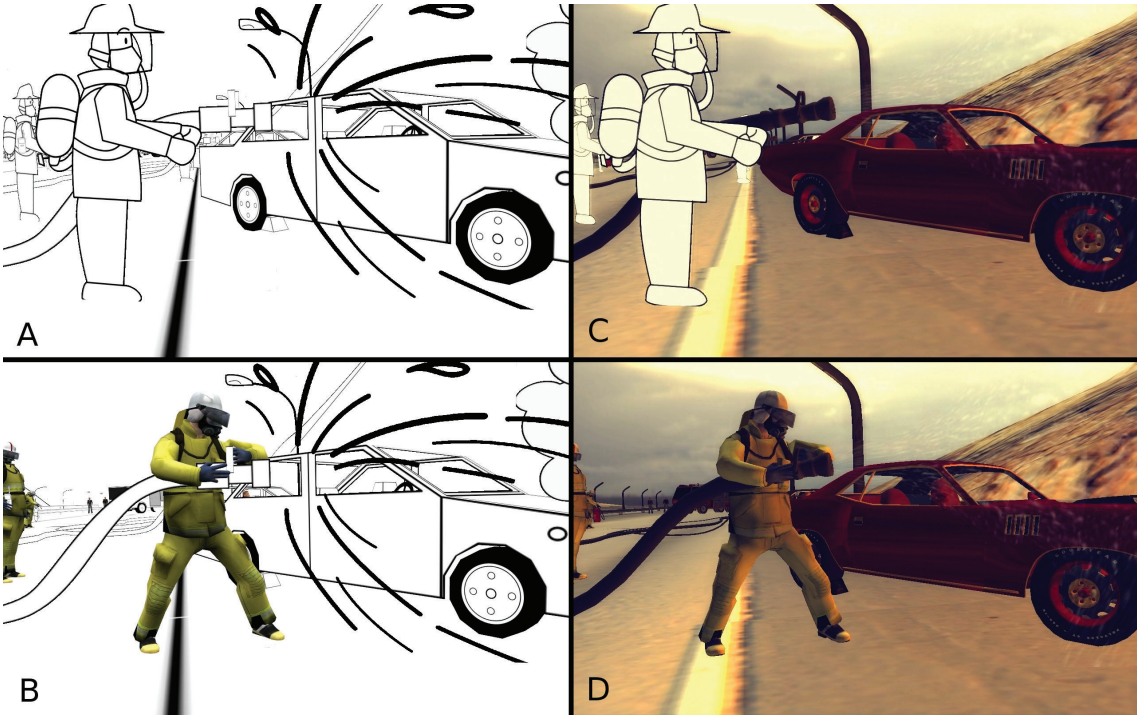


Figure 1.4: The four experimental conditions of VEAR of alternating visual complexity: (A) simple, (B)simple AVAs, (C) simple World and (D) lifelike.

sonality traits, etc) (Witmer & Singer, 1998; Slater, 1999; Slater & Garau, 2007). Further studies are required to discover the relationship between each factor in hardware, behavioural and psychological elements in order to formulate better subjective measures. Slater (1999) argues that PQ only assesses the perception of system properties and lacks in measuring presence psychologically. Therefore, the author employed Perceived Affective Quality questionnaire in conjunction with PQ to give holistic response of users being in VR simulations.

- **Perceived Affective Quality** (Russell & Pratt, 1980, PAQ) indicated perceived affective quality of the participants in the VR simulation. PAQ measures the affect or the emotional response of the user being in artificial worlds. Thus, in conjunction with PQ, both questionnaires give better measure of users' response in VR simulations. For example, PQ only asks "Do you feel like being a room?" and the answer would show tendency towards "yes" or "no" response along a likert scale. PQ does not ask "How was the room/Did you like the room?" but PAQ does.

- **Program Ratings** (Moreno, 2007, PR) evaluated students learning outcomes by measuring the perception of learning by providing affective and cognitive load scores that measured students' affect on learning how motivating, interesting and helpful it was as well as their perceived difficulty of learning.
- **Retention and transfer tests** also measured students' learning outcomes. The participants had to remember verbal information in the retention test and they had to problem-solve different scenarios in the transfer test.

1.5 Structure of the thesis

The thesis contains three main parts consisting of smaller chapters. The first part contains the introduction. The second part contains the chapters of published journals and papers, a book chapter as well as manuscripts submitted to journals. The chapters in the second part are presented in the chronological order of publication. The last part of this thesis only contains the conclusion that maps all the research questions to the empirical results, and gives an overview for the future directions of research. The structure of this thesis is shown in Figure 1.5.

The section gives an overview of the subsequent chapters in this thesis. Since the form of this thesis is by publications, the length of each chapter is relatively small. This thesis started with the finishing of two research projects conducted by Associate Professor Manolya Kavakli. The first was Simulator Sickness in VR simulations. The other was the development of a risk assessment simulator called RiskMan (Richards & Barles, 2005; Kavakli, 2006, Risk Management), upgraded and named as BOSS (BOrder Security Simulation) by using the latest immersive VR technology (see Richards et al., 2009; Dras et al., 2010). The author merged the findings of these two projects that he was involved in while exploring visual complexity, which became the focus of this thesis. Therefore, the author is not listed as the first author in the first 4 publications of part II (see Chapter 2,

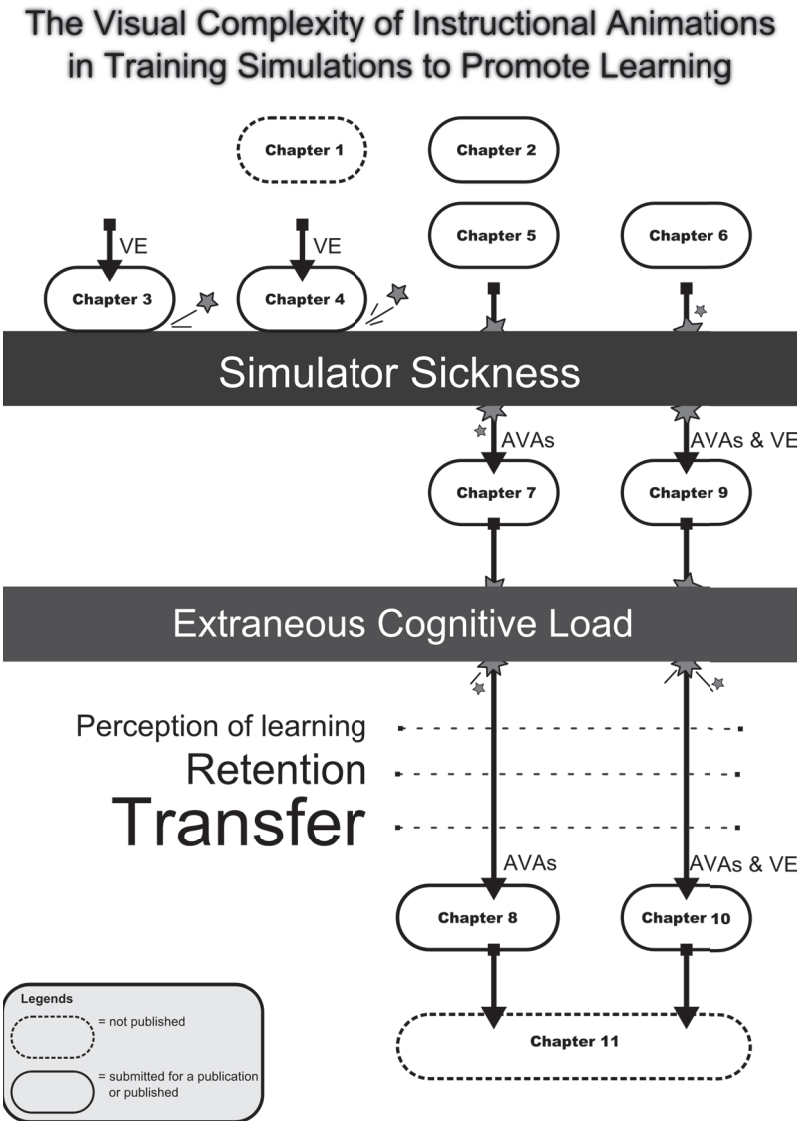


Figure 1.5: The structure of the chapters in the thesis.

3, 4 and 5). However, the author took an active part in the research, experimental design, data collection and the preparation of these manuscripts for publications. The author is listed as the first name in the following 6 chapters (see Chapter 6, 7, 8, 9, 10 and Appendix A. To keep the content of the thesis concise, the following information can be found in appendices: (1) presentation slides delivered at the 9th Meeting of the Australasian Society for Cognitive Science 2009; (2) the development of ANTECATALYST and VEAR; (3) storyboards; (4) screenplays; (5) questionnaires; (6) python codes of the simulations; (7) R codes of statistical analyses; (8) final ethics approval.

Appendix B only outlines the development of ANTECATALYST and VEAR by the author for examining the effects of visual complexity in learning materials on SS and learning. The author believe that Chapter 2, 3, 4, and 5 sufficiently describe the implementations of the other VR simulations. The author was not working alone as the only Research Associate in the studies conducted by A/Prof. Manolya Kavakli on SS (see Chapter 3 and 4) and learning with VR technology (see Chapter 2 and 5). There were also other researchers contributing to the development of those platforms.

1.5.1 Overview of the Chapters

Chapter 1. Introduction

This chapter begins with a brief overview of the two problems when learning with visual complexity in VR-based instructional animations: Simulator Sickness (SS) and the limited capacity of working memory in processing extraneous information. Following the research problems, this chapter presents the research questions of the thesis and a brief methodology to answer each question. This chapter closes with the author's contribution and the overview of the following chapters of the thesis.

Chapter 2. Avatar Creation for Believable Agents

Kavakli, M., & Kartiko, I. (2007). Avatar creation for believable agents. In Proceedings of the 10th international conference on computer graphics and artificial intelligence - 3ia'2007 (pp. 1 – 6).

Initially, the plan was to utilise the existing RiskMan (Risk Management) simulation, which was later upgraded as BOSS (BOrder Security Simulation). RiskMan and BOSS train customs and police officers in handling risk situations in a safe VR environment. Thus, the initial efforts were directed to migrate from RiskMan to BOSS, before conducting experiments to assess the impact of visual complexity on SS and learning outcomes.

Softimage—XSI 3.5 (Avid, 2007) that we used at the completion of this initial study

does provide realistic humanoid meshes, and Blender (Blender Foundation, 2011) offers node-based compositing to enhance believable-ness of agents. The author could have used these features to create believable agents easily in simulations. However, Softimage—XSI 3.5 does provide neither a realistic human skin texture nor reliable file format to transfer bone structures for animations. On the other hand, Blender does not have humanoid generator, but able to translate one 3D format to another easily. Furthermore, Blender's node-based compositing is applicable to non real-time applications or pre-rendered animations.

In overcoming these limitations, the chapter presents the development of a visually complex AVA with lifelike gait animation to simulate suspicious behaviour in a training simulation. This chapter also provides practical suggestions to improve the work-flow of AVAs' creation for VR simulations. This chapter is linked with Chapter 5 that provides the system architecture of RiskMan.

Chapter 3. Effects of Digital Content on Motion Sickness in Immersive Virtual Environments

Kavakli, M., Kartiko, I., Porte, J., & Bigoin, N. (2007). Effects of digital content on motion sickness in immersive virtual environments. In Proceedings of the 3rd international conference on computer science and information systems (pp. 1 – 12).

The study examines the various levels VE's visual complexity in a VR simulation with many objects and visual details. Many studies examine the levels of SS by changing the hardware characteristics of the VR technology, such as the Field-of-View (Lin et al., 2002; Mourant, Ahmad, Jaeger, & Lin, 2007, FOV), display size and movement (Shigemasu et al., 2006), duration (Stanney & Zyda, 2002; Bruck & Watters, 2009a), and movement speed (Mourant, Rengarajan, Cox, Lin, & Yaeger, 2007). This thesis complement the existing studies by examining the effects of the digital content on SS. This chapter is a

stepping-stone for making hypotheses or predictions in later publications. Although the published chapter does not mention about group allocation, it is important to note that the participants are randomly allocated to the groups.

Chapter 4. Effects of Depth Cues on Simulator Sickness

Bigoin, N., Porte, J., Kartiko, I., & Kavakli, M. (2007). Effects of depth cues on simulator sickness. In Proceedings of the 1st international conference on immersive telecommunications - immerscom 2007 (pp. 1 – 4). (ACM SIGMM Technically sponsored by Create-Net and EURASIP) [ERA C].

This chapter follows the findings of Chapter 3 and investigates the effects of depth cues of visually complex VEs on SS. The study compares the level of discomfort between a scene of smooth rolling hills that give less depth perception and a scene with extrusions and sharp edges that give a greater depth perception.

Chapter 5. RiskMan: A Multi-agent System for Risk Management

Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). Architectural design of multi-agent systems: Technologies and techniques. In H. Lin (Ed.), (pp. 356 – 376). Information Science Reference.

This chapter links with chapter 2, and discusses the design of a VR system and the development of a VE with AVAs. The chapter describes the modular design approach of RiskMan as a multi-agents system architecture. The modules include an expert training system, narrative engine, a game engine and a graphics engine. These modules communicate with each another in delivering complex multi-agent (multi-AVAs) learning scenarios, and thus it creates a lifelike learning environment. BOSS retains the same modular architecture as RiskMan (see Richards et al., 2009; Dras et al., 2010). RiskMan utilises an older Unreal Tournament engine 2004, while BOSS runs on the newer and flexible VR engine called Vizard (Worldviz, 2010).

Chapter 6. Virtual Reality Police Training: How much visual information is too much for knowledge transfer?

Kartiko, I., & Kavakli, M. (2008). Virtual reality police training: How much visual information is too much for knowledge transfer? In Proceedings of the 2nd international conference on cybercrime forensics education and training - cfet 2008 (pp. 1 – 13).(ISBN 1899253-19x). [ERA C].

This chapter investigates SS, presence, PAQ and CTML for the development of VR simulations that promote retention and transfer. The study explores the best practice for designing VR simulations to train police officers in the domain of Crime Prevention Through Environmental Design (Cozens, Saville, & Hillier, 2005, CPTED) in the light of CTML. Coherence Principle, a principle that Mayer (2001) derived from CTML, suggests the importance of removing the extraneous visual cues to assist the learning process. Although reducing visual details may ease SS, this may also reduce the motivational factor and thus affect learning outcomes adversely. This project could not continue with CPTED, since the author could not locate any guidelines from police officers due to its confidential nature.

Chapter 7. The Impacts of Animated - Virtual Actors' Visual Complexity and Simulator Sickness in Virtual Reality Applications

Kartiko, I., Kavakli, M., & Cheng, K. (2009b). The impacts of animated - virtual actors' visual complexity and simulator sickness in virtual reality applications. In B. Werner (Ed.), Proceedings of the 6th international conference computer graphics, imaging and visualization (pp. 147 – 152). IEEE Computer Society. (ISBN 978-0-7695-3789-4). [ERA C].

To continue further exploration on the effects of visual complexity on SS and learning, the author developed *Cataglyphis* ants simulation, known as ANTECATALYST, for undergraduate psychology students at Macquarie University in 2008. The author decided to present the results of visual complexity and SS, separately from learning outcomes.

If a learning simulation induces SS, learning outcomes would be hindered. Therefore, it is necessary to assess SS in VR simulations first. The chapter only presents the impact of visually complex AVAs and SS of ANTECATALYST simulation. ANTECATALYST presents the navigation behaviour of *Cataglyphis* ants in featureless desert. The simulation displays three types of AVAs with differing visual complexity, as shown in Figure 1.3.

In contrast to previous and other studies in similar fields, the author employed non-parametric statistical tests rather than parametric statistical tests. Chiefly, the box-plot and histogram of the dependent variables revealed outliers and skewed distributions. Although parametric tests are desirable to provide greater statistical power, the presence of outliers promote type I error, as parametric tests depend on mean values. Before applying non-parametric statistical tests, the author attempted to remove outliers, which reveals second, and third outliers. Although it was possible to do further outlier deletions to satisfy the assumptions of normal distribution in parametric tests, the samples were greatly reduced and resulted in lower statistical power. On the other hand, non-parametric tests are not affected by outliers nor skewness of distribution (Burke, 2001; Skovlund & Fenstad, 2001; Neuhäuser, 2010). Since sample size was small to begin with, the author did neither deem it expedient to delete samples nor transform the data (Neuhäuser, 2010). Moreover, Gardner & Martin (2007) suggest that the use of small number likert-scaled questionnaires (e.g., Presence Questionnaire), which value tend to cluster around a certain value in creating a "lump" effect, nonparametric test is better suited to get a reliable result. Hence, non-parametric tests were employed to answer proposed thesis questions as presented in Chapter 7, 8, 9 and 10.

Chapter 8. Learning Science in a Virtual Reality Application: The Impacts of Animated-Virtual Actors' Visual Complexity

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts

of animated-virtual actors visual complexity. Computers & Education, 55(2), 881 – 891. [ERA A]. Impact factor: 2.190.

This chapter follows the findings of Chapter 7 and describes the effects of AVAs' visual complexity on learning outcomes in greater detail. This chapter examines the impacts of AVAs' visual complexity on presence, PAQ, and the learning outcomes, by using PR, retention and transfer tests. According to Riva, Mantovani, & Gaggioli (2004) and Takatalo et al. (2008), presence is an integral component of flow in VR applications. Therefore, this chapter links the sense of presence with positive affective quality toward an optimal experience or flow for learning (Csikszentmihalyi, 1997, 1990; Ghani & Deshpande, 1994).

Apart from display factor, literature agrees that direct interaction is one of the substantial factor in inducing presence and flow (Witmer & Singer, 1998; Riva et al., 2004). Since the study is solely focusing on the issue of visual complexity, the author did not provide direct interaction in ANTECATALYST to provide each participant with matching visual stimulus within an experimental group. For example, one student may pick up a cartoon ant from the top view, and another may pick up the same ant at an oblique angle. The cartoon ant looks like a sphere from the top, while it looks like a humanoid ant with complex gaits at an oblique angle. Clearly, direct interaction cannot ensure that one participant would receive, approximately, the same visual stimulus as the other participants within the same learning condition. Thus, the discussion on direct interaction is excluded from this chapter.

Chapter 9. Correlations between Simulator Sickness and Visual Complexity of Animated-Virtual Actors and Virtual Environments

Kartiko, I., Kavakli, M., & Cheng, K. (2011b). Correlations between simulator sickness and visual complexity of animated-virtual actors and virtual environments. *Journal of Computer Animation and Virtual World*, 1 – 27. (Submitted on Sept 30, 2011). [ERA B]. Impact factor: 0.629.

To confirm the findings with ANTECATALYST and explore further the impacts of AVAs and VEs' visual complexity on SS and learning outcomes, the author developed a firefighter training simulation known as VEAR. Similar to Chapter 7, the chapter only presents the study of scenes' visual complexity and SS by using a firefighting simulation (VEAR). VEAR trains firefighters about the safe procedures in handling fire in car accidents. ANTECATALYST does not allow further examination of the effects of the VE's visual complexity, while VEAR makes this possible. The author used guidelines for handling fire from New South Wales Rural Fire Service (Service, 1999, 2001b,a, NSWRFS) to develop VEAR.

Chapter 10. The Impacts of Visual Complexity of Animated-Virtual Actors and the Virtual Environment in Virtual Reality-based Instructional Animations

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. IEEE Transactions on Affective Computing, 1 14. (Submitted on Oct 14, 2011).

This chapter closes the second part of this thesis by presenting the effects of scenes' visual complexity on learning outcomes by using VEAR. Extending ANTECATALYST, this chapter presents the impacts of AVAs' and the VEs' visual complexity on presence, perceived affective quality, and the learning outcomes in program ratings, retention and transfer tests. Similar to the study in Chapter 8, direct interaction is excluded in an attempt to provide the same visual stimuli for each participants within the same learning condition.

Chapter 11. Conclusion

This final chapter summarises all the key findings of the empirical results as guidelines for VR designers and gives future directions of research to develop VR simulations that *"work"* (Mayer, 2001).

Part II

Submitted and published articles

This part includes an accepted conference presentation, submitted and published articles (in chronological order):

1. Kavakli, M., & Kartiko, I. (2007). Avatar creation for believable agents. *In Proceedings of the 10th international conference on computer graphics and artificial intelligence* (pp. 1 – 6).

2. Kavakli, M., Kartiko, I., Porte, J., & Bigoin, N. (2007). Effects of digital content on motion sickness in immersive virtual environments. *In Proceedings of the 3rd international conference on computer science and information systems* (pp. 1 – 12).

3. Bigoin, N., Porte, J., Kartiko, I., & Kavakli, M. (2007). Effects of depth cues on simulator sickness. *In Proceedings of the 1st international conference on immersive telecommunications* (pp. 1 – 4). (ACM SIGMM Technically sponsored by Create-Net and EURASIP) [ERA C].

   Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). Riskman: A multi-agent system for risk management. In H. Lin (Ed.), Architectural design of multi-agent systems: Technologies and techniques (pp. 356 – 376). IGI Global.

4. Kartiko, I., & Kavakli, M. (2008). Virtual reality police training: How much visual information is too much for knowledge transfer? *In Proceedings of the 2nd international conference on cybercrime forensics education and training* (pp. 1 – 13).(ISBN 1899253-19x). [ERA C].

5. Kartiko, I., Kavakli, M., & Cheng, K. (2009b). The impacts of animated - virtual actors' visual complexity and simulator sickness in virtual reality applications. In B. Werner (Ed.), *Proceedings of the 6th international conference computer graphics, imaging and visualization* (pp. 147 – 152). IEEE Computer Society. (ISBN 978-0-7695-3789-4). [ERA C].

6. Kartiko, I., Kavakli, M., & Cheng, K. (2009a). *Does the visual complexity of animated-virtual actors in virtual reality applications affect learning?* Presented

at the 9th Meeting of the Australasian Society for Cognitive Science. Sydney. (paper was presented, but withdrawn - not published). [ERA C].

7. Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. *Computers & Education*, 55(2), 881 – 891. [ERA A]. Impact factor: 2.190.

8. Kartiko, I., Kavakli, M., & Cheng, K. (2011b). Correlations between simulator sickness and visual complexity of animated-virtual actors and virtual environments. *Journal of Computer Animation and Virtual World*, 1 – 27. (Submitted on Sept 30, 2011). [ERA B]. Impact factor: 0.629.

9. Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. *IEEE Transactions on Affective Computing*, 1 - 14. (Submitted on Oct 24, 2011).

Chapter 2

Avatar creation for believable agents

Pages 30-35 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Kavakli, M., & Kartiko, I. (2007). Avatar creation for believable agents. In *3IA 2007: Proceedings of the 10th 3IA International Conference* (pp. 1-6). Germany: Springer, Springer Nature.

Chapter 3

Effects of digital content on motion sickness in immersive virtual environments

Pages 38-49 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Kavakli, M., Kartiko, I., Porte, J., & Bigoin, N. (2008). Effects of digital content on motion sickness in immersive virtual environments. In P. Petratos, & P. Dandapani (Eds.), *Current advances in computing, engineering and information: papers from the third International Conference on Computer Science and Information Systems on July 23-24, 2007* (pp. 331-343). Athens: Athens Institute for Education and Research (ATINER).

Chapter 4

Effects of depth cues on simulator sickness

Pages 52-55 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Bigoin, N., Porte, J., Kartiko, I., & Kavakli, M. (2007). Effects of depth cues on simulator sickness. In *IMMERSCOM 2007: Proceedings of the First International Conference on Immersive Telecommunications* (pp. 1-4). Hong Kong: The Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering(ICST).

DOI: 10.4108/ICST.IMMERSCOM2007.2278

Chapter 5

RiskMan: A multi-agent system for risk management

Pages 58-78 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). RiskMan: a multi-agent system for risk management. In H. Lin (Ed.), *Architectural design of multi-agent systems: technologies and techniques* (pp. 356-376). Hershey, PA; London: IGI Global.

DOI: 10.4018/978-1-59904-108-7.ch018

Chapter 6

Virtual reality police training: How much visual information is too much for knowledge transfer?

Pages 80-92 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Kartiko, I., Kavakli, M., 2008: Virtual Reality Police Training How much visual information is too much for knowledge transfer?, In *2nd International Conference on Cybercrime Forensics Education & Training, CFET 2008***,** Canterbury Christ Church University, Canterbury, United Kingdom, 1-2 September 2008, 1-13.

Chapter 7

The Impacts of Animated - Virtual Actors' Visual Complexity and Simulator Sickness in Virtual Reality Applications

Pages 94-99 of this thesis have been removed as they contain published material. Please refer to the following citation for details of the article contained in these pages.

Kartiko, I., Kavakli, M., & Cheng, K. (2009). The impacts of animated-virtual actors' visual complexity and simulator sickness in virtual reality applications. In E. Banissi, M. Sarfraz, J. Zhang, A. Ursyn, W. C. Jeng, M. W. Bannatyne, J. J. Zhang, L. H. San, ... M. L. Huang (Eds.), *Proceedings of the 2009 6th International Conference on Computer Graphics, Imaging and Visualization: New Advances and Trends, CGIV2009* (pp. 147-152). [5298247] Los Alamitos, CA, USA: Institute of Electrical and Electronics Engineers (IEEE).

DOI: 10.1109/CGIV.2009.66

Chapter 8

Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity. Computers and Education, 55(2), 881-891. https://doi.org/10.1016/j.compedu.2010.03.019

Computers & Education 55 (2010) 881–891

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

Learning science in a virtual reality application: The impacts of animated-virtual actors' visual complexity

Iwan Kartiko [a,*], Manolya Kavakli [a], Ken Cheng [b]

[a] Department of Computing, Macquarie University, North Ryde, Sydney, NSW 2109, Australia
[b] Department of Brain, Behaviour and Evolution, Macquarie University, North Ryde, Sydney, NSW 2109, Australia

ARTICLE INFO

Article history:
Received 29 November 2009
Received in revised form
29 March 2010
Accepted 30 March 2010

Keywords:
Applications of animated-virtual actors
Evaluation of CAL systems
Human-computer interface
Multimedia/hypermedia systems
Virtual reality

ABSTRACT

As the technology in computer graphics advances, Animated-Virtual Actors (AVAs) in Virtual Reality (VR) applications become increasingly rich and complex. Cognitive Theory of Multimedia Learning (CTML) suggests that complex visual materials could hinder novice learners from attending to the lesson properly. On the other hand, previous studies have shown that visual complexity correlates with presence and may increase the perceived affective quality of the virtual world, towards an optimal experience or flow. Increasing these in VR applications may promote enjoyment and higher cognitive engagement for better learning outcomes. While visually complex materials could be motivating and pleasing to attend to, would they affect learning adversely? We developed a series of VR presentations to teach second-year psychology students about the navigational behaviour of *Cataglyphis* ants with flat, cartoon, or lifelike AVAs. To assess learning outcomes, we used Program Ratings, which measured perception of learning and perceived difficulty, and retention and transfer tests. The results from 200 students did not reveal any significant differences in presence, perceived affective quality, or learning outcomes as a function of the AVA's visual complexity. While the results showed positive correlations between presence, perceived affective quality and perception of learning, none of these correlates with perceived difficulty, retention, or transfer scores. Nevertheless, our simulation produced significant improvements on retention and transfer scores in all conditions. We discuss possible explanations and future research directions.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The application of Animated-Virtual Actors (AVAs) is an integral part of any Virtual Reality (VR) application. AVAs are either two-dimensional (2D) or three-dimensional (3D) entities in a virtual scene and may depict humans or other creatures (Nowak & Rauh, 2008). There are a number of interesting examples of AVAs developed for learning. Johnson and Rickel (1997) showed an AVA known as Steve, to help the user to run a high-pressure air compressor. Lester et al. (1997) and Lester, Stone, and Stelling (1999) showed the ability of an AVA, Herman the bug, to help students to learn about climate and plants. Wagner, Billinghurst, and Schmalstieg (2006) let Mr. Virtuoso guide students to learn art history in a museum. Moreno and Flowerday (2006) employed various AVAs with different ethnicities to help students' learning. Peddy the parrot guided the students to solve mathematical problems (Lusk & Atkinson, 2007). An AVA called "Dr. Bob" helped students learn the human cardiovascular system (Dunsworth & Atkinson, 2007). The BOSS used AVAs to teach the procedure to examine an incoming passenger in an airport (Kavakli, Szilas, Porte, & Kartiko, 2007, chap. 18; Richards, Dras, Porte, & Meredith, 2009).

Based on previous studies, there are four main arguments to use AVAs in an educational setting, apart from the graphics and computer technology. Firstly, AVAs show actions or certain behaviours that would be difficult to describe verbally (e.g., Kavakli et al., 2007; Richards et al., 2009). Secondly, they are able to guide the learning process (Holmes, 2007; Lester et al., 1997; Lusk & Atkinson, 2007; Moreno & Flowerday, 2006). Thirdly, they motivate the users to attend to the lesson (Gulz & Haake, 2006; Lester et al., 1997, 1999; Sadowski & Stanney, 2002, pp. 791–806; Wagner et al., 2006). Lastly, a recent study by Dunsworth and Atkinson (2007) shows that the use of AVAs improves retention. As the computer graphics technology advances, with many affordable and open source 3D engines and modelling tools (Kartiko & Kavakli, 2008,

* Corresponding author. Tel.: +61 2 9850 9081.
E-mail address: ikartiko@science.mq.edu.au (I. Kartiko).

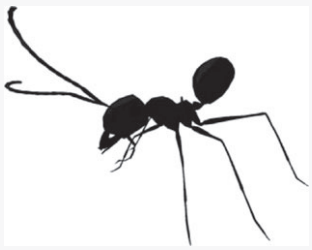
0360-1315/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2010.03.019

pp. 1–13; Kartiko, Kavakli, & Cheng, 2009), so do its applications in creating complex AVAs. An obvious example of this trend is the contrast between Steve and BOSS' AVAs. Steve (Johnson & Rickel, 1997) consists of primitive polygons, basic colours and some basic animations, as the technology allowed at that time. However, BOSS' AVAs (Kavakli et al., 2007; Richards et al., 2009) were developed with much more visual complexity, such as: lifelike facial features, life like skin tones, detailed clothing, wrinkles, along with motion-captured animations. Nevertheless, despite the feasibility of creating visually complex AVAs, the designer must question if it would affect learning? If creating simple AVAs is sufficient for learning, would it benefit the learners to have visually complex AVAs, as the current graphics hardware allows?

This study proposes to examine the impact of AVAs' visual complexity on learning from the view of Cognitive Theory of Multimedia Learning (Mayer, 2001, 2003, CTML), and the sense of presence and perceived affective quality within the framework of optimal experience or flow theory (Csikszentmihalyi, 1990, 1997). Unlike other multimedia means, such as web-learning or textbook, users in VR applications are subject to continuous streams of visual, auditory and sometimes, haptic information. Previous studies (e.g., Baddeley, 1998) show that humans have limited capacity of working memory to temporarily store and process information. In particular, Miller (1956) showed that short-term memory can hold about seven *chunks* of information, where a chunk is the amount of information in a schema — a single bit, a decimal digit, a word, or a phrase. Kavakli and Gero (2003) have considered Miller's (1956) magical number 7 +/− 2 in designing protocols. They found that those with 25 years experience of architectural sketching used at most 6 cognitive actions efficiently at a time, in producing many and detailed sketches. However, the novice group was unable to structure their task and used up to 16 cognitive actions at a time in producing fewer and poorer quality sketches. Studies about CTML by Harp and Mayer (1997, 1998), Mayer, Heiser, and Lonn (2001), Moreno (2007) suggest that complex visual materials, as a whole — not only AVAs but also the background environment — undermine learning performance under some circumstances. Therefore, from CTML's view, creating AVAs that are visually more complex could hinder learning. At the same time, the key advantage of VR applications is that they induce presence and affective quality, which may promote an optimal experience or flow. According to Csikszentmihalyi (1990) flow improves cognitive ability, which renders the task more pleasant, enjoyable, and easy. Earlier studies show that the visual complexity of the virtual world affects presence and humans constantly experience affective quality (Russell, 2003; Russell & Pratt, 1980). Furthermore, humans have shown a preference for complex visual stimuli (Roberts, 2007). Therefore, presence and the perceived affective quality should increase when viewing AVAs that are more complex. Thus, contrary to CTML's view, increasing visual complexity of AVAs may perhaps promote higher flow and better learning outcomes.

In this study our aim was to answer the following questions: (1) Does the visual complexity of AVAs increase presence and affective quality ratings? (2) Does an increase of the AVA's visual complexity affect learning? (3) Do the sense of presence and perceived affective quality affect the learning outcomes? To answer these questions, this study employed ANTECATALYST (ANimaTEd CATAgLYphiS anTs), a recently developed non-interactive VR learning module. ANTECATALYST was used in the curriculum of second-year psychology students in 2008. ANTECATALYST took the students through a 3D North African saltpan to learn about the navigational strategies of *Cataglyphis* ants (Wehner, 2003). Three groups learned with flat, cartoon, or lifelike AVAs (Table 1). We used the standardised Presence Questionnaire

Table 1
Visual complexity of AVAs.

AVA	Visual	Properties	
		Shape	Animation
Flat		Flat	none
Cartoon		Humanoid	Human gait
Lifelike		Lifelike	Tripods gait

(Witmer & Singer, 1998, PQ) and Perceived Affective Quality (Russell & Pratt, 1980, PAQ) ratings. PQ measured the students' subjective experience of being physically present in the virtual world. PAQ measured students' perceived affective quality in viewing the VR. We employed Program Ratings (Moreno, 2007, PR), and retention and transfer tests to measure students' learning outcomes. PR provided affective and cognitive load scores, which measured students' perception of the learning — how motivating, interesting and helpful it was — and the perceived difficulty of learning. The students then had to recall information in a retention test and they had to infer the behaviour of *Cataglyphis* ants, based on what they had learned, in a transfer test.

In this article, we refer to an AVA's visual complexity as the amount of visual information in depicting an AVA. For example, adding life-like facial features and hair adds visual complexity, but increasing the degree of the AVA's pictorial realism does not always increase visual complexity. In Fig. 1, we see an example of a realistic representation of a fighter in a bright room, which is less complex than the non-realistic counterpart. In addition, the file-size of the non-realistic fighter is approximately 2.5 times more than that of the realistic one at the same image resolution. The file-size comparison confirms and shows that the depiction of the non-realistic fighter contains more visual information than the realistic one (Donderi, 2006). The *Cataglyphis* ants in this study (Table 1), with their gaits, can provide more complex shapes and visual information than typical human-shaped AVAs. Table 1 summarises the different learning conditions with the three different levels of the AVA's visual complexity. The flat AVA was an extreme simplification of the shape of *Cataglyphis* ants, only a simple drawing placed on a one-polygon plane with no gait animations. The cartoon AVA was a simplified or cartoon model of *Cataglyphis* ants and consisted of fewer polygons, fewer joints and simpler animations than the lifelike AVA. The lifelike AVA was modelled after and animated according to the real *Cataglyphis* ant and this was the complex AVA.

2. Theoretical background

2.1. Cognitive theory of multimedia learning

As the use of multimedia learning systems became common with a lot of information being given at a time, concern was raised about how all the information could be processed simultaneously. We studied the effect of an AVA's visual complexity in a VR presentation in the framework of the Cognitive Theory of Multimedia Learning (Mayer, 2001, 2003, CTML). CTML assumes 1) separate visual and auditory channels for receiving information (Baddeley, 1998; Paivio, 1986); 2) limited capacity of working memory (Baddeley, 1998; Sweller, 1999); and 3) active learning: the learner must actively select, organise and store information for long-term memory (Wittrock, 1989). CTML attempts to overcome working memory's limitation by splitting the information into visual and verbal parts, to promote more active processing and to reduce cognitive load. Substantial empirical evidence suggests that CTML improves learning (Hoffler & Leutner, 2007; Mayer, 2001, 2003).

In a classic book, *Interest and Effort in Education*, Dewey (1913, p. ix) had already warned that adding interesting, yet irrelevant details will interfere with learning. Studies testing coherence principles of CTML showed that the visual complexity of scenes with many interesting but irrelevant visual details hinders learning (Harp & Mayer, 1997, 1998; Mayer et al., 2001). In one study by Harp and Mayer (1997), students learned about lightning. Those who learned with a visually complex display, with many visual details such as an aeroplane, a swimmer, a golfer and a person with a damaged football jersey, found the material interesting but performed worse than students who learned with a simple visual presentation. In a series of follow-up experiments, Harp and Mayer (1998) showed that the students who learned with entertaining yet irrelevant details also found it interesting but performed poorly in retention and problem-solving tests. Similarly, Mayer et al. (2001) added entertaining yet conceptually irrelevant materials, such as bent trees because of strong wind, an ambulance rescuing

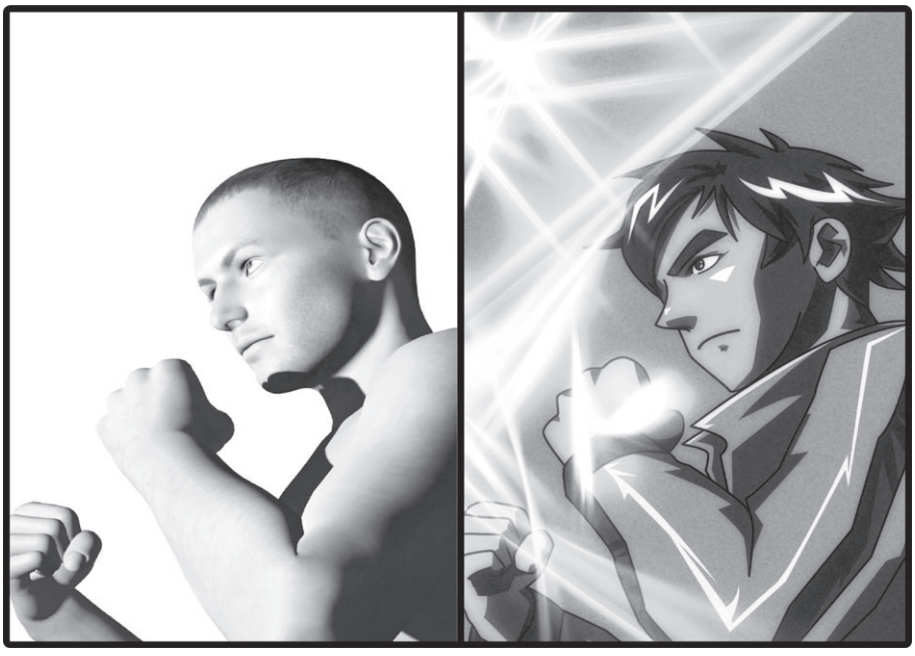


Fig. 1. A comparison of a realistic and a non-realistic AVA. Pictorial realism does not always convey more complexity.

a victim with a stretcher, and some onlookers standing by; these additions too resulted in poorer retention and transfer. CTML favours removing 'seductive' details that do not add to learning content to foster better learning (Mayer, 2003).

In ANTECATALYST, the visual information of an accurate 3D body shape and the tripod gait (Zollikofer, 1994) are irrelevant, because ANTECATALYST teaches the navigation behaviour of Cataglyphis ants and not their anatomical features. According to the Coherence principle of CTML, such visual complexity might overload the visual channel and hinder learning. Therefore, we predicted that the learning perception and the difficulty rating will increase with the AVA's visual complexity and the scores in retention and transfer tests will decrease with the AVA's visual complexity.

2.2. Flow theory

Writing data into a blank disc is as simple as to drag-and-drop the object folder into the destination medium, then clicking the '*Burn*' button for a perfect copy. For humans, gaining new knowledge or skills is not like burning information onto a blank disc. Instead, humans enter into a state of optimal experience or flow, which elevates cognitive engagement, which in turn renders the task easier and more enjoyable (Csikszentmihalyi, 1990, p. 74). According to Csikszentmihalyi (1990, p. 71), in this state, one becomes deeply immersed in a task, diverting attention away from the surroundings, other problems, or the passage of time.

The key feature of VR applications is that they are able to give the sense of presence, one of the integral components of flow (Mikropoulos & Strouboulis, 2004; Novak, Hoffman, & Yung, 2000; Takatalo, Nyman, & Laaksonen, 2008). However, the construct of presence alone does not guarantee flow for learning. Therefore this study also considered the affective quality of the virtual environment. The effects of the AVA's visual complexity on presence, the perceived affective quality of the virtual environments and learning outcomes within the framework of flow theory will be discussed next.

2.2.1. Presence

Witmer and Singer (1998, p. 225) defined presence as "the subjective experience of being in one place or environment, even when one is physically situated in another." Our study examined the impact of an AVA's visual complexity in increasing the sense of presence or simply, presence in VR applications. For many researchers, presence is the key feature for effective VR applications (Mikropoulos & Strouboulis, 2004; Witmer & Singer, 1998). Some studies suggest that presence contributes to motivation and enjoyment value in users, inducing them to attend to or perform tasks in VR (Lin, Duh, Parker, Abi-Rached, & Furness, 2002; Sadowski & Stanney, 2002).

How does an AVA's visual complexity influence presence? The study of the sense of presence in VR is new and the literature is limited. However, some studies suggest that presence increases with visual complexity. Witmer and Singer (1998) suggested that a VR scene with a great amount of visual stimuli should promote a higher sense of presence. Stanney, Kingdon, Graeber, and Kennedy (2002) added textures and higher ceilings to increase a scene's complexity, and that led to an increase in presence. Additionally, a recent study in media richness and flow by Liu, Liao, and Pratt (2009) showed that richer educational content led to higher flow. Also, as aforementioned, presence correlates with flow. Therefore, our hypothesis was that the AVA's visual complexity would also increase overall complexity of the scene, which in turn will increase the sense of presence. ANTECATALYST contains three different types of AVAs (Table 1). We predicted the sense of presence, which contributes to flow, would increase with the AVA's visual complexity.

How does presence influence learning? The literature on this topic is also rather limited (Mikropoulos, 2006) as well as inconsistent. Mikropoulos and Strouboulis (2004, pg. 1) stated that "presence is correlated with higher levels of cognitive performance and emotional development, factors that contribute to knowledge construction." Yet recent studies by Richards et al. (2009) and by Moreno and Mayer (2002) showed that presence does not correlate with learning. Whitelock, Romano, Jelfs, and Brna (2000) heightened the audio to increase presence, but found no benefit in learning scores. In addition, Moreno and Mayer (2002) suggested that presence matters little when the learning material adheres to the principles of CTML. These discrepancies in presence and learning outcomes might arise because presence did not consistently promote flow for learning. Although presence is an integral component of flow, its presence may not always increase motivation for learning. We will consider predictions after first discussing affective quality in the following section.

2.2.2. Perceived affective quality

Affect encompasses mood and emotions, which influence perception, cognition and behaviour (Pintrich, 2003, chap. 6, pp. 103–122; Russell, 2003). Affective quality is an environment's ability to change one's core affect (Russell, 2003; Russell & Pratt, 1980). Although these concepts are similar, Russell (2003) distinguishes them based on where they originated. Core affect exists within the person, while affective quality exists in the environment.

As previously mentioned, presence correlates with flow. Presence, however, is not always a pleasant experience. Virtual Reality Exposure Therapy (VRET) employs presence to induce discomfort, anxieties and fears. Regenbrecht, Schubert, and Friedmann (1998) showed that VR was able to induce fear of heights, even with minimal visualisation and presence (Krijn et al., 2004). For example, a user was placed on top of an eight-storey building in a virtual world. A question from the PQ (Witmer & Singer, 1998) asks: "How compelling was your sense of moving around inside the virtual environment?" This would elicit a positive answer from people with a fear of heights, but this answer does not indicate their enjoyment or motivation to learn. Therefore, in our study, we employed in addition the PAQ ratings to ask, "How was it, pleasant or unpleasant?" Affective quality is also an important factor in flow. Sanchez-Franco (2010) noted that affective quality correlates with flow. Thus, for a VR simulation to induce flow, presence must be accompanied by a pleasant and enjoyable experience. We hypothesised that PAQ influences presence, and predicted that those who reported higher PAQ will also report higher presence.

How might an AVA's visual complexity (Table 1) influence perceived affective quality in VR? Again, the literature on the relation between affective quality and visual complexity is limited. Recently, Roberts (2007) explored the dimensions of visual complexity and aesthetic preference in art works and paintings. His participants were asked to rate pictures on their beauty (Roberts, 2007, Annex A). Results showed a correlation between the amount of elements in a picture and aesthetic preference. Roberts' participants gave higher beauty ratings to pictures with more elements in them. Our hypothesis was the AVA's visual complexity will also increase the overall affective quality of the virtual world. We predicted the PAQ ratings would increase with the AVAs' visual complexity.

How might affective quality influence learning? Pintrich (2003) suggested that positive motivation and affect may increase cognitive engagement (see also Picard et al., 2004). Zhang and Li (2005) showed that affective quality correlates with motivation, enjoyment, and ease of use. In addition, Brand, Reimer, and Opwis (2007) showed that negative affect or emotion hinders transfer performance. Csikszentmihalyi (1997) stated that concentration requires more effort when motivation and mood are against it. When concentration to focus on the task at hand is hindered, then the learning process would be difficult. From these findings, it seems that a virtual environment that supports a strong level of presence but induces negative affect, could not promote flow and therefore, could undermine learning. Additionally, recent studies in flow by Lee, Yoon, and Lee (2009) and Liu et al. (2009), showed that flow also leads to an increase in users' acceptance of the system and willingness to learn using the system, factors that are crucial in e-learning systems. We argue that these are also crucial in every learning system, as one will be hindered to learn in an environment that induces repulsion or aversion. Therefore, we hypothesised that an increase in presence and PAQ in VR applications would increase the perception of the learning — how motivating, interesting and helpful it was — as well as retention and transfer scores, while it would decrease perceived difficulty.

3. Methods

3.1. Instructional material design

Vizard (Worldviz, 2009) displayed ANTECATALYST on a 6-m-wide canvas, an immersive semi-cylindrical projection system that provides a 160-degree field of view (FOV). We designed ANTECATALYST to teach students about the navigational strategies of *Cataglyphis* ants on a featureless saltpan (Wehner, 2003).

The VR simulation reproduced the field experiments and observations conducted by Zollikofer (1994), Wehner (2003), and Wittlinger, Wehner, and Wolf (2007). We organised it into five continuous scenes: 1 – introduction; 2 – overview of path integration; 3 – skylight compass; 4 – step counting behaviour; and 5 – summary. The simulation was 8 min long and the camera moved slowly from one place to another. The AVAs were visible in all scenes, except in scene 3. We used Blender (Blender Foundation, 2009) for 3D modelling and animation; Vue (e-on software, 2006), Gimp (Gimp Development Team, 2009) and Inkscape (Inkscape Development Team, 2009) for texturing the virtual world. As depicted in Fig. 2, and the supplementary movies [1], the scene consisted of AVAs, plain sand throughout, a clear blue sky, simulated polarisation patterns in the sky, food items and three test-channels. Supplementary movies 1, 2 and 3 show small portions of scene 4 with flat, cartoon and complex AVAs, respectively.

Throughout the experiment, the scenario, screen resolution, screen colour, camera's movements and the narration were the same. ANTECATALYST ran smoothly at maximum framerate. The only independent variable was the level of visual complexity of the AVAs, which was altered for each practical group. Since there was no interactivity within ANTECATALYST, everyone received equal visual exposure to the AVAs.

3.2. Participants and experiment design

The participants were 200 s-year psychology students at MacquarieUniversity (59 males and 141 females) with median age of 20.55 (IQR = 20–22). A practical group of 4–25 students participated in the experiment in the VR lab at the same time. Each practical group as a whole was randomly assigned to one of three conditions differing in the AVA's complexity (Table 1). 60 participants learnt with flat AVAs, 63 participants learnt with cartoon AVAs, and 77 participants learnt with lifelike AVAs. Since the session was part of the course, the student received no extra credit. None of the students had attended any lectures on the topic before.

3.3. Measurements

For each participant, the paper-and-pencil materials included the Simulator Sickness Questionnaire (Kennedy, Lane, Bream, & Lilienthal, 1993, SSQ), a ten-page retention and transfer pretest, the Presence Questionnaire (Witmer & Singer, 1998, PQ), Perceived Affective Quality (Russell & Pratt, 1980, PAQ) ratings, a Program Ratings (PR), and a ten-page retention and transfer posttest. We employed SSQ to measure simulator sickness and to assess its suitability for future use as VR applications are known to cause motion sickness sometimes. These materials, which served as the dependent measures, were typeset and presented on standard A4-size papers. All questionnaire items, with the retention and transfer post-test are included in the Appendix section of this article.

The PQ measured the subjective presence. We used only 10 relevant questions, out of 30 possible questions. The questionnaire contained 10 questions with the instruction "Please read the questions carefully and circle the appropriate number from 1 to 7." The scale from 1 to 7 ranged from Not Engaged to Moderately Engaged to Very Engaged. The internal consistency coefficient (Cronbach's alpha) of the PQ was 0.81.

To measures students' PAQ, we employed the shorter version of the PAQ ratings with twenty adjective words. This questionnaire was given to the students with the instruction "Below is a list of words that can be used to describe places. We would like you to rate how accurately each word below describes the Virtual World you just visited. Use the following 1–8 rating scale for your answer. Please be sure that you have given an answer for each word. 1 = extremely inaccurate; 2 = very inaccurate; 3 = quite inaccurate; 4 = slightly inaccurate; 5 = slightly accurate; 6 = quite accurate; 7 = very accurate; 8 = extremely accurate." The internal consistency coefficient (Cronbach's alpha) for ratings of the words ranged from 0.70 to 0.85.

The PR was adapted from a study by Moreno (2007). It contained eight questions to assess students' perceptions of learning on a five-point scale, e.g., 1 as boring and 5 as interesting. The first six questions measured the affective level during the presentation, that is, the perception of learning, motivation, helpfulness, and interest, with internal consistency of 0.91. The last two questions measured the cognitive load level or the perceived difficulty level, with internal consistency of 0.81.

[1] The field of view of the video clips was modified in order to fit three-screens of 6 m × 2 m size canvas into one 3:4 screen ratio for submission purpose.

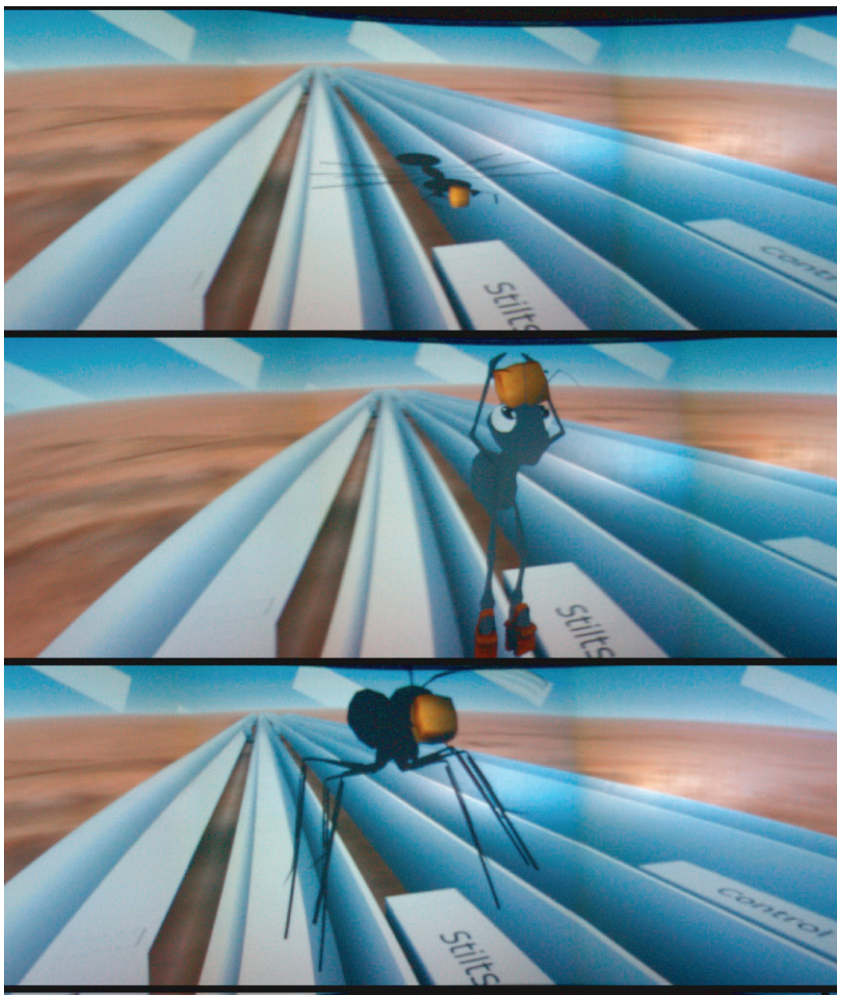


Fig. 2. AVAs as projected on canvas. From top to bottom: flat, cartoon and lifelike AVA.

The pretest and post-test in the retention test consisted of eight multiple-choice questions with an instruction "Please answer all 8 questions. Please circle the correct answer. This is a practice test and not part of course grades." In pretest and post-test, the questions remained the same, but the order of the multiple-choice answers were changed.

The pretest and post-test in the transfer, or problem-solving test consisted of four multiple-choice questions with a similar instruction "Please answer all 4 questions. Please circle the correct answer. This is a practice test and not part of course grades." The students received four questions that described four experimental manipulations on the ants. The students were expected to predict the ant's return path to the nest. In pretest and post-test, the questions were similar, but with different experimental manipulations on the ants. The order of the multiple-choice answers were changed as well.

3.4. Experimental procedures

On arrival at the VR lab, the students were asked to fill in the consent form and randomly assigned to one of three AVA conditions as a group. We told the participants the procedure to follow if they feel motion sickness during VR simulation. SSQ ratings were collected before and after viewing the simulation, to assess its fitness for future uses (see Kartiko et al., 2009). We gave the students 10 min to do the retention and transfer tests before presenting the VR simulation. After viewing the VR simulation, they filled in SSQ, PQ, PAQ ratings, PR, and the retention and transfer post-test. Once they completed the questionnaires, we thanked them for the participation and fully explained the nature of the study. Finally, a tutor took over the class in the VR lab to explain the material again in greater detail.

3.5. Scoring

A presence score was computed for each participant by adding the circled numbers on the sheet. For the negatively worded questions, the circled number on the sheet was multiplied by −1. Because of the unbalanced number of positive and negative question items, 8 points were deducted from the total score, so that the possible scores ranged from −30 to +30. Perceived affective quality scores were computed for each participant by adding the quality ratings next to the adjective words to give arousal, sleepy, pleasant and unpleasant scores. The arousal–sleepy and pleasant–unpleasant quality scores were obtained by subtracting the scores of negative items from the scores of the positive items, with a possible range from −35 to +35. The perception of learning score from the PR was computed for each participant by

calculating the means of the circled number of the first six questions on the PR sheet. Similarly, perceived difficulty score was computed by calculating the means of the circled number of the last two questions on the PR sheet. The affective and cognitive load scores ranged from 1 to 5. A retention score for each participant was calculated by multiplying the number of correct answer by 6.25. The transfer score was computed for each participant by multiplying the number of correct answer by 12.5. Possible retention and transfer scores ranged from 0 to 50. Thus, the highest test score that each participant could attain was 100 points.

4. Results

We found violations of the assumptions needed for parametric statistical tests from using the Shapiro–Wilk normality test and the Levene test for unequal variances. Therefore, we employed nonparametric tests in this study. For effect sizes, we calculated Cliff's *d* values (Cliff, 1993). All statistical analysis methods were performed by using R (R Development Core Team, 2008). Fig. 3 shows the box plots of the dependent variables. Additionally, Table 2 shows the medians (Mdn) and the interquartile range (IQR) of the dependent variables.

Question 1: Does the visual complexity of AVAs increase presence and affective response? First of all, a Kruskal–Wallis nonparametric ANOVA test was performed with the AVAs' visual complexity as the factor, and the presence scores as the dependent variable. The test did not reveal any significant differences between the groups, χ^2 (2, $N = 200$) = 2.24, $p = 0.33$. The results did not support the proposed prediction and showed no differences between flat, cartoon and lifelike AVA in promoting a sense of presence.

Again, we ran the same test with the AVA's visual complexity as the factor, and the pleasant–unpleasant and arousal–sleepy scores as the dependent variables. Again, the test did not reveal any significant differences between the groups in the pleasant–unpleasant score, χ^2 (2, $N = 200$) = 2.24, $p = 0.33$ or the arousal–sleepy score, χ^2 (2, $N = 200$) = 1.08, $p = 0.58$. The results from PAQ ratings did not confirm our earlier prediction. There were no differences between types of AVAs in the perceived affective quality of VR simulation that they created.

Question 2: Does the increase in the AVA's visual complexity affect learning outcomes? To answer this question, we performed a Kruskal–Wallis nonparametric ANOVA test. We assigned the AVA's visual complexity as the factor, and the perception of learning, perceived difficulty, the difference in retention and transfer scores (post-test score minus pretest score) as the dependent variables.

Contrary to our predictions according to CTML, increasing the visual complexity of AVAs did not lead to higher perception of learning, χ^2 (2, $N = 200$) = 0.12, $p = 0.94$. The students who attended the practical session with flat AVAs reported the same perception of learning — motivation, helpfulness, and interest — as those who learned with the visually more complex AVAs. Overall, the perception of learning was high across the groups, Mdn = 4.00, IQR = 3.50–4.50.

There was also no significant differences in perceived difficulty ratings between the groups, χ^2 (2, $N = 200$) = 0.27, $p = 0.87$, and results did not support the prediction according to CTML. The perceived difficulty rating was average across the groups, Mdn = 2.00, IQR = 1.50–2.13.

Also, the visual complexity of AVAs did not affect retention, χ^2 (2, $N = 200$) = 0.42, $p = 0.81$. All groups of students, however, showed significant learning after viewing the presentation. The Wilcoxon signed rank test between post-test and pretest retention scores in each group revealed a significant increase after viewing the presentation (flat: $V = 1830$, $p = 0.00$, $d = 0.98$; cartoon: $V = 1891$, $p = 0.00$, $d = 0.94$; and life-like AVA: $V = 2926$, $p = 0.00$, $d = 0.99$).

In addition, the test did not show any significant differences in transfer between the groups, χ^2 (2, $N = 200$) = 0.85, $p = 0.65$. Increasing the AVA's visual complexity did not lower transfer score. The students' transfer scores also improved after viewing the presentation. The Wilcoxon signed rank test between post-test and pretest transfer scores in each group revealed a significant increase in transfer (flat: $V = 960$, $p = 0.00$, $d = 0.56$; cartoon: $V = 1232$, $p = 0.00$, $d = 0.61$; and lifelike AVA: $V = 1900$, $p = 0.00$, $d = = 0.65$). The VR presentation thus resulted in substantial amounts of immediate learning.

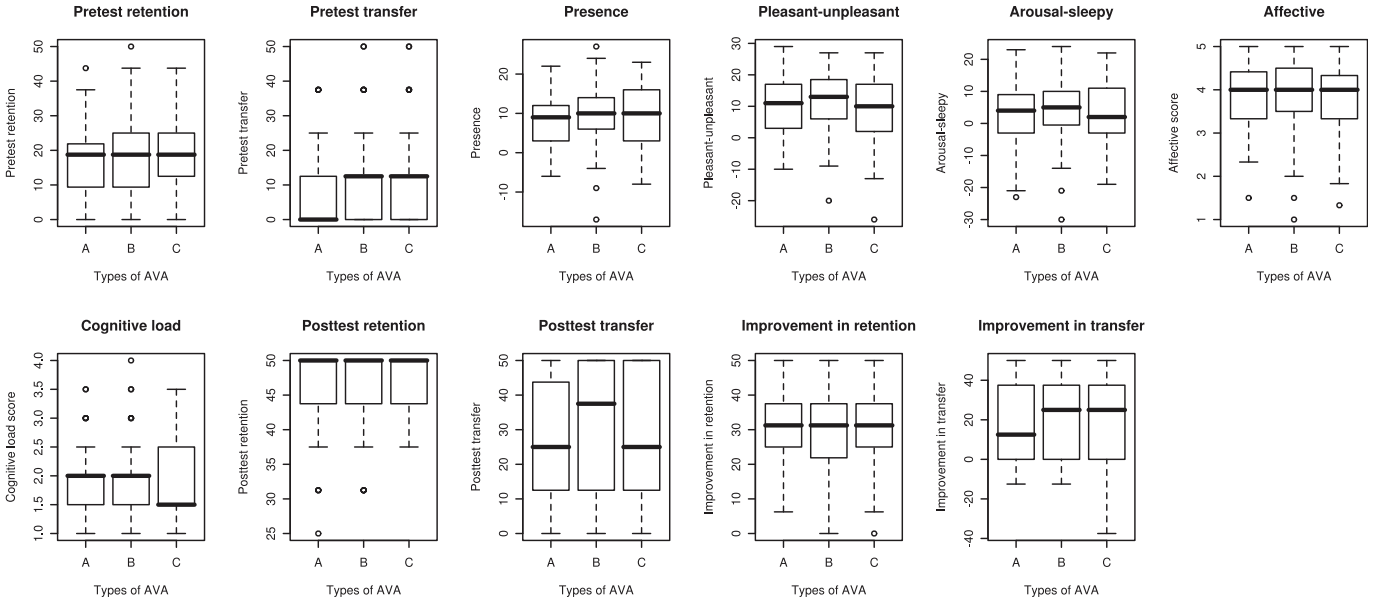


Fig. 3. Box plots of dependent variables. Types A, B and C AVAs refer to flat, cartoon and lifelike AVAs, respectively. The box represents the interquartile range (IQR) of the data. The upper hinge of the box represents the 3rd or upper quartile, and the lower hinge of the box represents the 1st or lower quartile. The thick line in the box shows the median value. The whiskers show the lower and upper values of 1.5 × lower quartile and 1.5 × upper quartile. Lastly, the white circles indicate the outliers in the data.

Table 2
Medians (Mdn) and interquartile range (IQR) of dependent variables.

Dependent measures	Group					
	Flat ($N = 60$)		Cartoon ($N = 63$)		Lifelike ($N = 77$)	
	Mdn	IQR	Mdn	IQR	Mdn	IQR
Pretest retention	18.75	10.94–20.31	18.75	9.38–25.00	18.75	12.50–25.00
Pretest transfer	0.00	0.00–12.50	12.50	0.00–12.50	12.50	0.00–12.50
Presence	9.00	3.00–12.00	10.00	6.00–14.00	10.00	3.00–16.00
Pleasant–unpleasant	11.00	3.00–17.00	13.00	6.00–14.00	10.00	2.00–17.00
Arousal–sleepy	4.00	−3.00–9.00	5.00	−0.50–10.00	2.00	−3.00–11.00
Affective	4.00	3.33–4.37	4.00	3.50–4.50	4.00	3.33–4.33
Cognitive load	2.00	1.50–2.00	2.00	1.50–2.00	1.50	1.50–2.50
Posttest retention	50.00	43.75–50.00	50.00	43.75–50.00	50.00	43.75–50.00
Posttest transfer	25.00	12.50–40.62	37.50	12.50–50.00	25.00	12.50–50.00
Increase in retention	31.25	25.00–37.50	28.17	21.88–37.50	31.25	25.00–37.50
Increase in transfer	12.50	0.00–37.50	25.00	0.00–37.50	25.00	0.00–37.50

Note. The score ranged from zero to 50 for retention and transfer test[s]; −30 to 30 for presence;-35 to 35 for arousal–sleepy and pleasant–unpleasant; 1–5 for affective.

Question 3: Does the sense of presence and perceived affective quality affect the learning outcomes? Firstly, we questioned if presence is related to perceived affective quality. A Spearman rank correlation test, as shown in Table 3, revealed moderate and significant correlations between presence and arousal–sleepy quality, r_s (200) = 0.50, $p < 0.001$, and between presence and pleasant–unpleasant quality, r_s (200) = 0.55, $p < 0.001$. With regard to learning outcomes, only perception of learning is moderately correlated with presence, r_s (200) = 0.59, $p < 0.001$, the arousal–sleepy scale, r_s (200) = 0.40, $p < 0.001$, and with the pleasant–unpleasant scale, r_s (200) = 0.49, $p < 0.001$. However, none of the dependent measures correlates significantly with perceived difficulty, or with retention or transfer scores.

4.1. Other results

We found gender effects in perceived affective quality. The Wilcoxon rank sum test was performed, using gender as the factor, and arousal–sleepy scores as the dependent variable. The analysis revealed that male participants felt more sleepy after viewing the VR simulation than did female participants, $W = 5146.5$, $p = 0.008$, $d = 0.24$. There was no significant gender difference in presence score, $W = 4796.5$, $p = 0.09$, $d = 0.15$, or in the pleasant–unpleasant score, $W = 4281$, $p = 0.75$, $d = 0.02$.

5. Discussion and conclusion

Recent development in computer graphics allows VR developers to create complex visual materials for learning. However, would it affect learning adversely? Every VR developer must consider this question in building a learning simulation. This study tested the effects of the visual complexity of AVAs on learning. Complex visual materials have been shown to hinder learning in novice learners (Mayer, 2001, 2003). Simultaneously, complex visual materials may increase presence and perceived affective quality, which promote an optimal experience or flow, for better learning outcomes. This study presented three different types of AVAs to the students, teaching them the navigational strategies of *Cataglyphis* ants.

Our findings contradict most of our predictions. The perception of learning and the perceived difficulty did not increase with the AVA's visual complexity. In addition, the retention and transfer scores did not decrease with the AVA's visual complexity. In contrast to the previous studies in CTML (Harp & Mayer, 1997, 1998; Mayer et al., 2001), the extraneous details of our AVAs did not detract from learning. The students showed a high perception of learning, below-average difficulty ratings, and significant improvements in both retention and transfer tests after viewing the presentation. Also, our findings do not show any increase in the sense of presence or perceived affective quality with increasing AVA visual complexity. In fact, the scores are moderate (Table 2) in all visual conditions. There is no correlation between presence and affective quality on the one hand, and learning outcomes on the other hand. Our findings are, however, in line with the findings of Regenbrecht et al. (1998) and Zhang and Li (2005) showing that even the least amount of visual complexity of the virtual world is enough to promote moderate levels of presence and affective quality in VR applications, which correlates with motivation, enjoyment, and ease of use of a learning material. At the same time, the results do not support the notion that the increase or decrease of motivation and affect may affect the cognitive engagement in retention or transfer outcomes (Brand et al., 2007; Pintrich, 2003). Those who reported low presence and low perceived affective quality performed just as well in retention and transfer tests as those who reported high presence and high perceived affective quality.

We can think of several explanations for our results that are not mutually exclusive. One possibility is that an increase in the visual complexity of AVAs does not add much complexity to the presentation as a whole, not enough to affect flow and learning outcomes. We only manipulated the AVAs' complexity. A virtual world, however, consists of the AVAs and their habitat, that is, the rest of the virtual environment. The virtual environment contains 3D objects besides the AVAs. For example, the virtual airport in the study by Richards et al. (2009) contains: AVAs, aeroplanes, desks, floor reflections, clouds, and many other visual objects to depict an airport. Our complex AVAs moved around in a simple setting, filled with features that are relevant to the lesson. In contrast, the manipulations effected in previous studies increased the visual complexity of the entire scene, making the virtual world as a whole complex and often cluttered with irrelevant features (Harp & Mayer, 1997, 1998; Mayer et al., 2001). Our scenes with the complex AVAs do not match the complexity of the scenes used by Roberts (2007, Annex B, 3213, 3227) or the rich learning content used by Liu et al. (2009). As a result, increasing our AVAs' visual complexity (Table 1) might not be enough to increase the overall visual complexity of the virtual world, to induce a stronger sense of presence, to increase affective quality, or to affect learning outcomes.

Table 3
Spearman's (r_s) correlation of dependent variables.

	Presence	Arousal–sleepy	Pleasant–unpleasant	Affective	Cognitive load	Increase in retention
Arousal–sleepy	0.5***					
Pleasant–unpleasant	0.55***	0.44***				
Affective	0.59***	0.40***	0.49***			
Cognitive load	−0.16*	−0.12	−0.14*	−0.07		
Increase in retention	−0.13	−0.07	−0.14	−0.14	0.07	
Increase in transfer	0.01	−0.01	−0.02	0.01	−0.04	0.04

Note. *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

A second possible explanation is that our AVAs also did not appear in every scene, unlike earlier studies (Dunsworth & Atkinson, 2007; Gulz & Haake, 2006; Holmes, 2007; Kavakli et al., 2007; Lester et al., 1997; Moreno & Flowerday, 2006; Richards et al., 2009; Wagner et al., 2006). At times, they were small because the viewpoint of the camera was distant. It remains to be seen if increased exposure to even more complexity in AVAs might affect learning adversely.

A third possibility is that an AVA may be somehow 'chunked' or independently processed as a unit, so that increasing the internal complexity of an AVA adds little cognitive load that might detract from learning. Such a hypothesis remains to be tested.

Finally, we used a slow camera movement to minimise Simulator Sickness (SS) and succeeded (Kartiko et al., 2009). The slow camera movement also slowed the scenes down, and might have allowed the students to better process the materials presented before continuing to the next scene, thus mitigating any effects of AVA complexity. This hypothesis too deserves to be investigated.

In conclusion, this study assessed the impact of the visual complexity of AVAs on presence, affective quality and their correlation with learning outcomes in VR applications. Results showed that in our VR simulation, the visual complexity of AVAs did not play a significant role at all. The VR simulation overall induced moderate presence and pleasant quality (Table 2) with all levels of the AVA's visual complexity. We have developed a pleasing and enjoyable VR experience that is useful for future educational use.

The results also have important implications for artists and designers. Sometimes, it requires a much smaller amount of work for creating simple AVAs that can produce the same experiential outcomes as complex AVAs in VR applications. In our case, it took less than 120 s to create the flat AVAs that achieved the same presence, pleasant quality scores, and retention and transfer scores as the lifelike AVAs, which took much longer to make. Even if we want presence and pleasant quality in a VR simulation, a simple AVA — flat and animation-less — may suffice for an enjoyable VR experience.

We have the privilege of the latest technology in computer graphics for creating complex AVAs, but simple AVAs can work as well. We are building another VR simulation, VEAR (VEhicle Accident and fiRes), to train future firefighters and to extend our findings. Our plans are to manipulate the complexity of entire scenes in this work.

Acknowledgements

This project has been sponsored by The Australian Research Council Discovery Grant titled Risk Management Using Agent Based Systems (DP0558852) to Debbie Richards, Manolya Kavakli, and Mark Dras. Special thanks to John Porte for the support in troubleshooting, and running the VR system. We are also grateful to John Porte, Mine Cicek, Susan Bruck and the anonymous reviewers for comments on earlier drafts of this manuscript, to Eric Fassbender and Meredith Taylor for helping to make the audio recording, and to Noah Beamer for maintaining Blender's Cal3D exporter. Some portions of the data were presented at the 9th Conference of the Australasian Society for Cognitive Science 2009 (ASCS 2009).

Abbreviations

AVA: Animated-Virtual Actor
CTML: Cognitive Theory of Multimedia Learning
PQ: Presence Questionnaire
PR: Program Ratings
PAQ: Perceived Affective Quality
SS: Simulator Sickness
SSQ: Simulator Sickness Questionnaire
VR: Virtual Reality
VRET: Virtual Reality Exposure Therapy

Appendix. Questionnaire items

Presence Questionnaire (PQ)

(1) "How completely were all of your senses engaged?"
(2) "How much did the visual aspects of the environment involve you?"
(3) "How aware were you of events occurring in the real world around you?"
(4) "How inconsistent or disconnected was the information coming from your various senses?"
(5) "How compelling was your sense of moving around inside the virtual environment?"
(6) "To what degree did you feel confused or disoriented at the beginning of breaks or at the end of the experimental session?"
(7) "How involved were you in the virtual environment experience?"

(8) "How quickly did you adjust to the virtual environment experience?"
(9) "How much did the visual display quality interfere or distract you from the training experience?"
(10) "Were you involved in the experimental task to the extent that you lost track of time?"

Perceived affective quality (PAQ)

Intense, inactive, pleasant, dissatisfying, arousing, drowsy, nice, displeasing, active, idle, pleasing, repulsive, alive, lazy, pretty, unpleasant, forceful, slow, beautiful, and uncomfortable.

Program ratings (PR)

(1) "How interesting was it to learn about *Cataglyphis* ants today";
(2) "How entertaining was it to learn about *Cataglyphis* ants today?"
(3) "How eager would you be to learn about a different animal behaviour topic in the same conditions you learned today?"
(4) "How motivating was it to learn about *Cataglyphis* ants today?"
(5) "How much did the VR learning system help you understand about navigation of *Cataglyphis* ants?"
(6) "How helpful was the VR simulation for learning about navigation of *Cataglyphis* ants?"
(7) "How difficult was it to learn about navigation of *Cataglyphis* ants with the VR learning system?"
(8) "How much effort did you have to invest to learn about navigation of *Cataglyphis* ants with the VR learning system?"

Retention and transfer (post) tests

Post_expQuestionnaires.pdf

Appendix. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compedu.2010.03.019.

References

Baddeley, A. D. (1998). *Human memory: Theory and practice*. Needham Heights, MA, revised edition: Allyn and Bacon.
Blender Foundation. (2009). Blender. Available from. http://www.blender.org/ Accessed on November 2009.
Brand, S., Reimer, T., & Opwis, K. (2007). How do we learn in a negative mood? Effects of a negative mood on transfer and learning. *Learning and Instruction, 17*(1), 1–16.
Cliff, N. (1993). Dominance statistics: ordinal analyses to answer ordinal questions. *Psychological Bulletin, 114*(3), 494–509.
Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: BasicBooks.
Csikszentmihalyi, M. (1997). *Finding flow: The psychology of engagement with everyday life*. New York: Harper and Row.
Dewey, J. (1913). *Interest and effort in education*. Cambridge Boston, MA: Houghton Mifflin. 1913.
Donderi, D. C. (2006). Visual complexity: a review. *Psychological Bulletin, 132*(1), 73–97.
Dunsworth, Q., & Atkinson, R. K. (2007). Fostering multimedia learning of science: exploring the role of an animated agent's image. *Computers & Education, 49*(3), 677–690.
E-on software. (2006). Vue 6 esprit: the art of natural 3D!. Available from. http://www.e-onsoftware.com/ Accessed on November 2009.
Gimp Development Team. (2009). GIMP – GNU image manipulation program. Available from. http://www.gimp.org/ Accessed on November 2009.
Gulz, A., & Haake, M. (2006). Design of animated pedagogical agents—a look at their look. *International Journal of Human-Computer Studies, 64*(4), 322–339.
Harp, S. F., & Mayer, R. E. (1997). The role of interest in learning from scientific text and illustrations: on the distinction between emotional interest and cognitive interest. *Journal of Educational Psychology, 89*(1), 92–102.
Harp, S. F., & Mayer, R. E. (1998). How seductive details do their damage: a theory of cognitive interest in science learning. *Journal of Educational Psychology, 90*(3), 414–434.
Hoffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: a meta-analysis. *Learning and Instruction, 17*(6), 722–738.
Holmes, J. (2007). Designing agents to support learning by explaining. *Computers & Education, 48*(4), 523–547.
Inkscape Development Team. (2009). Inkscape – open source scalable vector graphics editor. Available from. http://www.inkscape.org/ Accessed on November 2009.
Johnson, W. L., & Rickel, J. (1997). Steve: an animated pedagogical agent for procedural training in virtual environments. *SIGART Bulletin, 8*(1–4), 16–21.
Kartiko, I., & Kavakli, M. (2008). *Virtual reality police training how much visual information is too much for knowledge transfer?*. 2nd International Conference on Cybercrime Forensics Education & Training Canterbury, United Kingdom: Canterbury Christ Church University. 1–2 September.
Kartiko, I., Kavakli, M., & Cheng, K. (2009). The impacts of animated-virtual actors' visual complexity and simulator sickness in virtual reality applications. In *Proceedings of the 2009 6th international conference computer graphics, imaging and visualization* (pp. 147–152). IEEE Computer Society.
Kavakli, M., Gero, J. S., et al. (2003). Strategic knowledge differences between an expert and a novice designer. In U. Lindemann (Ed.), *Human behaviour in design* (pp. 42–51). Springer, ISBN 3-540-40632-8.
Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). *Riskman: A multi-agent system for risk management*. Architectural Design of Multi-Agent Systems. Information Science Reference.
Kennedy, R. S., Lane, N. E., Bream, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology, 3*(3), 203–220.
Krijn, M., Emmelkamp, P. M. G., Biemond, R., de Wilde de Ligny, C., Schuemie, M. J., & van der Mast, C. A. P. G. (2004). Treatment of acrophobia in virtual reality: the role of immersion and presence. *Behaviour Research and Therapy, 42*(2), 229–239.
Lee, B.-C., Yoon, J.-O., & Lee, I. (2009). Learners' acceptance of e-learning in South Korea: theories and results. *Computers & Education, 53*(4), 1–44.
Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., & Bhogal, R. S. (1997). The persona effect: affective impact of animated pedagogical agents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 359–366). ACM Press.
Lester, J. C., Stone, B. A., & Stelling, G. D. (1999). Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and User-Adapted Interaction, 9*(1), 1–44.
Lin, J. J.-W., Duh, H. B., Parker, D. E., Abi-Rached, H., & Furness, T. A. (2002). Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Proceedings of the IEEE virtual reality 2002 (VR'02)*.
Liu, S.-H., Liao, H.-L., & Pratt, J. A. (2009). Impact of media richness and flow on e-learning technology acceptance. *Computers & Education, 52*(3), 599–607.
Lusk, M. M., & Atkinson, R. K. (2007). Animated pedagogical agents: does their degree of embodiment impact learning from static or animated worked examples? *Applied Cognitive Psychology, 21*(6), 747–764.
Mayer, R. E. (2001). *Multimedia learning*. Cambridge Press.
Mayer, R. E. (2003). The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction, 13*(2), 125–139.
Mayer, R. E., Heiser, J., & Lonn, S. (2001). Cognitive constraints on multimedia learning: when presenting more material results in less understanding. *Journal of Educational Psychology, 93*(1), 187–198.

Mikropoulos, T. (2006). Presence: a unique characteristic in educational virtual environments. *Virtual Reality, 10*(3), 197–206.
Mikropoulos, T. A., & Strouboulis, V. (2004). Factors that influence presence in educational virtual environments. *Cyber Psychology & Behavior, 7*(5), 582–591.
Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*(2), 81–97.
Moreno, R. (2007). Optimising learning from animations by minimising cognitive load: cognitive and affective consequences of signalling and segmentation methods. *Applied Cognitive Psychology, 21*(6), 765–781.
Moreno, R., & Flowerday, T. (2006). Student's choice of animated pedagogical agents in science learning: a test of the similarity-attraction hypothesis on gender and ethnicity. *Contemporary Educational Psychology, 31*(2), 186–207.
Moreno, R., & Mayer, R. E. (2002). Learning science in virtual reality multimedia environments: role of methods and media. *Journal of Educational Psychology, 94*(3), 598–610.
Novak, T. P., Hoffman, D. L., & Yung, Y.-F. (2000). Measuring the customer experience in online environments: a structural modeling approach. *Marketing Science, 19*(1), 22–42.
Nowak, K. L., & Rauh, C. (2008). Choose your "buddy icon" carefully: the influence of avatar androgyny, anthropomorphism and credibility in online interactions. *Computers in Human Behavior, 24*(4), 1473–1493.
Paivio, A. (1986). *Mental representations: A dual coding approach*. Oxford, England: Oxford University Press.
Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., et al. (2004). Affective learning – a manifesto. *BT Technology Journal, 22*(4), 253–269.
Pintrich, P. R. (2003). *Motivation and classroom learning*. New York: Wiley.
R Development Core Team. (2008). *R: A language and environment for statistical computing*. Available from. Vienna, Austria: R Foundation for Statistical Computing. http://www.r-project.org/ Accessed on November 2009.
Regenbrecht, H. T., Schubert, T. W., & Friedmann, F. (1998). Measuring the sense of presence and its relations to fear of heights in virtual environments. *International Journal of Human-Computer Interaction, 10*(3), 233–249.
Richards, D., Dras, M., Porte, J., & Meredith, T. (2009). *The role, measurement and delivery of immersion in training environments*. Adelaide, Australia: SimTecT 2009 Simulation Conference: Simulation – Concepts, Capability and Technology.
Roberts, M. N. (2007). Complexity and aesthetic preference for diverse visual stimuli, PhD thesis, Departament de Psicologia, Universitat de les Illes Balears.
Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological Review, 110*(1), 145–172.
Russell, J. A., & Pratt, G. (1980). A description of the affective quality attributed to environments. *Journal of Personality and Social Psychology, 38*(2), 311–322.
Sadowski, W., & Stanney, K. M. (2002). *Handbook of virtual environments: Design, implementation, and applications*. Lawrence Erlbaum Associates, Inc. chapter Presence in virtual environments.
Sanchez-Franco, M. J. (2010). WebCT – the quasimoderating effect of perceived affective quality on an extending technology acceptance model. *Computers & Education, 54*(1), 37–46.
Stanney, K. M., Kingdon, K. S., Graeber, D., & Kennedy, R. S. (2002). Human performance in immersive virtual environments: effects of exposure duration, user control, and scene complexity. *Human Performance, 15*(4), 339–366.
Sweller, J. (1999). *Instructional design in technical areas*. Camberwell, Australia: ACER.
Takatalo, J., Nyman, G., & Laaksonen, L. (2008). Components of human experience in virtual environments. *Computers in Human Behaviour, 24*(1), 1–15.
Wagner, D., Billinghurst, M., & Schmalstieg, D. (2006). How real should virtual characters be?. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*.
Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural and Behavioural Physiology, 189*(8), 579–588.
Whitelock, D., Romano, D., Jelfs, A., & Brna, P. (2000). Perfect presence: what does this mean for the design of virtual learning environments? *Education and Information Technologies, 5*(4), 277–289.
Witmer, B. G., & Singer, M. J. (1998). Measuring presence in virtual environments: a presence questionnaire. *Presence: Teleopeator and Virtual Environment, 7*(3), 225–240.
Wittlinger, M., Wehner, R., & Wolf, H. (2007). The desert ant odometer: a stride integrator that accounts for stride length and walking speed. *Journal of Experimental Biology, 210*(2), 198–207.
Wittrock, M. C. (1989). Generative process of comprehension. *Educational Psychologist, 24*(4), 345–376.
Worldviz. (2009). Vizard: virtual reality toolkit. Available from. http://www.worldviz.com/ Accessed on November 2009.
Zhang, P., & Li, N. (2005). The importance of affective quality. *Communications of the ACM, 48*(9), 105–108.
Zollikofer, C. P. E. (1994). Stepping patterns in ants: influence of speed and curvature. *Journal of Experimental Biology, 192*(1), 95–106.

Chapter 9

Correlations between Simulator Sickness and Visual Complexity of Animated-Virtual Actors and Virtual Environments

Correlations between Simulator Sickness and Visual Complexity of Animated-Virtual Actors and Virtual Environments

Iwan Kartiko[1], Manolya Kavakli[1], Ken Cheng[2]

VISOR Research Group, Dept. of Computing[1]

Dept. of Biological Sciences[2]

Macquarie University

North Ryde, Sydney, Australia

Tel. (+61)02-9850-{9081, 9572, 8613} Fax. (+61)02-9850-{9551[1]/9231[2]}

email: {iwan.kartiko, manolya.kavakli, ken.cheng}@mq.edu.au

Abstract

This study examined the effects on Simulator Sickness (SS) due to the visual complexity of Animated-Virtual Actors (AVAs) and the Virtual Environment (VE) in Virtual Reality (VR) simulations. Visually complex stimuli may be attractive, but studies on

SS show a possibility of significant health risk outweighing the expected benefit in Virtual Reality (VR) simulations. We used a series of simulations to teach the basic skill of Village Firefighting in handling fire in car accidents. The participants learned with simple (simple AVAs and a simple VE), simple VE (lifelike AVAs and a simple VE), simple AVAs (simple AVAs and a lifelike VE) and lifelike (lifelike AVAs and a lifelike VE) experimental conditions. We predicted that: (1) SS ratings would increase with VR simulation's visual complexity and (2) Simpler VEs would compensate for the effects of visually complex AVAs. The results contradicted our predictions with no significant effects of AVA or VE complexity on simulation sickness. We discuss possible explanations and future research directions to design safe VR simulations.

Keywords: Animated-Virtual Actors, Virtual Environment, Simulator sickness, Virtual Reality, Virtual Scene, Visual Complexity, Visualisation, Vection.

Introduction

Animated Virtual Actors (AVAs) are two or three-dimensional entities in a Virtual Environment (VE) of a Virtual Reality (VR) application [1]. The task of developing, implementing and deploying AVAs in VR applications is becoming easier with advanced 3D modelling tools [2, 3, 4] and 3D engines [5, 6, 7, 8, 9], of which many are freely available.

Similar to the application of VR simulations, the application of AVAs and the VE is often technologically or market-driven, without following proper ergonomic guidelines and the technology usually outstrips the studies investigating it [10, 11]. In the domain of education and training with multimedia systems such as VR simulation, Moreno and Mayer [12] present two conflicting hypotheses: media-affect-learning and method-affects-learning. The media-affect-learning hypothesis states that better technology in delivering learning content promotes learning. On the other hand, the method-affect-learning hypothesis argues that as long as the methods promote cognitive learning, the medium does not matter [13].

In examining both the media-affect-learning and the method-affect-learning hypotheses, Moreno and Mayer [12] compared the impact of different levels of immersion in VR devices and personalised narration. The experimental results with college students indicated that the use of standard desktop displays vs. head mounted displays (HMD) did not affect the level of learning, but those who learnt with personalised narration performed better than those who learnt with non-personalised narration. Moreno [14] analysed the impacts of different levels of immersion in various VR devices and the method of delivery. The experimental results

with college students also revealed that those who learned with Head Mounted Displays were more motivated but this did not contribute to learning outcomes. However, students performed better when they learned with narration rather than on-screen text. From these empirical results, Moreno and Mayer [12] concluded that as long as the learning material promotes cognitive processing, the medium of delivery does not seem to matter. Mayer [15, p.8] and Schnotz and Bannert [16] also added that research on comparing media is largely fruitless as methods to promote cognitive processing from one medium can be applied to another. Furthermore, Choi and Clark [17] considered that AVAs as a medium of delivering information can be replaced by other means. Although AVAs can be replaced with text or narration on screen, the presence of AVAs is necessary in certain scenarios for learning to take place (e.g., in the interpersonal skills education of medical student from observing the behavioural cues of the virtual patients, a program developed by Johnsen et al. [18]).

VR simulations offer advantages that would be difficult to duplicate by using traditional teaching methods or media, advantages such as presenting time-based phenomena that are not readily observable in real life or too dangerous and costly to simulate in real life. Although the authors as VR-based education and training developers incline towards the method-affect-learning hypothesis, the visual complexity in VR simulations is known to induce Simulator Sickness (SS). The literature agrees that an effective VR simulation for learning must facilitate memorisation and more importantly, problem solving or transfer of knowledge to another scenario [15, 13, 19]. These outcomes cannot be achieved when the learning material promotes SS. Therefore, the impact of the visual complexity of AVAs and

VEs must be examined. This study extends the previous investigations by Kartiko et al. [20] by posing the following questions:

- Is SS correlated with the AVA's visual complexity?
- Does compensating for visually complex AVAs with simpler VEs lower SS?

In an attempt to examine the impact of digital contents' visual complexity on SS, this study employed a recently-built non-interactive VR simulation called VEAR (VEhicle Accident and Fire) as shown in Figure 1. VEAR showed the hazards for the firefighter in handling car accidents and the appropriate actions to ensure safety. This study used a standardised Simulator Sickness Questionnaire [21, SSQ] to obtain a subjective rating of SS. VEAR provided more complex visual stimuli, continuous camera movements, and consistent AVAs throughout the scene, compared to the educational VR scenario that we developed in a previous study [20].

Simulator Sickness

Simulator Sickness is a feeling of discomfort similar to motion sickness [21]. It may begin with a mild dizziness, worsen to sensations of nausea, and may lead to vomiting. In VR simulations, the continuous stream of visual stimuli creates vection or self-induced motion, which can lead to SS [22, 23]. In most VR applications the user is stationary and the illusion of movement is caused by the movement of the virtual camera. When the brain perceives the inconsistency between the sense of self-motion derived from visual stimuli and the lack

of self-motion derived from the body senses (vestibular, proprioceptive, and kinaesthetic stimuli), it can trigger the symptoms of SS. Similar to real-world motion sickness, SS has devastating effects, such as nausea, headache and vomiting [21, 24, 25]. Various factors might affect whether and how much SS is induced by visual stimuli, including the field of view [26], display size and movement [27], viewing angle [28], length of exposure [29, 30], and the visual complexity of the scene [31, 32].

Humans derive a sense of position by integrating various spatial cues in the environment [33], including visual motion cues such as those found in a VR simulation. More visual complexity in a scene, such as realistic looking roads, dirt and trees, would increase optic flow and heighten the visual perception of self-motion [34, 31, 23]. Thus, reducing visual complexity or details in a virtual scene may decrease SS.

Visual Complexity of Animations and SS

Most studies in VR-based instructional animation do not use a moving or non-linear camera that navigates in a VE. In addition, the visual appearance does not follow the trend in the entertainment industry, and thus previous studies have found low levels of SS [35, 36]. A VR simulation with the a non-linear virtual camera in a visually complex VE with many elements and extraneous details, however, may result in high ratings of SS [31, 32].

In one study, Kavakli et al. [32] examined the impact of environmental details on SS in a virtual rollercoaster. The participants who were exposed to a complex scene with myriads

of details in the textures, such as signs, tiles, windows, doors, surface imperfections, glare on windows, and detailed roads, experienced higher SS than the group exposed to a simple black and white scene. The path of the roller coaster ride was otherwise equivalent in the two groups. Similarly, So et al. [31] examined the impact of spatial velocity and visual complexity by varying both factors. Their results also showed that scenes with textures and additional objects increase SS rating. Kartiko et al. [20] found that this was not the case. The participants were subjected to flat, cartoon and lifelike AVAs in ANTECATALYST simulation that taught the navigational behaviour of *Cataglyphis* ants. Their results showed that AVAs' visual complexity does not contribute to SS. They argued that the negative outcome might be due to the absence of AVAs in some scenes, minimal camera movements and the insufficient level of visual complexity of the AVAs and VE. In light of these studies, we predicted that the increase of the visual complexity of AVAs and VEs will increase overall complexity of the scene, which will in turn increase vection and produce stronger SS ratings.

VEAR (VEhicle Accident and Fire) was designed with a constantly moving and non-linear camera movement that travels from one location to another. It allowed us to vary both the AVAs' visual complexity and the VE's visual complexity. Figure 1 shows the different learning conditions in VEAR. Based on the theories and literature on SS, we predicted that:

- SS ratings will follow the order of the VR simulation's complexity: simple, simple VE, simple AVAs, lifelike
- Compensating for AVAs' visual complexity with simpler VEs will lower SS

This study refers to "visual complexity" as the visual information in depicting an entity or a scene [37]. A scene may be realistic and not complex, such as a photograph of a real tomato in the middle of a white-painted room. However, a watercolour painting of a lecture theatre with 300 students in it would be far more complex and yet not necessarily realistic [38]. Similarly, imperfections on the clothing of AVAs' add visual complexity, but not necessarily pictorial realism. This study is concerned with visual complexity, not realism.

Methods

Participants and Experimental Design

The eighty-eight participants were Macquarie University students and staff as well as members of the public (43 males and 45 females) with a median age of 22 (IQR = 20 - 25). A group of 1 to 6 people participated in the experiment at the same time. Each person was randomly assigned to one computer with one of four experimental conditions differing in the AVAs' and VE's visual complexity (Figure 1). Twenty-three participants learnt with simple condition (visually simple AVAs and a visually simple VE), 21 participants with simple VE (visually complex AVAs and a simple VE), 21 participants with simple AVAs (simple AVAs and a visually complex VE), and 23 participants with lifelike visually complex AVAs and a visually complex VE.

Material and Apparatus

VEAR, was run on an Intel Core2 Duo E8400 desktop PC with nVidia GeForce GTX 260 graphic card, 4GB RAM. The VR simulation ran smoothly at maximum frame-rate. The simulation was displayed full-screen at 1680 x 1050 pixels resolution on a 473 mm x 300 mm viewable screen area. The participants were seated at approximately 60 cm away from the computer screen. Thus, the physical horizontal FOV was approximately 43°. The VEAR simulation taught the participants about the hazard in motor accidents and the best practice to approach safely and prevent the spread of fire. The learning material was developed based on Village Firefighter handbooks: Village Firefighter (VF) Manual [39, Chapter 3, section 9], Specialist Training Manual: Compressed Air Breathing Apparatus [40, CABA], and Compressed Air Breathing Apparatus Standard Operating Procedures [41].

VEAR provides four different levels of visual complexity for learning, as shown in Figure 1. The simple condition is the simplest visually, created in black and white by using simple circles and lines only. We used simple AVAs and a simple VE, such as flat and static water, flat and static fire, flat and static smoke, simple car objects with flat components, and a plain white sky. In the most visually complex lifelike condition, we used lifelike complex AVAs and a lifelike VE, including lifelike clothing with details, detailed car objects, animated water, fire and smoke, cloudy skies, reflections on cars, helmet, and glass, and complex textures to depict the rocks and roads, along with other surface imperfections in the VE. Two conditions with intermediate visual complexity were also created. The simple

VE condition contained visually complex AVAs and a simple VE. The simple AVAs condition was the converse of the simple VE condition, with simple AVAs but a visually complex VE.

The learning material was organised into five continuous scenes: 1 - Introduction; 2 - Compressed Air Breathing Apparatus (CABA); 3 - Vehicle roll and tip hazard; 4 - Explosion hazard; 5 - Hazardous material; 6 - Electrical hazard; 7 - Vehicle with no fire; 8 - Summary. The duration was 8 minutes and the camera moved from one location to another following the narration at normal speaking pace. The AVAs were visible on all scenes. We used Blender [2] for 3D modelling and animation; GIMP [42] and Inkscape [43] for texturing the VEs. The 3D engine used to display the simulation was Blender itself. Throughout the experiment, the scenario, screen resolution, camera's speed and movement, and the narration were kept the same for all conditions. The independent variable was the visual complexity of the scene, which was altered for each group. None of the students had attended the simulation before. Since there was no interactivity within the VR simulation, the participants had an equal duration of visual exposure.

This study employed the Simulator Sickness Questionnaire [21, SSQ] to assess the severity of SS in the simulation. The SSQ was administered before and after the exposure to the VR presentation. The participants were required to give a scale from 0 to 3 on each of 16 symptoms listed: General discomfort, Fatigue, Headache, Eyestrain, Difficulty focusing, Increased salivation, Sweating, Nausea, Difficulty concentrating, Fullness of head, Blurred vision, Dizzy (eyes open), Dizzy (eyes closed), Vertigo, Stomach awareness, and Burping.

The ratings were summed to obtain three sub-scores of SS: Nausea, Disorientation, and Oculomotor, which can be added for a total severity score.

Procedure

Upon the arrival in the computer lab, a group of participants was asked to fill in the consent form and each participant was randomly assigned to one computer with one of four experimental conditions. The participants were then reminded about SS and the procedure to follow if they feel sick during the simulation. SSQ ratings were collected before and after viewing the simulation.

Results

Initial assessment using the Shapiro-Wilk normality test and the Levene test for unequal variances found violations for assumptions needed for parametric statistical tests. Therefore, we employed nonparametric tests in this study. For effect sizes, we calculated Cliff's *d* values [44, 45, 46]. All statistical analysis methods were performed by using R [47]. Figure 2 shows the box plots of dependent variables and table 1 lists the medians (*Mdn*) and the interquartile range (*IQR*) of the three sub-scale measures (nausea, disorientation, and oculomotor) and the total SSQ score for each group .

Question 1: Is SS correlated with the AVA's visual complexity? To answer this question, a Kruskal-Wallis nonparametric ANOVA test was performed with experimental conditions

as the factor, and the difference in total SS severity scores (post-exposure score minus pre-exposure score), as the dependent variable. The test did not reveal any significant differences between the groups, χ^2 (3, N = 88) = 3.11, p = 0.37. The results did not support the proposed prediction and showed no increase in SS in any experimental condition.

Question 2: Does the compensation of visually complex AVAs with simpler VEs lower SS? The Kruskal- Wallis nonparametric ANOVA test did not support the prediction that lowering VE's visual complexity lowers SS rating with complex AVAs.

Discussion

Recent developments in computer technology allow digital artists and VR developers to create complex visual presentations more easily. More and more AVAs and VEs are being used in educational settings since their cost has diminished. Visual complexity is attractive, particularly to the entertainment industry whose purpose is to meet market demands [10, 48]. However, the concern is that greater visual complexity can lead to higher SS ratings. This study examined the effects of the visual complexity of AVAs and of VEs on SS. Our findings reveal that (1) there is no correlation between AVA's visual complexity and SS, and (2) the VE's visual complexity also does not contribute to SS. The first result replicates our recent study [20], which also showed that there is no correlation between the AVAs' visual complexity and SS. The results of this study revealed in addition that the visual complexity of a VE also does not correlate with SS. Overall our results contradict the findings of So et

al. [31] and Kavakli et al. [32]. Our participants reported low SS ratings at the end of the simulation.

Drawing from studies in SS, we can draw some explanations of our results. Firstly, there might be an interaction between visual complexity and camera speed. While VEAR used more camera movement than ANTECATALYST [20], the camera speed was not significantly faster than that in ANTECATALYST, and significantly slower compared to the rollercoaster ride presented by Kavakli et al. [32]. Kavakli et al. [32] had the participants in the virtual rollercoaster at higher speed in a shorter presentation of approximately 6 minutes. In 6 minutes, their study obtained a significant increase of SS. VEAR might not have generated sufficient optic flow to induce SS due to the slow camera movement. The camera movement followed the spoken narration at a normal pace. At high camera speeds, visual complexity adds to SS, but when camera speed is low enough, visual complexity has no effect, perhaps because it is below a certain SS threshold.

Secondly, VEAR was a short presentation at less than 10 minutes, whereas SS increases with exposure duration [29, 30]. Stanney and Zyda [29] showed that participants who had 45 - 60 minutes exposure gave higher SS ratings. So and Yuen [30] also revealed that longer duration of exposure led to higher SS ratings. Results from our study suggest that by combining slow camera movement with a short presentation, the visual complexity of the digital contents might not lead to adverse effects in SS. However, this issue demands further examination as computer technology progresses with time.

A third possible explanation is that the participants of this study might have been habit-

uated to viewing digital media, such as movies, games and other VR simulations [49, 50]. Habituation might lessen SS. Hill and Howarth [49] exposed 26 subjects to a hovercraft racing game called "Wipeout" for 20 minutes on 5 consecutive days and the subjects reported less nausea over days. A recent study by Bigoin et al. [50] also showed that those who were familiar (and played) computer games were not affected by SS as much as those who had little familiarity. In the present study, we have no measure of how familiar our participants were with playing computer games.

Finally, the low SS ratings might could be attributed to the lack of sudden or fast vertical camera movement and relatively s physical field of view, due to the conventional computer screen used in the study. Previous studies in SS indicated that vertical acceleration and the frequency of sinusoidal movement in a VE is significant in inducing SS [51, 52]. Kavakli et al. [32] used a large 6m x 2m semi-cylindrical projection screen with approximately 150-160 degrees of FOV to display the roller-coaster simulation. Thus, the participants who attended the rollercoaster simulation were likely to experience greater stimulus of vertical sinusoidal acceleration and frequency, which led to greater SS ratings. Again, we have no quantitative measure of these variables, and this demands further attention in future studies.

Conclusion

This study examined the correlations between SS and the visual complexity of AVAs and VEs. Similar to the previous experiments by Kartiko et al. [20], the results confirmed that

SS is not correlated with AVAs' visual complexity. Furthermore, the results also showed that SS is not correlated with the level of VEs' visual complexity. VEAR produced very low SS ratings.

We have developed a safe VR simulation for future education and training use in fire-fighting. Knowing that VR simulations might induce SS, it is neither right nor wrong to ask "do they have a better role for learning than other media?" Instead, the main concern for VR simulation developers ought to be: "how should they be designed to facilitate learning?" There are many research opportunities in improving the usability of VR simulations to improve users' performance [53, 54]. The privileges of free and open-source 3D modelling tools and 3D engines should encourage more research investigating further the applications of AVAs and VEs in VR applications to support learning [55, 56, 57]. We plan to address the aforementioned issues to examine the correlation between SS and the visual aspects of digital contents under various conditions and scenarios. Additionally, we are attempting to quantify visual complexity to facilitate comparisons between VR simulations.

Acknowledgement

This project has been sponsored by the Australian Research Council Discovery Grant, DP 0558852 (Richards, Kavakli, Dras) titled Risk Management Using Agent Based Systems. Special thanks to Matt Cabanag and Darius Taslim for their assistance in troubleshooting, and running the computers in the labs. We are also grateful to Susan Bruck for the fruitful

discussion on SS, and Geoff Plummer for the audio recording session.

References

[1] Iwan Kartiko, Manolya Kavakli, and Ken Cheng. Learning science in a virtual reality application: The impacts of animated-virtual actors? visual complexity. *Computers & Education*, 55(2):881 – 891, 2010.

[2] Blender Foundation. Blender, 2011. Last accessed on September 2011.

[3] Björn Gustavsson, Jakob Cederlund, Dan Gudmundsson, Vlad Dumitrescu, Patrik Nyblom, Anthony D'Agostino, Gregg Griffin, Chris Osgood, Howard Trickey, Anders Conradi, Sean Hinde, Danni Coy, and Mick Balaban. Wings3d, 2011. Last accessed on September 2011.

[4] Timothy M. Shead. K-3d, 2011. Last accessed on September 2011.

[5] Carnegie Mellon University. Panda3d, 2011. Last accessed on September 2011.

[6] Delta3D Group. http://www.delta3d.org/. MOVES Institute, Naval Postgraduate School, Monterey, California, USA, 2011. Last accessed on September 2011.

[7] Irrlicht Team. Irrlicht, 2011. Last accessed on September 2011.

[8] Steve Streeting. Ogre3d, 2011. Last accessed on September 2011.

[9] Jorrit Tyberghein. Crystalspace 3d, 2011. Last accessed on September 2011.

[10] Michael Zyda. From visual simulation to virtual reality to games. *Computer*, 38(9):25–32, 2005.

[11] J. M Hoc. Cognitive ergonomics: a multidisciplinary venture. *Ergonomics*, 51(1):71 – 75, 2008.

[12] Roxana Moreno and Richard E. Mayer. Personalized messages that promote science learning in virtual environments. *Journal of Educational Psychology*, 96(1):165–173, 2004.

[13] Richard E. Mayer. The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction*, 13(2):125–139, 2003.

[14] Roxana Moreno and Richard E. Mayer. Virtual reality and learning: Cognitive and motivational effects of students' sense of presence. In *the Ninth International Conference on Human-Computer Interaction. 2001*, pages 65–67, 2001.

[15] Richard E. Mayer. *Multimedia Learning*. Cambridge Press, 2001. ISBN 9780521787499.

[16] W. Schnotz and M. Bannert. Construction and interferences in learning from multiple representation. *Learning and Instruction*, 13(2):141–156, 2003.

[17] S. Choi and R.E. Clark. Cognitive and affective benefits of an animated pedagogical agent for learning english as a second language. *Journal of Educational Computing Research*, 34(4):441–466, 2006.

[18] Kyle Johnsen, Andrew Raij, Amy Stevens, D. Scott Lind, and Benjamin Lok. The validity of a virtual human experience for interpersonal skills education. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1049–1058, New York, NY, USA, 2007. ACM.

[19] K.S. Hale, K.M. Stanney, and L. Malone. Enhancing virtual environment spatial awareness training and transfer through tactile and vestibular cues. *Ergonomics*, 52(2):187 – 203, February 2009.

[20] Iwan Kartiko, Manolya Kavakli, and Ken Cheng. The impacts of animated - virtual actors' visual complexity and simulator sickness ?in virtual reality applications. In Bob Werner, editor, *Proceedings of the 6th International Conference Computer Graphics, Imaging and Visualization*, pages 147 – 152. IEEE Computer Society, 2009. ISBN 978-0-7695-3789-4.

[21] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[22] Lawrence J. Hettinger, Kevin S. Berbaum, Robert S. Kennedy, William P. Dunlap, and Margaret D. Nolan. Vection and simulator sickness. *Military Psychology*, 2:3:171–181, 1990.

[23] Bernhard E. Riecke, J. Schulte-Pelkum, Marios N. Avraamides, Markus von der Heyde, and Heinrich H. B???lthoff. Scene consistency and spatial presence increase the sensation of self-motion in virtual reality. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, pages 111–118, New York, NY, USA, 2005. ACM Press.

[24] Capt L Sinclair. Motion sickness - it's in the bag! Touchdown - Royal Australian Navy Publication, December 2003.

[25] T.N. Jones. *Warefare Officers Career Handbook*. Royal Australian Navy, September 2006.

[26] Ronald R. Mourant, Najla Ahmad, Beverly K. Jaeger, and Yingzi Lin. Optic flow and geometric field of view in a driving simulator display. *Displays*, 28(3):145–149, 2007.

[27] Hiroaki Shigemasu, Toshiya Morita, Naoyuki Matsuzaki, Takao Sato, Masamitsu Harasawa, and Kiyoharu Aizawa. Effects of physical display size and amplitude of oscillation on visually induced motion sickness. In *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 372–375, New York, NY, USA, 2006. ACM Press.

[28] Masaki Emoto, Masayuki Sugawara, and Yuji Nojiri. Viewing angle dependency of visually-induced motion sickness in viewing wide-field images by subjective and autonomic nervous indices. *Displays*, 29(2):90–99, 2008.

[29] Kay M. Stanney and Michael Zyda. *Handbook of Virtual Environment: Design, Implementation and Applications*, chapter Virtual Environments in the 21st Century, pages 1–14. Lawrence Erlbaum Associates, 2002.

[30] Richard H. Y. So and S. L. Yuen. Comparing symptoms of visually induced motion sickness among viewers of four similar virtual environments with different color. In *ICVR'07: Proceedings of the 2nd international conference on Virtual reality*, pages 386–391, Berlin, Heidelberg, 2007. Springer-Verlag.

[31] Richard H. Y. So, Andy Ho, and W. T. Lo. A metric to quantify virtual scene movement for the study of cybersickness: Definition, implementation, and verification. *Presence: Teleoperator and Virtual Environment*, 10(2):193–215, 2001.

[32] Manolya Kavakli, Iwan Kartiko, John Porte, and Nolwenn Bigoin. Effects of digital content on motion sickness in immersive virtual environments. In *Proceedings of the 3rd International Conference on Computer Science and Information Systems*, pages 1 – 12, 2007.

[33] Ken Cheng, Sara J. Shettleworth, Janellen Huttenlocher, and John J. Rieser. Bayesian integration of spatial information. *Psychological Bulletin*, 133(4):625–637, July 2007.

[34] M.E. McCauley and T.J. Sharkey. Cybersickness: perception of self-motion in virtual environments. *Presence*, 1(3):311 – 318, 1992.

[35] Debbie Richards, Mark Dras, John Porte, and Meredith Taylor. The role, measurement and delivery of immersion in training environments. SimTecT 2009 Simulation Conference: Simulation - Concepts, Capability and Technology. Adelaide, Australia, 15 - 18 June 2009.

[36] Mark Dras, Debbie Richards, Meredith Taylor, and Mary Gardiner. Generating and detecting deceptive language in virtual agents. In *The 9th International Conference on Autonomous Agents and Multiagent Systems: Workshop 19 – International Workshop on Interacting with ECAs as Virtual Characters. Toronto, Canada*, 2010.

[37] Don C. Donderi. Visual complexity: A review. *Psychological Bulletin*, 132(1):73–97, 2006.

[38] Catharine Abell. Pictorial realism. *Australasian Journal of Philosophy*, 85(1):1–17, March 2007.

[39] NSW Rural Fire Service. VF/1-6 (1999). Village Firefighter (VF) Manual, 1999. ISBN 0731097303x.

[40] NSW Rural Fire Service. Compressed Air Breathing Apparatus Standard Operating Procedures, 2001.

[41] NSW Rural Fire Service. SP/3(1995). Specialist Training Manual: Compressed Air Breathing Apparatus (CABA)., 2001. ISBN 0731064321.

[42] Gimp Development Team. Gimp, 2011. Last accessed on September 2011.

[43] Inkscape Development Team. Inkscape, 2011. Last accessed on September 2011.

[44] Norman Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509, 1993. ISSN: 0033-2909.

[45] Africa Borges del Rosal, Concepción San Luis, and Alfonso Sánchez-Bruno. Dominance statistics: A simulation study on the d statistic. *Quality and Quantity*, 37(3):303–316, 2003.

[46] Jeanine Romano, Jeffrey D. Kromrey, Jesse Coraggio, and Jeff Skowronek. Appropriate statistics for ordinal level data : Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys? Paper presented at the annual meeting of the Florida Association of Institutional Research, February 1 -3, 2006, Cocoa Beach, Florida, 2006.

[47] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[48] Marcos Nadal Roberts. *COMPLEXITY AND AESTHETIC PREFERENCE FOR DIVERSE VISUAL STIMULI*. PhD thesis, Departament de Psicologia, Universitat de les Illes Balears, 2007.

[49] K. J. Hill and P. A. Howarth. Habituation to the side effects of immersion in a virtual environments. *Displays*, 21(1):25–30, March 2000.

[50] Nolwenn Bigoin, John Porte, Iwan Kartiko, and Manolya Kavakli. Effects of depth cues on simulator sickness. In *Proceedings of the 1st International Conference on Immersive Telecommunications*, pages 1–4, 2007. ACM SIGMM Technically sponsored by Create-Net and EURASIP.

[51] J.E. Bosa and W. Bles. Modelling motion sickness and subjective vertical mismatch detailed for vertical motions. *Brain Research Bulletin*, 47(5):537–542, 1998.

[52] M. E. Maccawley. Motion sickness incidence as a function of frequency and acceleration of vertical sinusoidal motion. 1974.

[53] John R. Wilson. Virtual environments and ergonomics: needs and opportunities. *Ergonomics*, 40(10):1057 – 1077, 1997.

[54] Kay Stanney and Gavriel Salvendy. Aftereffects and sense of presence in virtual environments: Formulation of a research and development agenda. *International Journal of Human-Computer Interaction*, 10(2):135–187, June 1998.

[55] Daniel Wagner, Mark Billinghurst, and Dieter Schmalstieg. How real should virtual characters be? In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 57, California, USA, 2006. ACM Press.

[56] Yu-Wei Lin and Enrico Zini. Free/libre open source software implementation in schools: Evidence from the field and implications for the future. *Computers & Education*, 50(3):1092 – 1102, 2008.

[57] Alexis Heloir and Michael Kipp. Real-time animation of interactive agents: Specification and realization. *Applied Articial Intelligence*, 24(6):510 – 529, July 2010.

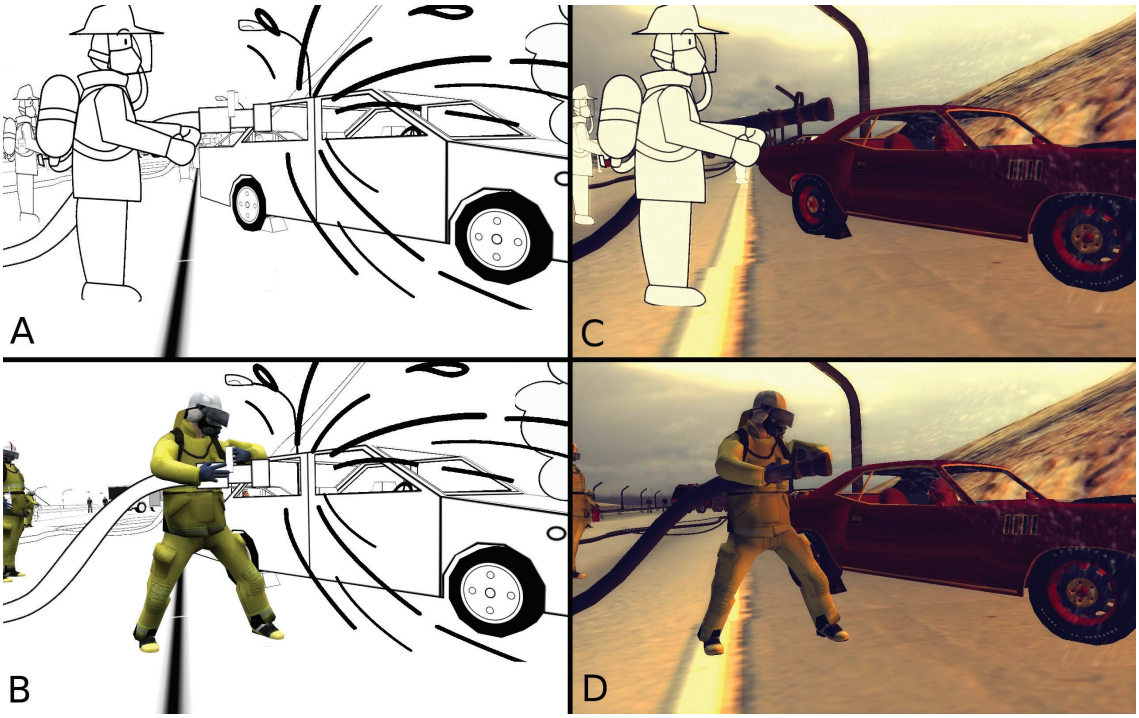


Figure 1: The four experimental conditions of VEAR. Anticlockwise from top left: (1) simple, (2) simple AVAs, (3) lifelike and (4) simple VE.

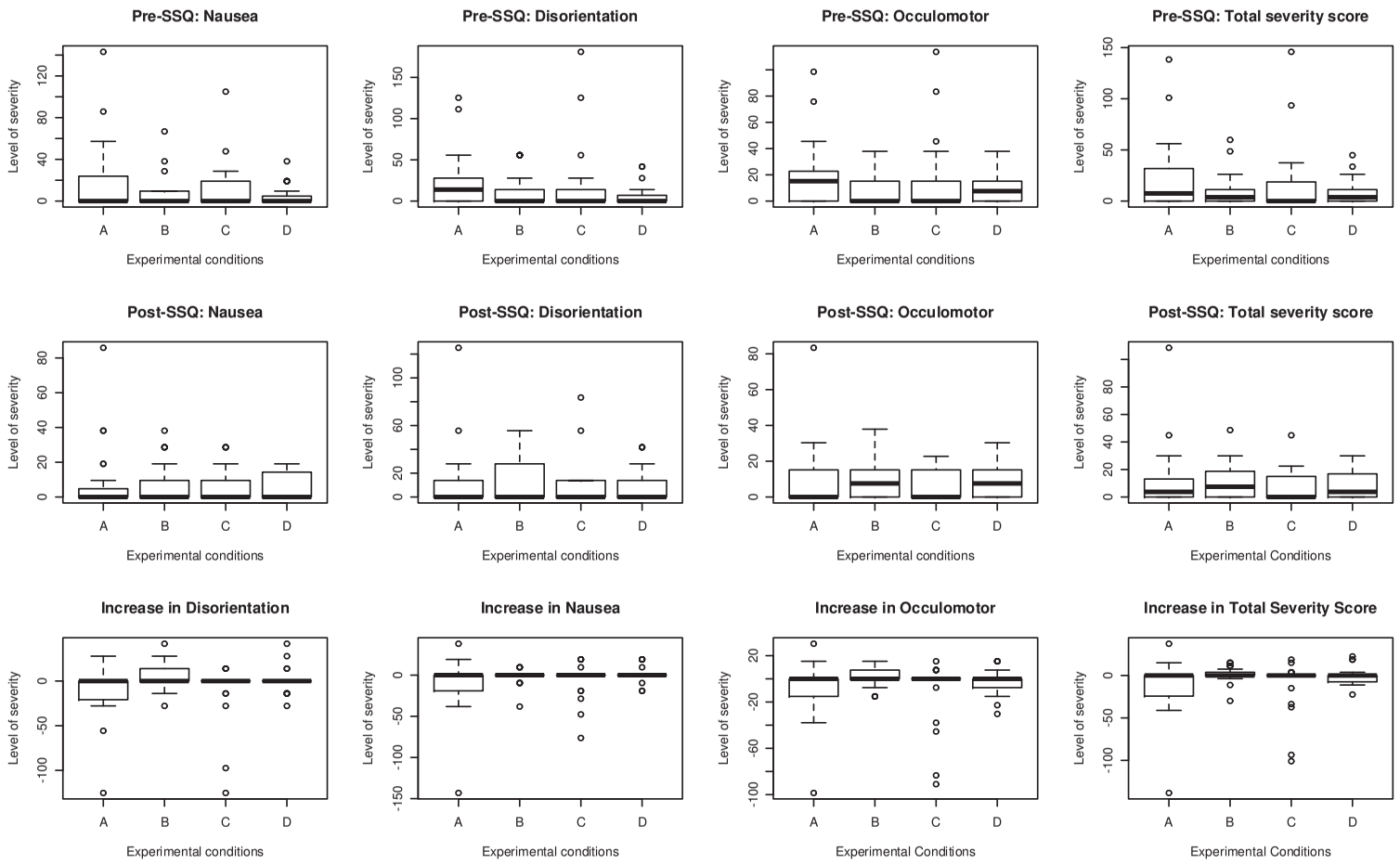


Figure 2: Box plots of the dependent variables. Experimental condition A, B, C and D refer to simple, simple VE, simple AVAs and lifelike condition respectively. The box represents the interquartile range (*IQR*) of the data. The upper hinge of the box represents the 3rd or upper quartile, and the lower hinge of the box represents the 1st or lower quartile. The bold line in the box indicates the median value. The whiskers show the lower and upper values of 1.5 x lower quartile and 1.5 x upper quartile. Lastly, the white circles indicate the outliers in the data.

Table 1: *Medians* and *IQR* scores of Dependent Variables in each groups

		Pre-SSQ				Post-SSQ			
		Nausea	Disorientation	Occulomotor	Total severity	Nausea	Disorientation	Occulomotor	Total severity
Group	*N*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*
A-Simple	23	0.00(0.00-23.85)	13.92(0.00-27.84)	15.16(0.00-22.74)	7.48(0.00-31.79)	0.00(0.00-4.77)	0.00(0.00-13.92)	0.00(0.00-15.16)	3.74(0.00-13.09)
B-Simple VE	21	0.00(0.00-9.54)	0.00(0.00-13.92)	0.00(0.00-15.16)	3.74(0.00-11.22)	0.00(0.00-9.54)	0.00(0.00-27.84)	7.58(0.00-15.16)	7.48(0.00-18.70)
C-Simple AVAs	21	0.00(0.00-19.08)	0.00(0.00-13.92)	0.00(0.00-15.16)	0.00(0.00-18.70)	0.00(0.00-9.54)	0.00(0.00-13.92)	0.00(0.00-15.16)	0.00(0.00-14.96)
D-Lifelike	23	0.00(0.00-4.77)	0.00(0.00-6.96)	7.58(0.00-15.16)	3.74(0.00-11.22)	0.00(0.00-14.31)	0.00(0.00-13.92)	7.58(0.00-15.16)	3.74(0.00-16.83)

		Increase (Post-SSQ - Pre-SSQ)			
		Nausea	Disorientation	Occulomotor	Total severity
Group	*N*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*	*Mdn(IQR)*
A-Simple	23	0.00(-19.08-0.00)	0.00(-20.88-0.00)	0.00(-15.16-0.00)	0.00(-24.31-0.00)
B-Simple VE	21	0.00(0.00-0.00)	0.00(0.00-13.92)	0.00(0.00-7.58)	0.00(0.00-3.74)
C-Simple AVAs	21	0.00(0.00-0.00)	0.00(0.00-0.00)	0.00(0.00-0.00)	0.00(0.00-0.00)
D-Lifelike	23	0.00(0.00-0.00)	0.00(0.00-0.00)	0.00(-7.58-0.00)	0.00(-7.48-0.00)

Chapter 10

The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

The Affective and Cognitive Impacts of the Animated-Virtual Actors' and the Virtual Environment's Visual Complexity in Virtual Reality-based Instructional Animations

Iwan Kartiko, Manolya Kavakli, and Ken Cheng

Abstract—There is a trend of creating visually complex scenes, which consists of Animated-Virtual Actors (AVAs) and Virtual Environments (VEs) for learning. Flow theory suggests that visually complex scenes may stimulate motivational and affective appeal to improve learning. However, the Coherence Principle, derived from Cognitive Theory of Multimedia Learning (CTML) suggests that extraneous visual details may hinder learning. This study examines the impact of the scenes' visual complexity on affective and cognitive responses for learning. The results obtained from eighty-eight participants showed that increasing the visual complexity of the scene increased the sense of presence and perceived affective quality on an arousal-sleepy scale, and lowered perceived difficulty. At the same time, it impaired retention. All participants showed high perception of learning, and importantly, significant improvements in retention in all conditions. All groups also showed significant improvements in transfer, except for those who attended with complex AVAs and VE. Implications for instructional design in using visually complex scenes for learning are discussed.

Index Terms—Animated-Virtual Actors, Cognitive learning, Coherence Principle, Emotion, Instructional Animation, Virtual Reality, Visual Complexity, Virtual Environments.

1 Introduction

Animated-Virtual Actors (AVAs) are animated two or three-dimensional (2D/3D) entities in Virtual Environments (VEs). In this article, we define VEs as virtual stages or background where scenarios take place. VEs may consist of animated objects such as windmills and waterfalls. AVAs and the VEs are found not only in the entertainment industry [1], but also in instructional animations for learning [2], [3], [4], [5], [6], [7], [8]. As the computer graphics technology progresses, there is a tendency to create more visually complex scenes in instructional animations.

Visually complex scenes are fun, pleasing, attractive and a desired commodity in the entertainment industry [3], [9], [10]. At the same time, the literature suggests that complex visual materials may increase overall animation complexity and the extraneous cognitive load for learning and this may overwhelm novices, hindering them in directing their attention to extract the relevant information [11], [12], [13]. On the other hand, lowering visual complexity to a simpler animation may "*underwhelm*" learning as the material may not be sufficiently engaging or motivating to attend to [11].

- *Iwan Kartiko and Manolya Kavakli are with the Department of Computing, Macquarie University, Sydney NSW 2109, Australia. E-mail: iwan.kartiko@mq.edu.au*
- *Ken Cheng is with the Department of Biological Sciences, Macquarie University, Sydney NSW 2109, Australia.*

In this study, we ask: (1) Does the presence and affective quality ratings follow the order of scenes' visual complexity? (2) Does an increase of scenes' visual complexity affect learning? (3) Do the sense of presence and perceived affective quality affect the learning outcomes? We examine the answers by considering the Coherence Principle, derived from Cognitive Theory of Multimedia Learning [14], [15, CTML], and the sense of presence and perceived affective quality of VR simulations within the framework of flow theory [16], [17].

We employed a recently built non-interactive firefighter VR simulation called VEAR (VEhicle Accident and Fire). VEAR showed the hazards for firefighters in car accidents and the appropriate actions to ensure rescue safety. VEAR has four visual conditions of increasing visual complexity, as shown in Figure 1, from simple to simple VE, simple AVAs and lifelike conditions. These visual conditions can be used to investigate the aforementioned questions. Similar to the previous studies by Kartiko et al. [8], [18], we refer to visual complexity as the visual information in depicting an entity or a scene [19]. This study concerns visual complexity and not visual realism [20]. A scene may not be realistic but be visually complex, such as an oil painting of a city scene with plenty of details in it, e.g., people, taxis, buses, and environmental effects. On the other hand, a scene may be realistic and not visually complex, such as a photograph of a little drop of tomato sauce on white-painted concrete. There are three visual components that alter the visual complexity of a scene in VR simulations:

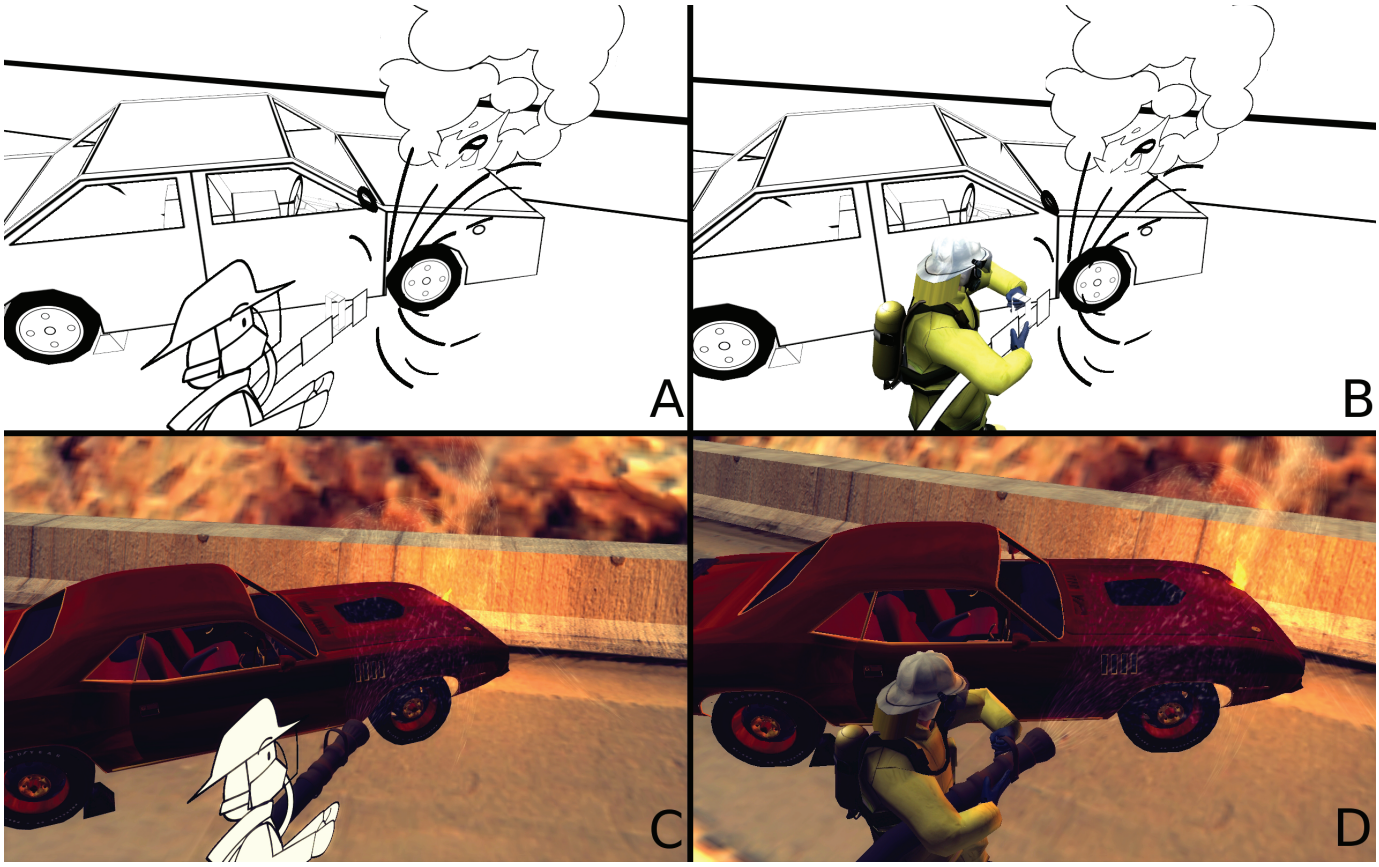


Fig. 1. The four experimental conditions of VEAR in the order of increasing visual complexity: (A) simple, n = 23, (B) simple VE, n = 21, (C) simple AVAs, n = 21, and (D) lifelike, n = 23.

the cinematography, the AVAs and the VEs [21]. In this study, we only focus on the visual complexity of the AVAs and the VEs.

1.1 VEs in Instructional Animations

Virtual Environments (VEs) are artificial spaces where actions take place in depicting a scenario. The application of VE is straightforward in comparison to the application of AVAs. In conjunction with AVAs, VE gives a finer depiction of a scene [21]. In the absence of AVAs, a VE is often sufficient in depicting a scene. For example, by implementing only the customs officers in BOSS (BOrder Security Simulation), with an empty VE, the VR simulation does not convey the notion of simulation concerned with border security. However, the presence of a VE that resembles an airport, with desks, chairs, window panels, tabletop computers, aeroplanes, and airport signs, would clearly convey that the VR simulation is about border security. Furthermore, the design of the VE must correspond to the learning material or it may hinder learning [22]. For example, having BOSS with a VE looking like an abattoir may distract the learners. The presence of VE is influential in VR simulations in depicting an accurate representation of a scenario.

1.2 AVAs in Instructional Animations

AVAs in instructional animations may represent humans, animals, and anthropomorphic entities such as the talking paperclip in Microsoft® Office suites [23], [24]. Choi and Clark [25] compared Microsoft's ® genie as the virtual actor and an arrow to give lessons about the English language. Choi and Clark [25] assert that learning occurs mainly as a function of the instructional method rather than the medium used to deliver the learning material, such as the presence and nature of the virtual actor. This assertion may be true in light of some recent studies in which AVAs were replaced by other objects [25], narration or onscreen text [3], [4], [7], [26]. Nonetheless, literature shows that there are four main arguments for using AVAs for learning, apart from the technological feasibility.

First, AVAs often depict a scene better than a description of a scene. When required, AVAs are able to portray entities with their characteristic behaviour that would be difficult to describe verbally. According to Schnotz [27], depictions are concrete and specific, useful for drawing inferences, always complete (possessing shape, size and orientation), robust against degradation (e.g., a picture of a car with a part of the wheel missing still represents a car), and easier for updating prior knowledge than using on-screen or printed text. Johnsen et al. [5] used

an AVA for interpersonal skills education where medical students must interact, learn and analyse the complex behaviour of the virtual patient. Kavakli et al. [6] and Richards et al. [28] also used AVAs in a Border Security Simulation (BOSS) to teach customs officers to identify potential risks in suspiciously behaving incoming passengers. When such detailed and visually rich cues are needed, AVAs are able to depict the notion of sickness and anxiety better than onscreen text. Furthermore, replacing these AVAs with any contextually irrelevant object, e.g., an arrow may impede learning [22], [29].

Secondly, AVAs are able to scaffold the learning process. Novice learners are often overwhelmed by complex illustrations or animations and fail to direct their attention to the relevant material [11], [12], [13], [30]. AVAs benefit learning as they are able to scaffold the learning process by guiding learners with gestures or verbal information as needed. For example, Lester et al. [2] created Herman the bug with lifelike animations to give students problem-solving guidance in designing a plant. Dunsworth and Atkinson [4] also employed an AVA known as "Dr. Bob" to guide narration with eye gaze and gestures in assisting students to learn about blood in the human circulatory system. Moreover, Hongpaisanwiwat [31] showed that the use of an AVA in multimedia learning to direct students attention to the relevant material is as effective as masking, framing and zooming techniques.

Thirdly, the application of AVAs motivates the users to attend to the lesson. If masking, framing and zooming techniques are as effective as AVAs [31], why do we still require AVAs if it can be replaced with text or other means of delivering information? It is because of AVAs' motivational factor. Gulz and Haake [32] assert that AVAs' visual and aesthetic factors are important for motivating and engaging users. Lester et al. [2, p.359] refer to this motivational factor as the *Persona Effect*: "the presence of a lifelike character in an interactive learning environment, even one that is not expressive, can have a strong positive effect on students' perception of their learning experience." AVAs are more attractive and motivating than static Virtual-Actors or text on-screen [33]. Furthermore, in the study by Wagner et al. [3], the group who learned art history with an AVA called Mr. Virtuoso rated the learning experience to be more engaging and fun than the text-only group. Moreno and Flowerday [26] also showed that when the participants were given a choice to use AVAs or not, they preferred to learn with the presence of AVAs. According to Social Agency Theory, the richer implementation of social cues in AVAs leads to greater engagement for the learning process [4], [34].

Lastly, the presence of AVAs improves retention. This is one of the expected outcomes of the motivational effects of AVAs. Moreno et al. [34] showed that the students who learned with the interactive AVA, Herman the bug, outperformed those who learned without Herman in interest ratings and transfer tests. Furthermore, a recent study by Dunsworth and Atkinson [4] showed that the presence of AVAs fosters learning by improving retention. Those who learnt with the virtual actor, known as "Dr. Bob", who coordinated the narration with gaze and pointing behaviour, scored higher in memorisation than those who learn with onscreen text or narration only.

1.3 Visual complexity and Cognitive Theory of Multimedia Learning

Given these privileges and the trend of using increasingly complex visual materials in instructional animations, the main agenda of educators is still to promote learning in retention and most importantly, the highly valued problem-solving [11], [14], [35]. Educators must examine the effects of scenes' visual complexity in promoting retention and transfer.

Mayer [14], [36] derives a set of seven design principles from Cognitive Theory of Multimedia Learning (CTML) and from empirical experiments. The fourth principle, Coherence Principle suggests that students learn better when extraneous words and pictures are excluded from learning materials. CTML provides insight into how the human mind processes streams of information in multimedia learning systems. CTML draws on the assumption that (1) there are separate channels in processing visual and verbal information [37], [38], (2) there is limited capacity of working memory in processing information [38], [39], and (3) learners must actively select relevant information, organise both visual and verbal information and integrate information with prior knowledge [40], [41]. Substantial empirical evidence suggests that the application of the design principles of CTML promotes retention and transfer for novice learners, even in passive media that are without user and social interactions [14], [15], [36], [42].

Baddeley [43] explains that sports fans are able to remember various matches with all the score details because their strong interest and passion drive them to attend to all sports-related information. Earlier, Dewey [44] in a classic educational book entitled "Interest and efforts in education" also recognised the importance of interest for learning. However, Dewey [44, p. ix] cautioned us not to treat interest as a flavouring, as this may interfere with the learning process. Working memory has an extremely limited capacity for processing information before changes in long-term knowledge take place [38], [45]. In overcoming this limitation, CTML facilitates better learning by splitting the information into separate visual and auditory channels. However, a scene with complex visual materials might overload the visual channel and increase the cognitive load. Under the burden of cognitive load, learners would find it difficult to select relevant information, to integrate both visual and auditory information and to store them in long-term memory. This cognitive load is also commonly referred to as extraneous cognitive load [39], [46], due

to extraneous elements that do not contribute to the learning process. Studies behind the Coherence Principle of CTML showed that the removal of amusing but unnecessary visual details fosters better learning [47], [48], [49].

Harp and Mayer [47] conducted a study on the effects of emotional interest and cognitive interest. They had the students learn about lightning by using static illustrations with text captions. Those who learned with myriads of extraneous and unnecessary details such as an aeroplane struck with lightning, swimmers in a pool with detailed background, a burnt tree, a golfer with lightning marks on the ground, and a footballer with a burnt jersey, found these features appealing yet scored lower on retention and transfer tests than those who attended the lesson with simple illustrations.

Similarly, Harp and Mayer [48] also showed that the students who learned with pleasing, entertaining but irrelevant visual details were interested by the visual materials but performed poorly in retention and transfer tests. Moreover, Mayer et al. [49, experiment 3 and 4] extended these findings by using simple animation and complex video clips. They added entertaining and conceptually irrelevant visual details, such as lightning flashes in the sky and clouds, a group of scientists launching rockets into the clouds, ambulance with a stretcher and a crowd of onlookers; this also resulted in poorer retention and transfer compared with those who learnt with lean and simple animation. CTML, which relies heavily on Cognitive Load Theory, favours the use of simple yet coherent visual materials. In the context of this study, CTML suggests that lowering the visual complexity of scenes would foster better retention and transfer than using visually complex scenes.

1.4 Visual complexity and Flow Theory

Humans are not like machines that simply accept and write information in memory [14]. Instead, humans are in constant interaction with the surroundings in receiving and processing information. Earlier, Lewin [50, p.25] stated that "In principle it is everywhere accepted that behaviour (B) is a function of the person (P) and the environment (E), B=f(P,E) and that P and E in this formula are interdependent variables." In VR-based educational context, learning (B) is a function of the learners (P) and the VR simulation (E) as the virtual place of learning. Furthermore, Csikszentmihalyi [16, p.74] suggested that humans are able to enter the state of optimal experience or flow that raises cognitive engagement, which leads to easier and more enjoyable tasks. Therefore, this study also examines the key factors in VR simulations (E) that may influence the user (P) and promote flow for learning (B): presence and perceived affective quality.

According to Witmer and Singer [51], presence is the subjective experience of being in one place when physically being elsewhere. Novak et al. [52] and Takatalo et al. [53] showed that presence is correlated with flow, which promotes the sense of enjoyment and motivation to attend to the VR simulations [54], [55]. Witmer and Singer [51] suggested various factors in VR simulations that may increase the sense of presence and one of them is the visual complexity or the amount of details in the virtual environment. Stanney et al. [56] also showed that increasing scene complexity by adding textures and high ceilings in the environment leads to a higher sense of presence. Thus, there is a suggestion that increasing visual complexity in AVAs and the VE could increase the overall presence of the scene.

Although Mikropoulos and Strouboulis [57] made a claim that presence increases cognitive engagement for learning, the results have been inconsistent [28], [58], [59]. Kartiko et al. [8] suggested that although the sense of presence alone is correlated with flow, measuring presence is not enough to indicate flow as the sense of presence in VR simulations could be a motivating [59], [60] or traumatising experience [61], [62]. In other words, presence questionnaires only determine the degree of someone being in another place (e.g., Did you feel like being in a room?), but it does not indicate the emotion and feeling induced by the environment (e.g., How was the room? Was it pleasant?). In sum, for flow to occur the VR simulation must stimulate the sense of presence with positive ratings of moods and emotion induced by the surrounding environment or positive perceived affective quality.

Affective quality is the ability of an environment to alter one's core affect [63], [64]. Core affect, also commonly called affect, is the state of mood or feeling, e.g., happy, sad, pleasant, etc., that occurs within a person and influences perception, cognition and behaviour [63], [65], [66], [67], [68]. How does visual complexity in VR simulation influence users' perceived affective quality? A recent study by Roberts [10] indicated that visual complexity correlates with perceived affective quality. Roberts [10] explored humans' aesthetic preference of various art works and paintings of differing visual complexities and found a correlation between the amount of elements or visual details and aesthetic preference. The participants of Roberts [10] rated a visually complex illustration with higher beauty ratings than a simple illustration. Thus, there is an indication that increasing visual complexity of the scene may increase the overall perceived affective quality.

With regard to learning, Csikszentmihalyi [17] suggested that learning performance would be hindered if motivation and emotion are against the task. A recent study by Brand et al. [68] showed that negative affect hinders learning and transfer performance on various problem-solving tasks. On the other hand, Zhang et al. [67] showed that perceived affective quality is positively correlated with cognitive absorption. According to Zhang et al. [67], cognitive absorption consists of five dimensions, temporal dissociation, focused immersion, heightened enjoyment, control, and curiosity. These dimensions serve as building blocks for perceived useful-

ness and ease of use, the important factor in technology acceptance [66], [67]. Furthermore, recent studies by Lee et al. [69] and Liu et al. [70] also showed users' acceptance of and willingness to use a learning system increase with flow.

Thus, contrary to CTML, the flow-theory approach suggests that the increase of presence and perceived affective quality, might promote better learning in instructional animations. Stevenson [71] summed it up beautifully as "Just a spoonful of sugar helps the medicine go down, in a most delightful way" in Mary Poppins.

1.5 The present study

In the present study, we used a similar method as used by Kartiko et al. [8]. The previous findings showed that the visual complexity of AVAs, such as flat, cartoon or lifelike, does not affect learning outcomes. In this study, we used VEAR, a firefighting simulation, to confirm and extend the findings of the previous study. VEAR showed greater visual complexity of the AVAs and the VE (Figure 1), compared with the previous study using ANTECATALYST [8]. Similar to ANTECATALYST, VEAR offered no direct interactivity to manipulate objects in the simulation, visual cues or feedback to direct users' attention.

In VEAR, the participants learnt with (A) a simple condition with visually simple AVAs and VE, (B) a simple VE condition with visually complex AVAs and simple VE, (C) a simple AVAs condition with simple AVAs and visually complex VE, or (D) a lifelike condition with visually complex AVAs and VE (Figure 1). We used the standardised Presence Questionnaire [51, PQ] and Perceived Affective Quality [64, PAQ] ratings. PQ measured participants' experience of being in the virtual world. PAQ measured students' perceived affective quality of the VR simulation (e.g., how forceful or beautiful it was). In assessing learning outcomes, we used Program Ratings [72, PR], and retention and transfer tests. PR provided affective scores that measured participants' perception of learning (e.g., how motivating, interesting and helpful it was) and cognitive load scores that measured their perceived difficulty of learning. The retention test required the participants to recall the information as given in the simulation. Lastly, the transfer test required the participants to infer safe firefighting procedures in different fire or car accident scenarios based on what they had learned.

1.6 Hypotheses

We conceptualised the hypotheses to answer the three questions proposed earlier: (1) Does the presence and affective quality ratings follow the order of scenes' visual complexity? (2) Does an increase of scenes' visual complexity affect learning? (3) Do the sense of presence and perceived affective quality affect the learning outcomes? Unlike the previous study by Kartiko et al. [8], VEAR enabled us to examine the effects of scenes' complexity by altering the visual complexity of the VEs in addition to that of the AVAs to achieve greater visual complexity.

With regard to the sense of presence and perceived affective quality ratings, we hypothesised that these measures would increase in the order of scenes' visual complexity (Hypothesis 1). This is in line with the recent studies on presence, human aesthetic preference for complex visual stimuli, market demands, and affect and emotion that visually complex AVAs and VEs are pleasing and attractive [9], [10], [51], [56].

In line with studies on the Coherence Principle [47], [48], [49], it is also hypothesised that the learning perception and the difficulty rating will increase with the scenes' visual complexity and the scores in retention and transfer tests will decrease with the scenes' visual complexity (Hypothesis 2). Previous studies on the Coherence Principle of CTML showed that complex visual materials are entertaining and motivating, but impair learning.

Following the prior discourse on the sense of presence and perceived affective quality within the framework of flow theory, we hypothesised that PAQ would correlate with presence: those who report higher PAQ ratings would report higher presence scores (Hypothesis 3). Furthermore, we hypothesised that greater scenes' visual complexity would increase the perception of the learning (how motivating, interesting and helpful the content is), retention and transfer scores, while it would decrease perceived difficulty (Hypothesis 4). Hypothesis 4 is similar to hypothesis 2 with the opposite predictions on test scores and perceived difficulty. Since both hypotheses conceptualised from two different perspectives, therefore both cannot be true at the same time.

2 METHOD

2.1 Participants and design

The eighty-eight participants were Macquarie University students and staff as well as members of the public (43 males and 45 females) with a median age of 22 ($IQR = 20 - 25$), who were paid AUD$ 20.00 for their voluntary participation. A group of 1 to 6 people participated in the experiment at the same time. Each person was randomly assigned to one of 4 experimental conditions differing in the AVAs' and VE's visual complexity (Figure 1).

2.2 Instruments

VEAR was run on an Intel Core2 Duo E8400 desktop PC with nVidia GeForce GTX 260 graphic card, 4GB RAM. The VR simulation ran smoothly at maximum frame-rate. The simulation was displayed full-screen at 1680 x 1050 pixels resolution on a 473 mm x 300 mm viewable screen area. The participants were seated at approximately 60 cm away from the computer screen. Thus, the physical horizontal FOV was approximately 43°.

The VEAR simulation taught the participants about the hazard in motor accidents and the best practice to

approach safely and prevent the spread of fire. The learning material was developed based on Village Firefighter handbooks: Village Firefighter (VF) Manual [73, Chapter 3, section 9], Specialist Training Manual: Compressed Air Breathing Apparatus [74, CABA], and Compressed Air Breathing Apparatus Standard Operating Procedures [75]. VEAR enabled learners to learn dangerous scenes in a physically, ethically and financially safe environment. VEAR was able to present vehicle accident scenes in greater details with camera movements capturing point of interests, than the printed materials.

The learning material was organised into eight continuous scenes: 1 - Introduction; 2 - Introduction to Compressed Air Breathing Apparatus (CABA); 3 - Vehicle roll and tip hazard; 4 - Explosion hazard; 5 - Hazardous material; 6 - Electrical hazard; 7 - Vehicle with no fire; 8 - Summary. The duration was 8 minutes and the camera moved from one location to another following the narration. The virtual actors were visible in all scenes. We used Blender [76] for 3D modelling and animation; GIMP [77] and Inkscape [78] for texturing the virtual world. The 3D engine used to display the simulation was Blender [76] itself. The scene was composed of AVAs and a VE displaying the car accidents. The VE consists of roads, a tunnel, hills, vehicles and the environmental effects, such as smoke, water and fire. Throughout the experiment, the scenario, screen resolution, screen colour, camera speed and movement, and the narration were kept the same for all conditions. The independent variable was the visual complexity of the scene, which was altered for each group. None of the students had attended the simulation before. Since there was no interactivity within the VR simulation, the participants had an equal duration of visual exposure.

Figure 1 and Table 1 show the learning conditions in VEAR. The simple condition is the simplest visually, created in black and white by using simple circles and lines only. We used simple AVAs and a simple VE, such as flat and static water, flat and static fire, flat and static smoke, simple car objects with flat components, and a plain white sky. In the most visually complex lifelike condition, we used lifelike complex AVAs and a visually complex VE, including lifelike clothing with details, detailed car objects, animated water, fire and smoke, cloudy skies, reflections on cars, helmet, and glass, and complex textures to depict the rocks and roads, with other surface blemishes in the VE. Two conditions with intermediate visual complexity were also created. The simple VE condition contained visually complex AVAs and a simple VE. The simple AVAs condition was the opposite of the simple VE condition, with simple AVAs but a visually complex VE.

2.3 Measurements

For each participant, the paper-and-pencil materials included the Simulator Sickness Questionnaire [79, SSQ], a ten-page retention and transfer pretest, the Presence Questionnaire [51, PQ], Perceived Affective Quality [64, PAQ] ratings, a Program Ratings [72, PR], and a ten-page retention and transfer posttest. We employed SSQ to assess the safety of the presentation as VR simulations cause simulator sickness. These materials served as the dependent measures, typeset and presented on standard A4-size papers. All questionnaire items, with the retention and transfer post-test, are included in the Appendix.

We used only 10 questions from PQ [51] that related to this study, out of 30 possible questions to measure the sense of presence. The questionnaire had the instruction "Please read the questions carefully and circle the appropriate number from 1 to 7." The scale from 1 - 7 ranged from Not Engaged to Moderately Engaged to Very Engaged. The internal consistency coefficient (Cronbach's alpha) of the PQ was 0.81.

In assessing perceived affective quality, we only used the shorter version of the PAQ ratings that contains twenty adjective words [64]. This questionnaire was given to the participants with the instruction "Below is a list of words that can be used to describe places. We would like you to rate how accurately each word below describes the Virtual World you just visited. Use the following 1-8 rating scale for your answer. Please be sure that you have given an answer for each word. 1 = extremely inaccurate; 2 = very inaccurate; 3 = quite inaccurate; 4 = slightly inaccurate; 5 = slightly accurate; 6 = quite accurate; 7 = very accurate; 8 = extremely accurate." The internal consistency coefficient (Cronbach's alpha) for ratings of the words ranged from 0.70 - 0.85.

The PR was adapted from a study by Moreno and Flowerday [72]. It contained eight questions to assess participants' perceptions of learning on a five-point scale, e.g., 1 as boring and 5 as interesting. The first six questions measured the affective level during the presentation, that is, the perception of learning, motivation, helpfulness, and interest, with internal consistency of 0.91. The last two questions measured the cognitive load level or the perceived difficulty level, with internal consistency of 0.81.

The pretest and posttest in the retention test consisted of nine multiple-choice questions with an instruction "Please answer all 9 questions. Please circle the correct answer. This is a practice test and not part of course grades." In pretest and posttest, the questions remained the same, but the order of the questions and the multiple-choice answers were changed.

The pretest and posttest in the transfer, or problem-solving test consisted of six multiple-choice questions with a similar instruction "Please answer all 6 questions. Please circle the correct answer. This is a practice test and not part of course grades." The students received six problem-solving questions and some are not specifically related to Village Firefighting. The questions and the multiple-choice answers were similar in both pretest and posttest but presented in different order.

TABLE 1
Visual complexity between experimental conditions

			Experimental Condition			
			Simple	**Simple AVAs**	**Simple VE**	**Lifelike**
	AVAs	Shape	Flat - 2D, 1 polygon	Flat - 2D, 1 polygon	Lifelike - 3D	Lifelike - 3D
		Colour	Black and white	Black and white	Full colour	Full colour
		Animations	Spin only	Spin only	Lifelike breathing and turning	Lifelike breathing and turning
		Details	Simplified	Simplified	Lifelike skin Creases and imperfections Sky reflection	Lifelike skin Creases and imperfections Sky reflection
Environment	**Cars**	Shape	Simplified	3D - real vehicle	Simplified	3D - real vehicle
		Colour	Black and White	Full colour	Black and White	Full colour
		Details	Simplified	Tyre markings Detailed interior Env. reflection	Simplified	Tyre markings Detailed interior Env. reflection
	Fire, Smoke Water, Foam	Shape	Flat	3D	Flat	3D
		Colour	Black and White	Full colour	Black and White	Full colour
		Animation	None	Texture - lifelike Normal mapped	None	Texture - lifelike Normal mapped
	Sky	Shape	None	3D dome	None	3D dome
		Colour	White	Photograph	White	Photograph
	Lighting	Lightsource	Directional only	Multiple lights	Directional only	Multiple lights
	Road & Landscape	Shape	Flat	3D	Flat	3D
		Colour	Black and White	Full	Black and White	Full
		Details	None	Cracks and imperfections Normal mapped	None	Cracks and imperfections Normal mapped
	Post-processing		None	Colour curve and Glow	None	Colour curve and Glow

2.4 Experimental procedures

Upon arrival in the computer lab, a group of participants was asked to fill in the consent form and each participant was randomly assigned to one computer with one of four experimental conditions. The participants were then reminded about SS and the procedure to follow if they feel sick in the simulation. SSQ ratings were collected to ensure VEAR's suitability for future use. We gave the participants 15 minutes to do the retention and transfer tests before attending the VR simulation. After learning with the VR simulation, they filled in SSQ, PQ, PAQ ratings, PR, and the retention and transfer posttest. Once they completed the questionnaires, we thanked them for the participation. At the end of the experiment, they were given the correct answers and the explanations to the posttest before leaving the computer lab.

2.5 Scoring

Presence score was computed for each participant by adding the circled numbers on the sheet. For the negatively worded questions, the circled number on the sheet was multiplied by negative one. Due to the unbalanced number of positive and negative question items, 8 points was deducted from the total presence score, so that the possible scores ranged from -30 to +30. Perceived affective quality scores were computed for each participant by adding the quality ratings next to the adjective words to give arousal, sleepy, pleasant and unpleasant scores. The arousal-sleepy and pleasant-unpleasant quality scores were obtained by subtracting the scores of negative items from the scores of the positive items, with a possible range from -35 to +35. The perception of learning score from the PR was computed for each participant by calculating the means of the circled number of the first six questions on the PR sheet. Similarly, perceived difficulty score was computed by calculating the means of the circled number of the last two questions on the PR sheet. The affective and cognitive load scores ranged from 1 to 5. A retention score for each participant was calculated by multiplying the number of correct answer by 5.55, and then 0.05 was added to the sum for a total score out of 50. The transfer score was computed for each participant by multiplying the number of correct answer by 8.33, then 0.02 was added to the sum for a total score out of 50. Thus, the highest test score that each participant could achieve was 100 points.

3 Results

There were violations of the assumptions for parametric statistical tests from the results of the Shapiro-Wilk normality test and the Levene test for unequal variances. Therefore, the data were analysed by employing non-parametric tests. We calculated Cliff's d values [80], [81], [82] to show the effect sizes. All statistical analysis methods were performed by using R [83]. Additionally, we used npmc package in R to run non-parametric post-hoc tests [84] with $alpha = 0.05$. Figure 2 and Table 2 show the box plots, and the medians (*Mdn*) and interquartile ranges *IQR* of the dependent variables, respectively. Table 3 shows the correlations between the dependent variables.

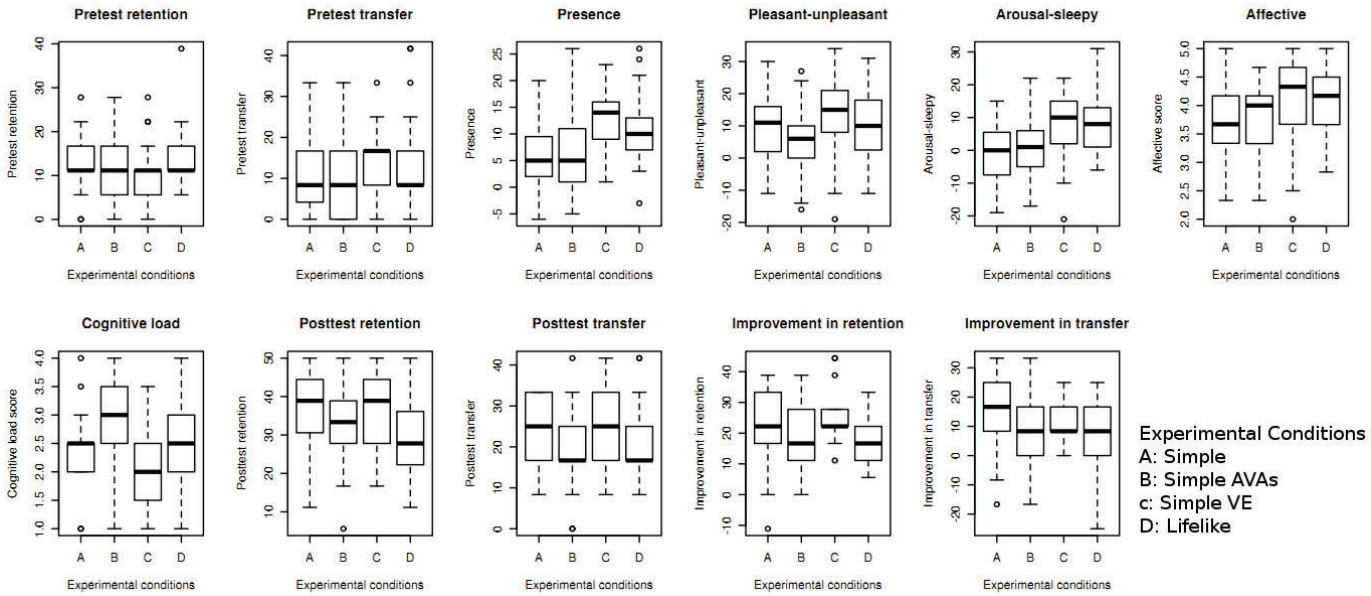


Fig. 2. Box plots of dependent variables. Experimental condition A, B, C and D refer to simple, simple VE, simple AVAs and lifelike condition respectively. The box represents the interquartile range (IQR) of the data. The upper hinge of the box represents the 3rd or upper quartile, and the lower hinge of the box represents the 1st or lower quartile. The thick line in the box shows the median value. The whiskers show the lower and upper values of 1.5 x lower quartile and 1.5 x upper quartile. Lastly, the white circles indicate the outliers in the data.

TABLE 2
Medians (*Mdn*) and Interquartile Range (*IQR*) of Dependent Variables

	Group							
	Simple (N=22)		Simple VE (N=20)		Simple AVAs (N=20)		Lifelike (N=20)	
Dependent measures	*Mdn*	*IQR*	*Mdn*	*IQR*	*Mdn*	*IQR*	*Mdn*	*IQR*
Pretest retention	11.15	11.15-16.70	11.15	5.60-16.70	11.15	11.15-16.70	11.15	5.60-16.70
Pretest transfer	8.35	2.10-16.67	8.35	8.35-16.68	8.35	2.10-16.68	8.35	6.27-16.68
Presence	5.50	2.00-9.75	9.00	3.00-14.00	5.50	2.0-9.75	4.50	1.00-11.25
Pleasant-unpleasant	11.00	1.50-16.00	10.50	1.75-17.25	11.00	1.5-16.00	6.00	-0.75-10.75
Arousal-sleepy	1.00	-7.75-5.75	5.00	-1.25-11.50	1.00	-7.75-5.75	1.50	-5.50-6.75
Affective	3.67	3.5-4.17	4.00	3.50-4.50	3.67	3.5-4.17	4.00	3.46-4.17
Cognitive load	2.50	2.00-2.50	2.50	2.00-3.00	2.50	2.00-2.50	3.00	2.5-3.5
posttest retention	38.90	29.19-43.06	33.35	27.80-38.90	38.90	29.19-43.06	33.35	27.80-40.29
posttest transfer	22.20	16.65-33.30	25.01	16.68-33.34	25.01	16.68-33.34	16.68	16.68-27.09
Increase in retention	22.20	16.65-33.30	22.20	16.65-27.72	22.20	16.65-33.30	16.65	11.10-27.75
Increase in transfer	16.66	8.33-24.99	8.33	0.00-16.66	16.66	8.33-24.99	12.49	0.00-16.66

Question1: Does the presence and affective quality ratings follow the order of scenes' visual complexity? The Kruskal-Wallis nonparametric ANOVA test showed significant differences between the groups, $\chi^2(3, N = 88) = 16.18, p = 0.00$. The npmc post-hoc test showed significant difference only between simple and simple AVAs groups, $p = 0.00$, and between simple VE and simple AVAs groups, $p = 0.00$, as shown in Table 4. The results only showed that increasing scene complexity from simple to simple AVAs condition and from simple VE to simple AVAs conditions increased presence in VR simulations.

Furthermore, using pleasant-unpleasant and arousal-sleepy scores as the dependent variables, the Kruskal-Wallis test indicated significant differences between the groups in arousal-sleepy score, $\chi^2(3, N = 88) = 13.32, p = 0.00$, but not in pleasant-unpleasant score, $\chi^2(3, N = 88) = 5.58, p = 0.13$. The npmc post-hoc test revealed significant difference between simple and simple VE groups, $p = 0.01$, and between simple and lifelike, $p = 0.01$, as shown in Table 5. The result indicated that increasing scenes' visual complexity from simple to simple AVAs condition, and from simple to lifelike condition increased perceived affective quality in

TABLE 3
Spearman's (r_s) Correlation of Dependent Variables

	PQ	PAQ.AS	PAQ.PU	PR-A	PR-Cl	retention⇑
PQ						
PAQ.AS	0.52***					
PAQ.PU	0.47***	0.30**				
PR-A	0.53***	0.59***	0.38***			
PR-Cl	-0.46***	-0.10	-0.34**	-0.20		
retention⇑	0.14	0.00	0.14	0.09	-0.29**	
transfer⇑	-0.07	-0.33**	-0.10	-0.17	-0.22*	0.22*

$*p < .05, * * p < .01, * * *p < .001$.. retention⇑: increase in retention. transfer⇑: increase in transfer.

TABLE 4
Post-hoc npmc test of presence score.

Comparison	effect	lower.cl	upper.cl	p-value 1s	p-value 2s
1-2	0.48	0.23	0.73	1.00	1.00
1-3	0.80	0.60	1.00	0.00	0.00
1-4	0.69	0.47	0.91	0.073	0.13
2-3	0.79	0.6	0.99	0.00	0.00
2-4	0.67	0.43	0.90	0.14	0.23
3-4	0.35	0.10	0.60	1.00	0.35

Experimental condition 1, 2, 3 and 4 refer to simple, simple VE, simple AVAs and lifelike condition respectively. cl: confidence level. 1s/2s: one/two sided.

VR simulations.

Question 2: Does an increase of scenes' visual complexity affect learning? There are three learning outcomes in this study: (1) program ratings (PR) that give affective ratings or perception of learning, and cognitive load or perception of difficulty, (2) retention and (3) transfer scores. How does the perception of learning differ between experimental groups? We performed a Kruskal-Wallis nonparametric ANOVA test with experimental conditions as the factor, and the affective scores as the dependent variable. The test showed no significant difference in perception of learning between groups, $\chi^2(3, N = 88) = 6.29, p = 0.09$. The results did not support predictions according to CTML. All groups reported a similar level of perception of learning (motivation, helpfulness, interest) and the level was relatively high, $Mdn = 4.00, IQR = 3.50 - 4.50$.

Does perceived difficulty of learning differ between experimental groups? We performed Kruskal-Wallis non-

TABLE 5
Post-hoc npmc test of arousal-sleepy rating.

Comparison	effect	lower.cl	upper.cl	p-value 1s	p-value 2s
1-2	0.56	0.31	0.80	0.77	0.93
1-3	0.77	0.56	0.98	0.00	0.00
1-4	0.75	0.55	0.95	0.00	0.01
2-3	0.70	0.46	0.94	0.07	0.12
2-4	0.68	0.45	0.92	0.1	0.16
3-4	0.44	0.19	0.68	1.00	0.90

Experimental condition 1, 2, 3 and 4 refer to simple, simple VE, simple AVAs and lifelike condition respectively. cl: confidence level. 1s/2s: one/two sided.

TABLE 6
Post-hoc npmc test of cognitive load score.

Comparison	effect	lower.cl	upper.cl	p-value 1s	p-value 2s
1-2	0.56	0.31	0.80	0.77	0.93
1-3	0.77	0.56	0.98	0.00	0.00
1-4	0.75	0.55	0.95	0.00	0.01
2-3	0.70	0.46	0.94	0.07	0.12
2-4	0.68	0.45	0.92	0.094	0.16
3-4	0.44	0.19	0.68	1.00	0.90

Experimental condition 1, 2, 3 and 4 refer to simple, simple VE, simple AVAs and lifelike condition respectively. cl: confidence level. 1s/2s: one/two sided.

TABLE 7
Post-hoc npmc test of improvement of retention score.

Comparison	effect	lower.cl	upper.cl	p-value 1s	p-value 2s
1-2	0.41	0.18	0.65	1.00	0.78
1-3	0.53	0.29	0.78	1.84	0.99
1-4	0.35	0.12	0.58	1.00	0.31
2-3	0.66	0.42	0.90	0.17	0.29
2-4	0.45	0.20	0.69	1.00	0.95
3-4	0.24	0.05	0.43	1.00	0.00

Experimental condition 1, 2, 3 and 4 refer to simple, simple VE, simple AVAs and lifelike condition respectively. cl: confidence level. 1s/2s: one/two sided.

parametric ANOVA with experimental conditions as factor and the cognitive load score as the dependant variable. The test revealed significant differences between groups, $\chi^2(3, N = 88) = 9.85, p = 0.02$. The npmc post-hoc test only showed significant difference between simple VE and simple AVAs groups, $p = 0.02$, as shown in Table 6. Contrary to our predictions according to CTML, the result showed that increasing scenes' visual complexity from simple VE to simple AVAs condition decrease perceived difficulty in VR simulations.

To assess the impacts of scenes' complexity on retention, we conducted a Kruskal-Wallis nonparametric ANOVA test with the experimental conditions as the factor, and the difference in total retention scores (post-learning score minus pre-learning score) as the dependent variable. The test showed a significant difference across groups in improvement of retention from pre- to post-test, $\chi^2(3, N = 88) = 8.25, p = 0.04$. The npmc post-hoc test revealed a significant difference between lifelike and simple AVAs groups in favour of the simple AVAs group, $p = 0.00$, as shown in Table 7. The result indicated that increasing scenes' complexity from simple AVAs to lifelike condition decreased retention, which supported the prediction according to CTML.

In addition, the Wilcoxon signed rank test between post and pre retention scores in each group revealed a significant increase after attending the VR simulations (simple: $V = 249, p = 0.00, d = 0.86$, simple AVAs: $V = 231, p = 0.00, d = 0.96$, simple VE: $V = 210, p = 0.00, d = 0.84$ and lifelike: $V = 276, p = 0.00, d = 0.84$).

In investigating the impacts of scenes' visual com-

plexity on transfer, we performed a Kruskal-Wallis non-parametric ANOVA test on the groups as the factor and the difference in transfer scores (post-learning score minus pre-learning score) as the dependent variable. The test did not show any significant differences in transfer between the groups, $\chi^2(3, N = 88) = 3.53, p = 0.32$. Furthermore, the Wilcoxon signed rank test between post and pre transfer scores in each group also revealed a significant increase in transfer in all conditions except for the lifelike condition (simple: $V = 226, p = 0.00, d = 0.63$, simple AVAs: $V = 136, p = 0.00, d = 0.63$, simple VE: $V = 120.5, p = 0.00, d = 0.51$, and lifelike: $V = 122.5, p = 0.11, d = 0.37$).

Question 3: Do the sense of presence and perceived affective quality affect the learning outcomes? We began with the question of whether presence correlates with perceived affective quality. Spearman rank correlation test, as shown in Table 3, revealed moderate and significant correlations between presence and arousal-sleepy quality, $r_s(88) = 0.52, p < 0.001$, and between presence and pleasant-unpleasant quality, $r_s(88) = 0.47, p < 0.001$. With regard to learning outcomes, affective score is moderately and significantly correlated with presence, $r_s(88) = 0.52; p < 0.001$, arousal-sleepy ratings, $r_s(88) = 0.60, p < 0.001$, and pleasant-unpleasant ratings $r_s(88) = 0.38, p < 0.001$. Cognitive load was moderately and inversely correlated with presence, $r_s(88) = 0.45, p < 0.001$, and pleasant and unpleasant ratings, $r_s(88) = 0.33, p < 0.001$. Arousal-sleepy rating was also negatively correlated with increase in transfer $r_s(88) = 0.31, p < 0.001$. In addition, cognitive load was weakly and inversely correlated with retention, $r_s(88) = 0.28, p < 0.05$, and transfer scores, $r_s(88) = 0.20, p < 0.001$.

4 Discussion

The present study examined the impacts of visually complex AVAs and VEs on VR-based instructional animation. The visual representation of AVAs and VEs for learning become visually rich and complex as the computer graphics technology progresses. Complex visuals are motivating, pleasing, and could improve learning according to flow-theory [10], [17], [57], [63], [67], but Cognitive Theory of Multimedia Learning [14], [15], [36] raises the possibility that visually complex presentations may hinder the learning process. This study began by asking: (1) Does the presence and affective quality ratings follow the order of scenes' visual complexity? (2) Does an increase of scenes' visual complexity affect learning? (3) Do the sense of presence and perceived affective quality affect the learning outcomes? The summary of the outcomes is listed in Table 8.

In partial agreement with Hypothesis 1, the sense of presence and perceived affective quality increased with the visual complexity of the scene, but not in the order of experimental conditions A-B-C-D. Firstly, the findings suggest that increasing the visual complexity of VE would significantly increase the sense of presence in VR simulations compared to increasing the visual complexity of the AVAs. Secondly, the findings show that increasing the visual complexity of the VE or the visual complexity of the AVAs and the VE would increase the perceived affective quality ratings along the arousalsleepy scale.

With regard to learning outcomes, Hypothesis 2 is partially supported. There was no difference in perception of learning and transfer scores between experimental groups. However, the participants subjectively rated greater visual complexity as being easier to comprehend. Furthermore, this finding also suggests that lowering the visual complexity of VE might promote better retention. The outcomes in retention are in line with the previous studies on the Coherence Principle of CTML [47], [48], [49] that complex visual materials hinder learning. In addition, the participants showed high perception of learning, and importantly, significant improvements of retention scores after viewing VEAR in all experimental conditions. All groups also showed significant improvements in transfer after exposure to VEAR, except for those who attended with the visually complex scene with complex AVAs and VE. Thus, the learning outcomes with VEAR encourage the removal of extraneous visual information, in accordance with Coherence Principle, to ensure retention and transfer outcomes. It seems that a rich VE could be especially detrimental to learning.

The hypothesis that presence and perceived affective quality are correlated was confirmed in this study, as shown in Table 3 (Hypothesis 3). This shows that these two constructs measure some common components contributing to flow or optimal experience. Together with the results from the previous study using ANTECATALYST simulation [8], the results strengthen the notion that these two motivational factors in VR simulations contribute to some of the same components in promoting flow. In relation to learning outcomes, Hypothesis 4 is not confirmed in this study. The results, as shown in Table 3, suggest that motivational factors in VR simulation made the task appear simple, which is in line with the prediction of flow theory [16]. With regard to learning outcomes, these findings are similar to other previous studies in CTML and multimedia learning, that motivational factors arising from the learning contents do not contribute to learning outcomes, but may hinder the learning process [8], [15], [28], [36], [47], [48], [49], [85]. Thus, as with the findings on learning outcomes (Hypothesis 2), these findings also encourage the application of CTML in promoting retention and transfer by keeping extraneous details to a minimum in VR simulations. How do we explain these findings since other studies clearly show the impact of mood and emotion on learning outcomes [57], [60], [67], [68]? We suspect that every individual has a threshold at which presence and affective quality would be sufficient to promote flow for learning. Beyond a certain threshold, these factors might not add any more to the positive effects of flow, while

TABLE 8
The summary of the hypotheses, the questions and the outcomes

	Hypothesis	Confirmed	Outcomes
1	The sense of presence and perceived affective quality ratings would increase in the order of scenes' visual complexity	Yes - partial	• Only increasing scene's complexity from simple to simple AVAs condition (A-C) and from simple world to simple AVAs (B-C) conditions increased presence. • Only increasing scenes' visual complexity from simple to simple AVAs condition (A-C), and from simple to lifelike condition (A-D) increased perceived affective quality on sleepy-arousal ratings.
2	Learning perception and the difficulty rating will increase with the scenes' visual complexity and the scores in retention and transfer tests will decrease with the scenes' visual complexity	Yes-partial	• There was no difference in perception of learning scores between groups. • Only increasing scenes' visual complexity from simple world to simple AVAs (B-C) condition decreased perceived difficulty in VR simulations. • Only increasing scenes' visual complexity from simple AVAs to lifelike condition (C-D) decreased retention. • Scenes' visual complexity between groups did not affect transfer.
3	PAQ correlates with presence	Yes	• Those who reported higher presence also reported higher PAQ ratings.
4	Learning perception, the scores in retention and transfer tests will increase with the scenes' visual complexity and the difficulty rating will decrease with the scenes' visual complexity	No	• See the outcomes of hypothesis 2.
	Questions	**Confirmed**	**Outcomes**
1	Does the presence and affective quality ratings follow the order of scenes' visual complexity?	Yes - partial	• Hypothesis 1 is partially supported. The result indicate that increasing the visual complexity of VE increases presence and affective quality on sleepy-arousal scale.
2	Does an increase of scenes' visual complexity affect learning?	Yes - partial	• Hypothesis 2 is partially supported. The result indicated that lowering the visual complexity of VE might promote better retention.
3	Do the sense of presence and perceived affective quality affect the learning outcomes?	Yes	• Presence and perceived affective quality are correlated with perception of learning. • Presence and pleasant-unpleasant ratings of perceived affective quality are inversely correlated with perceived difficulty scores. • Perceived difficulty scores are negatively correlated with retention and transfer scores. • Perceived affective quality along arousal – sleepy ratings are inversely correlated with transfer.

adding to the negative effects of extraneous cognitive load. Such a hypothesis remains to be tested.

Although this study does not fully confirm the proposed hypotheses, its limitations should be pointed out. First, this study did not take the virtual camera into consideration as a factor which may influence affective and cognitive responses for learning outcomes. There are three main sources where an instructional animation may become complex, apart from the subject of learning [21]: (1) Cinematography or the art of capturing the scene by the virtual camera: the nature of VR simulations is dynamic visualisation where users transverse a 3D world by automatic guidance or free exploration, (2) Virtual Environments (VEs): the residence of the AVAs in depicting a scenario and may contain animations such as pleasant moving clouds, frightening blazing fire, and rotating windmills, and (3) AVAs: AVAs may perform predefined tasks in VEs to assist learning. Compared to many 2D animations [2], [4], [26], [86], VR-based instructional animations are visually more complex. The application of the virtual camera may or may not promote learning. The movement of the virtual camera can be unnecessarily complex and overwhelm learners. On the other hand, the virtual camera in VR simulations

[8], [28], [60] does not normally show the whole scene at one time, moving instead from one area to another to focus the view on the object or scene of interest. This idea is similar to framing, masking, zooming [31], and segmentation [72], techniques that guide users' attention to the relevant material. These techniques are commonly found in movies [87], [88]. In addition, the placement of the virtual camera in a scene might trigger a certain affective response that may not be desirable for learning, e.g., a camera close to the blazing fire might induce a traumatic experience (VEAR scene 4 - Explosion hazard). Second, it is necessary to develop a program to quantify the visual complexity of VR-based instructional animations based on empirical results by Roberts [10]. With regard to static images, Donderi [19] shows that visual complexity can be estimated by the size of the file at a constant compression ratio. Is this true also for VR-based instructional animations? A definitive measure of the level of visual complexity would enable VR developers to gauge the visual complexity and investigate whether it would impede or facilitate affective and cognitive responses toward better retention and transfer.

5 Conclusion

This study resembles the previous study using ANTECATALYST simulation [8]. VEAR extends the findings by using greater visual complexity, consistent AVAs throughout the scenes and taking in to account the visual complexity of the VE as well as that of the AVAs in assessing the affective and cognitive impacts for learning. In this study, we used VEAR (VEhicle Accident and Fire) to assess the impacts of visually complex scenes in VR-based instructional learning. The previous study with ANTECATALYST had only one simple VE and increasing the visual complexity of AVAs did not impact either affective or cognitive responses for learning. However, the current findings suggest that increasing scenes' visual complexity has the potential of increasing affective ratings and undermining retention and transfer, especially increasing the visual complexity of the VE. The availability of free 3D modelling and animation tools (e.g., [76], [89], [90]) and affordable VR hardware encourage VR developers to create visually rich and complex scenes in VR simulations. The outcomes of this study, however, warn us that making both visually complex AVAs and VEs in a scene would put retention and transfer outcomes in jeopardy. Increasing the visual complexity of VE seems to increase the overall scene complexity a lot more to impair learning than increasing the visual complexity of the AVAs. Therefore, if visually complex AVAs are required in a training simulation, having a simple VE might ensure optimal retention and transfer. Visually simple AVAs and VEs are simpler and quicker to develop compared with visually complex AVAs and VEs. This also benefits educational projects with short time-frame or small labour budgets for delivering effective or balanced affective-cognitive learning materials. In achieving an optimal learning experience, a medium level of visual complexity is probably best to stimulate flow without hindering learning outcomes. Since only one scenario is tested in this study, these outcomes must be scrutinised under different experimental conditions to test the generality of the results, including testing the effects of even greater visual complexity, different learning materials, longer lessons and target audience at differing academic levels.

Acknowledgments

This project has been sponsored by the Australian Research Council Discovery Grant, DP0558852 (Richards, Kavakli, Dras) titled Risk Management Using Agent Based Systems. Special thanks to Matt Cabanag and Darius Taslim for their assistance in troubleshooting, and running the computers in the labs. We are also grateful to Susan Bruck for the fruitful discussion on SS, and Geoff Plummer for the audio recording session. We also thank anonymous reviewers for their comments.

References

[1] G. Borshukov and J. P. Lewis, "Realistic human face rendering for "the matrix reloaded"," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*. New York, NY, USA: ACM Press, 2005, p. 13.

[2] J. C. Lester, S. A. Converse, S. E. Kahler, S. T. Barlow, B. A. Stone, and R. S. Bhogal, "The persona effect: affective impact of animated pedagogical agents," pp. 359–366, 1997.

[3] D. Wagner, M. Billinghurst, and D. Schmalstieg, "How real should virtual characters be?" in *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. California, USA: ACM Press, 2006, p. 57.

[4] Q. Dunsworth and R. K. Atkinson, "Fostering multimedia learning of science: Exploring the role of an animated agent's image," *Computers & Education*, vol. 49, no. 3, pp. 677 – 690, 2007.

[5] K. Johnsen, A. Raij, A. Stevens, D. S. Lind, and B. Lok, "The validity of a virtual human experience for interpersonal skills education," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2007, pp. 1049–1058.

[6] M. Kavakli, N. Szilas, J. Porte, and I. Kartiko, *Architectural Design of Multi-Agent Systems: Technologies and Techniques*, ser. Architectural Design of Multi-Agent Systems. Information Science Reference, 2007, ch. RiskMan: A Multi-Agent System for Risk Management, pp. 356 – 376.

[7] M. M. Lusk and R. K. Atkinson, "Animated pedagogical agents: Does their degree of embodiment impact learning from static or animated worked examples?" *Applied Cognitive Psychology*, vol. 21, no. 6, pp. 747–764, 2007.

[8] I. Kartiko, M. Kavakli, and K. Cheng, "Learning science in a virtual reality application: The impacts of animated-virtual actors? visual complexity," *Computers & Education*, vol. 55, no. 2, pp. 881 – 891, 2010.

[9] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25–32, 2005.

[10] M. N. Roberts, "Complexity and aesthetic preference for diverse visual stimuli," Ph.D. dissertation, Departament de Psicologia, Universitat de les Illes Balears, 2007.

[11] R. K. Lowe, "Animation and learning: Value for money?" in *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*, R. Atkinson, C. McBeath, D. Jonas-Dwyer, and R. Phillips, Eds., 5-8 December 2004, pp. 558 – 561. [Online]. Available: http://www.ascilite.org.au/conferences/perth04/procs/lowe-r.html

[12] E. Ozcelik, I. Arslan-Ari, and K. Cagiltay, "Why does signaling enhance multimedia learning? evidence from eye movements," *Computers in Human Behavior*, vol. 26, no. 1, pp. 110 – 117, 2010.

[13] H. Jarodzka, K. Scheiter, P. Gerjets, and T. van Gog, "In the eyes of the beholder: How experts and novices interpret dynamic stimuli," *Learning and Instruction*, vol. 20, no. 2, pp. 146 – 154, 2010, eye tracking as a tool to study and enhance multimedia learning.
[14] R. E. Mayer, *Multimedia Learning*, R. E. Mayer, Ed. Cambridge Press, 2001, iSBN 9780521787499.
[15] R. Mayer and R. Moreno, "Animation as an aid to multimedia learning," *Educational Psychology Review*, vol. 14, no. 1, pp. 87–99, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013184611077
[16] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. Harper Collins, New York, 1990.
[17] ——, *Finding Flow: The Psychology of Engagement with Everyday Life*. Basic Books, 1997.
[18] I. Kartiko, M. Kavakli, and K. Cheng, "The impacts of animated - virtual actors' visual complexity and simulator sickness ?in virtual reality applications," in *Proceedings of the 6th International Conference Computer Graphics, Imaging and Visualization*, B. Werner, Ed. IEEE Computer Society, 2009, pp. 147 – 152, iSBN 978-0-7695-3789-4.
[19] D. C. Donderi, "Visual complexity: A review," *Psychological Bulletin*, vol. 132, no. 1, pp. 73–97, 2006.
[20] C. Abell, "Pictorial realism," *Australasian Journal of Philosophy*, vol. 85, no. 1, pp. 1–17, March 2007. [Online]. Available: http://dx.doi.org/10.1080/00048400601185529
[21] I. Kartiko and M. Kavakli, "Virtual reality police training: How much visual information is too much for knowledge transfer?" in *Proceedings of the 2nd International Conference on Cybercrime Forensics Education and Training*, 2008, pp. 1 – 13, iSBN 1899253-19x.
[22] B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: Can it facilitate?" *International Journal of Human-Computer Studies*, vol. 57, no. 4, pp. 247–262, 2002.
[23] B. Whitworth, "Polite computing," *Behaviour & Information Technology*, vol. 24, no. 5, pp. 353–363, 2005. [Online]. Available: http://www.informaworld.com/10.1080/01449290512331333700
[24] K. L. Nowak and C. Rauh, "Choose your "buddy icon" carefully: The influence of avatar androgyny, anthropomorphism and credibility in online interactions," *Computers in Human Behavior*, vol. 24, no. 4, p. 1473???1493, 2008. [Online]. Available: www.elsevier.com/locate/comphumbeh
[25] S. Choi and R. Clark, "Cognitive and affective benefits of an animated pedagogical agent for learning english as a second language." *Journal of Educational Computing Research*, vol. 34, no. 4, pp. 441–466, 2006.
[26] R. Moreno and T. Flowerday, "Student's choice of animated pedagogical agents in science learning: A test of the similarity-attraction hypothesis on gender and ethnicity," *Contemporary Educational Psychology*, vol. 31, no. 2, pp. 186–207, 2006.
[27] W. Schnotz, *Multimedia Learning: Cognitive and Instructional Issues*. Elsevier Science Ltd., UK, 2001, ch. Sign Systems, Technologies, and the Acquisition of Knowledge, pp. 9 – 29.
[28] D. Richards, M. Dras, J. Porte, and M. Taylor, "The role, measurement and delivery of immersion in training environments," SimTecT 2009 Simulation Conference: Simulation - Concepts, Capability and Technology. Adelaide, Australia, 15 - 18 June 2009.
[29] G. Veletsianos, "Cognitive and affective benefits of an animated pedagogical agent: Considering contextual relevance and aesthetics," *Journal of Educational Computing Research*, vol. 36, no. 4, pp. 373–377, 2007.
[30] R. Moreno, "Learning in high-tech and multimedia environments," *Current Directions in Psychological Science*, vol. 15, no. 2, pp. 63–67, 2006.
[31] C. Hongpaisanwiwat, "The comparison study of attention drawing techniques for multimedia learning," *Information Technology Based Higher Education and Training, 2006. ITHET '06. 7th International Conference on*, pp. 652–653, July 2006.
[32] A. Gulz and M. Haake, "Design of animated pedagogical agents-a look at their look," *Int. J. Human-Computer Studies*, vol. 64, no. 4, pp. 322–339, 2006.
[33] A. L. Baylor and J. Ryu, "The effects of image and animation in enhancing pedagogical agent persona." *Journal of Educational Computing Research*, vol. 28, no. 4, pp. 373–394, 2003. [Online]. Available: http://search.ebscohost.com.simsrad.net.ocs.mq.edu.au/login.aspx?direct=true&db=iih&AN=11299177&site=ehost-live
[34] R. Moreno, R. E. Mayer, A. H. Spires, and J. C. Lester, "The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents?" *Cognition and Instruction*, vol. 19, no. 2, pp. 177 – 213, 2001.
[35] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257 – 285, 1988.
[36] R. E. Mayer, "The promise of multimedia learning: using the same instructional design methods across different media," *Learning and Instruction*, vol. 13, no. 2, pp. 125–139, 2003.
[37] A. Paivio, "Mental representations: A dual coding approach," Oxford University Press, Oxford, England, 1986.
[38] A. D. Baddeley, *Human Memory: Theory and Practice*, revised ed. Allyn and Bacon, Needham Heights, MA, 1998, iSBN 0-205-27948-1.
[39] J. Sweller, "Instructional design in technical areas," ACER, Camberwell, Australia, 1999.
[40] M. C. Wittrock, "Generative process of comprehension," *Educational Psychologist*, vol. 24, pp. 345–376, 1989.
[41] R. E. Mayer, *The Promise of Educational Psychology Vol II: Teaching and Meaningful Learning*. Merrill Prentice Hall, 2002, iSBN 0-13-913013-6.
[42] T. N. Hoffler and D. Leutner, "Instructional animation versus static pictures: A meta-analysis," *Learning and Instruction*, vol. 17, no. 6, pp. 722 – 738, Dec 2007.
[43] A. D. Baddeley, *Essential to Human Memory*. Psychology press, 1999, iSBN 0-86377-544-6. [Online]. Available: http://www.psypress.com/9780863775444
[44] J. Dewey, *Interest and effort in education*. Cambridge Boston, MA : Houghton Mifflin, 1913.
[45] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956. [Online]. Available: http://cogprints.org/730/0/miller.html
[46] K. E. DeLeeuw and R. E. Mayer, "A comparison of three measures of cognitive load: Evidence for separable measures of intrinsic, extraneous, and germane load," *Journal of Educational Psychology*, vol. 100, no. 1, pp. 223–234, 2008.
[47] S. F. Harp and R. E. Mayer, "The role of interest in learning from scientific text and illustrations: On the distinction between emotional interest and cognitive interest." *Journal of Educational Psychology*, vol. 89, no. 1, pp. 92–102, 1997. [Online]. Available: http://visuallearningresearch.wiki.educ.msu.edu/file/view/Harp+%26+Mayer+%281997%29.pdf
[48] ——, "How seductive details do their damage: A theory of cognitive interest in science learning." *Journal of Educational Psychology*, vol. 90, no. 3, pp. 414–434, 1998. [Online]. Available: http://visuallearningresearch.wiki.educ.msu.edu/file/view/Harp+%26+Mayer+%281998%29.pdf
[49] R. E. Mayer, J. Heiser, and S. Lonn, "Cognitive constraints on multimedia learning: When presenting more material results in less understanding." *Journal of Educational Psychology*, vol. 93, no. 1, pp. 187–98, 2001.
[50] K. Lewin, *Field theory in social science; selected theoretical papers*, D. C. (ed., Ed. New York: Harper & Row, 1951.
[51] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoper. Virtual Environ.*, vol. 7, no. 3, pp. 225–240, 1998.
[52] T. P. Novak, D. L. Hoffman, and Y.-F. Yung, "Measuring the customer experience in online environments: A structural modeling approach," *Marketing Science*, vol. 19, no. 1, pp. 22–42, 2000. [Online]. Available: http://dx.doi.org/10.1287/mksc.19.1.22.15184
[53] J. Takatalo, G. Nyman, and L. Laaksonen, "Components of human experience in virtual environments," *Computers in Human Behaviour*, vol. 24, no. 1, pp. 1–15, 2008.
[54] J. J.-W. Lin, H. B. Duh, D. E. Parker, H. Abi-Rached, and T. A. Furness, "Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment," Proceedings of the IEEE Virtual Reality 2002 (VR'02), 2002.
[55] W. Sadowski and K. M. Stanney, *Handbook of virtual environments: Design, implementation, and applications*. Lawrence Erlbaum Associates, Inc., 2002, ch. Presence in virtual environments, pp. 791–806.
[56] K. M. Stanney, K. S. Kingdon, D. Graeber, and R. S. Kennedy, "Human performance in immersive virtual environments: Effects of exposure duration, user control, and scene complexity," *Human Performance*, vol. 15, no. 4, pp. 339–366, 2002.

[57] T. A. Mikropoulos and V. Strouboulis, "Factors that influence presence in educational virtual environments," *CyberPsychology & Behavior*, vol. 7, no. 5, pp. 582 – 591, 2004.
[58] D. Whitelock, D. Romano, A. Jelfs, and P. Brna, "Perfect presence: What does this mean for the design of virtual learning environments?" *Education and Information Technologies*, vol. 5, no. 4, pp. 277 – 289, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1012001523715
[59] R. Moreno and R. E. Mayer, "Learning science in virtual reality multimedia environments: Role of methods and media," *Journal of Educational Psychology*, vol. 94, no. 3, pp. 598 – 610, 2002.
[60] T. Mikropoulos, "Presence: a unique characteristic in educational virtual environments," *Virtual Reality*, vol. 10, no. 3, pp. 197–206, dec 2006.
[61] T. D. Parsons and A. A. Rizzo, "Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis," *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 39, no. 3, pp. 250 – 261, 2008.
[62] M. B. Powers and P. M. Emmelkamp, "Virtual reality exposure therapy for anxiety disorders: A meta-analysis," *Journal of Anxiety Disorders*, vol. 22, no. 3, pp. 561 – 569, 2008.
[63] J. A. Russell, "Core affect and the psychological construction of emotion," *Psychological Review*, vol. 110, no. 1, pp. 145–172, 2003.
[64] J. A. Russell and G. Pratt, "A description of the affective quality attributed to environments," *Journal of Personality and Social Psychology*, vol. 38, no. 2, pp. 311–322, 1980.
[65] P. R. Pintrich, *Motivation and classroom learning*. Wiley, New York, 2003, ch. 6, pp. 103–122.
[66] P. Zhang and N. Li, "The importance of affective quality," *COMMUNICATIONS OF THE ACM*, vol. 48, no. 9, pp. 105–108, September 2005.
[67] P. Zhang, N. Li, and H. Sun, "Affective quality and cognitive absorption: Extending technology acceptance research," Proceedings of the 39th Hawaii International Conference on System Sciences - IEEE 2006, 2006, iSBN 0-7695-2507-5.
[68] S. Brand, T. Reimer, and K. Opwis, "How do we learn in a negative mood? effects of a negative mood on transfer and learning," *Learning and Instruction*, vol. 17, no. 1, pp. 1 – 16, 2007.
[69] B.-C. Lee, J.-O. Yoon, and I. Lee, "Learners' acceptance of e-learning in south korea: Theories and results," *Computers & Education*, vol. 53, no. 4, pp. 1320 – 1329, 2009, learning with ICT: New perspectives on help seeking and information searching.
[70] S.-H. Liu, H.-L. Liao, and J. A. Pratt, "Impact of media richness and flow on e-learning technology acceptance," *Computers & Education*, vol. 52, no. 3, pp. 599 – 607, 2009.
[71] R. Stevenson, "Mary poppins," 1964, last accessed: July 2010. [Online]. Available: http://www.imdb.com/title/tt0058331/
[72] R. Moreno, "Optimising learning from animations by minimising cognitive load: Cognitive and affective consequences of signalling and segmentation methods," *Applied Cognitive Psychology*, vol. 21, no. 6, pp. 765–781, 2007.
[73] N. R. F. Service, "VF/1-6 (1999). Village Firefighter (VF) Manual," NSW Rural Fire Service, 1999, ISBN 0731097303x.
[74] ——, "Compressed Air Breathing Apparatus Standard Operating Procedures," NSW Rural Fire Service, 2001.
[75] ——, "SP/3(1995). Specialist Training Manual: Compressed Air Breathing Apparatus (CABA)." NSW Rural Fire Service, 2001, ISBN 0731064321.
[76] Blender Foundation, "Blender," 2011, last accessed on September 2011. [Online]. Available: http://www.blender.org
[77] Gimp Development Team, "Gimp," 2011, last accessed on September 2011. [Online]. Available: http://www.gimp.org
[78] Inkscape Development Team, "Inkscape," 2011, last accessed on September 2011. [Online]. Available: http://www.inkscape.org
[79] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness," *International Journal of Aviation Psychology*, vol. 3, no. 3, pp. 203–220, 1993.
[80] N. Cliff, "Dominance statistics: Ordinal analyses to answer ordinal questions," *Psychological Bulletin*, vol. 114, no. 3, pp. 494–509, 1993, iSSN: 0033-2909. [Online]. Available: www.florida-air.org/romano06.pdf
[81] A. B. del Rosal, C. S. Luis, and A. Sánchez-Bruno, "Dominance statistics: A simulation study on the d statistic," *Quality and Quantity*, vol. 37, no. 3, pp. 303–316, 2003.
[82] J. Romano, J. D. Kromrey, J. Coraggio, and J. Skowronek, "Appropriate statistics for ordinal level data : Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys?" Paper presented at the annual meeting of the Florida Association of Institutional Research, February 1 -3, 2006, Cocoa Beach, Florida, 2006. [Online]. Available: http://www.florida-air.org/romano06.pdf
[83] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2011, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org
[84] J. Helms and U. Munzel, "Nonparametric multiple comparisons," http://cran.r-project.org, April 2009. [Online]. Available: http://cran.r-project.org/web/packages/npmc
[85] R. Moreno and R. E. Mayer, "Personalized messages that promote science learning in virtual environments," *Journal of Educational Psychology*, vol. 96, no. 1, pp. 165–173, 2004.
[86] B. B. de Koning, H. K. Tabbers, R. M. Rikers, and F. Paas, "Attention guidance in learning from a complex animation: Seeing is understanding?" *Learning and Instruction*, vol. 20, no. 2, pp. 111 – 122, 2010, eye tracking as a tool to study and enhance multimedia learning.
[87] M. Simon, *Storyboards: Motion in Art*. Focal Press, 2000.
[88] M. Begleitter, *From word to image: Storyboarding and the filmmaking process*. Michael Wiese Production, 2001.
[89] DAZ3D Team, "Daz 3d," 2010, last accessed on March 2010. [Online]. Available: http://daz3d.com/
[90] T. M. Shead, "K-3d," 2011, last accessed on September 2011. [Online]. Available: http://www.k-3d.org/



Iwan Kartiko He is a finishing PhD student at Macquarie University. He is a VR developer and an experienced digital graphic artist. In his Ph.D. candidature, he examines the aspect of visual complexity that may affect learning in VR-based instructional animations, including Simulator Sickness (SS) and cognitive load. His research interests include Virtual Reality (VR), Computer Graphics (CG), Artificial Intelligence (AI), and Multimedia learning systems.



Manolya Kavakli She is an Associate Professor at Macquarie University in the department of computing. She obtained her Ph.D. from Istanbul Technical University, in Knowledge Based System Design. In 2003, she established Virtual Reality research group at Macquarie University. Her area of research covers visual perception and cognition, Human-Computer Interaction (HCI), User Interface (UI) design, computer games and Artificial Intelligence (AI).



Ken Cheng He is a Professor at Macquarie University in the department of Biological Sciences. He obtained the Ph.D. in Psychology from the University of Pennsylvania. He is a His research interests include comparative cognition, especially spatial cognition, ranging from humans' to desert ants'.

Part III

Closing Remarks

Chapter 11

Conclusion

11.1 Significance of the Outcomes

Virtual Reality (VR) simulations present system designers with unprecedented privileges for teaching purposes, which the traditional methods, e.g., chalk and blackboard, cannot provide. VR simulations can depict risky and dangerous scenarios in a safe environment for learning (e.g., Kavakli, Szilas, Porte, & Kartiko, 2007, Chapter 5). Furthermore, there are various affordable and free tools that enable the creation and the deployment of VR simulations (Kartiko et al., 2010, 2011a) that catalyst the use of VR simulations for instructional learning. The constant challenge for the designers is how to harness the latest technology to foster learning (Mantovani, 2003; Lee & Wong, 2008). Although, literature shows that VR simulations assist in gaining and developing psycho-motor skills (e.g., Grantcharov, 2006; Verdaasdonk, Stassen, Monteny, & Dankelman, 2006; G. Chen et al., 2010), educators must also strive to promote cognitive learning to foster retention and the highly valued transfer skills (Sweller, 1988; Moreno & Mayer, 2001).

Current studies in VR simulations as instructional animations yield mixed outcomes (see Winn, 1996; Whitelock, Romano, Jelfs, & Brna, 2000; T. Mikropoulos, 2006; Moreno & Mayer, 2004; Richards et al., 2009). Comparing one VR simulation to another can be complicated because one must consider many internal parameters, such as learning con-

tent, background music, stereoscopic images, duration, and interactions. Therefore, it is necessary to consider the principal characteristics of VR simulations. According to Kavakli & Lloyd (2005), visualisation, interaction and immersion are the three principal characteristics of VR simulations. The first principal characteristic on the list is visualisation. Although recent studies show that interaction (Korakakis, Pavlatou, Palyvos, & Spyrellis, 2009) and music (Richards, Fassbender, Bilgin, & Thompson, 2008) in VR simulations improve learning, visual complexity in VR applications is fundamental to improve learning.

By looking at humans' physiological and cognitive limitations, two problems may arise from visually complex learning materials (see Figure 1.5). First, some studies suggest that increasing visual complexity may increase optic flow and trigger Simulator Sickness (Hettinger, Berbaum, Kennedy, Dunlap, & Nolan, 1990; Ito & Takano, 2004; Mourant, Ahmad, et al., 2007, SS), which impairs learning. Second, literature shows that the extraneous information may overwhelm working memory's capacity in processing and storing information (Baddeley, 1998; Mayer, 2001; Kavakli & Gero, 2003; Lowe, 2004). On one hand, although lowering visual complexity might be a solution, sometimes, complex visuals are necessary in delivering learning material. For example, Raij et al. (2006) and Johnsen et al. (2007) used a VR simulation to train medical students in gaining interpersonal skills by observing the behavioural cues of the virtual patients. On the other hand, visual complexity may enhance presence and Perceived Affective Quality (PAQ) in VR simulations, which increase learners' motivation and interest towards an optimal experience to promote better learning (Csikszentmihalyi, 1990; Takatalo et al., 2008). How should VR system designers develop VR simulations to foster safe and meaningful learning outcomes? This chapter presents the main findings by the author during his PhD candidature, as detailed in Chapter 2-10 and summarised in Table 11.1, to answer the questions as posited in Chapter 1 in assessing the effects of visual complexity to promote retention and transfer.

Table 11.1: The summary of studies conducted, experimental conditions and the correlations

	Study of visual complexity in...	Independent variables	Distance	Display size	Physical H-FOV	Correlation observed
1	realism and SS (Chapter)	VE's visual complexity	approx. 2.7m	6m x 2m	160	• Discomfort increases with visual complexity
2	depth field and SS (Chapter)	VE's visual complexity	approx. 2.7m	6m x 2m	160	• None, however, familiarity with games reduces SS
3	AVAs and SS (Chapter)	AVAs' visual complexity	approx. 2.7m	6m x 2m	160	• No correlations
4	AVAs and learning (Chapter)	AVAs' visual complexity	approx. 2.7m	6m x 2m	160	• No correlations • Presence and PAQ correlate with perception of learning
5	scenes and SS (Chapter)	scene's visual complexity	approx. 600mm	473 mm x 300 mm	43	• No correlation
6	scenes and learning (Chapter)	scene's visual complexity	approx. 600mm	473 mm x 300 mm	43	• No correlation • Presence and PAQ correlate with perception of learning • Presence and perceived affective quality are correlated with perception of learning • Presence and pleasant-unpleasant ratings of perceived affective quality are inversely correlated with perceived difficulty scores • Perceived difficulty scores are negatively correlated with retention and transfer scores • Perceived affective quality along arousal - sleepy ratings are inversely correlated with transfer

11.2 Contribution of the thesis

Contribution of Chapter 2: Avatar creation for believable agents

Kavakli, M., & Kartiko, I. (2007). Avatar creation for believable agents. In Proceedings of the 10th international conference on computer graphics and artificial intelligence (pp. 1 – 6)

Chapter 2 (Kavakli & Kartiko, 2007) demonstrates that the creation of believable AVAs involves complex processes. It may require several 3D modelling and rendering package to create a believable AVA, as it contains 3D mesh information, and a 3D skeleton to drive the mesh, as well as lifelike textures derived from photographs and motion capture data. Believable AVAs have lifelike properties, which contain more visual information. This chapter shows that the template for believable AVAs (for RiskMan simulation) can be achieved by using human templates from Softimage—XSI 3.5 (Avid, 2007). The believable-ness or the lifelike-ness can be achieved by using photographs to aid the 3D modelling in Blender (Blender Foundation, 2011), and by using Gimp (Gimp Development Team, 2011) for texture creation. The paper demonstrates the feasibility of using MotionBuilder software to add the lifelike gait of walking animation into the AVA. Findings are:

F1-1a The polygon count increases when creating a visually complex AVA. It is indispensable to have a polygon budget of each 3D entities to overcome hardware limitations.

F1-2a Commercial packages can be expensive, an important issue to consider in a project. There might be a possibility to achieve 3D modelling, rigging and animation in an open-source software package in future (achieved in Chapter 7, A, 8, 9, 10).

F1-3a Creating AVAs often requires more than one software, and it is essential to optimise file transfer. Using an appropriate file format may eliminate problems arising from importing and exporting files in various 3D applications. For example, the skeleton

information from 3D Studio Max will be read properly by MotionBuilder but not the other way around. The paper also states what went wrong in the development of AVAs. For example, our original plan was to set the skeleton in Blender then export in .3DS format for 3D Studio Max to achieve the .FBX format. This method was not successful, since .3DS format does not carry skeleton information. At the writing of this thesis, the recent version of Blender 2.49b and 2.5x series export to .FBX and other common 3D formats for VR simulations.

Contribution of Chapter 5: Architectural design of multi-agent systems: Technologies and techniques

Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). Riskman: A multi-agent system for risk management. In H. Lin (Ed.), Architectural design of multi-agent systems: Technologies and techniques (pp. 356 – 376). IGI Global.

Chapter 5 (Kavakli, Szilas, et al., 2007) shows that the construction of a believable VE is a straightforward process, compared with the creation of AVAs as a VE does not carry any complex skeleton information for animation. A believable VE consists of 3D mesh information and lifelike textures to depict a scenario. Chapter 5 explains the stages of designing a modular VR system as well as creating a believable VE for RiskMan training simulation, a module of Unreal Tournament 2004 game engine (UT2004). RiskMan demonstrates a modular multi-agent architecture for police training in a safe environment. The VE of RiskMan contains stationary objects. Furthermore, UnrealEditor was the only tool used to create the complex VE (the airport). Because the VE did not come from external 3D applications, there were no problems of file exchange between 3D modelling applications. Creating a believable VE was streamlined in RiskMan with some limitations for its use in a training simulation:

F1-1b The players spawn randomly in the virtual world created by the game engine.

F1-2b RiskMan requires code modifications to enable multi-user collaboration over network.

F1-3b RiskMan depends on UT2004 to run it. UT2004 is a game, not a Software Development Kit (SDK). Therefore, it is necessary to code from scratch to overcome these limitations. Finally, it is planned in the paper to upgrade RiskMan into BOSS (BOrder Security Simulation) by using the Vizard Virtual Reality SDK that allows greater flexibility compared with UT2004.

Question 1: What is the correlation between VE's visual complexity and SS?

(a) Does a visually complex VE induce more SS than a visually simpler VE?

Kavakli, M., Kartiko, I., Porte, J., & Bigoin, N. (2007). Effects of digital content on motion sickness in immersive virtual environments. In Proceedings of the 3rd international conference on computer science and information systems (pp. 1 – 12).

Our findings in Chapter 3 (Kavakli, Kartiko, et al., 2007) state that:

F1-1a Both colourful and visually complex VEs, and black-and-white scenes induce some degree of SS. However, SS is greater in the visually complex scene.

F1-2a This is probably due to the fast-moving camera of the roller-coaster in the scene.

F1-3a Even the black-and-white scene is sufficient to cause SS as it generates high rate of optic flow and induces considerable level of SS.

(b) Does a scene with stronger depth cues induce more SS?

Bigoin, N., Porte, J., Kartiko, I., & Kavakli, M. (2007). Effects of depth cues on simulator sickness. In Proceedings of the 1st international conference on immersive telecommunications (pp. 1 – 4). (ACM SIGMM Technically sponsored by Create-Net and EURASIP) [ERA C].

It is expected that VR developers, especially with stereoscopic displays, would attempt to produce the depth sensation where possible to heighten the sense of presence and enjoyment in VR simulations. Our findings in Chapter 4 (Bigoin et al., 2007) states that:

F1-1b Stronger depth cues, due to countless extrusion of 3D faces, do not induce more SS than the soft landscape with lesser depth cues.

F1-2b Familiarity with computer games reduces the risk of SS. This phenomenon is known as habituation.

The statistical analysis of the effect of familiarity with computer games would have been better carried out by factorial ANOVA test using discomfort as the dependent variable and scene viewed and familiarity with computer games as the independent variables. This would allow further investigations into the interaction between computer games and scene viewed. This would also allow any problems with unequal distribution of game familiarity across the groups to be dealt with. And so would strengthen the findings in level of discomfort of participants viewing each scene.

Question 2: What are the correlations among the order of visual complexity, the level of SS, and the quality of learning?

(a) Does an increase in SS follow the order of AVAs' visual complexity?

Kartiko, I., Kavakli, M., & Cheng, K. (2009b). The impacts of animated - virtual actors' visual complexity and simulator sickness in virtual reality applications. In B. Werner (Ed.), Proceedings of the 6th international conference computer graphics, imaging and visualization (pp. 147 – 152). IEEE Computer Society. (ISBN 978-0-7695-3789-4). [ERA C].

Our findings states that:

F2-1a In ANTECATALYST, as detailed in Chapter 7 (Kartiko et al., 2009b), the increase of SS does not follow the order of AVAs' visual complexity.

F2-2a The results also showed unusually high pre-SS ratings. The author suspects this may be caused by the procedure of the experiment. The pre-SSQ was given to each participant the moment he/she arrived at the VR lab. Some participants came in rush to fill in the pre-SSQ. This may have influenced their level of discomfort prior to learning with ANTECATALYST. Future studies must establish the same "baseline" prior self-assessment with SSQ (see Bruck & Watters, 2009a,b). This could be done by asking the participants to sit and see a still image on the projection screen for 5 or 10 minutes. This would enable the participants to begin at approximately the same level of comfort or discomfort prior pre-SSQ, thus eliminating the issue of unusually high level of pre-SS ratings.

(b) Does an increase in SS follow the order of scenes' (AVAs and VEs) visual complexity?

Kartiko, I., Kavakli, M., & Cheng, K. (2011b). Correlations between simulator sickness and visual complexity of animated-virtual actors and virtual environments. Journal of Computer Animation and Virtual World, 1 – 27. (Submitted on Sept 30, 2011). [ERA B]. Impact factor: 0.629.

We found that:

F2-1b In the study with VEAR, as detailed in 9 (Kartiko et al., 2011b), the increase of SS does not follow the order of scenes' visual complexity.

F2-2b The similar results we obtained in both ANTECATALYST and VEAR is likely due to the slow motion of the virtual camera. Neither VR simulations generate sufficient optic flow to induce a high level of SS. On the contrary with the previous experiment in Chapter 4 (Kavakli, Kartiko, et al., 2007), a simple black-and-white scene is sufficient to induce SS, possibly because of the fast-moving camera of the

roller-coaster simulation.

(c) Does the visual complexity of AVAs increase presence and affective quality ratings?

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. Computers & Education, 55(2), 881 – 891. ISSN: 0360-1315 [ERA A]. Impact factor: 2.190.

The results from ANTECATALYST, as detailed in Chapter 8 (Kartiko et al., 2010), show that:

F2-1c The increase of AVAs' visual complexity does not increase presence and PAQ ratings in VR simulations.

(d) Does an increase in the AVAs' visual complexity affect learning?

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. Computers & Education, 55(2), 881 – 891. ISSN: 0360-1315 [ERA A]. Impact factor: 2.190.

There are three learning outcomes of interest in this project: (1) Program Ratings (PR) that gives perception of learning and perceived difficulty, (2) retention scores and (3) transfer scores. PR assesses perception of learning – how motivating, interesting, and helpful the content is – and perceived difficulty of learning material. In other words, PR gauges the subjective experience of learning. However, Mayer (2001) states that retention and transfer tests are the indicator whether a learning system "works." In the study with ANTECATALYST, as detailed in Chapter 8 (Kartiko et al., 2010) and Appendix A (Kartiko et al., 2009a), we found that:

F2-1d The increase of AVAs' visual complexity does not affect perception of learning, perceived difficulty, retention and transfer scores. It is important to note that the data reveals ceiling effects on the post-test retention scores. This could be an indicator that the retention test was not sufficiently difficult to allow differences between groups to be measured.

F2-2d The participants reported moderately high perception of learning and average level of learning difficulty in all conditions (simple, cartoon and lifelike AVAs).

F2-3d After exposure to ANTECATALYST, the results show a significant increase in retention and transfer scores.

(e) Does an increase in the scenes' (AVAs and VEs) visual complexity affect learning?

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. IEEE Transactions on Affective Computing, 1 - 14. (Submitted on Oct 24, 2011)

In the study with VEAR of higher visual complexity, as detailed in Chapter 10 (Kartiko et al., 2011a), we found that:

F2-1e There is no significant difference between experimental groups in perception of learning.

F2-2e Increasing scenes' visual complexity from simple world to AVAs condition decreases perceived difficulty.

F2-3e Increasing scenes' visual complexity from simple AVAs to lifelike condition decreases retention.

F2-4e There is a significant increase in retention score after attending the VR simulation.

F2-5e There is no significant difference between experimental groups in transfer scores.

F2-6e All experimental groups also showed significant improvements in transfer after

watching the VR simulation, except those who attended with visually complex AVAs and VEs.

F2-7e There is an evidence that the application of principles derived from CTML leads to improved learning.

(f) Does an increase in the scenes' (AVAs and VEs) visual complexity increase presence and affective quality ratings?

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. IEEE Transactions on Affective Computing, 1 – 14. (Submitted on Oct 24, 2011)

We found that:

F2-1f Increasing scene's complexity from simple to simple AVAs conditions and from simple world to simple AVAs conditions increases presence.

F2-2f Increasing scenes' visual complexity from simple to simple AVAs condition, and from simple to lifelike condition increases perceived affective quality on sleepy-arousal ratings.

(g) Does presence correlate with perceived affective quality?

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. Computers & Education, 55(2), 881 – 891. [ERA A]. Impact factor: 2.190.

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. IEEE Transactions on Affective Computing, 1 – 14. (Submitted on Oct 24, 2011)

Regarding the correlation between presence and perceived affective quality, we found that:

F2-1g The results with ANTECATALYST, as shown in Chapter 8 (Kartiko et al., 2010) and Appendix A (Kartiko et al., 2009a), clearly show that presence correlates with PAQ in VR applications. The results suggest that these two variables of VR simulations measure a common component of flow.

F2-2g The results with VEAR, as shown in Chapter 10 (Kartiko et al., 2011a), also indicate that these two variables have significant correlations, and also measure a common component of flow.

(h) Do the sense of presence and perceived affective quality affect the learning outcomes?

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. Computers & Education, 55(2), 881 – 891. [ERA A]. Impact factor: 2.190.

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. IEEE Transactions on Affective Computing, 1 – 14. (Submitted on Oct 24, 2011)

In investigating the effects of presence and perceived affective quality on learning, our findings state that:

F2-1h In the first experiment with ANTECATALYST, as shown in Chapter 8 (Kartiko et al., 2010), presence and perceived affective quality correlate with perception of learning.

F2-2h In the second experiment, as shown in Chapter 10 (Kartiko et al., 2011a), presence and perceived affective quality are also correlated with perception of learning.

F2-3h The results with VEAR also indicate that presence and pleasant-unpleasant ratings of perceived affective quality are inversely correlated with cognitive load scores. These findings are in-line with flow theory: presence elevates users' engagement and makes tasks easier. However, there is no correlation among these components of flow towards retention and transfer scores.

F2-4h The results with VEAR also indicate that the cognitive load scores are negatively correlated with retention and transfer scores, according to the prediction of CTML.

F2-5h As noted in section 1.5.1, both ANTECATALYST and VEAR do not offer direct interaction. Since literature suggests that direct interaction is a substantial factor in presence and ow, the author suspects this could be a possible reason for obtaining no correlation between presence, perceived affective quality with learning outcomes in both main studies. This issue demands further investigations.

11.3 Future Directions of research

This thesis seeks to bridge gaps between studies in disciplines of computer science, educational, cognitive-affective psychology, digital arts, and ergonomics in addressing the effects of complex visual materials. It aimed at a unified approach to foster learning approach with VR technology. The chapters in the second part of this thesis, collectively, address visual complexity in VR simulations by drawing on studies on Simulator Sickness (Kennedy et al., 1993, SS), on Cognitive Theory of Multimedia Learning (Mayer, 2001, 2003, CTML), and on presence (Witmer & Singer, 1998) and Perceived Affective Quality (Russell & Pratt, 1980, PAQ) within the framework of flow theory (Csikszentmihalyi, 1990). Although in general the results favour the applications of Coherence Principle derived from CTML, there are more areas to be explored about the content of VR simulations that account for visual complexity. This thesis only considers the visual complexity of AVAs and the VEs in assessing visual complexity.

Each chapter in part II of this thesis contains conclusion and the directions of future research. This section gives the future directions of research in general. These directions would allow VR system designers to assess visual complexity in greater detail to develop VR simulations to support learning.

1. **Cinematography**. Cinematography or the art of capturing the scene by the virtual camera is an encouraging direction of research as it may hinder or promote learning in VR simulation.

   The movement of the virtual camera is one of the contributors to complexity, apart from the subject of learning, and the visual complexity of AVAs and the VEs. However, a fast-moving virtual camera may hinder learning as learners may not have enough time to grasp the continuous streams of information (Tversky et al., 2002). Furthermore, a high-speed camera creates high optic flow, with fast turning movements; the resulting SS is likely to stop learning (Kavakli, Kartiko, et al., 2007; Mourant, Rengarajan, et al., 2007).

   At the same time, cinematography may promote learning. The virtual camera does not always show the entire scene or scenario at any once, instead, moving from one area to another to focus on the object or scene of interest. This concept is similar to framing, masking, zooming (Hongpaisanwiwat, 2006), and segmentation (Moreno, 2007), techniques that guide users' attention to the relevant material. These techniques are common in the production of movies (Simon, 2000; Begleitter, 2001).

2. **Quantifying Visual Complexity in VR simulations**. This topic for future research direction follows the cinematography of the VR simulations. Comparing the degree of visual complexity between images can be achieved by comparing the file-size of the images (Donderi, 2006). Drawing from the recent studies of Oliva, Mack, Shrestha, & Peeper (2004) and Roberts (2007) on still images, there are at least ten dimensions of visual complexity, which include quantity of objects, clutter, openness, symmetry, variety of colours, unintelligibility, disorganisation, amount

of elements, variety of elements, and three-dimensionality. The challenge is how do we compare quantitatively between viewing VEAR and ANTECATALYST at 8 minutes each? Depending on the display hardware, there are many factors that may influence visual complexity, such as screen resolution, screen size and stereoscopic images. A concrete comparison enables VR designers to gauge the visual complexity of VR simulations in assessing their severity in hindering learning.

Future research on these topics would allow VR system designers to assess the visual complexity in greater detail, before adding more advanced features (e.g., interactivity and additional visual cues) to help learning. VR simulations for training are favourable when they promote retention and transfer skills.

Part IV

Appendix

Appendix A

Does the visual complexity of animated-virtual actors in virtual reality applications affect learning?

Kartiko, I., Kavakli, M., & Cheng, K. (2009a). Does the visual complexity of animated-virtual actors in virtual reality applications affect learning? Presented at the 9th Meeting of the Australasian Society for Cognitive Science. Sydney. (paper was presented, but withdrawn - not published). [ERA C].

The publication was withdrawn from Australasian Society for Cognitive Science 2009, but became the foundational work of other published article (see Chapter 8). This chapter presents the abstract and the powerpoint presentation given during the conference.

Abstract. We explore the application of Cognitive Theory of Multimedia Learning (CTML) in Virtual Reality (VR) applications. CTML suggests that complex visuals hinder novice learners to attend the lesson properly. However, VR applications favour the use of complex visuals as the technology in computer graphics allows it. So, how much of visual complexity is too much? This study specifically investigates into the visual complexity of Animated Virtual Actors' (AVAs) for learning. A series of VR simulations were created to teach second-year psychology students about the navigational capabilities

of Cataglyphis ants with flat, cartoon, or life-like AVAs. Regarding learning outcomes, we predicted that visually more complex AVA would increase the perception of learning and the cognitive load, and decrease retention and transfer scores. The results did not reveal any significant differences between AVA's visual complexity and learning outcomes, contradicting our predictions. Nevertheless, our methodology significantly improved retention and transfer scores in all conditions. Possible explanations and future research direction are discussed.

Does the Visual Complexity of Animated-Virtual Actors affect learning in Virtual Reality Applications?

ASCS2009
Sydney, Australia

Iwan Kartiko
Dept. of Computing – Macquarie University

Supervisors:
Manolya Kavakli and Ken Cheng

Introduction: Examples of Animated-Virtual Actors



© 1989, Nintendo © 2009, Rori (Daz3D Gallery Jan2009) © 2008, KOEI © Yasuhiro Nightow

Introduction: Background

- Ever increasing visual complexity in Virtual Reality (VR) applications
- Cognitive Theory of Multimedia Learning (Mayer 2001, CTML) shows that '*seductive*' details impair learning (Harp & Mayer, 1997, 1998; Mayer et al., 2001)

Thesis question & predictions according to CTML

- Does the visual complexity of AVAs affect learning?

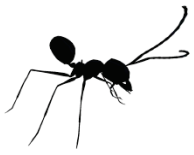
Table 1
The predictions of learning outcomes.

Learning outcomes	Predictions	Rationale
1. Perception of learning	Increases with AVAs visual complexity	Humans have preference over complex visual of many elements.
2. Learning difficulty	Increases with AVAs visual complexity	Humans have limited processing capacity.
3. Retention	Decreases with AVAs visual complexity	Difficulty of learning impairs memorisation.
4. Transfer	Decreases with AVAs visual complexity	Difficulty of learning impairs problem-solving.

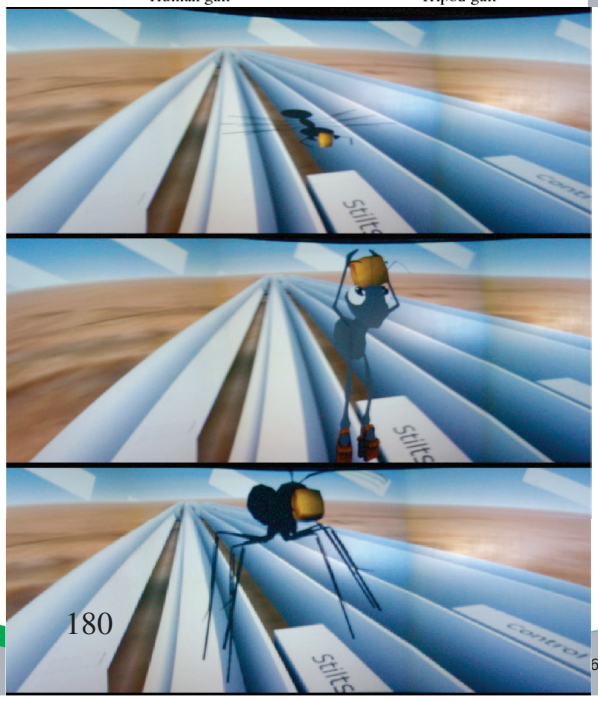
Methods

- *ANTECATALYST* – Animated *Cataglyphis* Ants, between group design
- Integrated into PSY236 in 2008



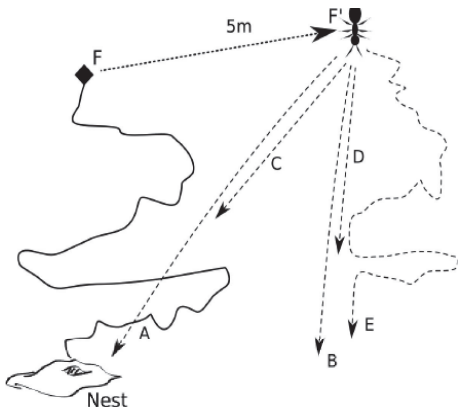
	Types of AVA		
	Flat	Cartoon	Life-like
			
Polygons	1	1755	2415
3D shape	None	Anthropomorphic	Life-like
Animated joints	None	18	25
Animation Type	None	Human gait	Tripod gait

AVA's visual complexity

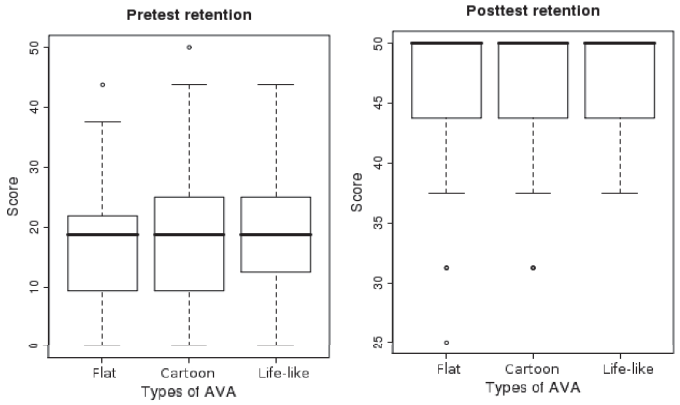


Retention and Transfer tests

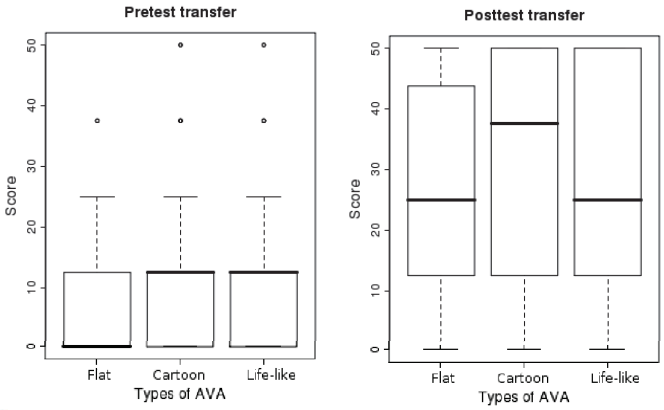
- Multiple choice test
- Based on material presented



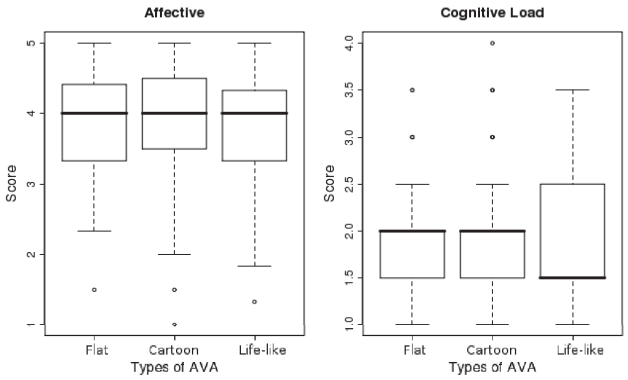
Results of Retention Tests



Results of Transfer Tests



Results of Affective and Cognitive Load Ratings

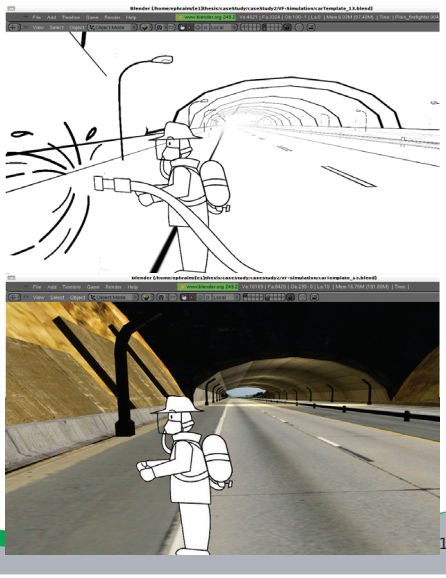


Discussion

- The visual complexity of AVAs does not affect learning outcomes
- Possible explanations:
  - Only AVAs' visual complexity manipulated
  - Only contained the required materials
  - Camera moved slowly
  - AVAs did not appear on each scene

Future Work

- Manipulate both AVA's and virtual environment's visual complexity
- Firefighting scenario



The end

- Thank you for listening!
- Any questions?



References

Harp, S. F., & Mayer, R. E. (1997). The role of interest in learning from scientific text and illustrations: On the distinction between emotional interest and cognitive interest. Journal of Educational Psychology, 89(1), 92–102.

Harp, S. F., & Mayer, R. E. (1998). How seductive details do their damage: A theory of cognitive interest in science learning. Journal of Educational Psychology, 90(3), 414–34.

Mayer, R. E.(2001). Multimedia learning (R. E. Mayer, Ed.). Cambridge Press. (ISBN 9780521787499)

Mayer, R. E., Heiser, J., & Lonn, S. (2001). Cognitive constraints on multimedia learning: When presenting more material results in less understanding. Journal of Educational Psychology, 93(1), 187–98.

Roberts, M. N. COMPLEXITY AND AESTHETIC PREFERENCE FOR DIVERSE VISUAL STIMULI. Departament de Psicologia, Universitat de les Illes Balears, 2007

Appendix B

Testing Platforms and Instructional Animations

This chapter outlines the development of ANTECATALYST and VEAR by the author for examining the effects of visual complexity in learning materials on SS and learning. The other VR simulations (Snow Simulation, Trenches Simulation, Roller-coaster, and BOSS) started before the PhD candidature of the author. Moreover, the author believe that Chapter 2, 3, 4, and 5 sufficiently describe the implementations of the other VR simulations. The author was not working alone as the only Research Associate in the studies conducted by A/Prof. Manolya Kavakli on SS (see Chapter 3 and 4) and learning with VR technology (see Chapter 2 and 5). There were also others researchers contributing to the development of those platforms.

This chapter explains the development of two animated Virtual Reality (VR) simulations known as ANTECATALYST (ANimaTEd CATAgLYphiS anTs) and VEAR (VEhicle Accident and fiRes). ANTECATALYST utilises Vizard 3D engine (Worldviz, 2010), while VEAR runs on the Blender 3D engine (Blender Foundation, 2011). This chapter demonstrates that both 3D engines could create VR simulations for examining the impacts of visual complexity on Simulator Sickness (SS) and learning outcomes. Although both 3D engines allow quick development and deployment of VR simulations, Vizard 3D

engine has a high-price tag to run the simulation full-screen with clustering computers to achieve the immersive display. Additionally, it requires some knowledge of coding in Python scripting. On the other hand, Blender 3D engine is open-source, but has a steep learning curve, and it requires some knowledge in 3D modelling, as well as Python scripting. Given the short project time-frame, limited budget, and familiarity with Blender and Python scripting, the author has chosen Blender.

B.1 Introduction

Since the goal of this thesis is to examine the impact of visual complexity on Simulator Sickness (SS) and learning outcomes, first, we need to accommodate a dynamic virtual camera that travels from one place to another to present the learning content (e.g., Chapter 3 and 4, Bigoin et al., 2007; Kavakli, Kartiko, et al., 2007). Second, the content must display various levels of visual complexity in depicting a scenario. The author could not use either RiskMan (Risk Management) or BOSS (BOrder Security Simulation). Like most studies in instructional animation and VR, they employ a stationary camera (see also Moreno & Mayer, 2001, 2002, 2004; Richards et al., 2009). Therefore, it is impossible to assess the effects of SS. This is due to the nature of the subject where the camera has limited movement and mostly stays on one space of interest or focuses on an AVA most of the time (similar to the mathematical addition experiment conducted by Lusk & Atkinson, 2007). The learning scenarios of RiskMan and BOSS contain mostly stationary cameras (see Richards et al., 2009; Dras et al., 2010), as the learners watch conversations and actions between AVAs in a fixed space.

Moreover, RiskMan and BOSS have insufficiently complex and redundant visual details as commonly found in the recent VR applications (see also Moreno & Mayer, 2002, 2004; Buisine & Martin, 2007; Wang et al., 2008) to promote learning. For example, in the study by Moreno & Mayer (2004), extraneous details were excluded to alleviate the cognitive load in working memory. RiskMan and BOSS also present the scenarios

concisely in VEs that resemble customs inspection areas, excluding extraneous visual information such as moving planes, rolling advertisements, luggage carousel, and other post-processing visual enhancements commonly used in the entertainment industry (see Trenholme & Smith, 2008). The accessibility of the guidelines for the development of virtual CPTED (Kartiko & Kavakli, 2008, Crime Prevention Through Environment Design), as detailed in Chapter 6, was limited. This forced us to developed other simulations to test our research questions.

This chapter focuses on the production of non-interactive VR simulations called ANTECATALYST (ANimaTEd CATAgLYphiS anTs) and VEAR (VEhicle Accident and fiRes) that overcome the limitations of RiskMan, BOSS and virtual CPTED to investigate the impacts of visual complexity, SS and learning outcomes. Both ANTECATALYST and VEAR contain a dynamic virtual camera and greater extraneous visual details than the others. The chapter begins with a brief description of developing ANTECATALYST and VEAR, and ends with a general discussion and future recommendation.

B.2 ANTECATALYST

The topic of ANTECATALYST is the navigational behaviour of *Cataglyphis* ants on the featureless desert. A 0.1 mg brain in the 10-mg ant deploys sophisticated navigation strategies (Wehner, 1999, 2003). With repeated experience along a path, visual cues (Schatz, Chameron, Beugnon, & Collett, 1999; Bisch-Knaden & Wehner, 2003; Knaden & Wehner, 2005; Knaden, Lange, & Wehner, 2006) or olfactory cues (Wolf & Wehner, 2000, 2005) along the route are used to set up route-based navigation. When these cues are removed, path integration is the primary means of navigation for desert ants over featureless terrain (Wehner, 2003; Narendra, Cheng, & Wehner, 2007). Path integration uses information from both the skylight compass as angular component of movement, and the odometer or step counter, and informs the ant about its current position relative to its point of departure(Wehner, 2003; Wittlinger, Wehner, & Wolf, 2006, 2007). *Cataglyphis* ants

could return (homebound journey) in an almost straight path after finding a food item on the desert plain. In ANTECATALYST, the students were expected to learn and understand path integration, the navigation behaviour of *Cataglyphis* in featureless plain. The lesson is organised into five continuous scenes (as detailed in Section C.1 and D.1):

1. **Introduction**. The simulation gives an introduction to the study and shows some *Cataglyphis* ants foraging for food items on the desert floor.
2. **Overview of path integration**. The second scene briefly explains that path integration is a mode of navigation by integrating skylight patterns used to keep directions and not visible to human eyes. The information from an internal odometer keeps track of distance travelled. This scene shows a red line that marks the ant's ability of returning to its nest in an almost straight line by adding vectors travelled during the ant's outbound journey.
3. **Skylight compass**. The third scene briefly shows what the skylight pattern might look like to were we sensitive to polarised light.
4. **Step counting behaviour**. The fourth scene shows an experiment on three ants being trained to gather food items along narrow and straight channels. Before performing the homebound journey, three experimental conditions are set: the legs of the first ant were left unmodified, the legs of the second were lengthened by using stilts (stilts condition), the legs of the third ant were shortened (stumps condition). This scene ends by showing that the ants with unmodified legs estimated the homebound distance correctly, while the ants on stilts and stumps overestimated and underestimated the homebound distance, respectively. Thus, it shows that *Cataglyphis* ants has internal odometer that count steps during outbound and homebound journey.
5. **Summary.** The last scene gives a summary of the entire lesson to reinforce learning.

Table B.1: Visual complexity between each experimental condition in ANTECATALYST.

			Experimental Condition		
			Flat	**Cartoon**	**Lifelike**
	AVAs	Shape	Flat - 2D, 1 polygon	3D - Humanoid	3D - Lifelike
		Colour	Black	Grey	Black
		Animations	None	Human gait	Tripods gait
Environment	**Tunnels**	Shape	3D	3D	3D
		Colour	White	White	White
		Details	Static shadow	Static shadow	Static shadow
	Food items	Shape	3D	3D	3D
		Colour	Full colour	Full colour	Full colour
		Detail	Bread-like texture	Bread-like texture	Bread-like texture
	Sky	Shape	3D cube	3D cube	3D cube
		Colour	Clear sky	Clear sky	Clear sky
	Lighting	Lightsource	None - Constant tex. value	None - Constant tex. value	None - Constant tex. value
	Landscape	Shape	Flat	Flat	Flat
		Colour	Full colour	Full colour	Full colour
		Details	Desert-like, plain	Desert-like, plain	Desert-like, plain

In ANTECATALYST, the visual complexity of AVAs can be varied from flat drawings to 3D lifelike ants with their complex gait animations (Table B.1). Each AVA contains idle, foraging, and home-bound animations. Figure 1.3 shows types of ants designed for the simulation. In addition, Table B.1 shows the detailed components of the simulation. The development of ANTECATALYST started with a screenplay (Section D.1) and storyboard (Section C.1), based on aforementioned scientific studies and observations by researchers in the domain of desert ants' navigation.

Blender was used for 3D modelling and animation, Vue (e-on software, 2006), Gimp (Gimp Development Team, 2011) and Inkscape (Inkscape Development Team, 2011) for texturing the virtual scene. Figure B.3, B.6, B.9 and B.11, and the supplementary movies[1] show the content of the scene, such as AVAs, plain sand throughout, a clear blue sky, simulated polarisation patterns in the sky, food items and three test-channels.

Vizard displays ANTECATALYST on a 6m-wide canvas, an immersive semi-cylindrical projection system that provides a 160-degree field of view (FOV). The simulation is about 8 minutes long and the virtual camera moves slowly from one place to another in capturing the behaviour of the *Cataglyphis* ants. The AVAs are visible in all scenes, except in scene 3.

B.2.1 Virtual Environment

The creation of the desert was as straightforward as the creation of the VE in RiskMan (Chapter 5, Kavakli, Szilas, et al., 2007). A random terrain was generated by using L3DT (Torpy, 2011), with its texture. This texture was then placed on a flat polygon in Blender. This polygon was duplicated many times to create the effect of vastness of the desert (Figure B.1 and B.3). To enhance the vastness effect of the desert, distant haze (fog) was added in the scene (Figure B.3).

The shadow effect in the tunnel was achieved by "baking" the sunlight information

[1]The field of view of the video clips was modified to fit three-screens of 6m x 2m size canvas into one 3:4 screen ratio for viewing purpose on normal desktop screens.

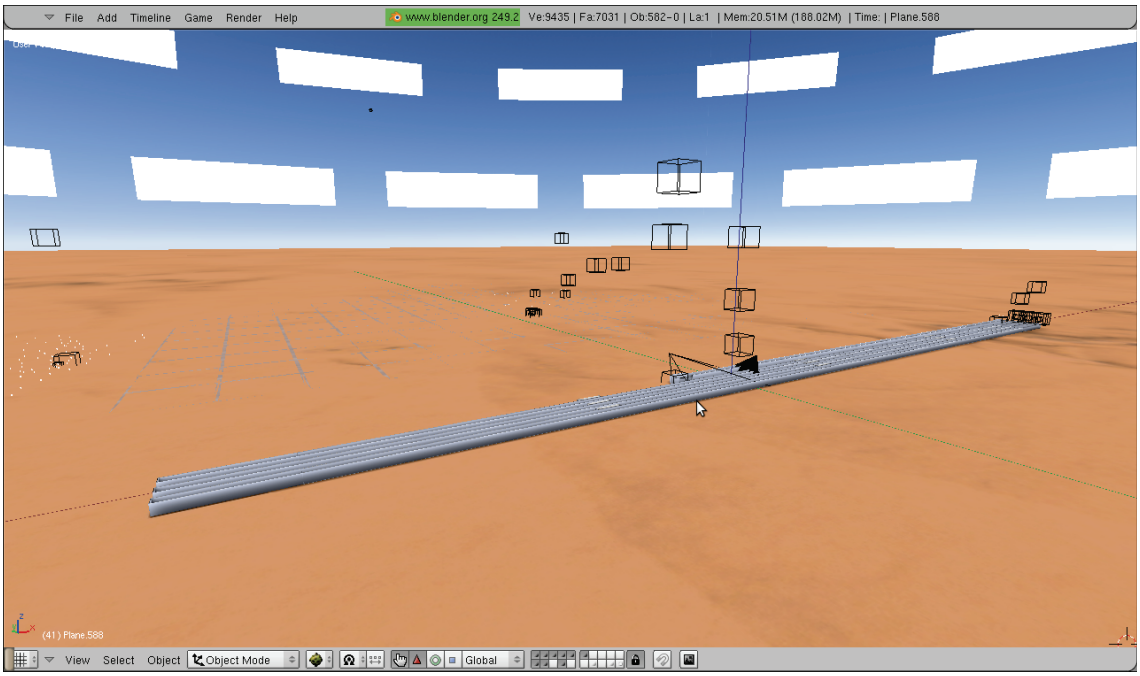


Figure B.1: The desert with a clear blue sky, test channels and a skylight pattern as shown in Blender with many wire-frame boxes, which served as camera positions and camera target points.

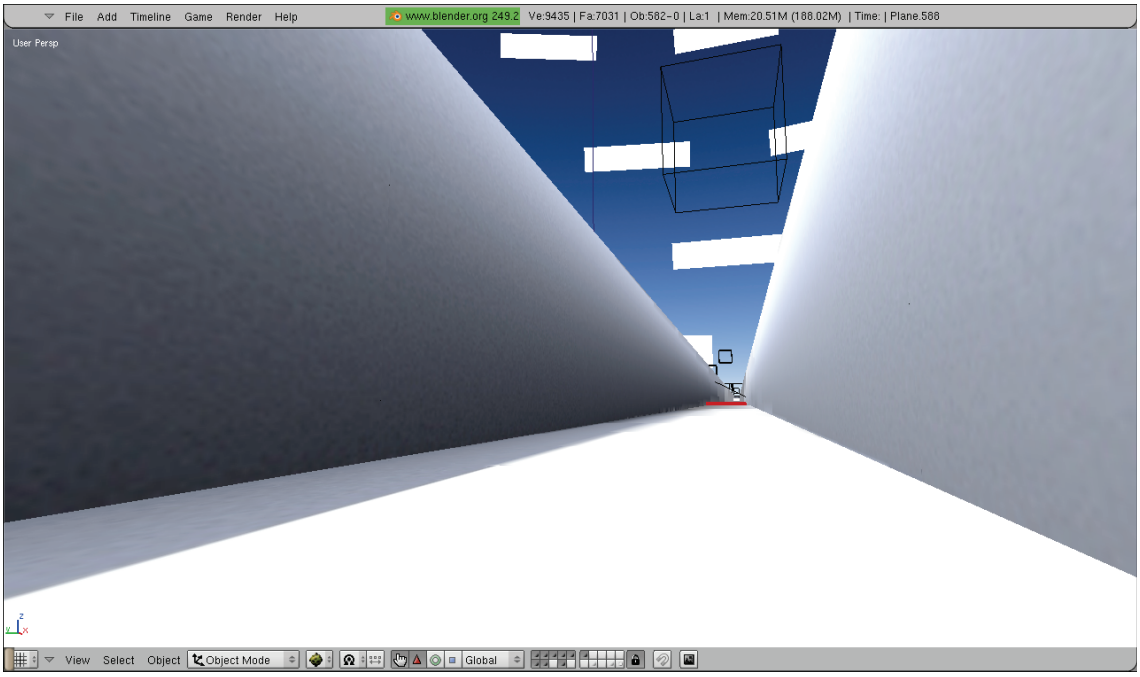


Figure B.2: The view of the virtual world from the test channel.

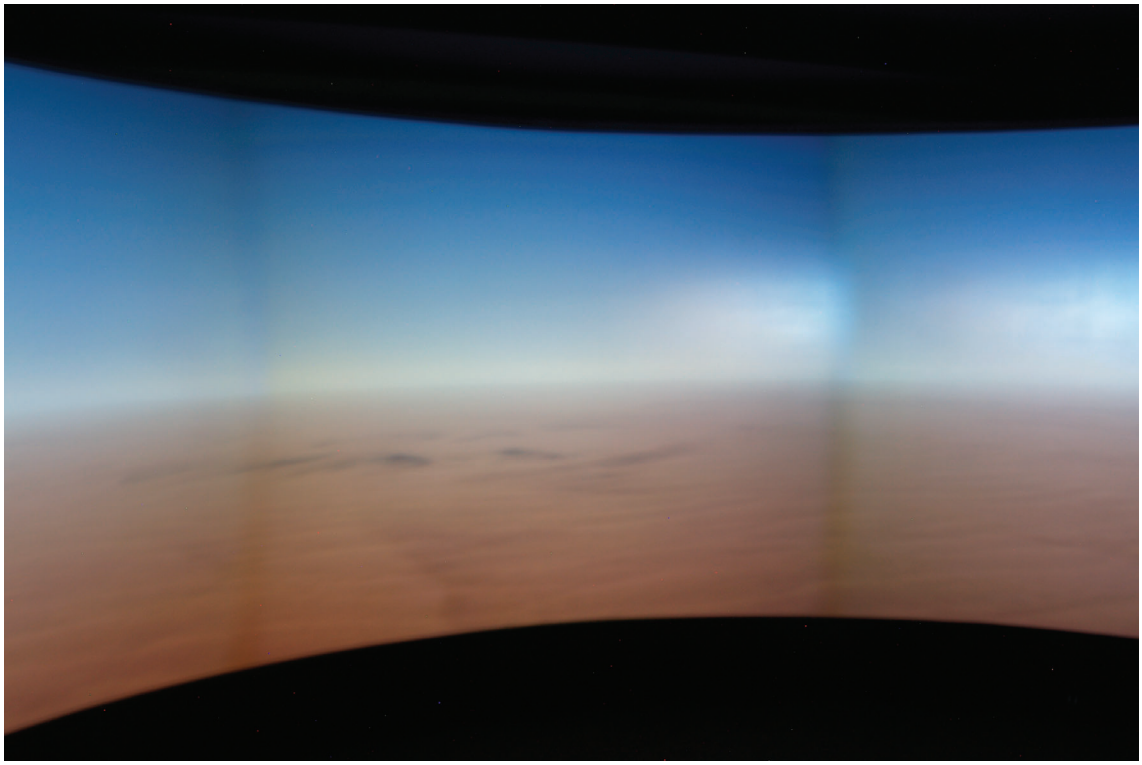


Figure B.3: The desert as rendered in VR lab by using Vizard.

on the textures of the tunnels (Figure B.2). All objects were orderly arranged in Blender before exporting them out in OSG file format that Vizard reads. The movements of the virtual camera was achieved by positioning the camera along the “place-holders” boxes in the scene (Figure B.1).

Panoramic sky was achieved by loading 6 images generated by Vue 6 Esprit edition (e-on software, 2006) into Vizard’s ”skydome.dml” module to create a seamless sky-cube or a skybox. Vue 6 Esprit is a landscape, sky and animation rendering package that has been used widely in movie productions. Vizard requires the filename of the images to be tagged as shown in Table B.2. Determining which camera position correspond to which tags can be difficult, after some trial and error, these camera settings of pitch, roll and yaw, as shown in Table B.2, create a seamless sky in Vizard. It was necessary to change the lens setting of the render camera in Vue from 35 mm, down to 18 mm, otherwise the images would not blend properly.

Table B.2: Filename tags and camera settings for a skybox in Vizard.

Tag on filename	Pitch	Roll	Yaw
*filename*_posz.bmp	$90.0°$	$0.0°$	$0.0°$
*filename*_posx.bmp	$90.0°$	$0.0°$	$-90.0°$
*filename*_negz.bmp	$90.0°$	$0.0°$	$-180.0°$
*filename*_negx.bmp	$90.0°$	$0.0°$	$-270.0°$
*filename*_posy.bmp	$0.0°$	$0.0°$	$0.0°$
*filename*_negy.bmp	$180.0°$	$0.0°$	$0.0°$

B.2.2 AVA 1: ANTique

ANTique was the codename of the flat ant model. As shown in figure B.4, Antique was the extreme simplification of lifelike *Cataglyphis* ants. This model was quick and easy to make, having neither a 3D model nor animations. As shown in figure B.5, the flat AVA was made from one polygon. The smaller rectangular polygons on the sides could be taken away to create the illusion of stumps condition. The long strips of polygons on the sides of the ant represented the flat stilts for the stilts condition. This model contained a set of armature with three bones. The smallest bone at the mandibles was the anchor point to attach the food during simulation. The other horizontal bone at the body controlled the rotation of the flat ants. The vertical bone was the root bone, the parent of other bones.

B.2.3 AVA 2: ANThony

ANThony was the codename of the cartoon ant model. ANThony was the simplification of lifelike *Cataglyphis* ants. Similar to the creation process of ANTique, it started with sketches of ANThony in each experimental conditions as shown in figure B.7. These sketches then were used as a reference to create the 3D model of ANThony (Figure B.5). The running animation was achieved by using a template by Williams (2002, pp.192, 193).

This ant contained more redundant visual information compared with flat AVAs, such as running animation with bouncing antennae and gaster. A pair of Japanese sandals was

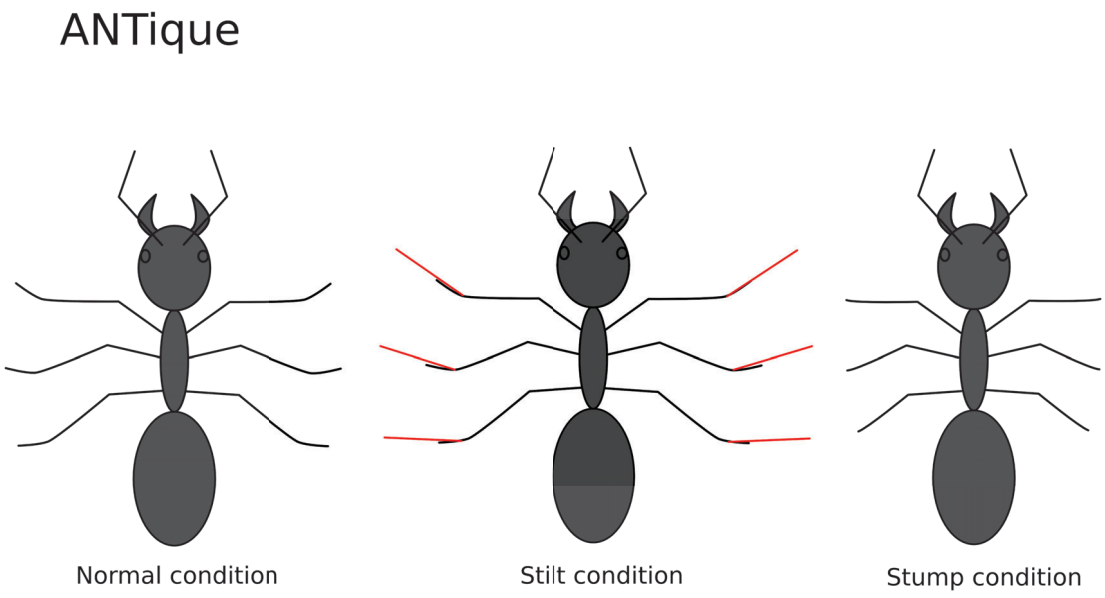


Figure B.4: The sketch of ANTique: the flat AVAs in ANTECATALYST on stumps and stilts.

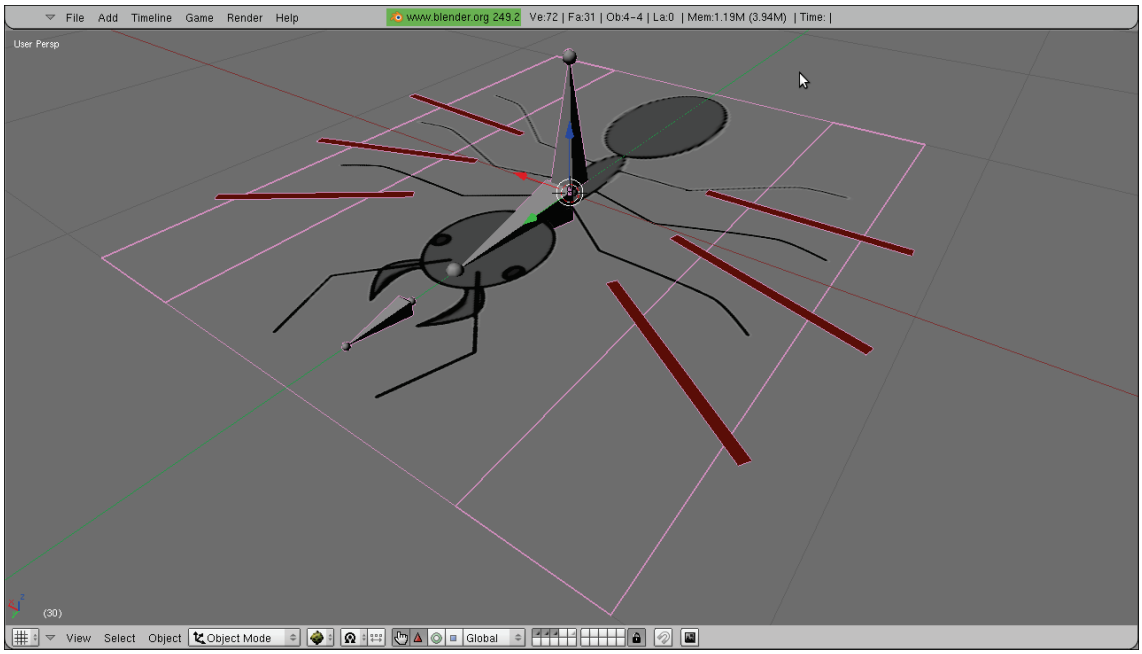


Figure B.5: ANTique: The flat AVAs as shown in Blender.

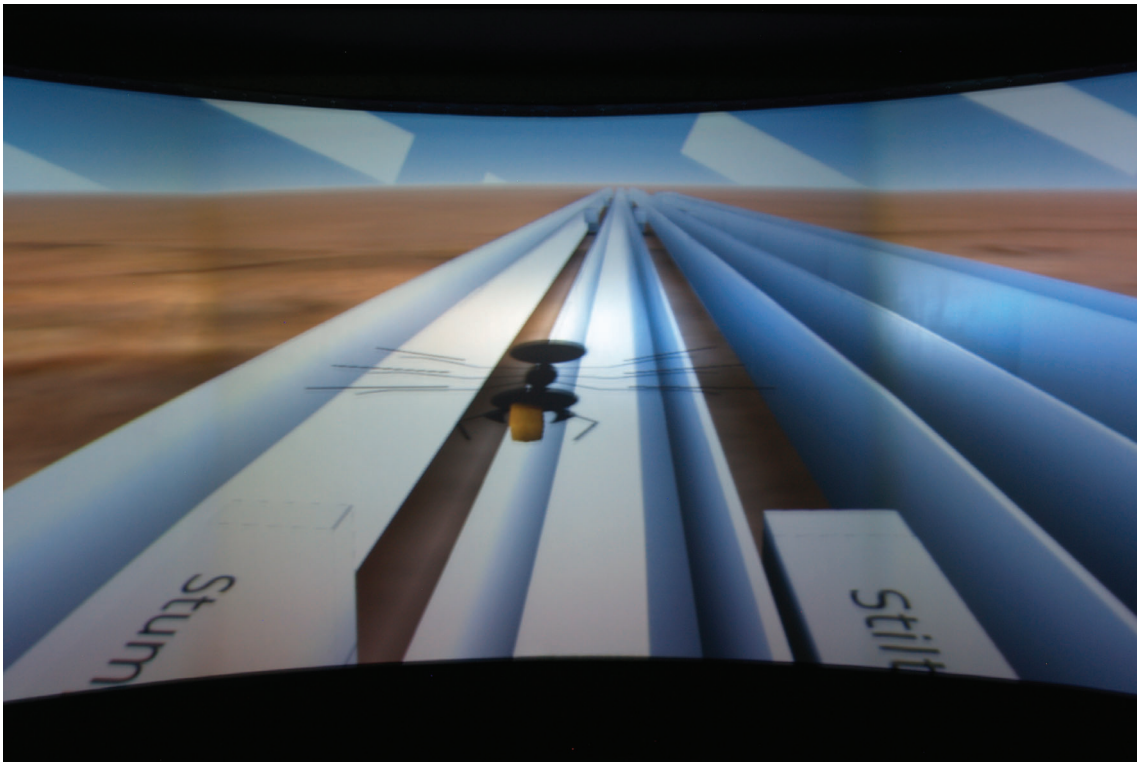


Figure B.6: ANTique: The flat AVAs as rendered in VR lab by using Vizard.

used instead of using plain stilts to extend the legs. Figure B.9 shows the cartoon ant as rendered in Vizard in the VR lab.

B.2.4 AVA 3: ANTlike

ANTlike was the codename of the lifelike *Cataglyphis* AVAs. ANTlike was modelled after the shape of *Cataglyphis* ants. Image references used to create ANTlike were gathered mainly from publications by Zollikofer (1994); Wehner (2003); Wittlinger et al. (2006, 2007) and Fisher (2010). These publications provided the photographs of the experimental set-up on the ants and the anatomy in high resolution.

ANTlike contained the most redundant visual information compared with flat or cartoon AVAs. ANTlike had complex 3D shape, animated mandibles, and running animation with moving antennae and tripods gait (Figure B.11). Compared with ANTique and ANThony, it took longer to create ANTlike due to its complex shape and tripods gait anima-

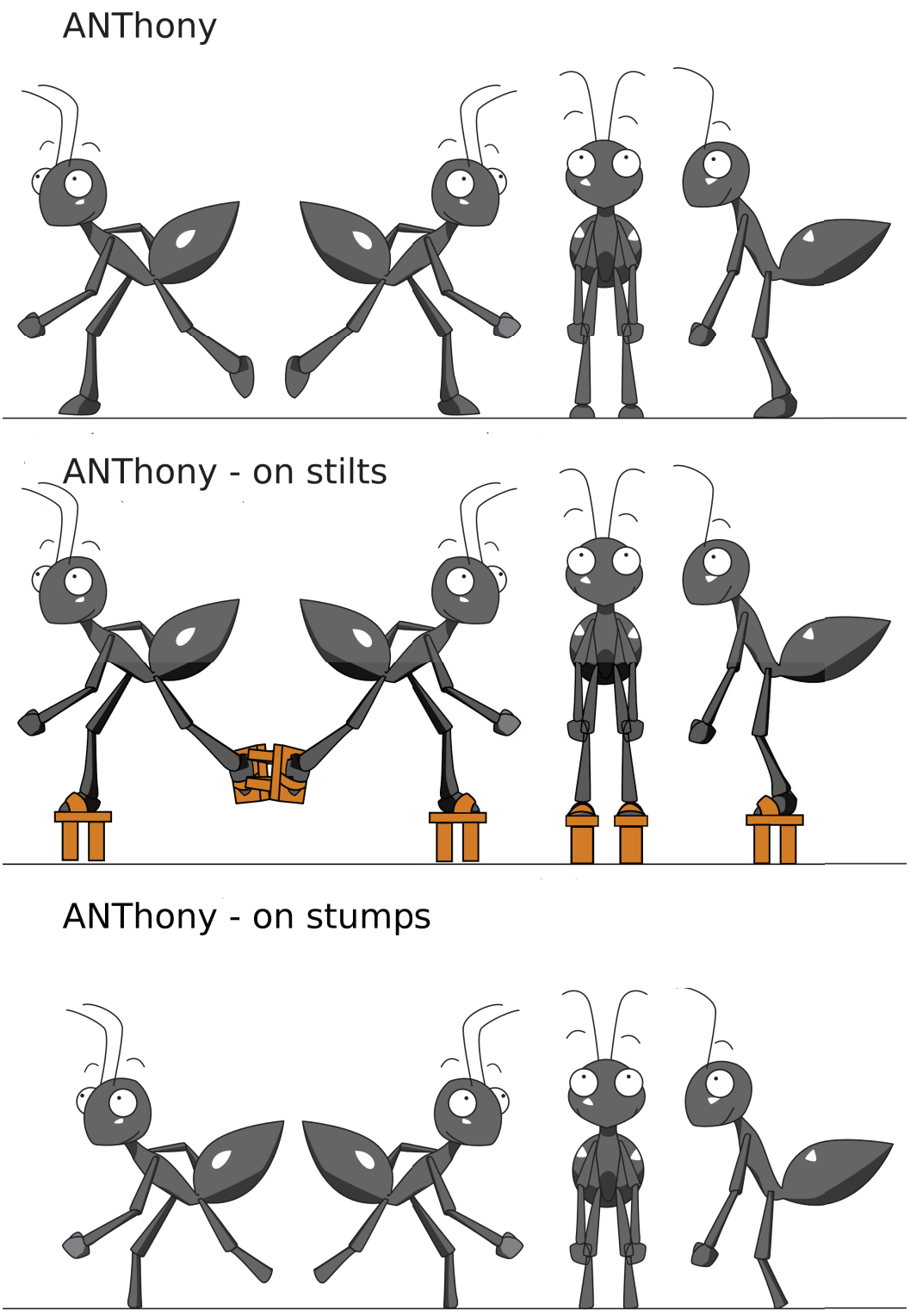


Figure B.7: The sketch of ANThony: cartoon AVA, on all experimental condition.

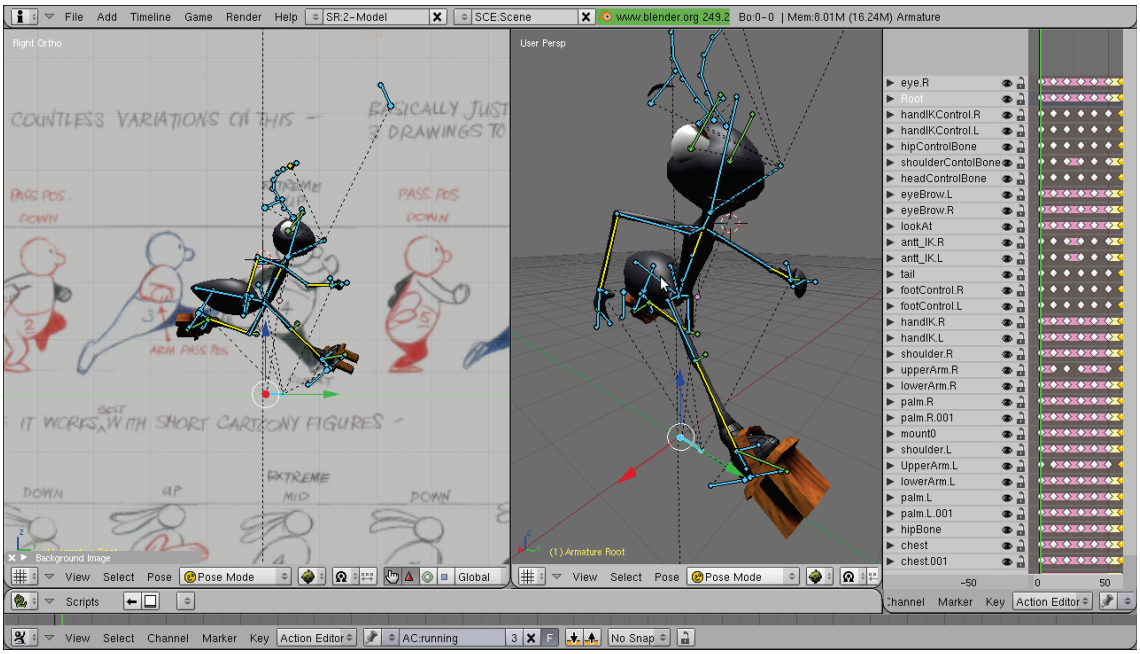


Figure B.8: ANThony: The cartoon AVA as shown in Blender with a pair of Japanese sandals as the stilts. It shows the rotoscoping technique by putting a reference image at the back of the 3D model to trace or duplicate animations from an animation handbook by Williams (2002, pp.192, 193).

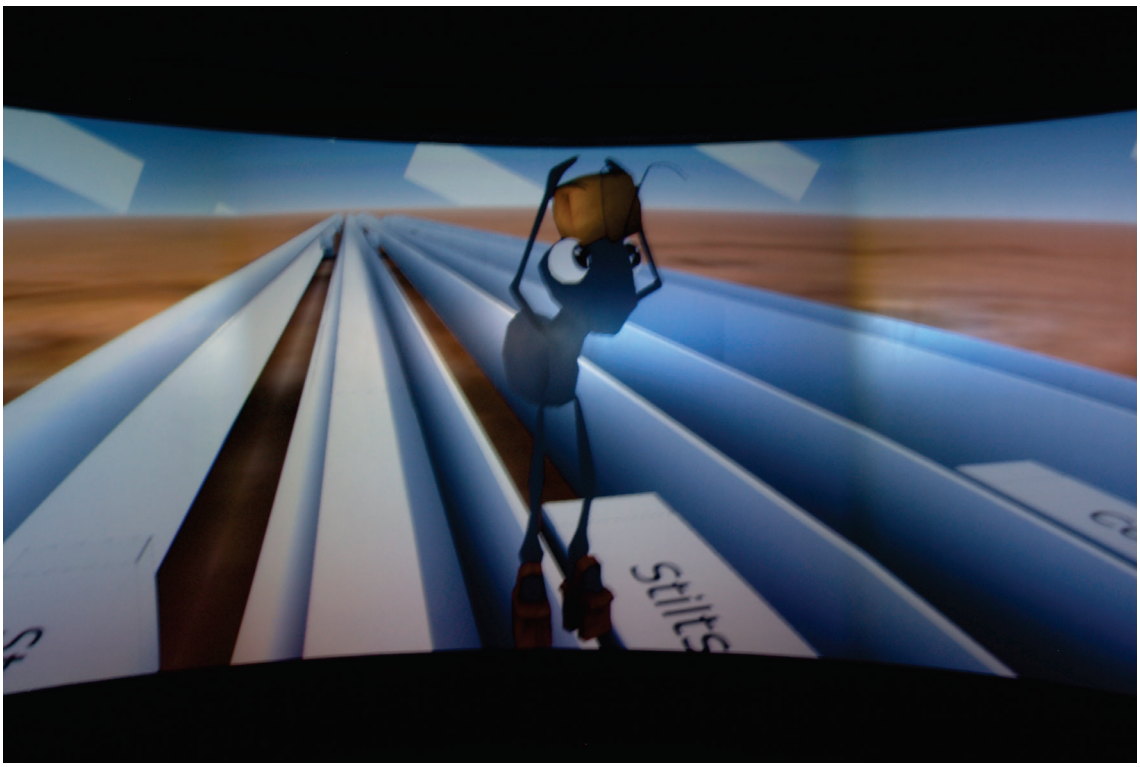


Figure B.9: ANThony: The cartoon AVA as shown in the VR lab by using Vizard.

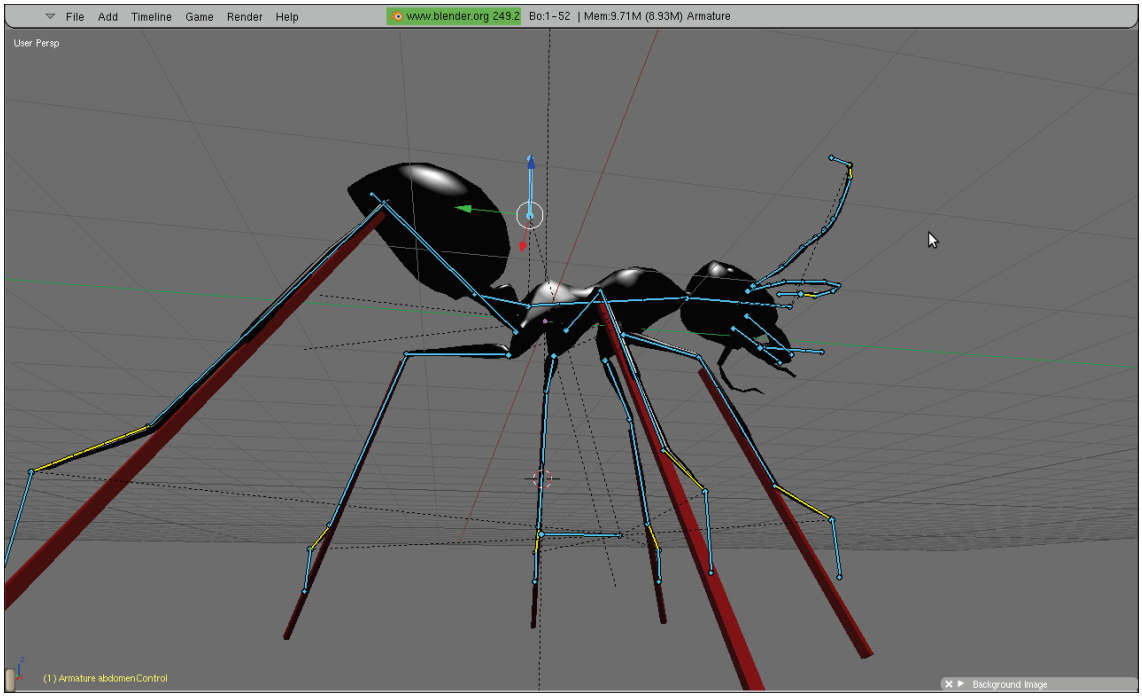


Figure B.10: ANTlike: The lifelike AVA as shown in Blender with the stilts.

tion, which required complex armature system to animate compared with the ANTique model.

B.3 VEAR

Would a scene with simple AVAs and a simple VE foster better learning than a scene with visually complex AVAs and a VE? It is impossible with ANTECATALYST to answer this question due to the limitations of the learning content. Adding a complex desert would require one to re-make the entire simulation with a different learning content as the navigation strategy of *Cataglyphis* ant would change. To confirm and extend the findings of ANTECATALYST, we need another simulation in which the VE's visual complexity can be modified without altering the learning content.

One appropriate learning topic is firefighting with the scenario of car accidents. Having a visually detailed vehicle with environmental reflection or a simplified black-and-white vehicle would not change the procedure of preventing the spread of the fire. There-



Figure B.11: ANTlike: The lifelike AVA as shown in the VR lab by using Vizard.

fore, the second VR simulation, known as VEAR (VEhicle Accident and fiRes), is designed to teach the participants the hazard in motor accidents and the safe practice to approach and prevent the spread of fire. VEAR contains greater visual complexity than ANTECATALYST, to confirm and to extend the findings.

Instead of using Vizard, VEAR uses Blender (Blender Foundation, 2011) to run the simulation. The main reason is that the three computers that drive ANTECATALYST on the semi-cylindrical screen by using three licences of Vizard, have older hardware (Processors and Graphics Cards) and unable to display the intended visual complexity in VEAR. Therefore, VEAR uses Blender 3D engine that can be deployed on any desktop computers with the latest hardware.

VEAR demonstrates the hazard in motor accidents, the safe procedures to approach and prevent the spread of fire (see Appendix D.2). The learning material was developed based on the New South Wales (NSW) Rural Firefighter's handbooks: Village Firefighter (VF) Manual (Service, 1999, Chapter 3, section 9), Specialist Training Manual: Com-

pressed Air Breathing Apparatus (Service, 2001a, CABA), and Compressed Air Breathing Apparatus Standard Operating Procedures (Service, 2001b). The learning material was organised into eight continuous scenes:

1. **Introduction**. The first scene introduces the topic of Village Firefighting. Village Firefighting means fighting fire of small structures or car accidents. The scene closes by showing car accidents near a tunnel.
2. **Compressed Air Breathing Apparatus**. Prior to containing the spread of fire in a car accident, the second scene brings the attention of the learners to Compressed Air Breathing Apparatus (CABA). CABA is crucial in any fire rescue as smoke and fumes pose danger to the human respiratory system. The scene shows a firefighter equipped with CABA.
3. **Vehicle roll and tip hazard**. Following from the importance of CABA, the scene shows a scenario of a vehicle on fire and the steps that must be taken to prevent the spread of fire, removing the ignition key, applying the handbrake and chocking the wheels. In addition, the scene also explains that the best position for approaching an accident scene is from the uphill side and not against the direction of the wind.
4. **Explosion hazard**. The fourth scene shows the explosion hazards for the firefighters. Vehicles contain components that may explode under constant heat of fire, such as a fuel tank, inflatable bumpers, tyres, airshocks and driveshafts.
5. **Hazardous material**. The fifth scene shows some vehicles' components that may turn into hazardous material when exposed to fire. This scene stresses the importance of having a full protective gear, as some hazardous material may cause serious skin injury without a sense of burning or stinging.
6. **Electrical hazard**. A car accident may involve fallen power lines, and this can be dangerous for the firefighters and the victim in the vehicle. The scene shows that the best action is to stay clear until an authorised electrician has confirmed that the power has been shut off.

7. **Vehicle with no fire**. Unlike the previous 5 scenes, this scene deals solely with a car accident with no apparent fire. The scene shows three steps that can be taken to reduce the chance of fire ignition, by blanketing the spillage with foam, prohibiting smoking and other ignition sources in the area, and disconnecting the vehicle's battery from the engine with the earth terminal first. The scene closes by showing a nearby firefighter with a hose on his hands.
8. **Summary**. The last scene reinforces learning by providing an overview of Village Firefighting in handling car accidents.

The duration of VEAR was about 8 minutes, and the camera moved from one location to another following the narration. The virtual actors were visible in all scenes. VEAR runs smoothly at maximum frame-rate on an Intel Core2 Duo E8400 desktop PC with nVidia GeForce GTX 260 graphic card, and 4GB RAM at 1680 by 1050 pixels resolution.

Unlike ANTECATALYST, VEAR enabled us to alter the visual complexity of the AVAs and the VEs to create a series of scenes with increasing visual complexity as shown in Figure 1.4:

1. **Simple** - visually simple AVAs and background objects.
2. **Simple AVAs** - simple AVAs and visually complex background objects.
3. **Simple world** - visually complex AVAs and simple background objects.
4. **Lifelike** - visually complex AVAs and background objects.

Table B.3 shows the components of VEAR in detail. Simple AVAs are made of flat polygons with simple lines and circles for drawing the texture. Lifelike AVAs resemble lifelike firefighters in detailed uniforms with sky reflections on the helmets. Each AVA contains idle animations and the degree of complexity depends on the visual complexity of each AVA. Simple AVAs spin on vertical axes while idling. On the other hand, lifelike AVAs show lifelike breathing and turning animations. The simple VE is composed of flat and polygon objects, with simple black and white textures. In the lifelike VE, objects were designed with extraneous visual details in mind. The lifelike VE contains myriads

Table B.3: Visual complexity of each experimental condition in VEAR.

			Experimental Condition			
			Flat	**Flat AVAs**	**Flat World**	**Lifelike**
	AVAs	Shape	Flat - 2D, 1 polygon	Flat - 2D, 1 polygon	Lifelike - 3D	Lifelike - 3D
		Colour	Black and white	Black and white	Full colour	Full colour
		Animations	Spin	Spin	Lifelike breathing and turning	Lifelike breathing and turning
		Details	Simplified	Simplified	Lifelike skin	Lifelike skin
					Creases and imperfections	Creases and imperfections
					Sky reflection	Sky reflection
Env.	**Cars**	Shape	Simplified	3D - real vehicle	Simplified	3D - real vehicle
		Colour	Black and White	Full colour	Black and White	Full colour
		Details	Simplified	Tyre marking	Simplified	Tyre marking
				Detailed interior		Detailed interior
				Env. reflection		Env. reflection
	Fire, Smoke	Shape	Flat	3D	Flat	3D
	Water, Foam	Colour	Black and White	Full colour	Black and White	Full colour
		Animation	None	Texture - lifelike	None	Texture - lifelike
				Normal mapped		Normal mapped
	Sky	Shape	None	3D dome	None	3D dome
		Colour	White	Photograph	White	Photograph
	Lighting	Lightsource	Directional	Multiple lights	Directional	Multiple lights
	Road & Landscape	Shape	Flat	3D	Flat	3D
		Colour	Black and White	Full	Black and White	Full
		Details	None	Cracks and imperfections	None	Cracks and imperfections
				Normal mapped		Normal mapped
	Post-processing		None	Colour curve and Glow	None	Colour curve and Glow

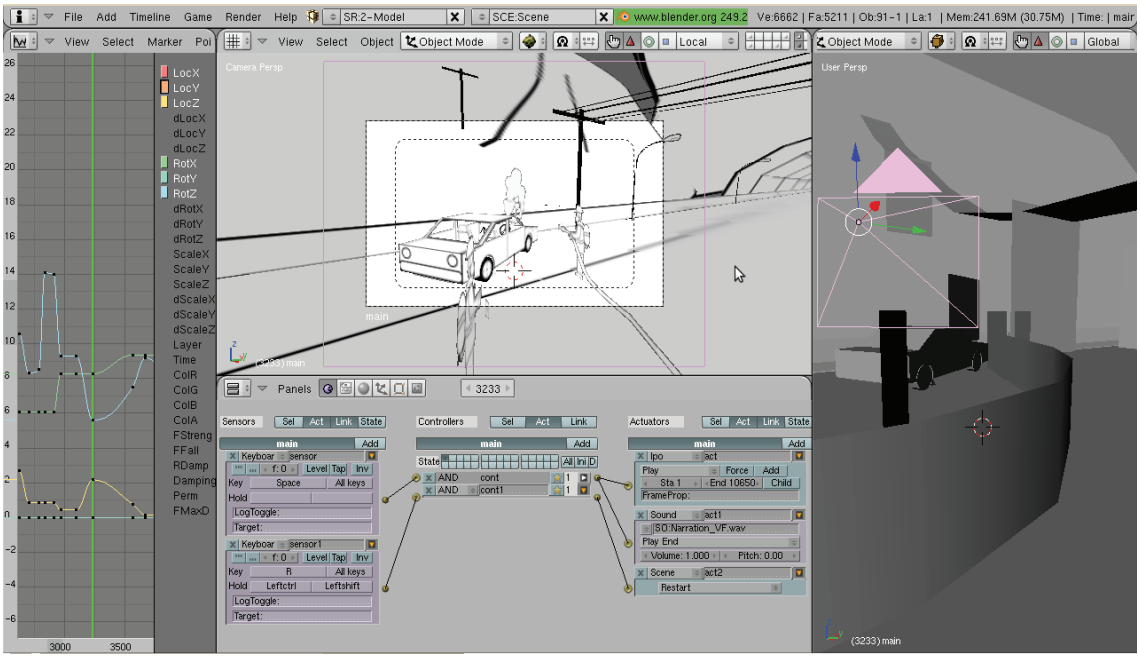


Figure B.12: A screenshot of the camera movement in Blender. The window on the left shows the graphs of key-framed camera movements around frame number 2500. The middle window shows what the user sees in VEAR. The middle bottom window shows the logic bricks of the camera. The top logic brick plays the narration and the movement of the camera from frame 0 to frame 10650, when the space bar button is pressed. The bottom logic brick sends a stop signal to Blender game engine when Ctrl, Shift and R buttons are pressed. The rightmost window shows the environment of VEAR without textures.

of details that are not available in the simple VE, such as reflections, twirling smokes, animated flame and water, and detailed textures that portray surface imperfections on various surfaces. Similar to ANTECATALYST, the development of VEAR also started with a screenplay (see section D.2) and storyboard (see C.2).

Blender was used (Blender Foundation, 2011) in developing, composing the scene and animating most 3D models, as well as the VR engine to run the simulation. Since Blender itself is a 3D engine, the objects do not need to be exported out into different formats. Therefore, Blender minimises the file transfer issue as previously encountered in RiskMan and BOSS (see Chapter 2, Kavakli & Kartiko, 2007). What we see in Blender is what we see in the VR simulation. In addition, most 3D objects needed for VEAR were readily available on-line at an affordable price. The movement of camera in VEAR

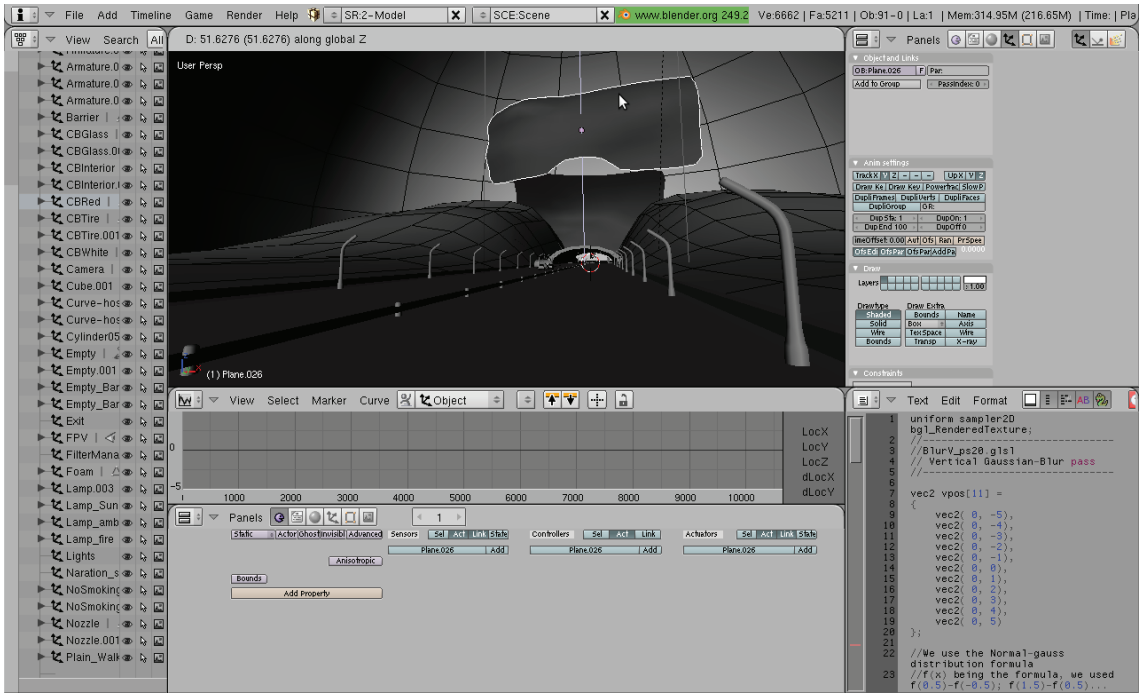


Figure B.13: A screenshot of composing the scene of VEAR prior to applying texture on the 3D objects.

is key-framed animations as shown in Figure B.12. The camera has many keys along the length of the simulation. In addition, the start and stop controls and other event-based animation, such as water, flames, utilise Blender's logic bricks as shown in Figure B.12, without the necessity of Python coding. Therefore, the development and the deployment processes of VEAR were considerably simpler than ANTECATALYST (Figure B.13).

B.3.1 Simple AVAs

The AVAs for simple AVAs condition were originally hand-drawn and re-traced manually by using Inkscape (Inkscape Development Team, 2011, Figure B.14). The simple AVA in VEAR was achieved by placing the image of the AVA on a flat polygon. Thus, it created the illusion of a flat AVA, as shown in Figure 1.4 (A, C) and B.15. Similar to ANTique, Simple AVA for VEAR was easy and fast to create.

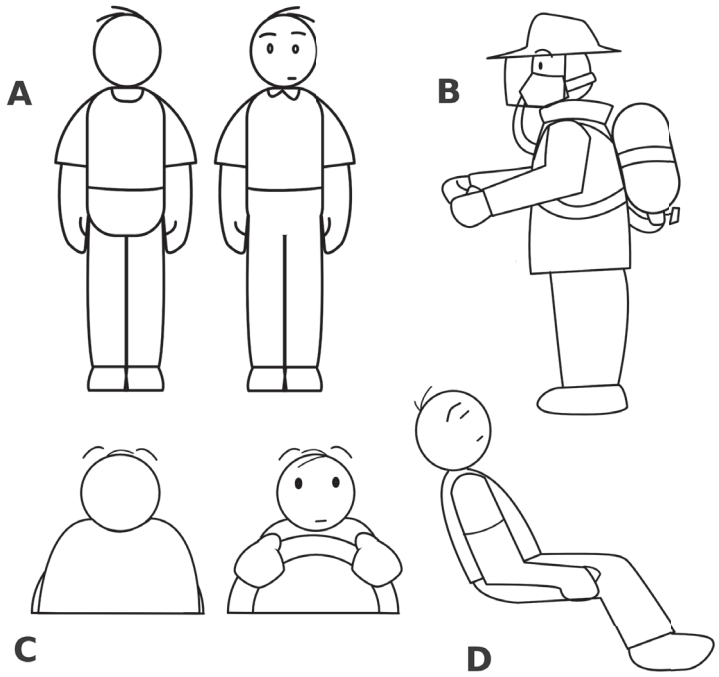


Figure B.14: The final sketch of simple AVAs in VEAR: (A) a bystander, (B) a firefighter, (C) a driver in a car, and (D) a victim.

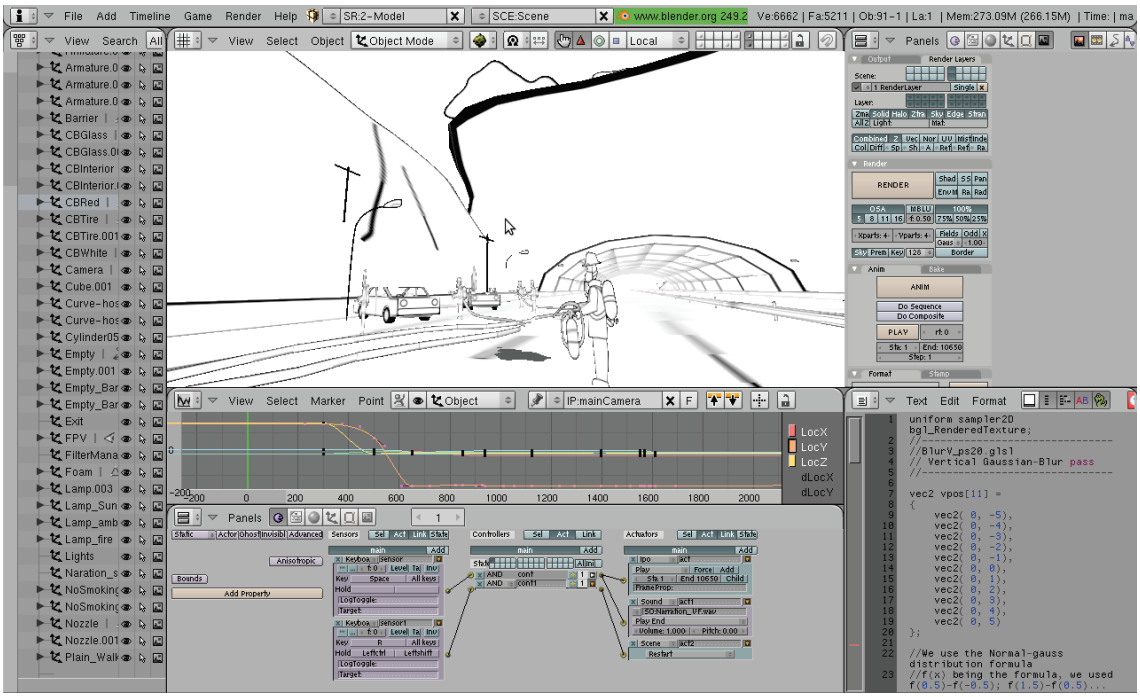


Figure B.15: Development screenshot of VEAR in Blender that shows simple AVAs and VE. The AVAs appear flat in the 3D world.

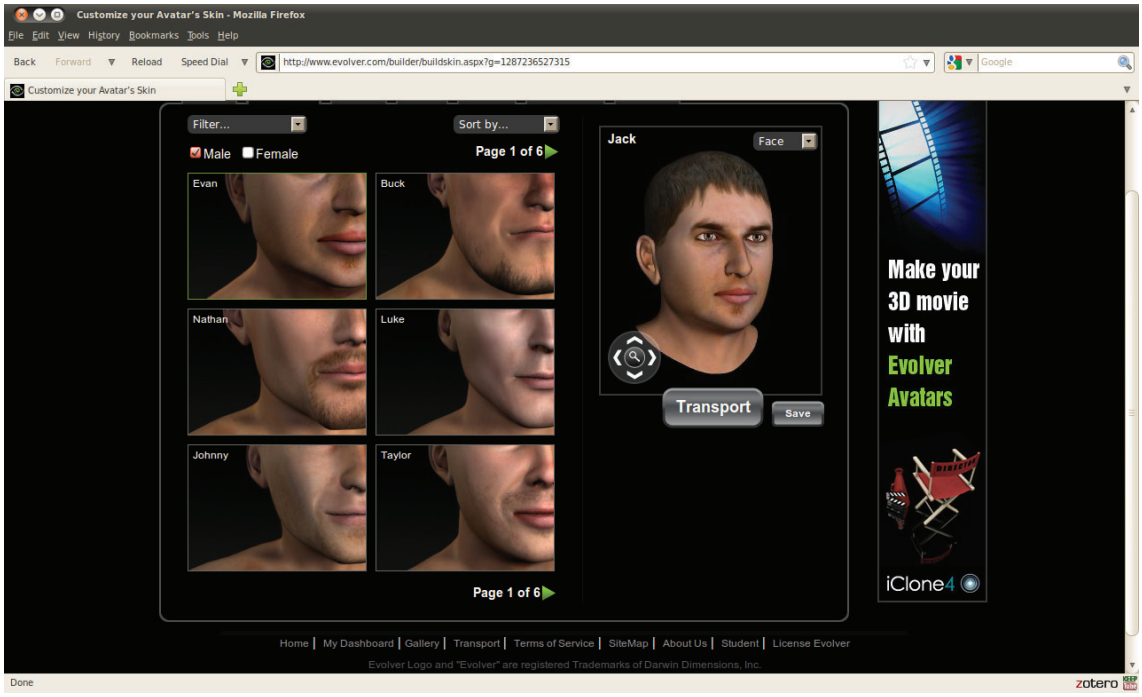


Figure B.16: The creation of the AVA that represents the victim of the car accident in VEAR in Evolver. The figure shows the step in choosing skin textures from the Evolver library.

B.3.2 Complex AVAs

The author designed and purchased the visually complex AVAs for the bystanders, drivers in the car and the victims from Evolver (**?**) as shown in Figure B.16 and B.17. The AVAs of the firefighters were purchased from TurboSquid[2]. Evolver pre-rigged the AVAs with armature systems, ready for animation and eased the development and deployment of VEAR. The firefighter, however, did not have armature system, and Blender was used to create one to deform the firefighter's mesh in making it lifelike. In contrast to the simple AVAs, the complex AVAs showed the 3D shape with other extraneous visual details as listed in Table B.3 (Figure B.19).

[2]http://www.turbosquid.com/3d-models/fireman-01-3d-model/289907

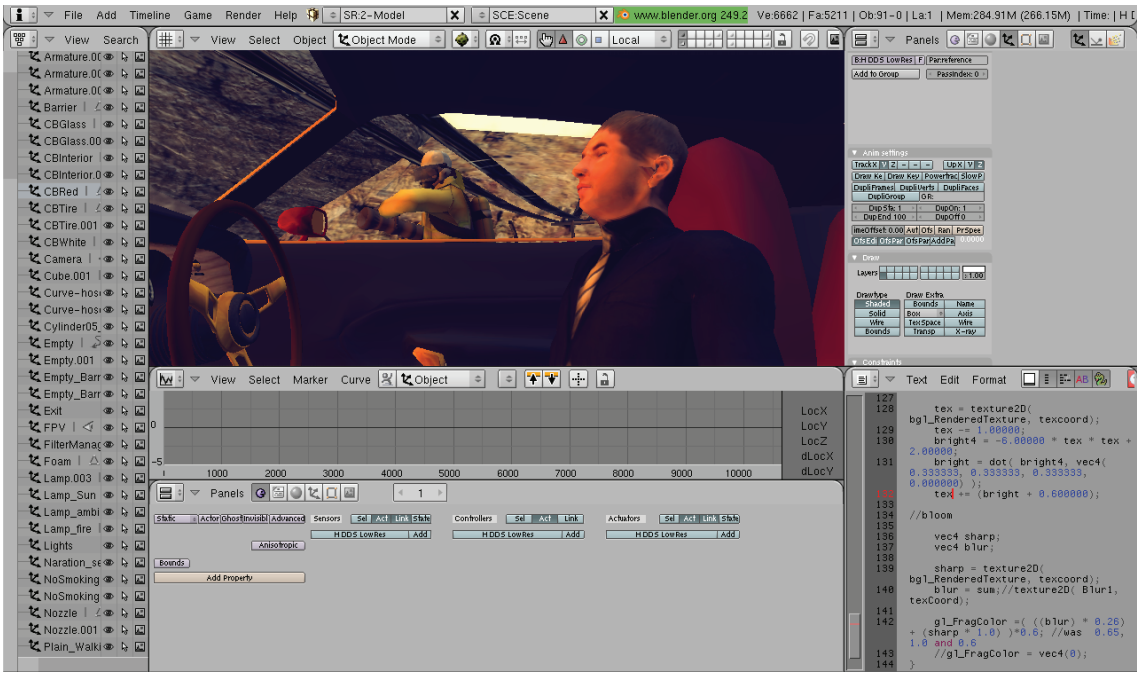


Figure B.17: The AVA that represents the victim of the car accident was seamlessly integrated into Blender as Evolver generated the armature system for Blender. In the background, the firefighter with armature system was implemented and the body was also deformable in Blender as the AVAs were also imported from Evolver.

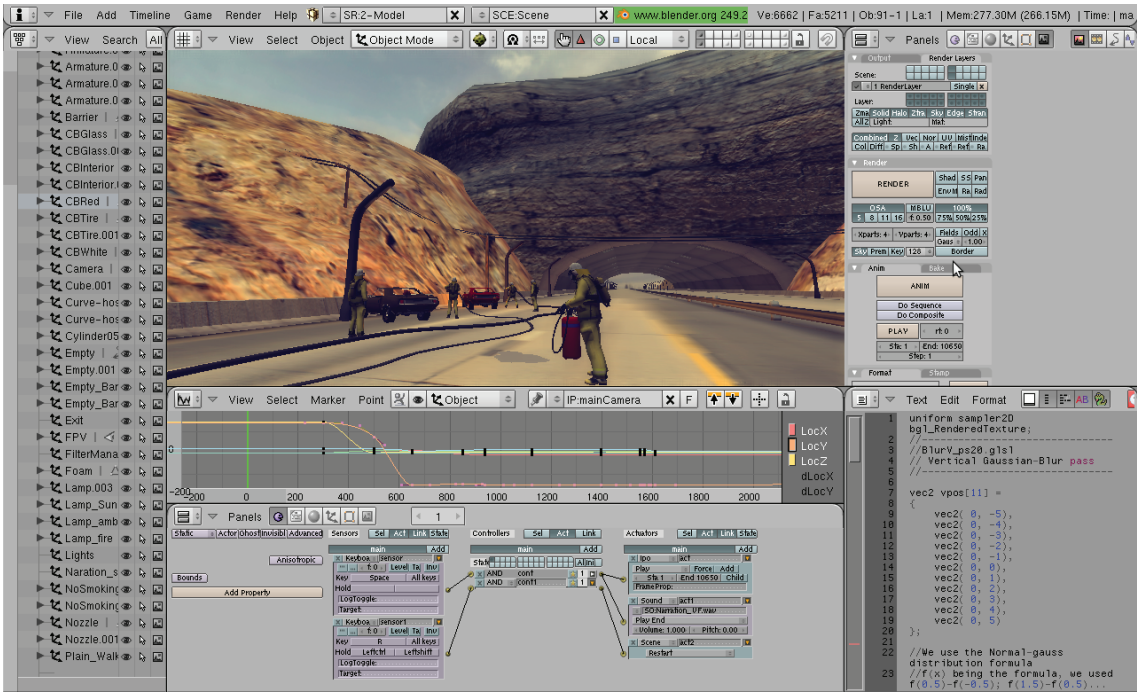


Figure B.18: Development screenshot of VEAR in Blender that shows visually complex AVAs and a complex VE. Visually complex AVAs give the 3D shape and other extraneous visual details that are not found in the simple AVAs.

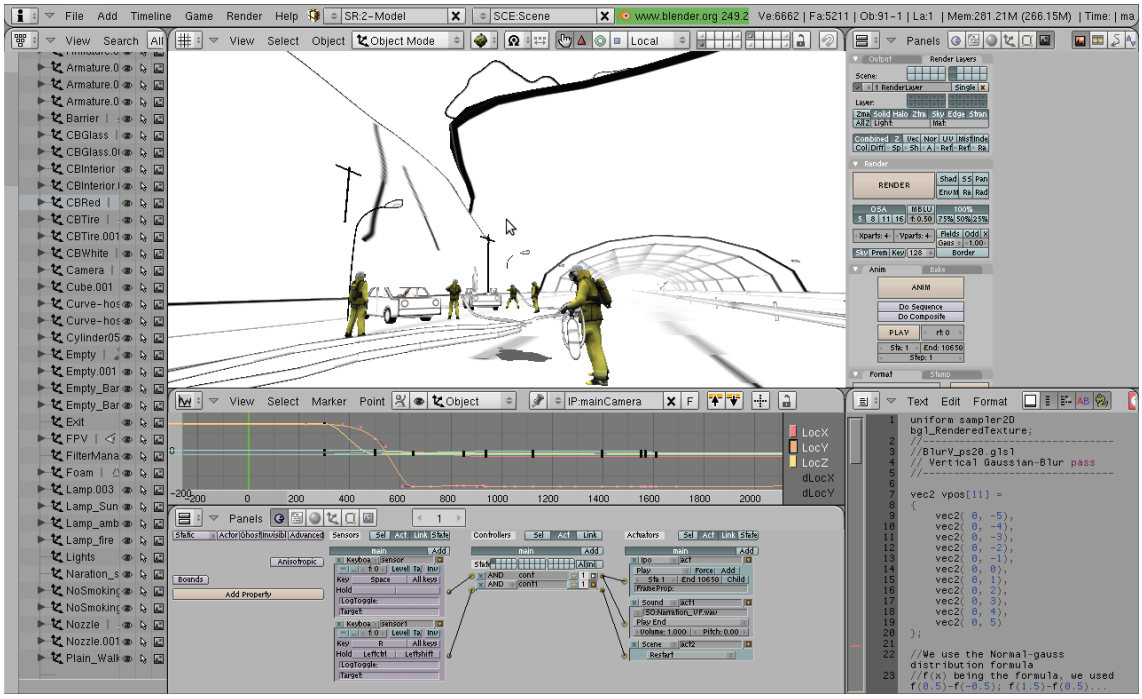


Figure B.19: Development screenshot of VEAR in Blender that shows visually complex AVAs and a simple VE. The simple VE is made of simple black-and-white drawings on flat polygons. Therefore, the simple VE appear flat in the 3D world.

B.3.3 Simple VE

The VE consisted of a tunnel, roads, hills, vehicles and the environmental effects, such as smoke, water and fire. The simple VE was the simplification of the visually complex VE in VEAR. The tunnel was textured with simple white textures. The simple VE was achieved by placing the 2D images of the hills, roads, clouds, fire, smoke, street lights, walkie-talkies and the street divider, on flat polygons. The vehicles are made simply by extruding cubes to form a basic shape of a vehicle, with simple and black-and-white textures on them. Therefore, the simple VE formed an illusion of a flat world, as shown in Figure 1.4 (A, B) and B.19.

B.3.4 Complex VE

The complex VE contained a plethora of visual details, as listed in Table B.3, not found in the simple AVAs (see Figure B.20). The textures were collected from various pho-

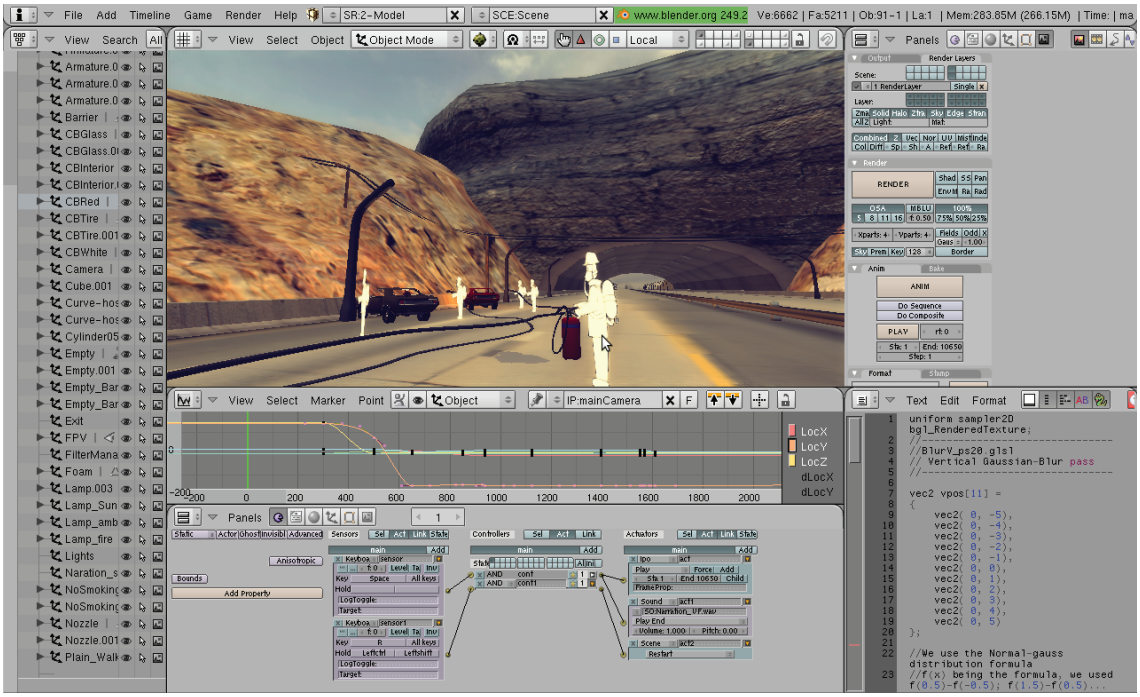


Figure B.20: Development screenshot of VEAR in Blender that shows simple AVAs and a visually complex VE. The visually complex VE is visually rich as it contains greater visual details than the simple VE.

tographs. The complex VE had animated fire[3], water[4] and twirling smokes[5]. The vehicles portrayed for the accident were purchased from DAZ3D[6] and contained a detailed engine compartment. Furthermore, the complex VE also showed the environmental reflection of the vehicles. The complex lighting was achieved by using multiple lights in the scene, coupled with post-processing effects, such as bloom and colour correction shaders[7].

B.4 Conclusion

This chapter demonstrates the creation of ANTECATALYST and VEAR, which can be used to investigate the effects of learning with complex visuals in VR simulations. AN-

[3]A fire GLSL shader demo by Martins Upitis, http://blenderartists.org/forum/showthread.php?t=159006
[4]A water demo by Mike Pan, http://mikepan.com/files/MatrixCourtyard.blend
[5]Adapted from a real-time smoke demo, http://blenderartists.org/forum/showthread.php?t=143292
[6]http://www.daz3d.com/i/shop/itemdetails/?item=2217
[7]Taken from a compilation of GLSL shaders by Martins Upitis, http://blenderartists.org/forum/showthread.php?t=156482

TECATALYST uses Vizard as the VR engine, while VEAR uses Blender. Vizard has many advantages as a VR engine. Vizard is a flexible SDK to create VR simulations, which requires some knowledge of Python scripting. Thus, for researchers without familiarity with Python scripting, Vizard provides an extensive set of examples that can be used to prototype VR simulations quickly. Vizard also supports many VR devices, including tracking devices, clustering (having multiple PCs to run a single simulation), and data gloves. The disadvantage of Vizard is the high-price tag to have a full screen and immersive simulation, which ANTECTALYST requires. The academic enterprise edition starts at USD$5,990 for one user at the time of the writing of this thesis[8]. Vizard is a scene composer, not a 3D modelling suite like Blender (Blender Foundation, 2011). Therefore, 3D objects need to be imported from external 3D applications.

Compared with ANTECATALYST, the development toolkit for VEAR cost nothing. Blender is not only a 3D modelling and animation software, but also a 3D engine. Therefore, Blender minimises file transfer issues between 3D applications (see Chapter 2 Kavakli & Kartiko, 2007). Blender also allows quick prototyping of VR simulations. However, it requires some knowledge of 3D modelling and Python scripting. Although it is possible to use Python scripting to create the flow of the VR simulation, VEAR is driven by logic bricks and key-framed animations as shown in figure B.12. VEAR also contains an interactive camera driven by Python codes[9], but it is disabled for experimental purposes. In contrast to ANTECATALYST, VEAR does not require any Python codes to drive the VR simulation.

Thus, researchers without familiarity with VR simulations and are willing to create a simulation quickly for research purpose, Vizard is a good candidate. Given some familiarity with Blender and Python scripting, the cost of the project can be reduced. In addition, within the same project time-frame, the author implemented more learning content (more information and scenes) and extraneous visual details by using Blender. Because of the

[8]http://www.worldviz.com/purchase/pricelist.html
[9]Implemented from http://www.tutorialsforblender3D.com

author's familiarity with Python scripting, this projects considered these two 3D engines. Information about other 3D engines can be retrieved from http://www.devmaster.net/engines/.

Appendix C

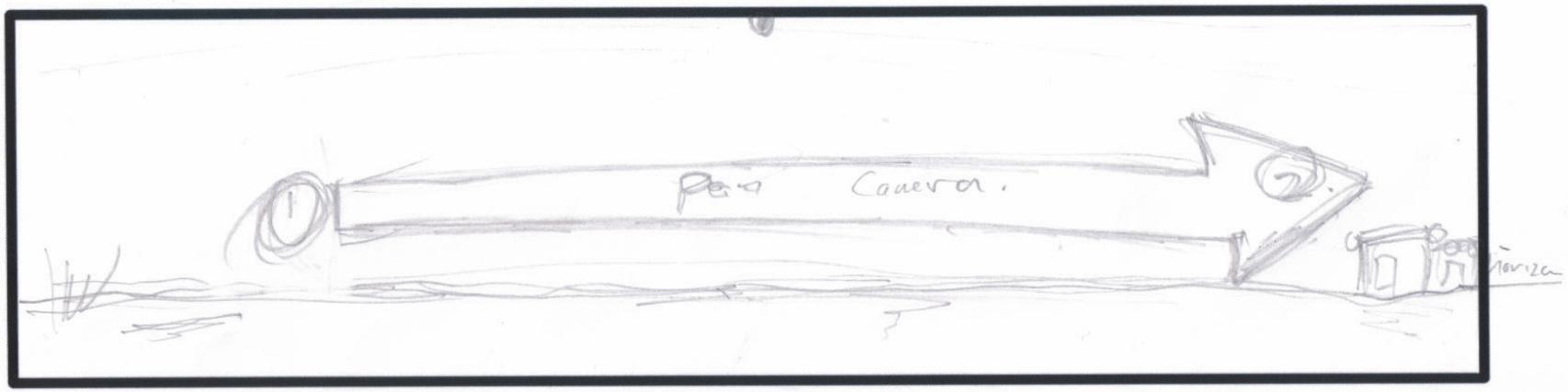
Storyboards

This chapter in appendix presents the storyboards that were made prior creating the Virtual Reality (VR) simulations. The layouts of each storyboard differs due to the output display. ANTECATALYST displays on 6m x 2m semi cylindrical projection system, so the layout of the storyboard resembles the screen ratio of the semi-cylindrical screen. Scene 4-3 of ANTECATALYST was removed from the production as it resembled scene 4-2, thus it was deemed unnecessary to have scene 4-3 independently, and merged into scene 4-2. The output of VEAR is 16:10 wide-screen desktop monitors, thus the panels of the storyboard resemble the shape of boxes at 16:10 ratio.

C.1 Storyboard of ANTECATALYST

Scene Title: EXT. DESERT - INTRODUCTION.

Scene No : 1-1 Revision No: A



Narration:

Notes:
Pan Camera across Dessert
1 2

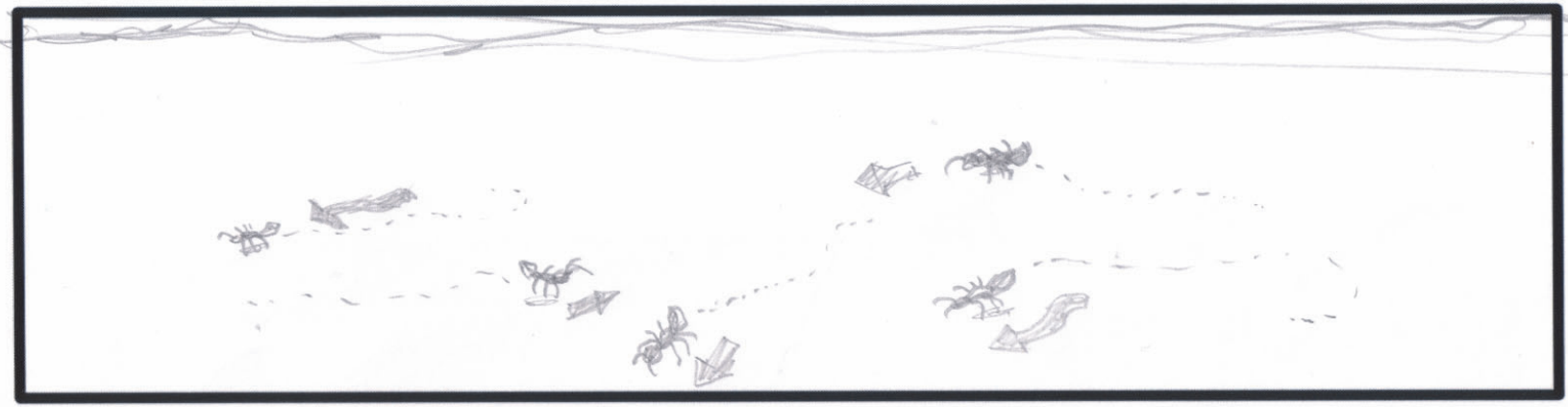
Prev. Scene: N/A

Next Scene : 1-2

Scene Title: EXT · DESERT · DAY : INTRO

Scene No : 1-2 Revision No: A

horizon



Narration:

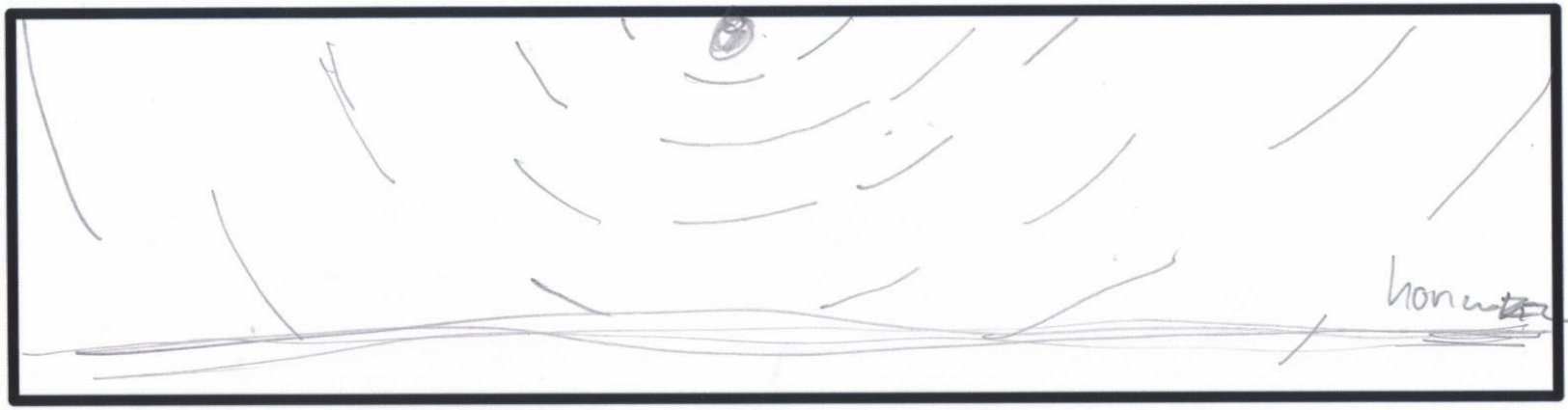
Notes: Camera looks down, and see Cone moving C for dis..

Prev. Scene: 1-1

Next Scene : 1-3

Scene Title: EXT-DESERT-DAY - sky light PATTERN.

Scene No : 3-1 Revision No:



Narration:

Notes: look up & fade in sky pattern over 2 sec

Prev. Scene: 2-4

Next Scene : 3-2

Scene Title: EXT. DESERT - DAY. Skylight Pattern.

Scene No : 3-2 Revision No:



Narration:

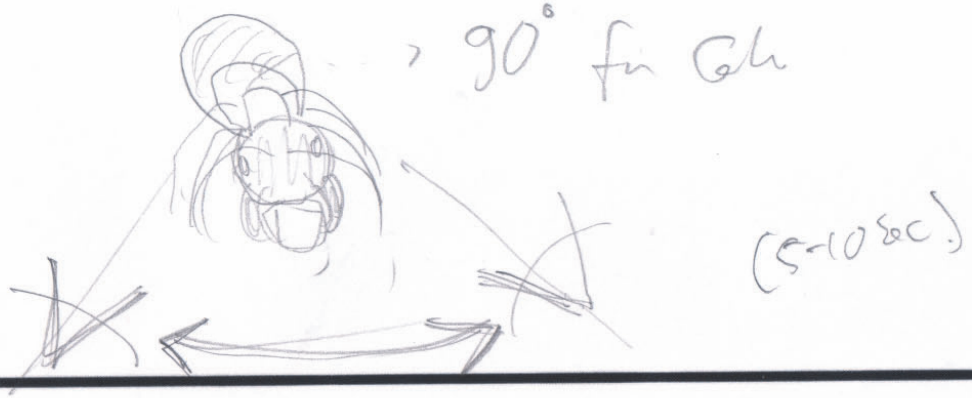
Notes: move the sun + sky pattern. from 75°-85°
(Sun should be almost overhead).
location:
leave the sky pattern throughout Demo

Prev. Scene: 3-1

Next Scene : 3-3

Scene Title: EXT - DESERT - DAY. ODOMETRY.

Scene No : 4-5 Revision No: 060208.



Narration:

Give food, state "Normal Condition" out

Notes:

Prev. Scene: 4-4

Next Scene : 4-6.

Scene Title: EXT. DESERT - DAY) - ODOMETRY.

Scene No : 4-6 Revision No: 060208



~~Narration:~~ Note:

- Move camera to Next location
- Show stills & attach them to ants legs.
- Give food them in Mandibles
(Swivel camera to approximately ~ 90°.)

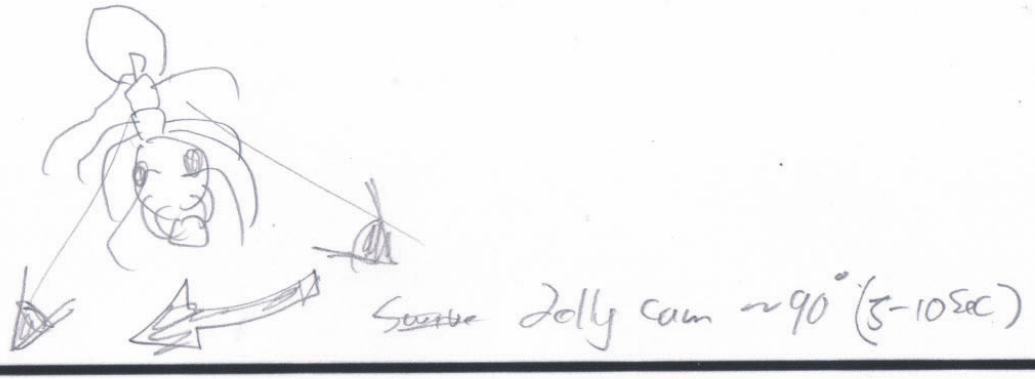
Notes: ↑ See above Note.

Prev. Scene: 4-5

Next Scene : 4-7.

Scene Title: EXT. DESERT. DAY ODOMETRY

Scene No : 4-7 Revision No: 060208



Narration:

- move cam to next location.
- apply stumps on the dirt, give food item (dolly camera)
-

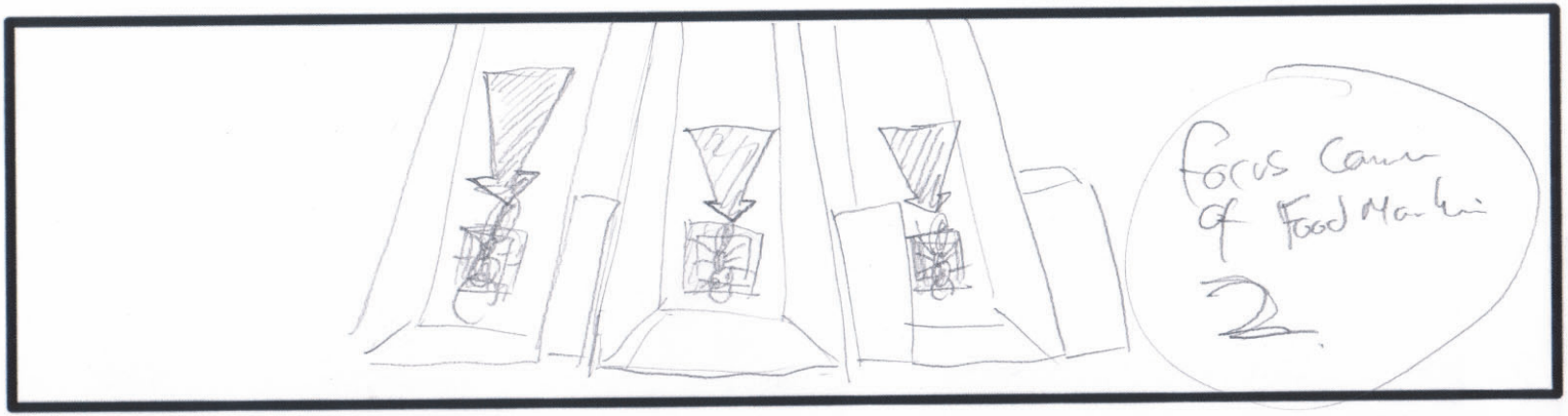
Notes:

Prev. Scene: 4-6

Next Scene : 4-8

Scene Title: EXT. DESERT. DAY. ODOMETRY

Scene No : 4-8 Revision No: 060208



Narration:

- lower all the ants
- Set them on inbound journey (double previous trail now crabble trail!)

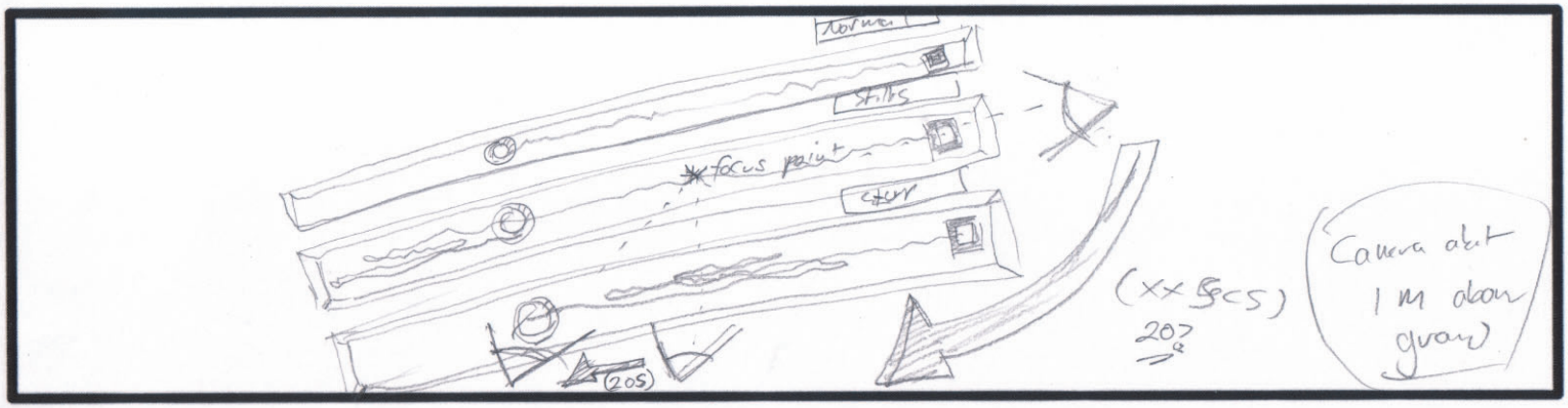
Notes:

Prev. Scene: 4-7

Next Scene : 4-9

Scene Title: EXT. DESERT. DAY. ODOMETRY

Scene No : 4-9 Revision No: 06 02 08



Narration:

once the act hit "Release" point, Remove out from View.



repeat

total durch = total to achieve
hit the "Release" point

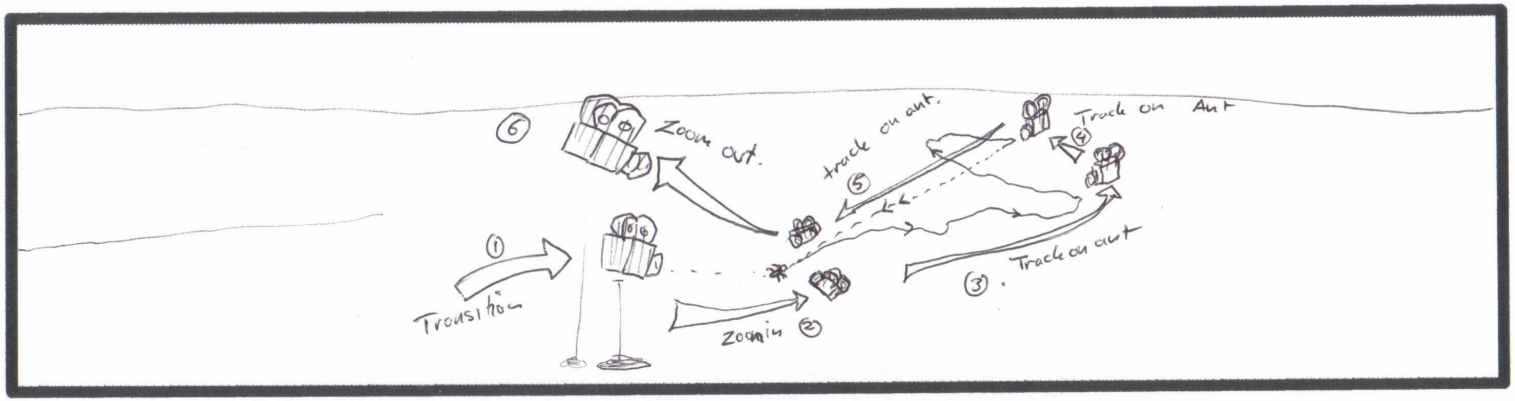
Notes: 3 location of camera with the same focus

Prev. Scene: 4-8

Next Scene : 5-1

Scene Title: EXT - DESERT - DAY. PATH INTEGRATION

Scene No : 2 Revision No: 240708



Narration:

6 Camera movement. Fade in Grid at (2)(4)
fade out grid at (6).

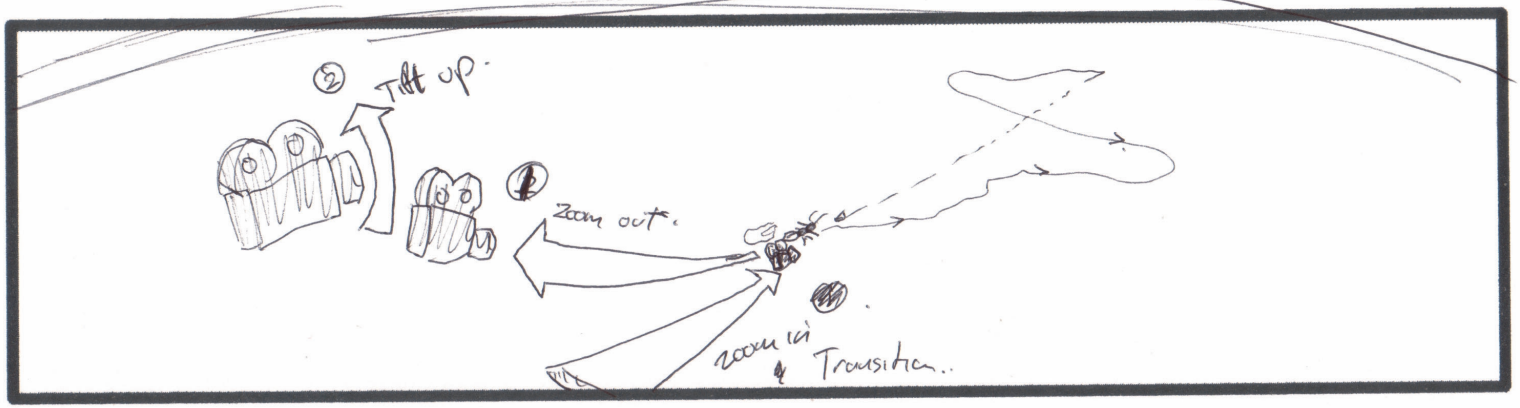
Notes: Slower Camera Movement.

Prev. Scene: 1

Next Scene : 3

Scene Title: EXT. DESERT. DAY. SUMMARY.

Scene No : 5 Revision No: 240708.



Narration:

During transition, fade in grid on the desert floor.
At ②, fade out sky patterns and the grid.

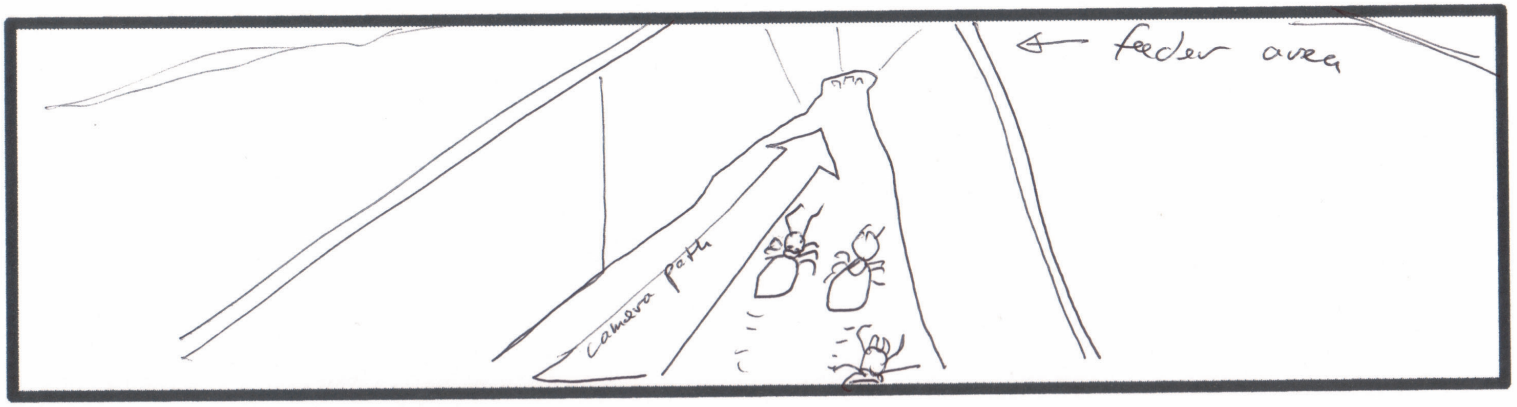
Notes:

Prev. Scene: 4

Next Scene : — —

Scene Title: EXT. DESERT - DAY. ODOMETRY EXPERIMENT.

Scene No : 4-2 Revision No: 240708.



Narration:

the camera glides inside the tunnel, while keeping track on the ants.

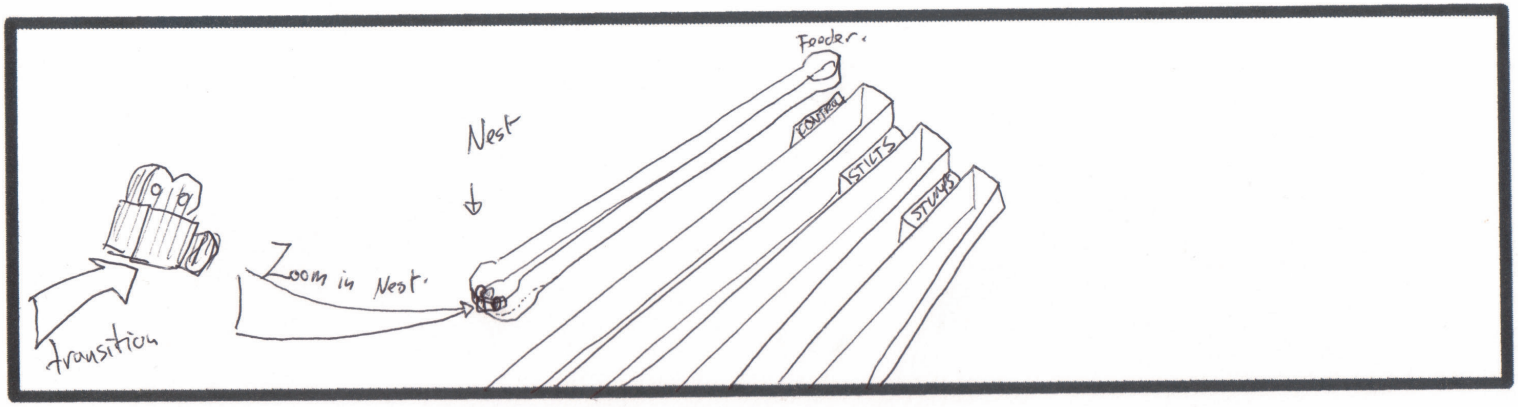
Notes:

Prev. Scene: 4-1

Next Scene : 4-3

Scene Title: EXT - DESERT - DAY . ODOMETRY EXPERIMENT.

Scene No : 4 - 1 Revision No: 240708,



Narration:

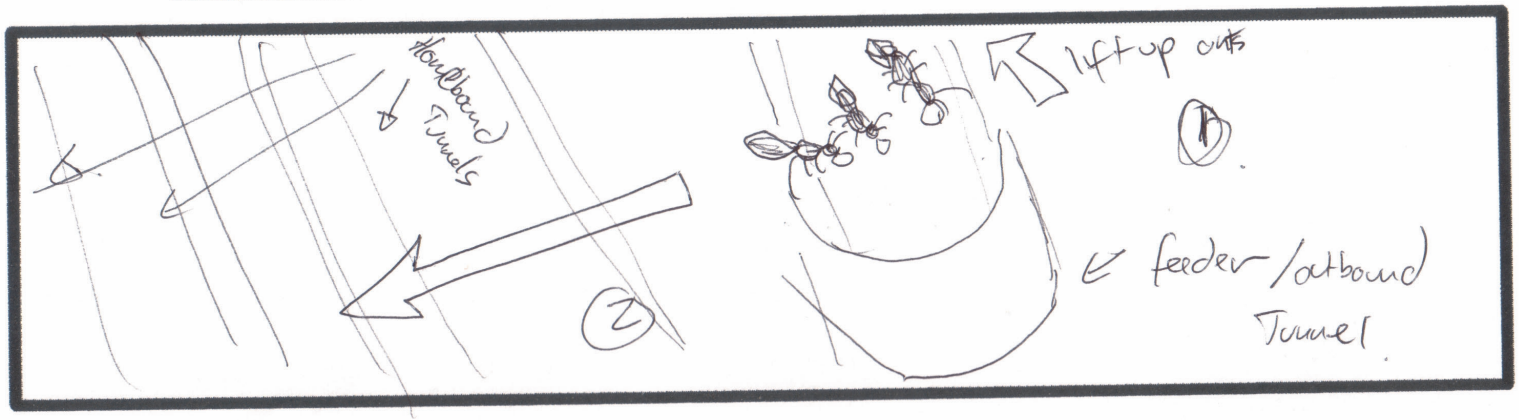
Notes:

Prev. Scene:

Next Scene :

Scene Title: EXT- DESERT- DAY - ODOMETRY.

Scene No : 4 -4. Revision No: 26-7-02



Narration:

① lift up the ants. & place them onto Separate channels.

Notes:

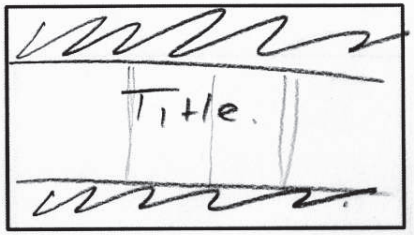
Prev. Scene:

Next Scene :

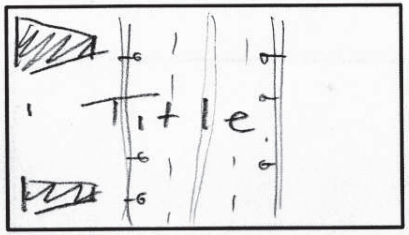
C.2 Storyboard of VEAR

Title: Introduction

Scene: 1

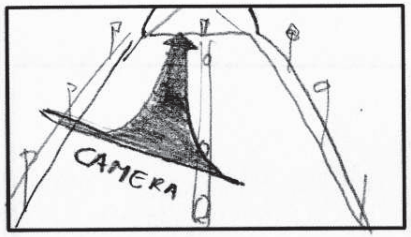


Top-down view on the road.

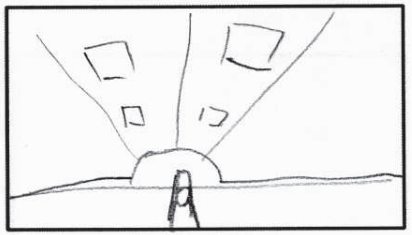


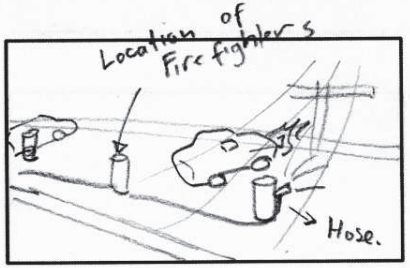
Remove black borders

Drop the camera into the "title"



Dolly Camera into tunnel

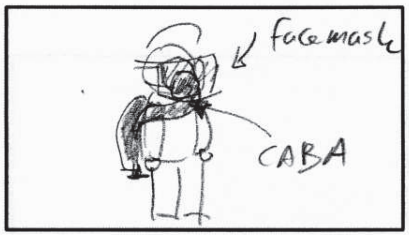




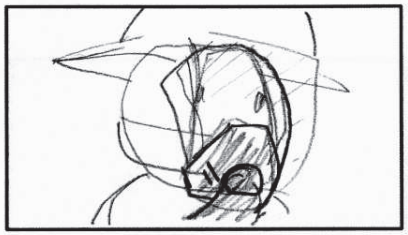
Capture the accident

Title: Preparation

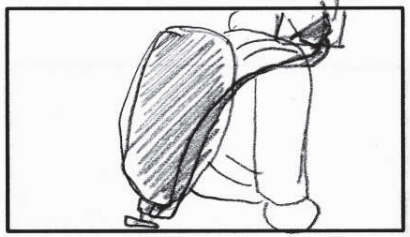
Scene: 2.



Capture a firefighter with CABA Close shot to face



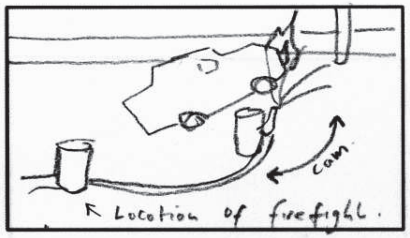
Dolly camera from face



to the body to capture the cylinder

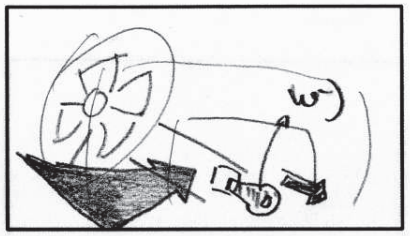
Title: Vehicle Roll

Scene: 3

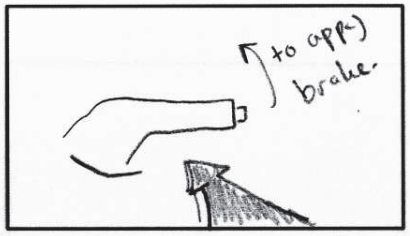


Dolly Camera on the Car at eye-height

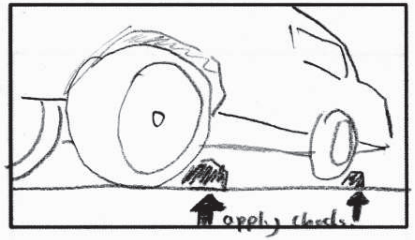
- Remove the Driver from Here



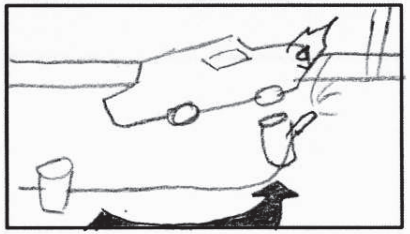
Zoom into key
Remove key
Put key on dashboard.



Move into handbrake
apply handbrake.



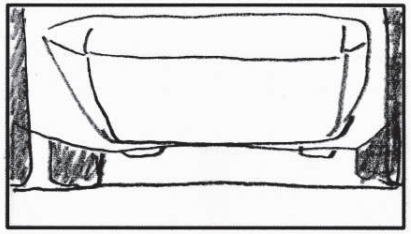
Close shot on rear wheel
Apply Chocks.



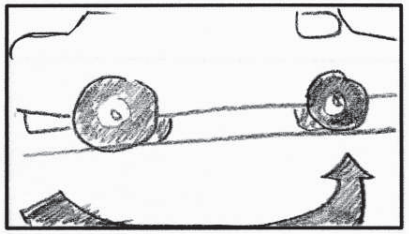
Dolly Camera out on the Car

Title: Explosions

Scene: 4



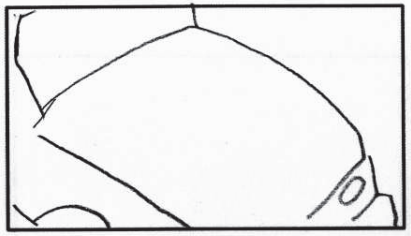
low angle - close shot on fuel tank



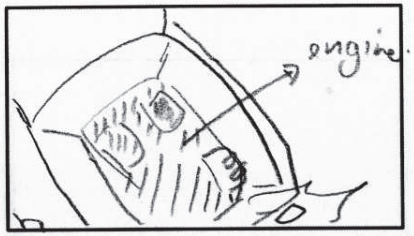
Dolly camera around capture all tyres

Title: Hazardous Material

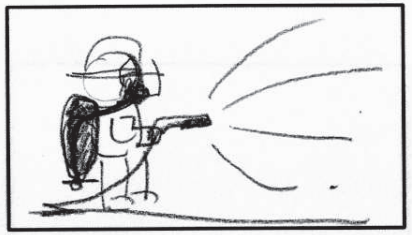
Scene: 5



Dolly camera into engine port



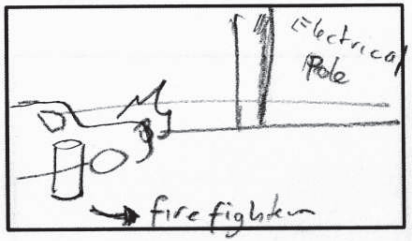
Remove cover
Show the engine
High angle shot.



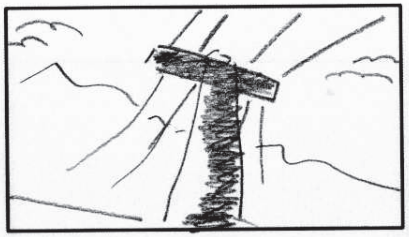
Dolly camera to capture nearest firefighter.

Title: Electrical Hazard.

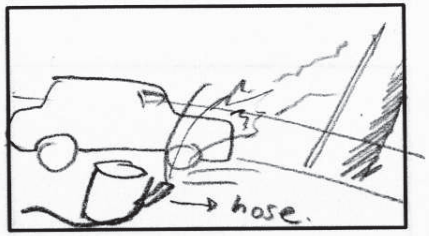
Scene: 6.



Dolly cam to front of car



Tilt camera up and down

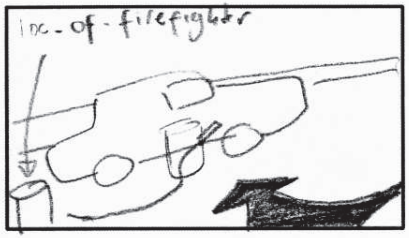


Capture the car & the Firefighter.

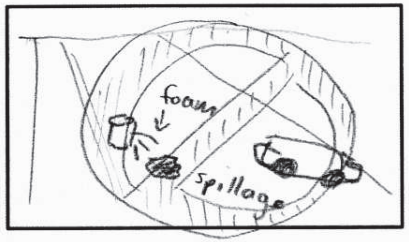
Title: Vehicle Incident - No fire.

Scene: 7

* DO not remove the victims from the car *

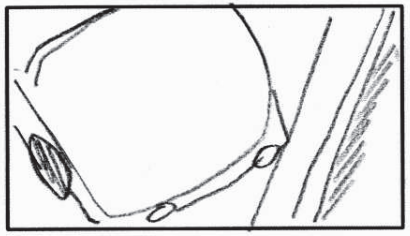


Dolly camera to the other car.

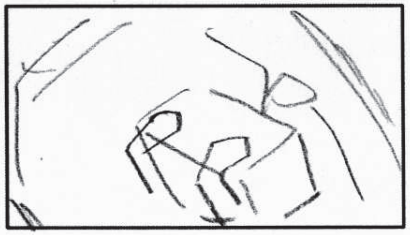


High angle shot on the spillage area

* Flash "no smoking" sign



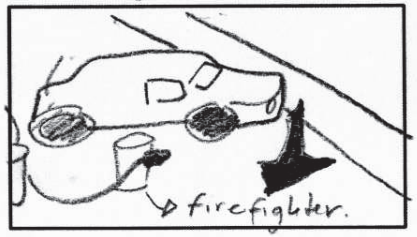
Dolly camera into engine compartment.



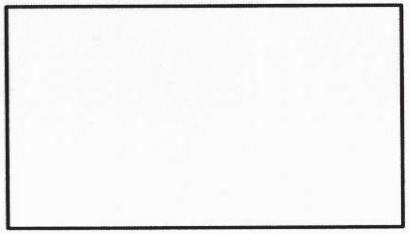
(1) lift the cover

(2) focus on car battery

* Disconnect Battery (Earth first)

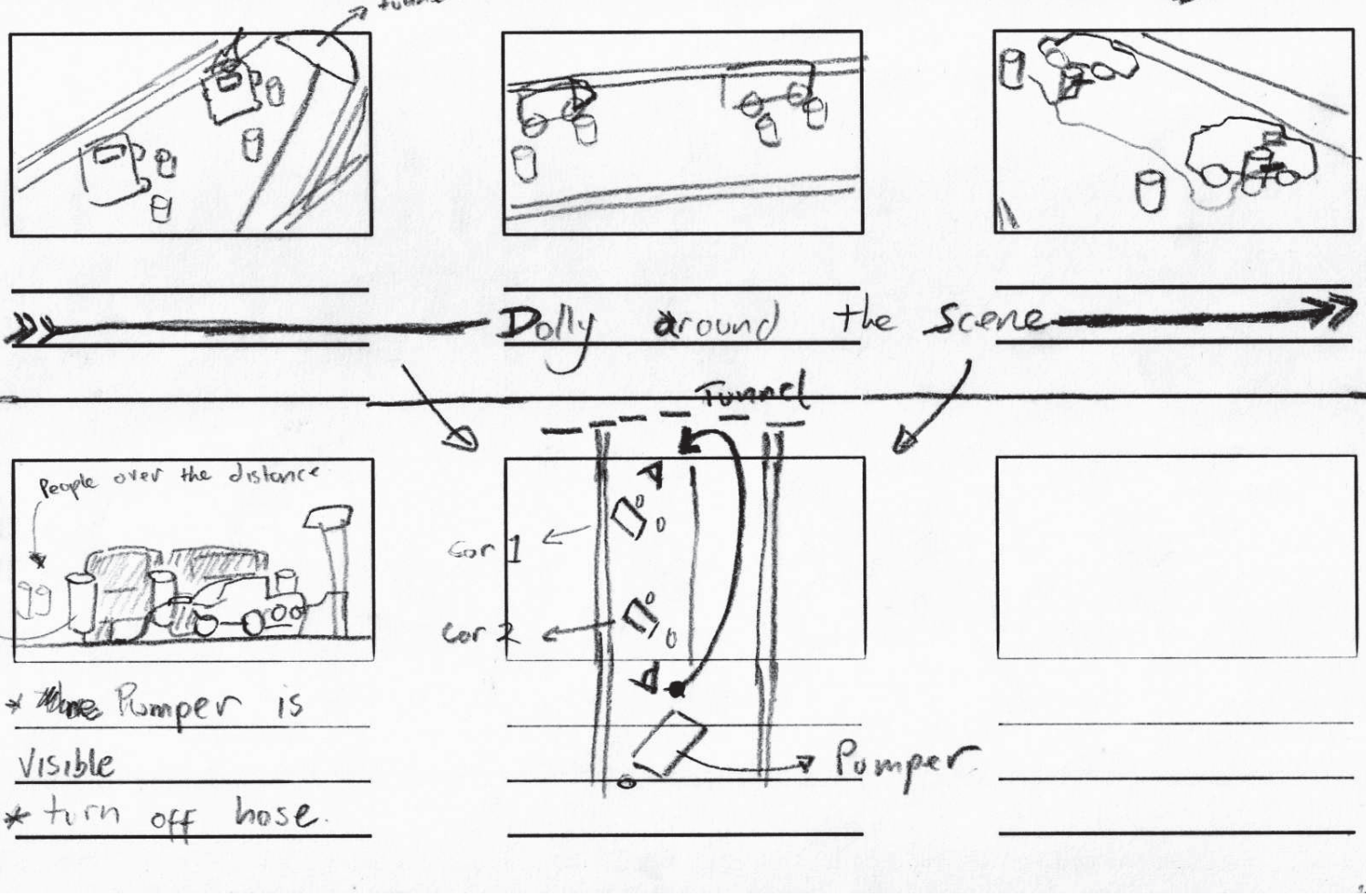


Dolly camera out & clasely around the car



Title: Summary – Final – scene.

Scene: 8



Appendix D

Screenplays

The screenplays of both studies, ANTECATALYST and VEAR, are provided here. Although both screenplays seem to be at the same length, the screenplay of VEAR inherently contains more learning material than the screenplay of ANTECATALYST.

D.1 Screenplay of ANTECATALYST

Screenplay of Case Study 1 - Odometric behaviour of Cataglyphis fortis

By Iwan Kartiko

Supervisors:
Dr. Manolya Kavakli and Dr. Kenneth Cheng

1 EXT.DESERT.DAY.INTRODUCTION

A vast and endless desert extends to the horizon. A contrast desert scene, a deep blue sky and bright white sands.

PAN CAMERA ACROSS DESERT, HORIZON IS BETWEEN MID AND BOTTOM FRAME

NARRATOR
Welcome to the Virtual Reality presentation of Cataglyphis ants in the North African salt-pan. Today, we will learn about their navigation strategies. We will look into how these small desert navigators accomplish their daily foraging task.

There are some Cataglyphis fortis ants roaming around on the desert floor

TILT CAMERA DOWN AT THE CROWDS OF ANTS

NARRATOR
Cataglyphis ants go out to find food items in the hottest part of the day, when other animals would rather rest in the shade.

During the hottest time of the day, the ground temperature in the desert may reach up to 70°C.

The are some Cataglyphis fortis ants idling and spinning around on the desert floor.

HIGH ANGLE AND ZOOM IN ON GROUP OF ANTS

NARRATOR
Cataglyphis ants rely on many cues to navigate, such as; olfactory cues, any available landmarks along a route, and path integration.

2 EXT.DESERT.DAY.PATH INTEGRATION

A Cataglyphis fortis is idling near a nest entrance.

HIGH ANGLE - CLOSE SHOT, ON BOTH NEST AND ANT

NARRATOR
In the absence of visual landmarks and olfactory cues, Cataglyphis ants rely solely on

(MORE)

(CONTINUED)

NARRATOR (cont'd)
path integration. Path integration allows Cataglyphis ant to keep track of its direction and distance from the nest.

C.fortis starts the foraging journey, not far from the nest. Trail movement is shown on screen.

DOLLY BACK - KEEP THE ANT ON THE FRAME

NARRATOR
Path integration requires the measurement of 2 parameters; the direction and the distance of travel. Directional information is provided by an external celestial compass. While an internal pedometer accomplishes distance measurement.

C.fortis is on foraging journey, quite far from the nest

DOLLY BACK - KEEP THE ANT ON THE FRAME

NARRATOR
Cataglyphis ants would go through a tortuous outbound journey in finding a food item. During the journey, Cataglyphis ants keep track of direction and distance from the nest.

C.fortis found a food item and is heading back home

DOLLY - FOLLOWS THE ANT

NARRATOR
Once a food item is found, the ant will go back to its nest by the shortest route. As we can see, the homebound journey is almost a straight line. This straight line represents the sum of the distance and the direction travelled. Path Integration is similar to a mathematical vector addition.

The ant is safely back to its nest with a food item

HIGH ANGLE - LONG SHOT ON GRID

TILT UP ON HORIZON

3 EXT.DESERT.DAY.SKYLIGHT PATTERN

A plain, featureless desert extends to the horizon. The sun shines brightly in the dark blue sky.

TILT UP TO THE SUN, FADE IN POLARIZATION PATTERN, HORIZON IS CLOSE TO BOTTOM FRAME

NARRATOR
Let us now look into Cataglyphis ant's celestial compass system. The compass system is based on the skylight pattern, the pattern of linearly polarized light that we cannot see. This pattern is caused by the scattering of sunlight by the air molecules in the atmosphere.

The Sun is rising about 20 degrees and polarization pattern is moving along with the sun as well.

KEEP CAMERA AT SAME HEIGHT

NARRATOR
Unlike us, Cataglyphis ants have visual receptors that are sensitive to this scatter. They are able to utilise the polarized skylight pattern to measure their direction relative to the nest while foraging.

DOLLY CAMERA TO NEXT LOCATION

4 EXT.DESERT.DAY.ODOMETRY EXPERIMENT

Four tunnels placed on the desert floor. Each tunnel has a marker nearby denoting which experiment will be applied on that particular channel. The tunnel is made from white-painted non-reflective aluminium.

HIGH ANGLE SHOT ON THREE CHANNELS (DIAGONALLY ON SCREEN)

NARRATOR
Let us now look into the distance estimation or the odometry behaviour of Cataglyphis ants. Cataglyphis ants use an internal pedometer to counts steps. The concept is similar to a car's odometer that registers the distance the car has travelled. These ants are trained to travel in a tunnel. The tunnels constrain ants' movements and

(MORE)

(CONTINUED)

NARRATOR (cont'd)
allow easy measurement of distance travelled.

Three ants are released and started foraging right away in the channel.

DOLLY - TRACK ON THREE ANTS

NARRATOR
In this experiment, we will modify the length of the ants' legs once they have picked up the food items in the feeder area ahead. The experimental manipulation will affect the stride length of the ants. So, if these ants count steps, shorter legs would cause the ant to underestimate distance travelled and longer legs would cause the ants to overestimate distance travelled.

The ants are running towards the feeder area.

DOLLY CAMERA TOWARDS THE FEEDER AREA

The ants finally reach the feeder.

DOLLY CAMERA ON THREE ANTS

NARRATOR
Let us pick them up before they head home with the food items. We will apply the experimental conditions by modifying the length of the legs.

Lift up and move the ants to different channels for their homebound journey.

DOLLY CAMERA AROUND CONTROL CONDITION ANT

NARRATOR
This is the control condition ant. No leg manipulation will be performed on this ant.

DOLLY CAMERA AROUND STILTS CONDITION ANT

NARRATOR
The Legs of this ant are extended artificially by gluing stilts to them. This is the stilts condition and the stride would be longer than the control condition.

Sticks are being glued on the legs.

DOLLY CAMERA AROUND STUMPS CONDITION ANT

NARRATOR
For this ant, the legs are shortened by cutting them physically. This is the stumps condition and the stride would be shorter than the control condition.

Legs are being shortened.

NARRATOR
Let us see how these conditions would affect distance estimation.

Ants are lowered down to the tunnels.

CAMERA TILTS DOWN ON THE ANTS

The ants are heading back to the nest.

DOLLY CAMERA AROUND TUNNELS - HIGH ANGLE SHOT

The normal (Control) condition ant estimates distance travelled from the nest rather accurately, however the other two experimental ants misgauge the distance travelled.

NARRATOR
The red markers are the fictive nest locations. They are at the same distance down the channel as the nest is from the feeder.

What are we looking at here? The ant on stilts overestimated the distance travelled, while the ant on stumps underestimated it. This shows that the ants use an internal pedometer in estimating distance and the basic premise of step counting is correct.

The ants are looking for the nest entrance around the fictive nest locations.

DOLLY CAMERA TO SUMMARY LOCATION

5 EXT.DESERT.DAY.SUMMARY

There are a group of moving ants on the desert floor near the nest entrance. Some are idling and some are moving about.

CAMERA TILTS DOWN - HIGH ANGLE SHOT ON CROWDS OF ANTS

NARRATOR
Let us summarise what we have learned so far. Cataglyphis ants have several navigation systems; olfactory cues, route based strategy and path integration. In a flat and featureless terrain, path integration is the only method of navigation.

DOLLY CAMERA BACK AND TILTS UP

NARRATOR
By relying on a skylight compass and an internal pedometer, the ant keeps track of its position relative to the nest. So, after foraging in a tortuous path to find a food item, Cataglyphis ants are able to sum all vectors travelled, and head home in almost a straight line.

TILT UP TO HORIZON AND FADE OUT POLARIZATION PATTERNS

NARRATOR
Thank you for watching this presentation and we hope that you've enjoyed learning about the Cataglyphis ant navigation system. This is the end of the presentation.

D.2 Screenplay of VEAR

VR-VILLAGE FIREFIGHTER (VF/3.9): VILLAGE FIRE BEHAVIOUR (CAR ACCIDENTS)

written by

Iwan Kartiko
ikartiko@science.mq.edu.au

Based on:
Village Firefighter Manual (VF/1-6, 1999)
ISBN: 0-7310-9730-3-x
SPECIALIST Training Manual (SP/3, 1995)
ISBN: 0-7310-6432-1
NSW Rural Fire Service
The New South Wales Government

1 EXT.FREEWAY.AFTERNOON.INTRODUCTION 1

A long freeway, which extends into a tunnel.

Dolly camera into the tunnel.

NARRATOR
Welcome to the Virtual Reality presentation of Village Firefighting part three, section nine. Village Firefighting consists of fighting fire in small structures and vehicle accident.

Tilts camera down, high angle shot on the scene.

Dolly camera closer.

NARRATOR
In this session, we will look into car accidents. You will learn about the various hazards, and ways to avoid them to ensure safe and effective firefighting.

There are two cars involved in an accident.

2 EXT.FREEWAY.AFTERNOON.PREPARATION 2

Dolly Camera around a firefighter.

NARRATOR
Before looking at the hazards in car accidents, we will look very briefly into the Compressed Air Breathing Apparatus or CABA. CABA must be used in environments where there is danger to the respiratory system. This includes, but is not restricted to toxic gases, oxygen deficient atmosphere, hot atmosphere and smoke.

3 EXT.FREEWAY.AFTERNOON.VEHICLE ROLL 3

Dolly Camera around the car at eye-height

NARRATOR
There are a number of hazards to be aware of in a car accident involving fire. The first one we will consider is vehicle roll or tip. A vehicle involved with fire may tip over, may continue to free-roll, or even drive under its
(MORE)

NARRATOR (CONT'D)
own power. This may endanger trapped persons in the vehicle and firefighters, and may contribute to the spread of the fire.

Open the driver's side door. Remove the unconscious driver.

Zoom camera into the interior of the car.

Zoom camera into the key.

NARRATOR
Vehicle roll can be prevented by switching off ignition if it is safe to do so,

Take out and fade out the key

Zoom Camera into the handbrake

NARRATOR
by applying the handbrake,

Apply handbrake.

Dolly camera out on the wheels.

NARRATOR
and by chocking the wheels.

Dolly camera out on the car on fire

NARRATOR
To avoid injuries from vehicle roll or tip, firefighters should approach the vehicle from the uphill side. Similarly, when there is wind fanning a fire, the vehicle should be approached from the windward side to avoid injury from smoke and flames.

4 EXT.FREEWAY.AFTERNOON.EXPLOSIONS 4

Low angle - close shot on the fuel tank

NARRATOR
The second hazard to consider is explosions. When exposed to a severe and sustained fire a vehicle's fuel tank may burst. If no-one is trapped,and it is believed that fire has impinged on the fuel container, concentrate on cooling the fuel container.

Dolly camera to capture ALL tyres from the side

NARRATOR
Sealed components, such as inflatable bumpers, tyres, airshocks and driveshafts may burst or explode during fire. If the vehicle is large, this can be very dangerous. This may cause the vehicle to drop several centimetres. Never put your hands or limbs between the wheel suspension and chassis.

5 EXT.FREEWAY.AFTERNOON.HAZARDOUS MATERIAL 5

Dolly camera into the engine part

NARRATOR
The third hazard is Hazardous Materials. Some vehicle components may become hazardous when involved in a fire.

Dolly camera around burned engine part with O ring and exposed electrical circuit.

NARRATOR
For example, Viton, a plastic used in some gaskets, ``O''rings, electrical items and other components can decompose into hydrofluoric acid when strongly heated. This is a powerful acid. Any contact with skin can lead to severe injury, even though it may not cause an immediate sensation of burning or stinging.

Dolly camera to capture the nearest firefighter.

NARRATOR
Always wear your full protective clothing and avoid skin contact with fire residues.

6 EXT.FREEWAY.AFTERNOON.ELECTRICAL HAZARD 6

Dolly Camera to the front of car

NARRATOR
The fourth hazard is electrical.

Tilt camera up to see the power lines.

NARRATOR
A vehicle fire or accident may involve fallen power lines. Always be on the alert for such hazard. You must remain well clear until an authorised electrician has confirmed that the power has been shut off.

Tilt Camera down on car and the firefighter

NARRATOR
This vehicle, however, has stopped before knocking down any power lines.

7 EXT.FREEWAY.AFTERNOON.VEHICLE INCIDENT -NO FIRE 7

Dolly camera to the other car.

A firefighter is foaming some spillage on the road.

NARRATOR
However, not all car accident involve fires. But when you see fuel spillage, be on guard for possible ignition. A few steps can be done to reduce the chance of fire ignition. Firstly, by blanketing the spillage with foam.

Dolly camera, high angle shot on spillage area.

Flash no smoking sign in the area.

NARRATOR
Secondly, prohibiting smoking and other ignition sources in the area.

Dolly camera to the engine compartment.

Zoom camera into the car battery.

NARRATOR
Thirdly, disconnecting the vehicle's battery lead with the earth terminal first.

Open cover. Disconnect leads with earth terminal first.

zoom out to cover.

Close cover.

Dolly Camera closely around the car.

A firefighter is standing by with the hose.

NARRATOR
You must have adequate and appropriate firefighting equipment ready for immediate use, if anyone could be endangered by fuel catching alight. Do not remove injured persons unless they are in imminent danger.

8 EXT.FREEWAY.AFTERNOON.SUMMARY 8

Dolly camera around the scene.

NARRATOR
Let us summarise what we have learned so far. In an accident with fire, there are four primary hazards: vehicle roll or tip, explosions,electrical hazards and Hazardous Materials. To prevent vehicle roll, switch off ignition if safe, apply the handbrake and chock the wheels. Fuel tanks may explode when they are, or have been severely impinged by fire. If no-one is trapped,and it is believed that fire has impinged on the fuel container, concentrate on cooling the fuel tank. If there are fallen power lines, remain clear until the power is disconnected. Wear the protective gear always, as vehicles' component may dissolve into Hazardous Materials at high temperatures. If possible, firefighters must position themselves uphill and not face into the direction of the wind. In an accident without a fire, the possibility of a fire starting can be reduced by blanketing spillage with foam, prohibiting smoking and isolating the ignition sources in the area, and disconnecting the vehicle's battery. You must know the hazards in car accidents to ensure safe and effective firefighting. Thank you for watching this presentation and we hope that you've enjoyed learning about Village Firefighting. See you next time.

Appendix E

Questionnaires

This chapter of the appendix contains all the questionnaires used in this thesis. Each section contains the information and consent forms, pretest questionnaires (part A), and posttest questionnaires (part B and C). The Information and Consent form was modified slightly in the second study due to formal changed that took place by the Ethics Committee.

E.1 Questionnaires of ANTECATALYST

Information Statement and Consent Form

Animation Complexity and Learning Outcome: Case Study I - Odometric Behaviour of Cataglyphis Ants

You are invited to participate in a study of navigation behaviour of Cataglyphis ants in an immersive Virtual Reality (VR). This study is linked to undergraduate unit PSY236. This study will take place during PSY236 practical session. **Participation is voluntary and will not affect progression in the course**.

The purpose of this study is to evaluate the effects of animation complexity and learning outcome in an immersive VR. VR has been used for many purposes, including learning and acquiring a new knowledge. However, more research is required in fine tuning VR technology to foster better learning outcomes.

The study is being conducted by Iwan Kartiko in the Computing Department as a part of his doctoral thesis. Iwan Kartiko is the chief investigator in this study.

The study will take up to seventy (70) minutes and will involve watching a scenario in VR. Before and after watching the scenario in VR you will be required to answer questionnaires. Iwan Kartiko will produce a report based on the outcome of this study. The overall (not individual) results will be posted on the password-protected course web-site. Do not include your name on the questionnaires. **Gender and age information will be collected in this study. Participants will not be identifiable in any publication of the results of the study**.

VR systems have known to induce motion-sickness and may trigger epileptic seizures. The likehood is minimal in this study. However, you **CANNOT** participate if you are sensitive to motion sickness **OR** if you ever had an epileptic seizure. A verbal warning will be given prior commencing the study.

Please read the following points **carefully**:

- Should you decide to participate, you may quit anytime during the study but please remain in the VR lab.
- Whenever you feel uncomfortable during the study, please take off your 3D goggles and let the chief investigator know. Chief investigator will be in the VR lab the entire session.
- Should you decide to stay during the study and experience severe discomfort, we will refer you to on-campus medical service.
- Should you decide to participate and experience prolonged discomfort of motion sickness after the study, please seek advice from on-campus medical service. The location of on-campus medical service is included in this information statement and consent form.

Medical Service on campus
Level 2 C8A Lincoln Student Services Building Macquarie University, Sydney NSW 2109 Australia (02) 9850 7497

For further queries about this study, please contact:

Name	Contact number	e-mail address	Office
Iwan Kartiko (Chief Investigator)	9850 9081	ikartiko@ics.mq.edu.au	E6A - 356
Dr. Manolya Kavakli (Supervisor)	9850 9572	manolya@ics.mq.edu.au	E6A - 372
Dr. Kenneth Cheng (Supervisor)	9850 8613	ken@galliform.bhs.mq.edu.au	W21A

I, _____ have read (or, where appropriate, have had read to me) and understand the information given and any questions I have asked have been answered to my satisfaction. I agree to participate in this research, knowing that I can withdraw from further participation in the research at any time without consequence. I have been given a copy of this signed form to keep.

Participant's Name:_____ (block letters)

Participant's Signature: _____ Date:_____

Investigator's Name:_____ (block letters)

Investigator's Signature: _____ Date: _____

The ethical aspects of this study have been approved by the Macquarie University Ethics Review Committee (Human Research). If you have any complaints or reservations about any ethical aspect of your participation in this research, you may contact the Committee through the Research Ethics Officer (telephone [02] 9850 7854, fax [02] 9850 8799, email: ethics@mq.edu.au). Any complaint you make will be treated in confidence and investigated, and you will be informed of the outcome

INVESTIGATOR'S COPY

Information Statement and Consent Form

Animation Complexity and Learning Outcome: Case Study I - Odometric Behaviour of Cataglyphis Ants

You are invited to participate in a study of navigation behaviour of Cataglyphis ants in an immersive Virtual Reality (VR). This study is linked to undergraduate unit PSY236. This study will take place during PSY236 practical session. **Participation is voluntary and will not affect progression in the course**.

The purpose of this study is to evaluate the effects of animation complexity and learning outcome in an immersive VR. VR has been used for many purposes, including learning and acquiring a new knowledge. However, more research is required in fine-tuning VR technology to foster better learning outcomes.

The study is being conducted by Iwan Kartiko in the Computing Department as a part of his doctoral thesis. Iwan Kartiko is the chief investigator in this study.

The study will take up to seventy (70) minutes and will involve watching a scenario in VR. Before and after watching the scenario in VR you will be required to answer questionnaires. Iwan Kartiko will produce a report based on the outcome of this study. The overall (not individual) results will be posted on the password-protected course website. Do not include your name on the questionnaires. **No personal details are gathered in this study. Participants will not be identifiable in any publication of the results of the study**.

VR systems have known to induce motion-sickness and may trigger epileptic seizures. The likehood is minimal in this study. However, you **CANNOT** participate if you are sensitive to motion sickness **OR** if you ever had an epileptic seizure. A verbal warning will be given prior commencing the study.

Please read the following points **carefully**:

- Should you decide to participate, you may quit anytime during the study but please remain in the VR lab.
- Whenever you feel uncomfortable during the study, please take off your 3D goggles and let the chief investigator know. Chief investigator will be in the VR lab the entire session.
- Should you decide to stay during the study and experience severe discomfort, we will refer you to on-campus medical service.
- Should you decide to participate and experience prolonged discomfort of motion sickness after the study, please seek advice from on-campus medical service. The location of on-campus medical service is included in this information statement and consent form.

Medical Service on campus
Level 2 C8A Lincoln Student Services Building Macquarie University, Sydney NSW 2109 Australia (02) 9850 7497

For further queries about this study, please contact:

Name	Contact number	e-mail address	Office
Iwan Kartiko (Chief Investigator)	9850 9081	ikartiko@ics.mq.edu.au	E6A - 356
Dr. Manolya Kavakli (Supervisor)	9850 9572	manolya@ics.mq.edu.au	E6A - 372
Dr. Kenneth Cheng (Supervisor)	9850 8613	ken@galliform.bhs.mq.edu.au	W21A

I, _____ have read (or, where appropriate, have had read to me) and understand the information given and any questions I have asked have been answered to my satisfaction. I agree to participate in this research, knowing that I can withdraw from further participation in the research at any time without consequence. I have been given a copy of this signed form to keep.

Participant's Name:_____ (block letters)

Participant's Signature: _____ Date:_____

Investigator's Name:_____ (block letters)

Investigator's Signature: _____ Date: _____

The ethical aspects of this study have been approved by the Macquarie University Ethics Review Committee (Human Research). If you have any complaints or reservations about any ethical aspect of your participation in this research, you may contact the Committee through the Research Ethics Officer (telephone [02] 9850 7854, fax [02] 9850 8799, email: ethics@mq.edu.au). Any complaint you make will be treated in confidence and investigated, and you will be informed of the outcome

PARTICIPANT'S COPY

Part A. Pre-test questionnaires

Demographic Questionnaire

Please specify your gender (M = Male/ F = Female):_____

Please specify your age:_____

Simulator Sickness Questionnaire

Please report the degree to which you experience each of the below symptoms as one of “None”, “Slight”, “Moderate” and “Severe”. Using the scale from "0" (none) to "3" (severe).

Symptom	None	Slight	Moderate	Severe
General discomfort	0	1	2	3
Fatigue	0	1	2	3
Headache	0	1	2	3
Eyestrain	0	1	2	3
Difficulty focusing	0	1	2	3
Increased salivation	0	1	2	3
Sweating	0	1	2	3
Nausea	0	1	2	3
Difficulty concentrating	0	1	2	3
Fullness of head	0	1	2	3
Blurred vision	0	1	2	3
Dizzy (eyes open)	0	1	2	3
Dizzy (eyes closed)	0	1	2	3
Vertigo	0	1	2	3
Stomach awareness	0	1	2	3
Burping	0	1	2	3

Pre-Test : Odometric Behaviour of Cataglyphis Ants

1 Retention Questionnaires

Please answer all 8 questions. Please circle the correct answer. This is a practice test and not part of course grades.

1. Following are methods of navigation of Cataglyphis ants;
   a) Path integration, path compensation and the use of olfactory cues.
   b) Distance estimation towards food source from the nest and memorization of terrain topography features.
   c) Landmark based route, use of olfactory cues and path integration.
   d) Path integration, olfactory cues and path factorization.
   e) Distance estimation towards food source from the nest and path integration.
2. On a flat desert with no landmarks, the only mode of navigation for Cataglyphis ants is;
   a) Path factorisation.
   b) Path memorization.
   c) Path integration.
   d) Path multiplication.
   e) Path compensation.
3. What are the components of Cataglyphis ants path integration?
   a) Memorization of 3D map and path multiplication.
   b) Integration of direction, distance and path compensation.
   c) Integration of direction, distance and path multiplication.
   d) Integration of direction, distance and optic-flow information that tells the ants about its location from the food source.
   e) Integration of direction and distance information that tells the ants about its location from the nest.

4. How do Cataglyphis ants accomplish direction measurement?

    a) By utilising a celestial compass based on polarized skylight pattern.

    b) By memorization of the 3D map.

    c) By measuring energy expenditure during foraging.

    d) By utilising ventral-field visual cues.

    e) By utilising lateral-field visual cues.

5. How do Cataglyphis ants accomplish distance measurement?

    a) By utilising celestial a compass based on polarized skylight pattern.

    b) By calculating time spent while traveling.

    c) By gauging energy expenditure while foraging.

    d) By memorization of the 3D map.

    e) By an internal pedometer, or a step counter.

6. Once an ant has picked up a food source on a flat and featureless desert, how would the ant go back to the nest?

    a) It traces the entire route traveled.

    b) It sends a signal to get another ant to show the way home.

    c) It heads directly to its nest in almost a straight line.

    d) It moves randomly to find its nest.

    e) It switches to landmark navigation method based on memorisation of the 3D terrain.

7. Once an ant picked up a food item, what will happen when the legs are extended artificially before it heads back to the nest?

    a) It will gauge the distance correctly to the nest.

    b) It will misgauge the distance traveled, underestimating the distance of the nest.

    c) The ant will be able to go back to the nest faster and better in approximating entry of the nest.

    d) It will misgauge the distance traveled, overestimating the distance of the nest.

    e) It will gauge the distance traveled properly, but cannot go back to the nest.

8. Once an ant picked up a food item, what will happen when the legs are shortened before it heads back to the nest?

   a) It will misgauge the distance traveled, underestimating the distance of the nest.

   b) It will misgauge the distance traveled, overestimating the distance of the nest.

   c) It will gauge the distance traveled properly, but cannot go back to the nest.

   d) The ant will go back to the nest slower and better in approximating entry of the nest.

   e) It will gauge the distance correctly to the nest.

2 Transfer Questionnaires

Please answer all 4 questions. Please circle the correct answer. This is a practice test and not part of course grades.

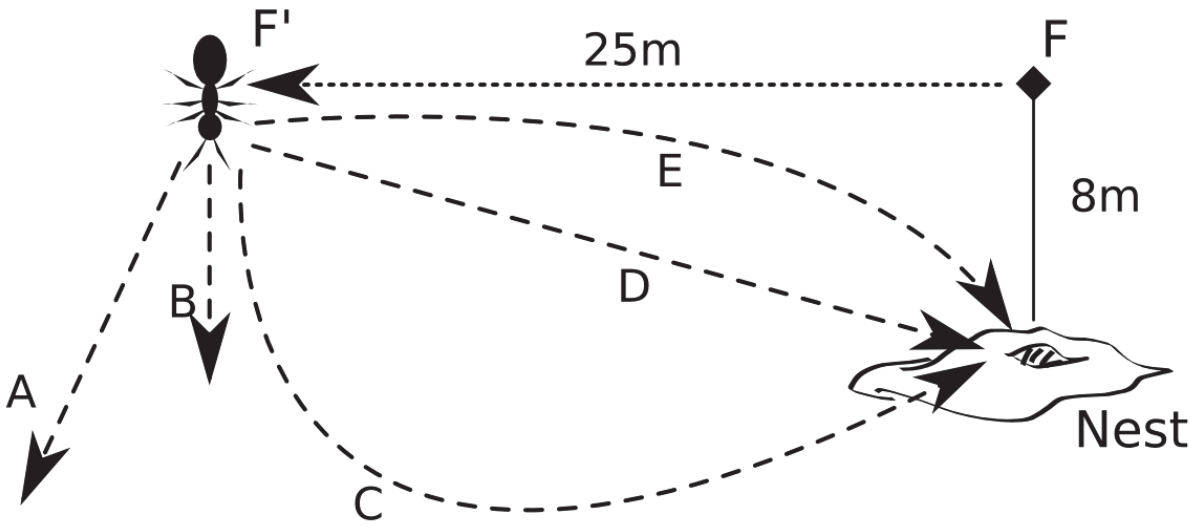


Figure 2.1: Question 3-1

1. After walking 8 meters towards the north, an ant found a food item at point F. Before the ant headed back to the nest, it was relocated 25 meters to point F'. Which path will the ant most likely take?

   a) Path A

   b) Path B

   c) Path C

   d) Path D

   e) Path E

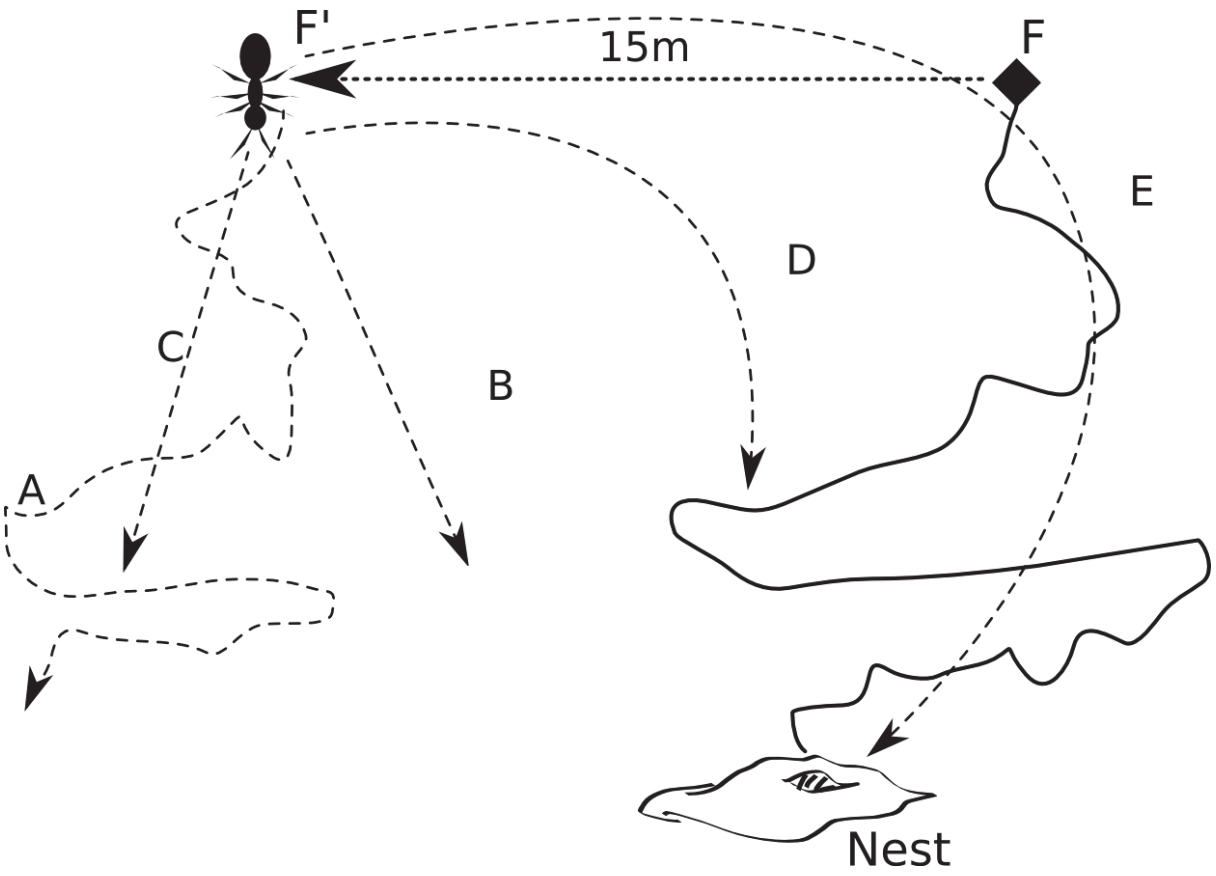


Figure 2.2: Question 3-2

2. After a long journey from the Nest, a Cataglyphis ant found a food item at point F. Before headed back to the nest, the legs were shortened, and the ant was relocated 15 meters to F'. Which path will the ant most likely take?

   a) Path A

   b) Path B

   c) Path C

   d) Path D

   e) Path E

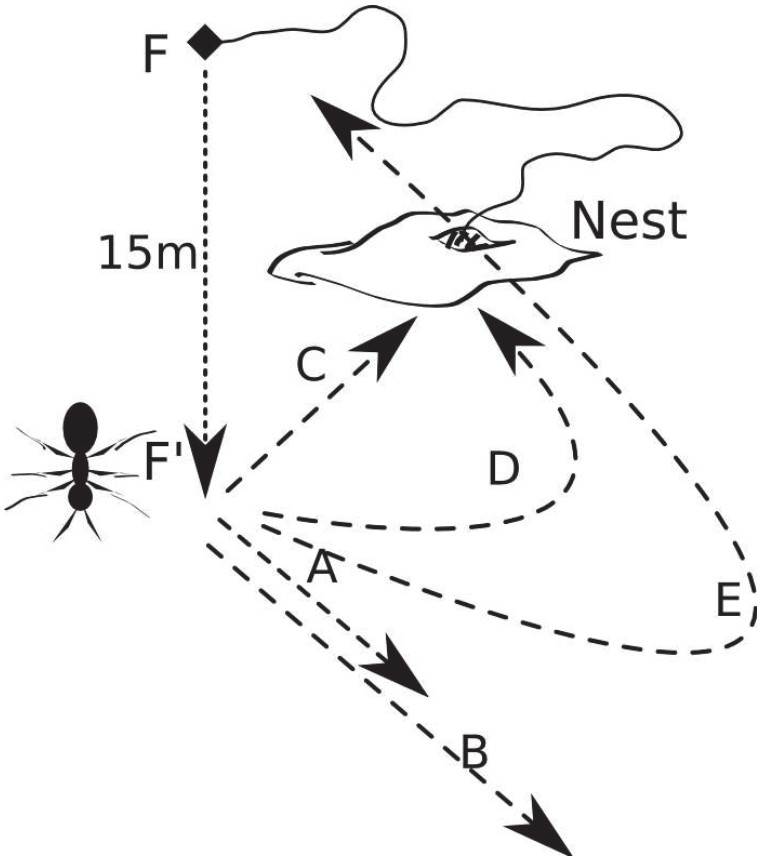


Figure 2.3: Question 3-3

3. An ant was captured at point F. Before headed back to its nest with a food item, stilts were attached to the legs and the ant was displaced 15 meters to point F'. Which path will the ant most likely take?

   a) Path A

   b) Path B

   c) Path C

   d) Path D

   e) Path E

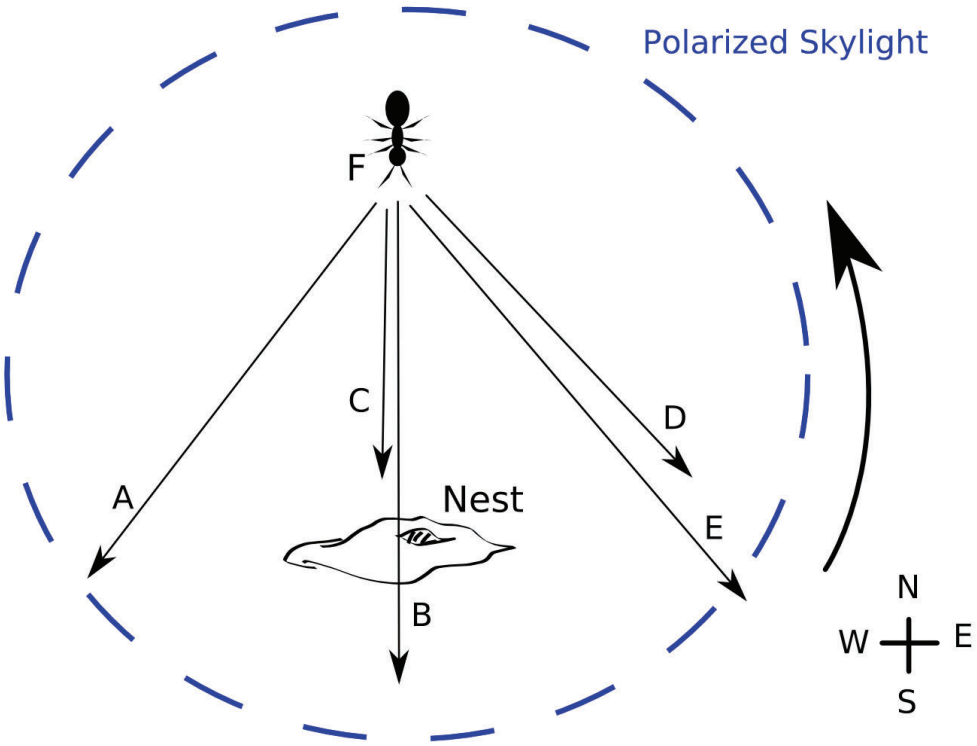


Figure 2.4: Question 3-4

4. A Cataglyphis ant travelled to the north and found a food item at point F. Before headed back to the nest the legs were shortened and the polarized skylight pattern was rotated 45 degrees in the direction shown in illustration above. Which path will the ant most likely take heading home?

   a) Path A

   b) Path B

   c) Path C

   d) Path D

   e) Path E

Part B. Post-test questionnaires: learning experience

1

Simulator Sickness Questionnaire

Instructions. Please report the degree to which you experience each of the below symptoms as one of “None”, “Slight”, “Moderate” and “Severe”. Using the scale from "0" (none) to "3" (severe).

Symptom	None	Slight	Moderate	Severe
General discomfort	0	1	2	3
Fatigue	0	1	2	3
Headache	0	1	2	3
Eyestrain	0	1	2	3
Difficulty focusing	0	1	2	3
Increased salivation	0	1	2	3
Sweating	0	1	2	3
Nausea	0	1	2	3
Difficulty concentrating	0	1	2	3
Fullness of head	0	1	2	3
Blurred vision	0	1	2	3
Dizzy (eyes open)	0	1	2	3
Dizzy (eyes closed)	0	1	2	3
Vertigo	0	1	2	3
Stomach awareness	0	1	2	3
Burping	0	1	2	3

Presence Questionnaires

Instructions. Please read the questions carefully and circle the appropriate number from 1 to 7.

1. How completely were all of your senses engaged?

1	2	3	4	5	6	7
Not engaged			Moderately			Very engaged

2. How much did the visual aspects of the environment involve you?

1	2	3	4	5	6	7
Not much			Moderately			Very much

3. How aware were you of events occurring in the real world around you?

1	2	3	4	5	6	7
Not aware			Moderately			Very aware

4. How inconsistent or disconnected was the information coming from your various senses?

1	2	3	4	5	6	7
Not inconsistent			Moderately			Very inconsistent

5. How compelling was your sense of moving around inside the virtual environment?

1	2	3	4	5	6	7
Not compelling			Moderately			Very compelling

6. To what degree did you feel confused or disoriented at the beginning of breaks or at the end of the experimental session?

1	2	3	4	5	6	7
Not disoriented			Moderately			Very disoriented

7. How involved were you in the virtual environment experience?

1	2	3	4	5	6	7
Not involved			Moderately			Very involved

8. How quickly did you adjust to the virtual environment experience?

1	2	3	4	5	6	7
Not quickly			Moderately			Very quickly

9. How much did the visual display quality interfere or distract you from the training experience?

1	2	3	4	5	6	7
Not much			Moderately			Very much

10. Were you involved in the experimental task to the extent that you lost track of time?

1	2	3	4	5	6	7
Not much			Moderately			Very much

Perceived Affective Quality

Instructions. Below is a list of words that can be used to describe places. We would like you to rate *how accurately* each word below describes the Virtual World you just visited. Use the following 1-8 rating scale for your answer. Please be sure that you have given an answer for *each word*.

1 = extremely inaccurate	5 = slightly accurate
2 = very inaccurate	6 = quite accurate
3 = quite inaccurate	7 = very accurate
4 = slightly inaccurate	8 = extremely accurate

intense

inactive

pleasant

dissatisfying

arousing

drowsy

nice

displeasing

active

idle

pleasing

repulsive

alive

lazy

pretty

unpleasant

forceful

slow

beautiful

uncomfortable

5

Program Ratings

Please answer all questions by circling corresponding answer.

1. How interesting was it to learn about Cataglyphis ants today?

1	2	3	4	5
Boring				Interesting

2. How entertaining was it to learn about Cataglyphis ants today?

1	2	3	4	5
Tiresome				Entertaining

3. How eager would you be to learn about a different animal behaviour topic in the same conditions you learned today?

1	2	3	4	5
Not eager				Very Eager

4. How motivating was it to learn about Cataglyphis ants today?

1	2	3	4	5
Not motivating				Motivating

5. How much did the VR learning system help you understand about navigation of Cataglyphis ants?

1	2	3	4	5
Not at all				Very much

6. How helpful was the VR simulation for learning about navigation of Cataglyphis ants?

1	2	3	4	5
Not helpful				Very helpful

7. How difficult was it to learn about navigation of Cataglyphis ants with the VR learning system?

1	2	3	4	5
Not difficult				Very difficult

8. How much effort did you have to invest to learn about navigation of Cataglyphis ants with the VR learning system?

1	2	3	4	5
No efforts				Lot of efforts

Would you like to give further comments?

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Part C. Post-test questionnaires: odometric behaviour of Cataglyphis ants.

1

1 Retention Questionnaires

Please answer all 8 questions. Please circle the correct answer. This is a practice test and not part of course grades.

1. Following are methods of navigation of Cataglyphis ants;

    a) Path integration, use of olfactory cues and landmark based route.

    b) Use of olfactory cues, path compensation and path integration.

    c) Landmark based route and path integration.

    d) Distance estimation towards food source from the nest and path integration.

    e) Distance estimation towards food source from the nest as well as mapping terrain feature in memory.

2. On a flat and featureless desert, the only mode of navigation for Cataglyphis ants is;

    a) Path randomisation.

    b) Path memorization.

    c) Path deduction.

    d) Path compensation.

    e) Path integration.

3. What are the components of Cataglyphis ants path integration?

    a) Integration of direction, distance and path compensation.

    b) Integration of direction and distance information that tells the ants about its location from the nest.

    c) Integration of direction, distance and path deduction.

    d) Memorization of the 3D map and path deduction.

    e) Integration of direction, distance and optic-flow information that tells the ants about its location from food source.

4. How do Cataglyphis ants accomplish direction measurement?

    a) By utilising lateral-field visual cues.

    b) By memorization of the 3D map.

    c) By utilising a celestial compass based on polarized skylight pattern.

    d) By measuring energy expenditure during foraging.

    e) By utilising ventral-field visual cues.

5. How do Cataglyphis ants accomplish distance measurement?

    a) By memorization of the 3D map.

    b) By utilising a celestial compass based on polarized skylight pattern.

    c) By gauging energy expenditure while foraging.

    d) By calculating time spent while travelling.

    e) By using an internal pedometer, or a step counter.

6. Once an ant has picked up a food source on a flat and featureless desert, how would the ant go back to the nest?

    a) It heads directly to its nest in almost a straight line.

    b) It traces the entire route traveled.

    c) It moves randomly to find its nest.

    d) It switches to landmark navigation method based on memorisation of 3D terrain.

    e) It sends a signal to get another ant to show the way home.

7. Once an ant picked up a food item, what will happen when the legs are extended artificially before it heads back to the point where it started foraging?

    a) It will misgauge the distance traveled, underestimating the distance of the nest.

    b) It will misgauge the distance traveled, overestimating distance of the nest.

    c) It will gauge the distance traveled properly, but cannot go back to the nest.

    d) The ant will go back to the nest slower and better in approximating entry of the nest.

    e) It will gauge the distance correctly to the nest.

8. Once an ant picked up a food item, what will happen when the legs are shortened before it heads back to the point where it started foraging?

   a) It will gauge the distance correctly to the nest.

   b) It will misgauge the distance traveled, underestimating the distance of the nest.

   c) The ant will be able to go back to the nest faster and better.

   d) It will misgauge the distance traveled, overestimating distance of the nest.

   e) It will gauge the distance traveled properly, but cannot go back to the nest.

Transfer Questionnaires

Please answer all 4 questions. Please circle the correct answer. This is a practice test and not part of course grades.

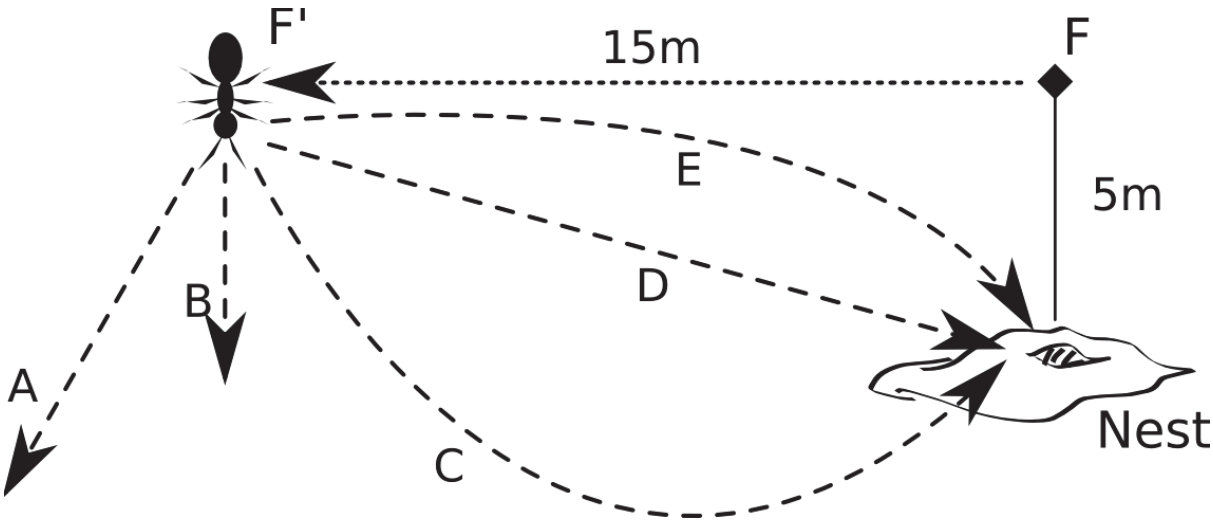


Figure 1.1: Question 3-1

1. After walking 5 meters towards the north, an ant found a food item at point F. Before the ant headed back to the nest, the experimenter relocated the ant 15 meters to point F'. Which path will the ant take that approximate its nest location?

   a) Path D

   b) Path C

   c) Path A

   d) Path B

   e) Path E

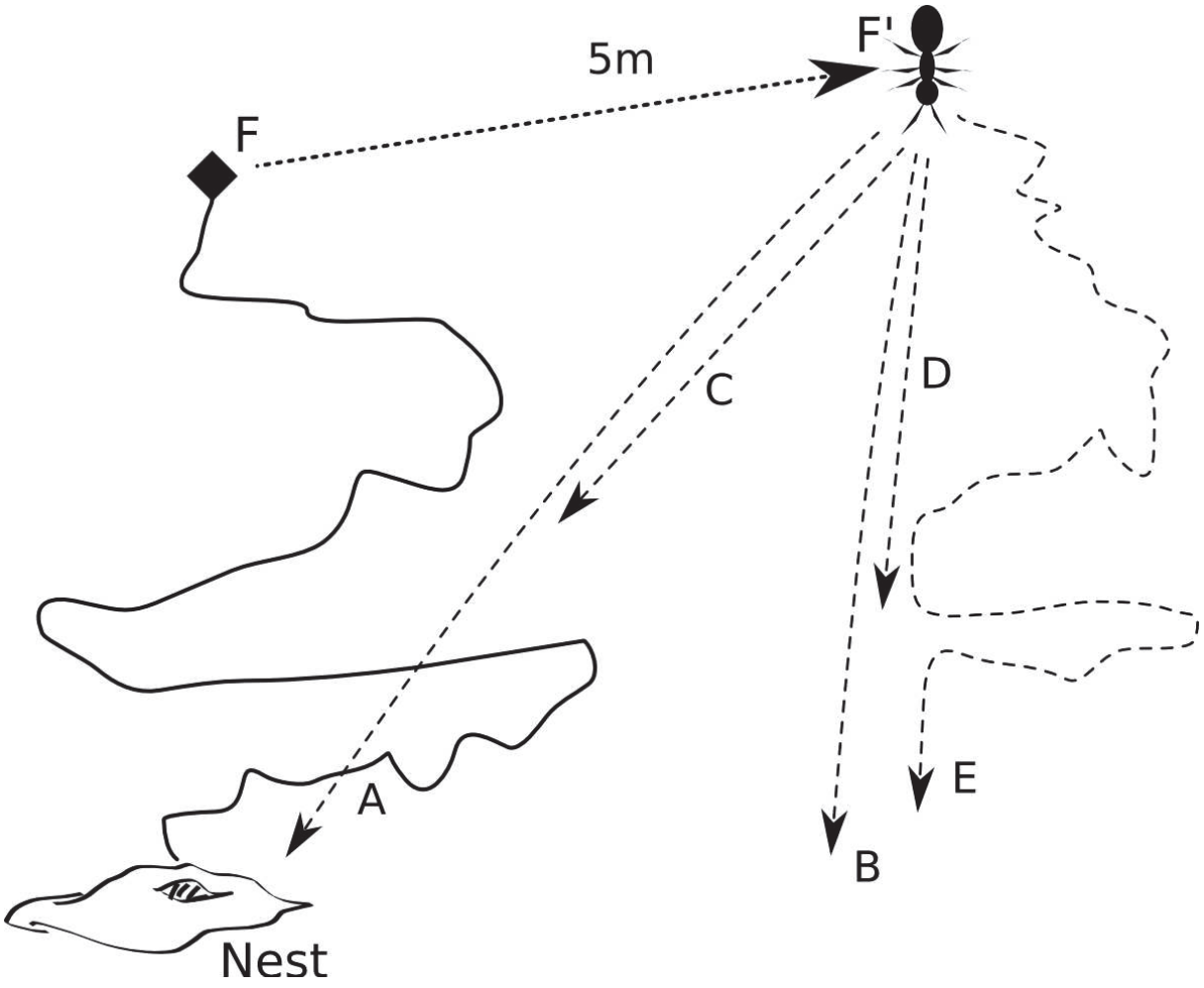


Figure 1.2: Question 3-2

2. After a long journey from the Nest, a Cataglyphis ant found a food item at point F. Before headed back to the nest, the ant was moved 5 meters to F' and the legs were also shortened. Which path will the ant take that approximate its nest location?

   a) Path B

   b) Path C

   c) Path E

   d) Path A

   e) Path D

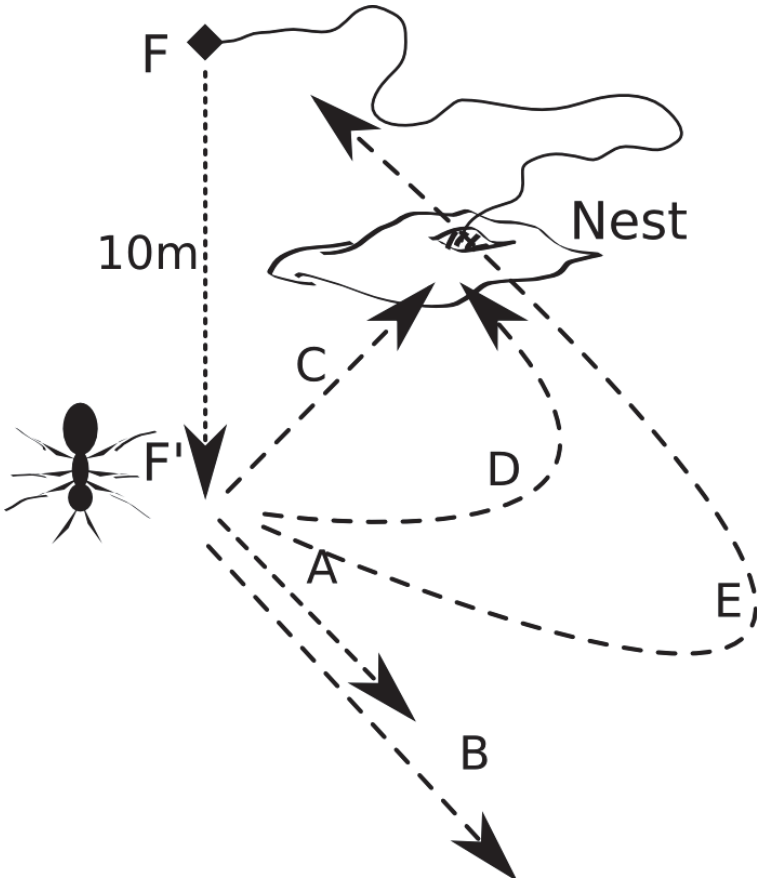


Figure 1.3: Question 3-3

3. An ant was captured at point F. Before headed back to its nest with a food item, it was displaced 10 meters to the south, at point F'. In this experiment, stilts were attached to the legs. Which path will the ant take that approximate its nest location?

   a) Path D

   b) Path A

   c) Path E

   d) Path C

   e) Path B

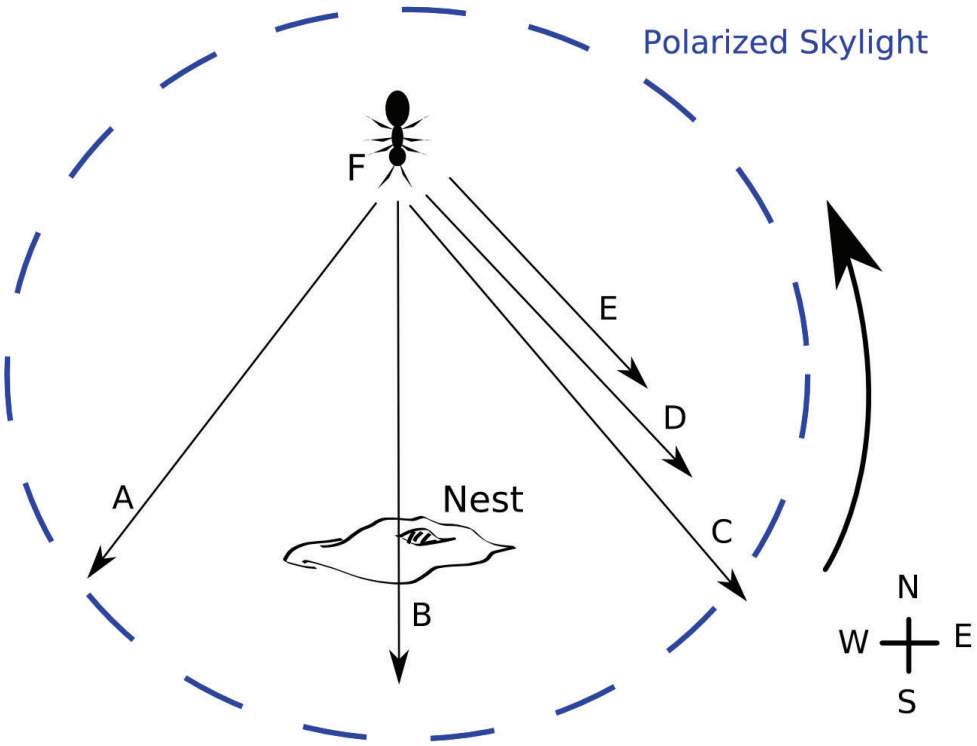


Figure 1.4: Question 3-4

4. A Cataglyphis ant travelled to the north and found a food item at point F. Before headed back to the nest the legs were lengthened and the polarized skylight pattern was rotated about 45 degrees in the direction shown in illustration above. Which path will the ant most likely take heading home?

   a) Path E

   b) Path D

   c) Path C

   d) Path B

   e) Path A

E.2 Questionnaires of VEAR

Information Statement and Consent Form

Animation Complexity and Learning Outcome: Case Study 2 - Village Firefighting

We invite you to take part in a study to learn about Village Firefighting in a Virtual Reality (VR).

The purpose of this study is to evaluate the effects of visual complexity and learning outcome in VR. VR has many purposes, including learning and gaining a new knowledge. However, we need more research to fine-tune VR technology to foster better learning outcomes.

Iwan Kartiko of the Department of Computing is conducting this study as a part of his doctoral thesis. Iwan Kartiko acts as the chief investigator in this study.

The study will take up to seventy (70) minutes and will involve watching a scenario in VR. Before and after watching the scenario in VR you will be required to answer questionnaires. At the end of the study, the participants will receive AUD$ 20 for their time. Iwan Kartiko will produce a report based on the outcome of this study. The overall (not individual) results will be emailed to those who are interested. Do not include your name on the questionnaires. **Gender and age information will be collected in this study. Participants will not be identifiable in any publication of the results of the study**.

VR may induce motion-sickness and may trigger epileptic seizures. The likelihood is minimal in this study. However, you **CANNOT** take part in this study if you are sensitive to motion sickness **OR** if you ever had an epileptic seizure. The chief investigator will give a verbal warning prior commencing the study.

Please read the following points **carefully**:

- Should you decide to take part in the study, you may quit any time during the study but please remain in the computer lab.
- Whenever you feel uncomfortable during the VR presentation, please let the chief investigator know. The chief investigator will be in the VR lab the entire session.
- Should you decide to stay during the study and experience severe discomfort, we will refer you to on-campus medical service.
- Should you decide to take part in the study and experience prolonged discomfort of motion sickness after the study, please seek advice from on-campus medical service. This information statement and consent form includes the location of on-campus medical service.

Medical Service on campus
Level 2 C8A Lincoln Student Services Building Macquarie University, Sydney NSW 2109 Australia (02) 9850 7497

For further queries about this study, please contact:

Name	Contact number	e-mail address	Office
Iwan Kartiko (Chief Investigator)	9850 9081	ikartiko@science.mq.edu.au	E6A - 256
Dr. Manolya Kavakli (Supervisor)	9850 9572	manolya@science.mq.edu.au	E6A - 372
Dr. Kenneth Cheng (Supervisor)	9850 8613	ken.cheng@mq.edu.au	W21A

I, _____ have read (or, where appropriate, have had read to me) and understand the information given and any questions I have asked have been answered to my satisfaction. I agree to take part in this research, knowing that I can withdraw without consequence. I have been given a copy of this signed form to keep.

Participant's Name:_____ (block letters)

Participant's Signature: _____ Date:_____

Investigator's Name:_____ (block letters)

Investigator's Signature: _____ Date: _____

The ethical aspects of this study have been approved by the Macquarie University Human Research Ethics Committee. If you have any complaints or reservations about any ethical aspect of your participation in this research, you may contact the Committee through the Director, Research Ethics (telephone (02) 9850 7854; email ethics@mq.edu.au). Any complaint you make will be treated in confidence and investigated, and you will be informed of the outcome.

INVESTIGATOR'S COPY

Information Statement and Consent Form

Animation Complexity and Learning Outcome: Case Study 2 - Village Firefighting

We invite you to take part in a study to learn about Village Firefighting in a Virtual Reality (VR).

The purpose of this study is to evaluate the effects of visual complexity and learning outcome in VR. VR has many purposes, including learning and gaining a new knowledge. However, we need more research to fine-tune VR technology to foster better learning outcomes.

Iwan Kartiko of the Department of Computing is conducting this study as a part of his doctoral thesis. Iwan Kartiko acts as the chief investigator in this study.

The study will take up to seventy (70) minutes and will involve watching a scenario in VR. Before and after watching the scenario in VR you will be required to answer questionnaires. At the end of the study, the participants will receive AUD$ 20 for their time. Iwan Kartiko will produce a report based on the outcome of this study. The overall (not individual) results will be emailed to those who are interested. Do not include your name on the questionnaires. **Gender and age information will be collected in this study. Participants will not be identifiable in any publication of the results of the study**.

VR may induce motion-sickness and may trigger epileptic seizures. The likelihood is minimal in this study. However, you **CANNOT** take part in this study if you are sensitive to motion sickness **OR** if you ever had an epileptic seizure. The chief investigator will give a verbal warning prior commencing the study.

Please read the following points **carefully**:

- Should you decide to take part in the study, you may quit any time during the study but please remain in the computer lab.
- Whenever you feel uncomfortable during the VR presentation, please let the chief investigator know. The chief investigator will be in the VR lab the entire session.
- Should you decide to stay during the study and experience severe discomfort, we will refer you to on-campus medical service.
- Should you decide to take part in the study and experience prolonged discomfort of motion sickness after the study, please seek advice from on-campus medical service. This information statement and consent form includes the location of on-campus medical service.

Medical Service on campus
Level 2 C8A Lincoln Student Services Building Macquarie University, Sydney NSW 2109 Australia (02) 9850 7497

For further queries about this study, please contact:

Name	Contact number	e-mail address	Office
Iwan Kartiko (Chief Investigator)	9850 9081	ikartiko@science.mq.edu.au	E6A - 256
Dr. Manolya Kavakli (Supervisor)	9850 9572	manolya@science.mq.edu.au	E6A - 372
Dr. Kenneth Cheng (Supervisor)	9850 8613	ken.cheng@mq.edu.au	W21A

I, _____ have read (or, where appropriate, have had read to me) and understand the information given and any questions I have asked have been answered to my satisfaction. I agree to take part in this research, knowing that I can withdraw without consequence. I have been given a copy of this signed form to keep.

Participant's Name:_____ (block letters)

Participant's Signature: _____ Date:_____

Investigator's Name:_____ (block letters)

Investigator's Signature: _____ Date: _____

The ethical aspects of this study have been approved by the Macquarie University Human Research Ethics Committee. If you have any complaints or reservations about any ethical aspect of your participation in this research, you may contact the Committee through the Director, Research Ethics (telephone (02) 9850 7854; email ethics@mq.edu.au). Any complaint you make will be treated in confidence and investigated, and you will be informed of the outcome.

PARTICIPANT'S COPY

Part A. Pretest questionnaires

Demographic Questionnaire

Please specify your gender (M = Male/ F = Female):_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Please specify your age:_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Simulator Sickness Questionnaire

Please report the degree to which you experience each of the below symptoms as one of "None", "Slight", "Moderate" and "Severe". Using the scale from "0" (none) to "3" (severe).

Symptom	**None**	**Slight**	**Moderate**	**Severe**
General discomfort	0	1	2	3
Fatigue	0	1	2	3
Headache	0	1	2	3
Eyestrain	0	1	2	3
Difficulty focusing	0	1	2	3
Increased salivation	0	1	2	3
Sweating	0	1	2	3
Nausea	0	1	2	3
Difficulty concentrating	0	1	2	3
Fullness of head	0	1	2	3
Blurred vision	0	1	2	3
Dizzy (eyes open)	0	1	2	3
Dizzy (eyes closed)	0	1	2	3
Vertigo	0	1	2	3
Stomach awareness	0	1	2	3
Burping	0	1	2	3

Pretest: Village Firefighting 3.9

1 Retention Questionnaires

Please answer all nine questions. Please circle the correct answer. This is a practice test and not part of course grades.

1. What is "Village Firefighting"?

    a) Firefighting in rural areas, at least 20km from the nearest town.
    b) Firefighting small structural fires in rural areas, at least 10km from the nearest town.
    c) Protecting villages from bushfire and vehicle accident.
    d) Firefighting within a small area such as structural fires and vehicle fires.
    e) Firefighting within a small area of 40km radius of nearest town.
    f) Firefighting within an open space such as structural fires and vehicle fires.

2. What does CABA stand for?

    a) Compressed Air Breathing Aid.
    b) Compressed Air Breathing Apparatus.
    c) Compressed Air Breathing Appliance.
    d) Compacted Air Breathing Assistance.
    e) Compacted Air Breathing Apparatus.
    f) Compacted Air Breathing Appliance.

3. When do you use CABA?

    a) Only within a 20m x 20m x 20m rescue space.
    b) When there is a danger to the respiratory system.
    c) Only when the Hazardous Materials are fully detected and identified.
    d) Only during rescue missions in rural areas where there is a danger to the respiratory system.
    e) When O_2 concentration drops below 90% and a hot environment within a rescue space of 20m x 20m x 20m.
    f) Only when you are told to do so during the briefing by your leader, due to the high-cost maintenance.

4. What should be done to reduce the risk of vehicle roll or tip in an accident?

    a) Place the firetruck right in front of the vehicle at 45 degrees angle.
    b) Switch off the engine when safe, and apply the handbrake.
    c) Switch off the engine when safe, apply handbrake and chock the wheels.
    d) Removing the driver immediately, switch off the engine when safe, apply handbrake and puncture the wheels to add surface friction with the road.
    e) Remove the driver immediately, switch off the engine when safe, apply handbrake, chock the wheels and place the firetruck at 45 degrees angle right in front of the vehicle.
    f) Remove the driver immediately, switch off the engine when safe, apply handbrake, puncture the wheels and place the firetruck at 45 degrees angle right in front of the vehicle.

5. What may contribute to a vehicle's explosions?

    a) The engine block.
    b) The vehicle's fuel tank and driveshafts.
    c) Inflatable bumpers, tyres and airshocks.
    d) The engine block and the vehicle's fuel tank.
    e) Inflatable bumpers, tyres, airshocks, driveshafts and vehicle's fuel tank.
    f) The engine block, vehicle's fuel tank, inflatable bumpers, tyres and airshocks.

6. If a vehicle accident involves fallen power lines...

    a) If the voltage is greater than 550kV, stay clear until an authorised staff has confirmed that electrical power has been shut off.
    b) If the voltage is less than 250kV, cut the powerlines off immediately.
    c) Remove the driver immediately at all cost to prevent further injuries.
    d) Stay clear until an authorised staff has confirmed that electrical power has been shut off.
    e) Stay clear until an authorised staff has confirmed that electrical power has been shut off when the power line configuration is tripole.
    f) Remove the driver immediately to prevent further burn injury when the power line configuration is less dangerous (for example, monopole and bipolar).

7. What will happen when Viton "O" rings and other electrical components are heated in a fire?

    a) They will turn into hydrofluoric acid and contact with skin may lead to a severe injury even if it may not cause immediate sensation of burning or stinging.
    b) They will turn into sulfuric acid and contact with skin may lead to a severe injury even if it may not cause immediate sensation of burning or stinging.
    c) They will turn into sulfuric acid and contact with skin may lead to an immediate sensation of burning or stinging.
    d) They will turn into hydrofluoric acid and contact with skin may lead to an immediate sensation of burning or stinging.
    e) They will turn into hydrofluoric acid and contact with skin may lead to an immediate cardiac arrest.
    f) They will turn into sulfuric acid and contact with skin may lead to an immediate cardiac arrest.

8. What can be done to reduce the likehood of a fire starting from a spillage on the road?

    a) Spray the spillage with water and quickly disconnect the vehicle's battery.
    b) Blanket the spillage with foam, prohibit smoking and other ignition sources.
    c) Blanket the spillage with CO_2 and disconnect vehicle's battery.
    d) Blanket the spillage with foam, prohibit smoking and other ignition sources, and disconnect vehicle's battery.
    e) Remove the driver immediately, blanket with CO_2, prohibit smoking and other ignition sources, and disconnect vehicle's battery.
    f) Remove the driver immediately, blanket with foam, prohibit smoking and other ignition sources.

9. How should firefighters approach a vehicle accident?

    a) From downhill side (when on a slope) and not facing the direction of the wind.
    b) 45 degrees angle in front of the casualty's vehicle and not facing the direction of the wind.
    c) From uphill side (when on a slope) and not facing the direction of the wind.
    d) Within the proximity of 5m radius of the accident and the nearest water hydrant.
    e) Not facing the direction of the wind.
    f) From uphill side (when on a slope) when the firetruck is downhill.

2 Transfer Questionnaires

Please answer all six questions. Please circle the correct answer. This is a practice test and not part of course grades.

1. Which incident below is most likely to be within the scope of Village Firefighting?

    a) A small-scale terrorist attack only within 10m of closest major public transport.

    b) A train stuck between two stations due to power failure.

    c) A small four-seater plane that made emergency landing in the sea due to fire on-board.

    d) A drifting 300-tonne carrier ship in a remote gulf due to engine failure.

    e) A five-seater plane caught on fire in rural area.

    f) All the above is beyond the scope of village firefighting.

2. You received a phone call about a burning storeroom (20m x 15m) of a warehouse that manufactures handheld computers. Which one of the following items would be most appropriate to ensure safe firefighting for the team?

    a) Compressed Air Breathing Assistance.

    b) Compressed Air Breathing Assistance and some non-acrylic based goggles.

    c) This is beyond the scope of village firefighting.

    d) Viton-based gloves to avoid sulfuric acid and Compressed Air Breathing Appliance.

    e) Compressed Air Breathing Apparatus.

    f) Compressed Air Breathing Apparatus and some non-acrylic based goggles.

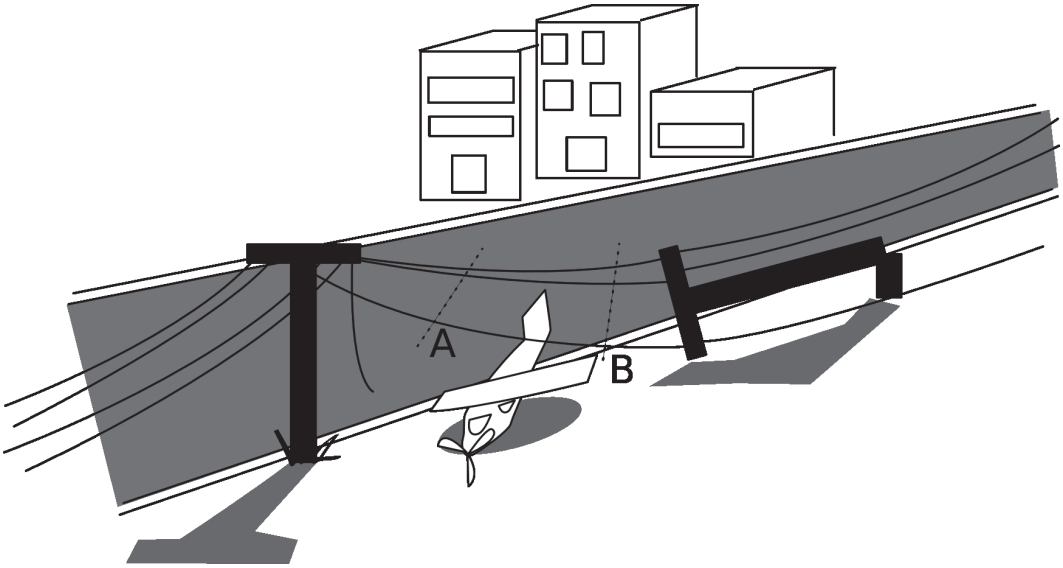


Figure 2.1: A small plane landed after an engine failure.

3. A small 2 seater plane landed in an emergency due to an engine failure shortly after taking off from a nearby rural airport. The situation is described in figure 2.1. The current condition of the pilot is unknown. What would be the best action upon your arrival at the scene?

   a) This question is beyond the scope of village firefighting.
   b) Cut the fallen lines at the dotted A and B line with appropriate equipment.
   c) Confirming if the electrical supply in the area has been cut off.
   d) Confirming if the electrical supply has been cut off and cut the fallen lines along the dotted A and B lines.
   e) Kerosene is highly flammable, focus the water on the engine and the fuel tank.
   f) Chock the wheels and the blades to avoid vehicle roll or tip.

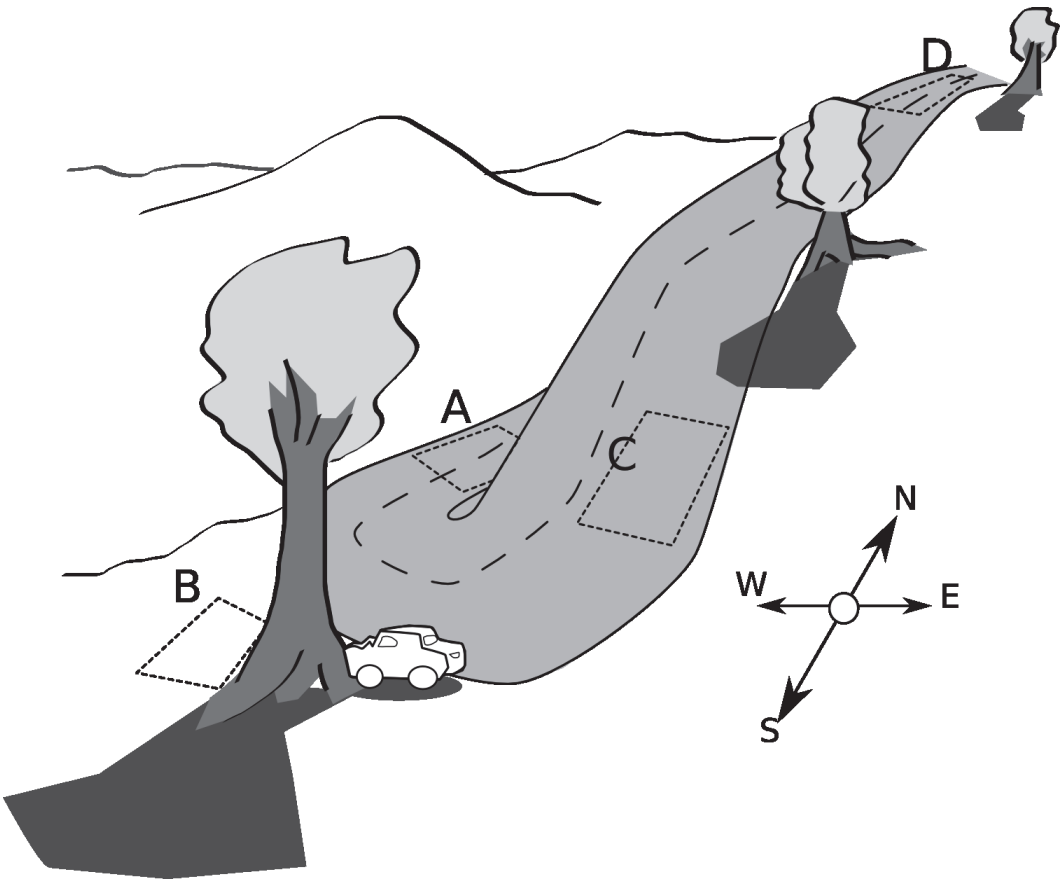


Figure 2.2: A vehicle accident on a hill.

4. A accident involving a small car is shown in figure 2.2. The driver appears as though he may have been speeding down the hill and wasn't quick enough for the sharp turn at the corner. A small amount of spillage is noticeable at the back of the car and on the road. The wind blows from the N direction. Which one of the following is the best option to handle the accident to ensure safe and effective firefighting?

   a) Approach from location A, B, C and D simultaneously with traffic control on these spots.
   b) Approach from location B with traffic control at location A.
   c) This is beyond the scope of village firefighting.
   d) Either approaching from C and A are fine.
   e) Approach from location D with traffic control at location A.
   f) Approach from above, the hill is too steep for such an emergency case.

5. Referring to figure 2.2, which one of the answers below represents the most appropriate action to prevent vehicle roll or tip and ignition of fire.

    a) The tree is big enough, just cut the vehicle battery and blanket the spillage with powder.

    b) Remove the driver immediately to reduce further neck injury and chock the wheels.

    c) Switch off the engine, remove the keys from the ignition, apply the handbrake, chock the wheels, blanket any spillage with foam, disconnect the battery, prohibit smoking and other ignition sources in the area.

    d) Chock the wheels and blanket the spillage on the road with foam.

    e) Switch off the engine, remove the keys from the ignition, apply the handbrake, chock the wheels, blanket any spillage with wet chemical, disconnect the battery, prohibit smoking and other ignition sources in the area.

    f) Any of (a), (c) and (e) after removing the driver from the seat.

6. What are some of the possible hazards for firefighters at a motor vehicle accident?

    a) Heart attack, trips, slips and falls, vehicle roll or tip, airbag deployment after the accident, electrical shock, and suffocating.

    b) Heart attack, airbag deployment after the accident, electrical shock, oil spillage, and psychological impacts.

    c) Heart attack, fuel spillage, exposure to casualty bodily fluids, airbag deployment before the accident, explosions, and suffocating.

    d) Trips, slips and falls, vehicle roll or tip, and electrical shock.

    e) Heart attack, trips, slips and falls, exposure to hazardous materials, oil spillage, psychological impacts, and third degree burns.

    f) Heart attack, trips, slips and falls, exposure to casualty bodily fluids, airbag deployment after the accident, vehicle roll or tip, and electrical shock.

Part B. Post-test questionnaires: learning experience

1

Simulator Sickness Questionnaire

Instructions. Please report the degree to which you experience each of the below symptoms as one of “None”, “Slight”, “Moderate” and “Severe”. Using the scale from "0" (none) to "3" (severe).

Symptom	None	Slight	Moderate	Severe
General discomfort	0	1	2	3
Fatigue	0	1	2	3
Headache	0	1	2	3
Eyestrain	0	1	2	3
Difficulty focusing	0	1	2	3
Increased salivation	0	1	2	3
Sweating	0	1	2	3
Nausea	0	1	2	3
Difficulty concentrating	0	1	2	3
Fullness of head	0	1	2	3
Blurred vision	0	1	2	3
Dizzy (eyes open)	0	1	2	3
Dizzy (eyes closed)	0	1	2	3
Vertigo	0	1	2	3
Stomach awareness	0	1	2	3
Burping	0	1	2	3

Presence Questionnaire

Instructions. Please read the questions carefully and circle the appropriate number from 1 to 7.

1. How completely were all of your senses engaged?

1	2	3	4	5	6	7
Not engaged			Moderately			Very engaged

2. How much did the visual aspects of the environment involve you?

1	2	3	4	5	6	7
Not much			Moderately			Very much

3. How aware were you of events occurring in the real world around you?

1	2	3	4	5	6	7
Not aware			Moderately			Very aware

4. How inconsistent or disconnected was the information coming from your various senses?

1	2	3	4	5	6	7
Not inconsistent			Moderately			Very inconsistent

5. How compelling was your sense of moving around inside the virtual environment?

1	2	3	4	5	6	7
Not compelling			Moderately			Very compelling

6. To what degree did you feel confused or disoriented at the beginning of breaks or at the end of the experimental session?

1	2	3	4	5	6	7
Not disoriented			Moderately			Very disoriented

7. How involved were you in the virtual environment experience?

1	2	3	4	5	6	7
Not involved			Moderately			Very involved

8. How quickly did you adjust to the virtual environment experience?

1	2	3	4	5	6	7
Not quickly			Moderately			Very quickly

9. How much did the visual display quality interfere or distract you from the training experience?

1	2	3	4	5	6	7
Not much			Moderately			Very much

10. Were you involved in the experimental task to the extent that you lost track of time?

1	2	3	4	5	6	7
Not much			Moderately			Very much

Perceived Affective Quality

Instructions. Below is a list of words that can be used to describe places. We would like you to rate *how accurately* each word below describes the Virtual World you just visited. Use the following 1-8 rating scale for your answer. Please be sure that you have given an answer for *each word*.

1 = extremely inaccurate	5 = slightly accurate
2 = very inaccurate	6 = quite accurate
3 = quite inaccurate	7 = very accurate
4 = slightly inaccurate	8 = extremely accurate

intense

inactive

pleasant

dissatisfying

arousing

drowsy

nice

displeasing

active

idle

pleasing

repulsive

alive

lazy

pretty

unpleasant

forceful

slow

beautiful

uncomfortable

Program Ratings

Please answer all questions by circling corresponding answer.

1. How interesting was it to learn about vehicle fires in Village Firefighting today?

1	2	3	4	5
Boring				Interesting

2. How entertaining was it to learn about vehicle fires in Village Firefighting today?

1	2	3	4	5
Tiresome				Entertaining

3. How eager would you be to learn about a different Village Firefighting topic in the same conditions you learned today?

1	2	3	4	5
Not eager				Very Eager

4. How motivating was it to learn about vehicle fires in Village Firefighting today?

1	2	3	4	5
Not motivating				Motivating

5. How much did the VR learning system help you understand about vehicle fires in Village Firefighting?

1	2	3	4	5
Not at all				Very much

6. How helpful was the VR simulation for learning about vehicle fires in Village Firefighting?

1	2	3	4	5
Not helpful				Very helpful

7. How difficult was it to learn about vehicle fires in Village Firefighting with the VR learning system?

1	2	3	4	5
Not difficult				Very difficult

8. How much effort did you have to invest to learn about vehicle fires in Village Firefighting with the VR learning system?

1	2	3	4	5
No efforts				Lot of efforts

Would you like to give further comments?

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Part C. Posttest questionnaires

1

Posttest: Village Firefighting 3.9

1 Retention Questionnaires

Please answer all nine questions. Please circle the correct answer. This is a practice test and not part of course grades.

1. How should firefighters approach a vehicle accident?

    a) Not facing the direction of the wind.
    b) From downhill side (when on a slope) and not facing the direction of the wind.
    c) 45 degrees angle in front of the casualty's vehicle and not facing the direction of the wind.
    d) From uphill (when on a slope) and not facing the direction of the wind.
    e) Within a 5m radius of the accident and the nearest water hydrant.
    f) From uphill side (when on a slope) when the firetruck is downhill.

2. When do you use CABA?

    a) When there is a danger to the respiratory system.
    b) Only when the Hazardous Materials are fully detected and identified.
    c) Only during rescue missions in a village where there is a danger to the respiratory system.
    d) When O_2 concentration drops less than 90% and a hot environment within a rescue space of 20m x 20m x 20m.
    e) Only when you are told to do so during the briefing by your leader, due to the high-cost maintenance.
    f) Only within a closed 20m x 20m x 20m rescue space.

3. What is "Village Firefighting"?

    a) Firefighting in rural areas, at least 20km from the nearest town.
    b) Firefighting small structural fires in rural areas, at least 10km from the nearest town.
    c) Protecting villages from bushfire and vehicle accident.
    d) Firefighting within a small area such as structural fires and vehicle fires.
    e) Firefighting within a small area of 40km radius of nearest town.
    f) Firefighting within an open space such as structural fires and vehicle fires.

4. What should be done to reduce the risk of vehicle roll or tip in an accident?

    a) Switch off the engine when safe, and apply the handbrake.
    b) Remove the driver immediately, switch off the engine when safe, apply handbrake and puncture the wheels to add surface friction with the road.
    c) Switch off the engine, remove the keys from the ignition, apply the handbrake, and chock the wheels
    d) Remove the driver immediately, switch off the engine when safe, apply handbrake, chock the wheels and place the firetruck at 45 degrees angle right in front of the vehicle.
    e) Place the firetruck right in front of the vehicle at 45 degrees angle.
    f) Remove the driver immediately, switch off the engine when safe, apply handbrake, puncture the wheels and place the firetruck at 45 degrees angle right in front of the vehicle.

5. What does CABA stand for?

    a) Compacted Air Breathing Appliance.
    b) Compressed Air Breathing Aid.
    c) Compressed Air Breathing Apparatus.
    d) Compressed Air Breathing Appliance.
    e) Compacted Air Breathing Assistance.
    f) Compacted Air Breathing Apparatus.

6. What may contribute to a vehicle's explosions?

    a) The engine block.
    b) Inflatable bumpers, tyres, airshocks, driveshafts and vehicle's fuel tank.
    c) Inflatable bumpers, tyres and airshocks.

d) The engine block and the vehicle's fuel tank.

e) The vehicle's fuel tank and driveshafts.

f) The engine block, vehicle's fuel tank, inflatable bumpers, tyres and airshocks.

7. If a vehicle accident involves fallen power lines...

   a) If the voltage is greater than 150kV, stay clear until an authorised staff has confirmed that electrical power has been shut off.

   b) Stay clear until an authorised staff has confirmed that electrical power has been shut off.

   c) If the voltage is less than 50kV, cut the powerlines off immediately.

   d) Remove the driver immediately at all cost to prevent further injuries.

   e) Remove the driver immediately to prevent further burn injury when the power line configuration is less dangerous (e.g. monopole and bipolar).

   f) Stay clear until an authorised staff has confirmed that electrical power has been shut off when the power line configuration is tripole.

8. What will happen when Viton "O" rings and other electrical components are heated in a fire?

   a) They will turn into sulfuric acid and contact with skin may lead to an immediate cardiac arrest.

   b) They will turn into sulfuric acid and contact with skin may lead to a severe injury even if it may not cause immediate sensation of burning or stinging.

   c) They will turn into sulfuric acid and contact with skin may lead to an immediate sensation of burning or stinging.

   d) They will turn into hydrofluoric acid and contact with skin may lead to a severe injury even if it may not cause immediate sensation of burning or stinging.

   e) They will turn into hydrofluoric acid and contact with skin may lead to an immediate sensation of burning or stinging.

   f) They will turn into hydrofluoric acid and contact with skin may lead to an immediate cardiac arrest.

9. What can be done to reduce the likehood of a fire starting from a spillage on the road?

   a) Remove the driver immediately, blanket with foam, prohibit smoking and other ignition sources.

   b) Spray the spillage with water and quickly disconnect the vehicle's battery.

c) Blanket the spillage with CO_2 and disconnect vehicle's battery.

d) Blanket the spillage with foam, prohibit smoking and other ignition sources, and disconnect vehicle's battery.

e) Remove the driver immediately, blanket with CO_2, prohibit smoking and other ignition sources, and disconnect vehicle's battery.

f) Blanket the spillage with foam, prohibit smoking and other ignition sources.

6

2 Transfer Questionnaires

Please answer all six questions. Please circle the correct answer. This is a practice test and not part of course grades.

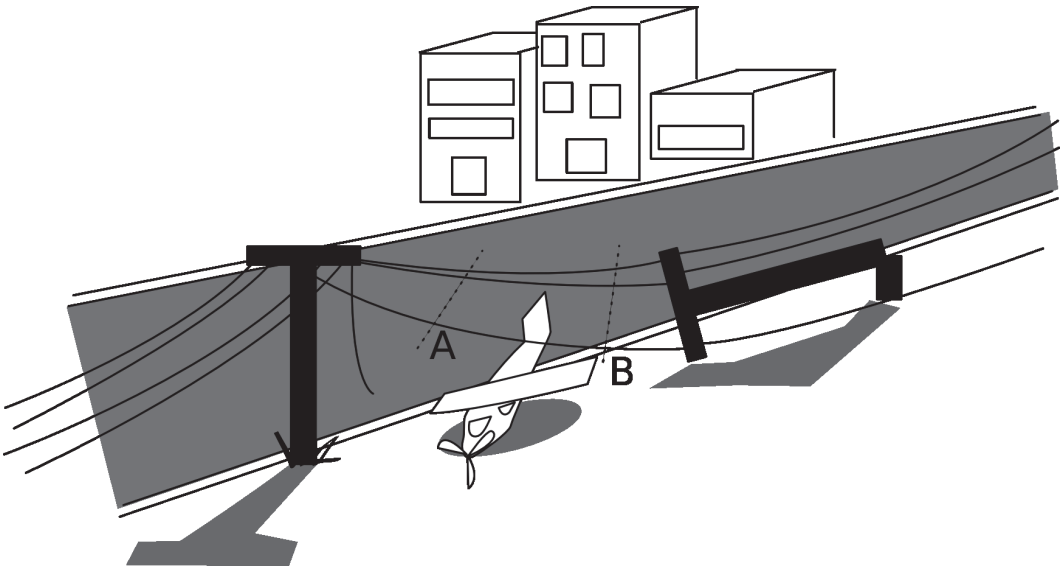


Figure 2.1: A small plane landed after an engine failure.

1. A small four seater plane landed in an emergency due to an engine failure shortly after taking off from a nearby rural airport. The situation is described in figure 2.1. The current condition of the pilot is unknown. What would be the best action upon your arrival at the scene?

   a) This question is beyond the scope of village firefighting.
   b) Approaching from location of dotted line B and remove the pilot immediately.
   c) Kerosene is highly flammable, focus the foam on the engine and the fuel tank.
   d) Confirming if the electrical supply has been cut off and cut the fallen lines along the dotted A and B lines.
   e) Confirming if the electrical supply in the area has been cut off.
   f) Chock the wheels and the blades to avoid vehicle roll or tip.

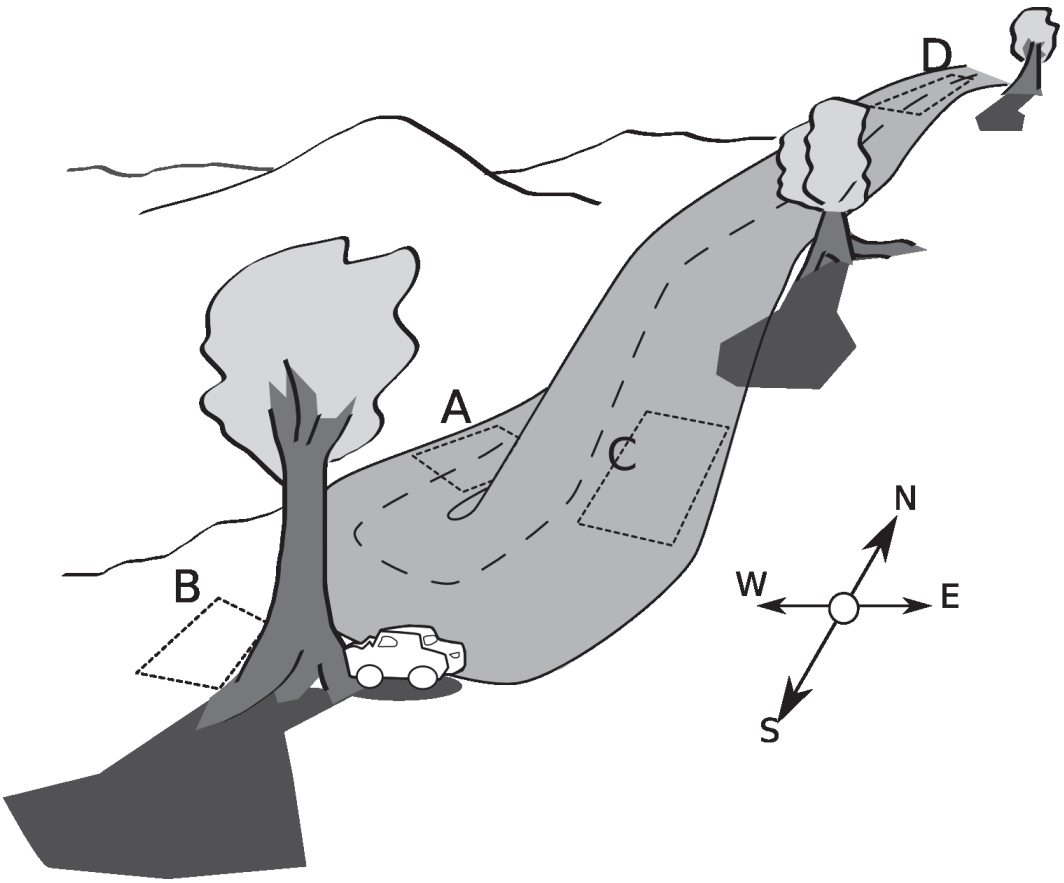


Figure 2.2: A vehicle accident on a hill.

2. A accident involving a small car is shown in figure 2.2. The driver appears as though he may have been speeding down the hill and wasn't quick enough for the sharp turn at the corner. A small amount of spillage is noticeable at the back of the car and on the road. The wind blows from the NE direction at 1 nautical mile per hour. Which one of the following is the best option to handle the accident to ensure safe and effective firefighting?

   a) Either approaching from C and A are fine with traffic control at location B.
   b) Approach from location D with traffic control at location A.
   c) Approach from location B with traffic control at location A.
   d) Approach from location A, B, C and D simultaneously with traffic control on these spots.
   e) Approach from location A with traffic control at location B.
   f) This is beyond the scope of village firefighting.

3. Referring to figure 2.2, which one of the answers below represents the most appropriate action to prevent vehicle roll or tip and ignition of fire.

    a) Switch off the engine, remove the keys from the ignition, apply the handbrake, chock the wheels, blanket any spillage with wet chemical, disconnect the battery, prohibit smoking and other ignition sources in the area.
    b) Switch off the engine, apply handbrake, blanket any spillage with foam, prohibit smoking and other ignition sources in the area.
    c) Switch off the engine, cut the vehicle battery and spray any spillage with wet chemical to reduce its combustion boiling point.
    d) Remove the driver immediately to reduce the mass of the vehicle and chock the wheels.
    e) Switch off the engine, remove the keys from the ignition, apply the handbrake, chock the wheels, blanket any spillage with foam, disconnect the battery, prohibit smoking and other ignition sources in the area.
    f) Any of (a), (b), (c) and (e) after removing the driver from the seat.

4. What are some of the possible hazards for firefighters at a motor vehicle accident

    a) Heart attack, trips, slips and falls, vehicle roll or tip, airbag deployment after the accident, electrical shock and suffocating.
    b) Heart attack, airbag deployment after the accident, electrical shock, oil spillage, and psychological impacts.
    c) Heart attack, fuel spillage, exposure to casualty bodily fluids, airbag deployment before the accident, explosions, and suffocating.
    d) Trips, slips and falls, vehicle roll or tip, and electrical shock.
    e) Heart attack, trips, slips and falls, exposure to hazardous materials, oil spillage, psychological impacts, and third degree burns.
    f) Heart attack, trips, slips and falls, exposure to casualty bodily fluids, airbag deployment after the accident, vehicle roll or tip, electrical shock.

5. You received a phone call about a burning storeroom 18m x 10m) of a warehouse that manufactures mobile phones. Which one of the following items would be most appropriate to ensure safe firefighting of the team?

    a) Compressed Air Breathing Assistance.
    b) Compressed Air Breathing Apparatus.
    c) This is beyond the scope of village firefighting.
    d) Viton-based gloves to avoid sulfuric acid and Compressed Air Breathing Appliance.
    e) Compressed Air Breathing Assistance and some non-acrylic based goggles.
    f) Compressed Air Breathing Apparatus and some non-acrylic based goggles.

6. Which of the incident below is most likely to be within the scope of Village Fire-fighting?

   a) A train stuck between two stations due to electrical failure.
   b) A small five-seater plane caught on fire in rural area.
   c) A small-scale a terrorist attack within 20m of closest major public transport.
   d) A small five-seater plane that made emergency landing in the sea due to fire on-board.
   e) A stranded 100-tonne ship in a remote gulf due to engine failure.
   f) All the above is beyond the scope of village firefighting.

Appendix F

Python codes of ANTECATALYST

This section contains all the codes that simulate the scenarios in the navigational behaviour of *Cataglyphis* ants in ANTECATALYST. **CataglyphisSimulationv463.py** file is the main file that runs the simulation. The other files manages other aspect of the simulation:

- **cameraNodes.py** manages the movement of the camera in the 3D world
- **desert.py** loads the 3D world: the land and the sky
- **narrations.py** manages the playback of the narration files throughout the simulation
- **actors.py** contains classes and methods to control the plain ants
- **actors2.py** contains classes and methods to control the cartoon ants
- **actors3.py** contains classes and methods to control the lifelike ants

Listing F.1: CataglyphisSimulationv463.py

```
1  import viz
2 import vizmenu
3 import viztask
4 import vizact
5 import vizmat
6
7 import cameraNodes as cn
8 import desert as d
9 import narrations as n
```

```
10
11 #import actors as a   #no animation conditions
12 import actors2 as a  #exaggeration animations
13 #import actors3 as a   #complex animations
14
15 STATE = 0 # Not sure why is it here... for now it has no purpose
16
17 viz.disable(viz.CULLING)
18 viz.disable(viz.CULL_FACE)
19
20 #Setting Up Camera for the scenes#####################################
21
22 #Grab the main view.
23 view = viz.MainView
24 view.setPosition(cn.S1_1)
25 #Setting up Blend mode
26 view.gotomode(viz.BLEND_ROTATE)
27
28 #Use spinto to rotate it back and forth. DEBUG ONLY##############
29 vizact.onkeydown( viz.KEY_LEFT, view.spinto , 0,1,0,-90,3,viz.TIME)
30 vizact.onkeydown( viz.KEY_RIGHT, view.spinto , 0,1,0,90,3,viz.TIME)
31
32 #List of scenes to be played during simulation#######################
33 def scene1():
34     '''Must be called from viz.director
35     '''
36
37     print 'executing scene 1-1'############
38     view.setPosition(cn.S1_1)
39     view.setEuler(-80,0,0)
40     yield viztask.waitTime(5.5)
41     view.spinto(0,1,0,-30,13,viz.TIME)
42     n.narration1_1.play()
43     yield viztask.waitTime(13)
44     view.pivot(cn.S1_la1)
45
46     #Putting Actors for scene1
47     a.actorsScene1Enable()
48
49     print 'executing scene 1-2'
50     view.pivot(cn.S1_la1)
51     view.goto(cn.S1_2, 12, viz.TIME)
52     n.narration1_2.play()
53
54     yield viztask.waitTime(12)
55
56     #executing actors movements
57     a.actorsScene1Action()
58     yield viztask.waitTime(3)
59
60     print 'executing scene 1-3'
61     n.narration1_3.play()
62     yield viztask.waitTime(9)
63     print 'Scene 1 ended...executing scene 2'
64
```

```
65      #Pausing before transition to next scene
66      yield viztask.waitTime(5)
67      scene2begin()
68
69  def scene2():
70      '''Must be called from viz.director
71      '''
72
73      #Initiating Scene 2
74      a.actorsScene2Enable()
75
76      print 'executing scene 2-1'
77      view.pivot(cn.S2_la1)
78      view.gotomode(viz.BLEND_ROTATE)
79      view.goto(cn.S2_t1,20,viz.TIME)
80      yield viztask.waitTime(20)
81
82      yield viztask.waitTime(2)
83
84      n.narration2_1.play()
85      d.myGrid.add(easeIn)
86
87      view.pivot(cn.S2_la1)
88      view.gotomode(viz.BLEND_ROTATE)
89      view.goto(cn.S2_t2,13,viz.TIME)
90      yield viztask.waitTime(14)
91
92      #Actions actors for scene 2
93      a.actorsScene2Action()
94
95      #yield viztask.waitTime(1)
96      n.narration2_2.play()
97
98
99      #Slowly move the camera to final foraging location##
100     view.gotomode(viz.PIVOT_ROTATE)
101     view.pivot(cn.S2_la1)
102     view.goto(cn.S2_1a, 16.5, viz.TIME) #was 24.5 #wqas 17
103     ######################################################
104
105     yield viztask.waitTime(19.0)
106
107     n.narration2_3.play()
108
109
110
111     #Slowly move the camera to final foraging location##
112     view.gotomode(viz.PIVOT_ROTATE)
113     view.pivot(cn.S2_la1)
114     view.goto(cn.S2_1b, 12, viz.TIME) #was 24.5
115     ######################################################
116     yield viztask.waitTime(12)
117
118     yield viztask.waitTime(3)
119
```

```
120     #Returning foraging actors to the nest
121     a.giveFoodScene2()
122     a.actorsScene2Return()
123 
124     yield viztask.waitTime(0.45)
125 
126     print 'executing scene 2-2a'
127     #Slowly move the camera to 2-2a camera location###
128     view.gotomode(viz.PIVOT_ROTATE)
129     view.pivot(cn.S2_la1)
130     view.goto(cn.S2_2a,14.65,viz.TIME) #was 18.75 #was 15.75 #was 15.5
131     ##################################################
132     yield viztask.waitTime(1)
133 
134 
135     print 'executing scene 2-2b'
136     n.narration2_4.play()
137     yield viztask.waitTime(15)
138 
139     a.removeFoodScene2()
140     #Slowly move the camera to 2-2b camera location#####
141     view.gotomode(viz.PIVOT_ROTATE)
142     view.pivot(cn.S2_la1)
143     view.goto(cn.S2_2b, 11, viz.TIME)
144     ##################################################
145 
146     yield viztask.waitTime(12)
147 
148     #a.actorsScene2Disable()
149     print 'Scene 2 is ended... executing scene 3'
150 
151     d.myGrid.add(easeOut)
152     #Pausing before transition to next scene
153     yield viztask.waitTime(5)
154     scene3begin()
155 
156 def scene3():
157     '''Must be called from viz.director
158     '''
159     print 'executing scene 3-1'
160     d.theSunSkyPatterns.setEuler([0,60,00],viz.ABS_GLOBAL)
161     view.gotomode(viz.BLEND_ROTATE)
162     view.pivot(cn.S3_la1)
163 
164     view.goto(cn.S3_1, 5, viz.TIME)
165     yield viztask.waitTime(5)
166     view.spinto(1,0,0,-16,4,viz.TIME) # was 28 #was 18
167 
168     yield viztask.waitTime(2)
169     n.narration3_1.play()
170 
171     fadeIn()
172     yield viztask.waitTime(16) #length of the audio
173     risingSun()
174 
```

```
175     print 'executing scene 3-2'
176     n.narration3_2.play()
177     yield viztask.waitTime(12) #length of the audio
178     view.pivot(cn.S4_la2)
179
180     #Pausing before transition to next scene
181     yield viztask.waitTime(5)
182
183     print 'Scene 3 is ended... executing scene 4'
184     scene4begin()
185
186 def scene4():
187     '''       This function MUST be called from viz.director
188           This function defines the control sequence
189           of scene 4
190     '''
191
192     #Enabling actors for scene 4
193     a.actorsScene4Enable()
194
195     print 'executing transtition to scene 4-1'
196     view.gotomode(viz.BLEND_ROTATE)
197     view.pivot(cn.S4_la1)
198     view.goto(cn.S4_t1,15  ,viz.TIME)
199     view.pivot(cn.S4_la1)
200     yield viztask.waitTime(15.5)
201
202     print 'executing scene 4-1'
203     view.pivot(cn.S4_la2)
204     view.goto(cn.S4_1,21  ,viz.TIME)
205     view.pivot(cn.S4_la2)
206     n.narration4_1.play()
207     yield viztask.waitTime(22)
208
209     print 'executing scene 4-2'
210     #a.controlAction()
211     a.stumpsAction()
212     yield viztask.waitTime(0.1)
213     a.stiltsAction()
214     yield viztask.waitTime(0.2)
215     #a.stumpsAction()
216     a.controlAction()
217
218     view.goto(cn.S4_3,33.55  ,viz.TIME) #was 32.5
219     view.pivot(cn.S4_la2)
220
221     yield viztask.waitTime(2)
222
223     n.narration4_2.play()
224     #yield viztask.waitTime(8) #was 23
225     view.pivot(cn.S4_la2)
226
227     yield viztask.waitTime(33)
228
229     #ants pick food items
```

```
230     a.giveFoodControl()
231     yield viztask.waitTime(0.5)
232     a.giveFoodStilts()
233     yield viztask.waitTime(0.5)
234     a.giveFoodStumps()
235 
236     yield viztask.waitTime(1)
237 
238 
239     print 'executing scene 4-3'
240     n.narration4_3.play()
241     #lift up ants 7, 8, 9
242     yield viztask.waitTime(1)
243     a.pickupAnts()
244     yield viztask.waitTime(0.5)
245 
246     view.goto(cn.S4_4,8  ,viz.TIME)
247     view.pivot(cn.S4_la2)
248     yield viztask.waitTime(8)
249 
250     view.pivot(cn.S4_la5)
251 
252     #Put ants over their individual tunnels
253     a.distributeAnts()
254     yield viztask.waitTime(5)
255 
256     #breaking here
257     #return
258     #breaking here
259 
260     #Scene 4-4#####################################
261     print 'executing scene 4-4'
262     view.goto(cn.S4_5a,5  ,viz.TIME)
263     view.pivot(cn.S4_la5)
264     yield viztask.waitTime(5)
265 
266     n.narration4_4.play()
267     view.pivot(cn.S4_la5)
268     view.goto(cn.S4_5b,9  ,viz.TIME)
269     #give food to control ant
270     #a.giveFoodControl()
271     yield viztask.waitTime(10)
272     view.pivot(cn.S4_la6)
273 
274     #Executing scene 4-5 (Stilts condition)################
275 
276     print 'executing scene 4-5'
277     view.goto(cn.S4_6a,5  ,viz.TIME)
278     view.pivot(cn.S4_la6)
279     yield viztask.waitTime(6)
280 
281     #Play narration
282     n.narration4_5.play()
283 
284     #applying stilts condition
```

```
285     a.applyStilts()
286
287     view.pivot(cn.S4_1a6)
288     view.goto(cn.S4_6b,9  ,viz.TIME)
289
290     yield viztask.waitTime(10)
291
292     #Give food
293     #a.giveFoodStilts()
294
295     #yield viztask.waitTime(8)
296     view.pivot(cn.S4_1a7)
297
298     #Executing scene 4-6 (Stumps condition)################
299     print 'executing scene 4-6'
300
301     view.goto(cn.S4_7a,5  ,viz.TIME)
302     view.pivot(cn.S4_1a7)
303     yield viztask.waitTime(6)
304     n.narration4_6.play()
305
306     #applying stilts condition
307     a.applyStumps()
308
309     view.pivot(cn.S4_1a7)
310     view.goto(cn.S4_7b,9  ,viz.TIME)
311
312     yield viztask.waitTime(12)
313
314
315
316     print 'executing scene 4-7'
317
318     view.pivot(cn.S4_1a6)
319     view.goto(cn.S4_8,6  ,viz.TIME)
320     view.pivot(cn.S4_1a6)
321
322     n.narration4_7.volume(14)
323     n.narration4_7.play()
324
325
326     yield viztask.waitTime(5)
327     view.pivot(cn.S4_1a6)
328
329
330     #breaking here
331     #return
332     #breaking here
333
334     #Lower down the ants##############
335     yield viztask.waitTime(2)
336     a.dropAnts()
337     yield viztask.waitTime(2)
338
339
```

```
340     #Let go ants towards fictive nest
341     a.controlReturn()
342     a.stiltsReturn()
343     a.stumpsReturn()
344     yield viztask.waitTime(0.15)
345     a.FINALSCENE = 1
346
347
348     yield viztask.waitTime(0.5)
349
350     print 'executing scene 4-8'################################
351
352 #   view.pivot([1.5, 0, -0.119])
353     view.pivot(cn.S4_la9)
354     view.goto(cn.S4_9a,37.65 ,viz.TIME) #was 36.55, was 37.25
355     #was view.goto([1.347, 1.263, -0.581],20 ,viz.TIME)
356
357     yield viztask.waitTime(36)
358     n.narration4_8.volume(14)
359     n.narration4_8.play()
360
361     #removing food items and the ants too
362     a.removeFoodControl()
363     a.removeFoodStilts()
364     a.removeFoodStumps()
365     a.actorsScene4Disable()
366
367     view.goto(cn.S4_9b,22 ,viz.TIME)
368     yield viztask.waitTime(25)
369
370
371     #Pausing before transition to next scene
372     yield viztask.waitTime(5)
373
374     print 'scene 4 is ended... starting scene 5'
375
376
377     scene5begin()
378
379
380 def scene5():
381     '''Must be called from viz.director
382     '''
383     ###Transtition scene 5_t1###################################
384     a.extraAntEnable()
385
386     view.gotomode(viz.BLEND_ROTATE)
387
388     view.pivot(cn.S5_la1)
389     view.goto(cn.S5_t1,22,viz.TIME)
390     yield viztask.waitTime(23) # time taken to reach camera destination
391     a.extraAntGo()
392     print 'executing scene 5-1'##################################
393     d.myGrid.add(easeIn)
394
```

```
395
396     view.pivot(cn.S5_la1)
397     view.goto(cn.S5_1, 40, viz.TIME)
398
399
400     n.narration5_1.play()
401
402     yield viztask.waitTime(17) # length of audio clip
403     n.narration5_2.play()
404
405     yield viztask.waitTime(18) # length of audio clip
406
407     yield viztask.waitTime(5)
408
409     #closing remarks#########################################
410     view.gotomode(viz.BLEND_ROTATE)
411     view.pivot(cn.S5_la2)
412     view.goto(cn.S5_2, 10, viz.TIME)
413     yield viztask.waitTime(3)
414     d.myGrid.add(easeOut)
415     fadeOut()
416     yield viztask.waitTime(3)
417     n.narration5_3.play()
418     yield viztask.waitTime(11)
419     print 'scene 5 is ended'
420
421 #Initiate scenes from viz.Director, or it will not work properly####
422
423 def scene1begin():
424     viztask.schedule(scene1())
425
426 def scene2begin():
427     viztask.schedule(scene2())
428
429 def scene3begin():
430     viztask.schedule(scene3())
431
432 def scene4begin():
433     viztask.schedule(scene4())
434
435 def scene5begin():
436     viztask.schedule(scene5())
437
438
439
440 #Setting up Scene Selection Menu#######################################
441 menu = vizmenu.add()
442 #Align the menu in the enter of the top of the screen.
443 menu.setAlignment( vizmenu.CENTER )
444 #Add two pop-downs menus within the main menu.
445 SceneMenu = menu.add( 'Scene Selection Menu' )
446
447 #Add sub-menu under the Ball1 menu and put radio buttons in it.
448 Scene1Button = SceneMenu.add( viz.BUTTON, 'Scene 1 - Introduction' )
```

```
449 Scene2Button = SceneMenu.add( viz.BUTTON, 'Scene 2 - Navigation on
    North African Salt Pan' )
450 Scene3Button = SceneMenu.add( viz.BUTTON, 'Scene 3 - Compass System' )
451 Scene4Button = SceneMenu.add( viz.BUTTON, 'Scene 4 - Estimating
    Distance' )
452 Scene5Button = SceneMenu.add( viz.BUTTON, 'Scene 5 - Summary' )
453
454 vizact.onbuttondown( Scene1Button, scene1begin )
455 vizact.onbuttondown( Scene2Button, scene2begin )
456 vizact.onbuttondown( Scene3Button, scene3begin )
457 vizact.onbuttondown( Scene4Button, scene4begin )
458 vizact.onbuttondown( Scene5Button, scene5begin )
459
460 # This section deals with general function that control the simulation
461 easeOut = vizact.fade(1,0,12)
462 easeIn = vizact.fade(0,1,12)
463 riseUp = vizact.spinto(1,0,0,28,30,viz.TIME) # was 20 not 28
464
465 def risingSun():
466     d.theSunSkyPatterns.setEuler([0,60,00],viz.ABS_GLOBAL)
467     d.theSunSkyPatterns.add(riseUp)
468
469 def fadeOut():
470     d.theSkyPatterns.add(easeOut)
471
472 def fadeIn():
473     d.theSkyPatterns.add(easeIn)
474
475 #########FOR DEBUGGING ONLY
    ##########################################################
476 def polarizationDisplay(key):
477     '''This function handles the fade in and fade out of the
        polarization patterns
478     '''
479     if key == 'q':
480         risingSun()
481
482     if key == 'w':
483         fadeIn()
484
485     if key == 'e':
486         fadeOut()
487 #########FOR DEBUGGING ONLY
    ##########################################################
488
489 viz.starttimer(0,0.5,-1)
490 viz.callback(viz.KEYBOARD_EVENT,polarizationDisplay)
491
492 #viz.window.setSize(1024,768)
493 #viz.window.setBorder(viz.BORDER_NONE)
494
495 if __name__ == "__main__":
496     viz.go()
497     viz.window.setSize(640,480)
498     #viz.window.setSize(1024,768)
```

```
499     viz.window.setBorder(viz.BORDER_NONE)
```

Listing F.2: cameraNodes.py

```
1  # This file only contains camera nodes and 'lookat' locations
2
3  import viz
4
5  #cn = viz.add('cameraNodes.osg')
6
7  #Camera locations for scene 1
8  S1_1 = [-8.3, 2, 6.152]
9  S1_2 = [-8.3, 0.055, 6.15]
10 S1_la1 = [-8.426, 0.002, 6.2]
11
12
13 #Camera locations for scene 2
14 S2_t1 = [0.618, 1.5, 3.5]
15 S2_t2 = [0.708, 0.03, 7.348]
16
17
18 S2_1a = [3.552, 0.05, 10.585]
19 S2_1b = [2.811, 0.12, 11.404]
20
21 S2_2a = [0.626, 0.01, 7.302]
22 #was S2_2a = [0.678, 0.01, 7.348]
23 S2_2b = [0.618, 0.9550, 5.5]
24 #was S2_2b = [0.618, 1.050, 5.5]
25 S2_la1 = [0.550, 0.002, 7.3]
26
27 #Camera locations for scene 3
28 S3_1 = [0.618, 3, 1.905]
29 S3_2 = S3_1
30 S3_la1 = [0.074, 8, 22.78]
31
32 #Camera locations for scene 4
33 S4_t1 = [-2,2, -0.855]
34
35 S4_1 = [-0.693, 0.03, 0.3]
36
37 S4_3 = [9.320, 0.05, 0.299]
38 S4_4 = [9.821, 0.6, -0.111]
39
40 S4_5a = [9.510, 0.214, 0.013]
41 S4_5b = [9.510, 0.214, -0.013]
42 S4_6a = [9.510, 0.214, -0.178]
43 S4_6b = [9.510, 0.214, -0.213]
44 S4_7a = [9.510, 0.214, -0.387]
45 S4_7b = [9.482, 0.214, -0.431]
46 S4_8 = [9.7, 0.5, -0.2]
47
48 S4_9a = [-0.6, 1.65, -0.591]
49 #was S4_9a = [-0.6, 1.45, -0.591]
50 S4_9b = [-0.6, 5.25, -0.601] #middle value was 1.8 #was 4.65
51 #was S4_9b = [-0.6, 2.85, -0.601] #middle value was 1.8
```

```
52
53 S4_la5 = [9.453, 0.202, 0]
54 S4_la6 = [9.453, 0.202, -0.2]
55 S4_la7 = [9.453, 0.202, -0.4]
56 S4_la9 = [-0.6, 0.01, -0.200]
57
58 S4_la1 = [-0.6,0,0.5]
59 S4_la2 = [9.450, 0.010,0.301]
60
61 #Camera locations for scene 5
62 S5_t1 = [0.479, 0.1, 7.071]
63
64 S5_1 = [-0.153, 1.5, 3.202]
65 S5_2 = S5_1
66 S5_la1 = S2_la1
67 S5_la2 = [-36.926, 15.5, 35.78]
```

Listing F.3: actors.py

```
1  import viz
2  import vizmat
3
4  n = 0
5  viz.startlayer(viz.LINES)
6
7  p1 = ((-7.25), 0.001, (18.161))
8  p2 = ((-7.25), 0.001, (3.161))
9  p3 = ((7.75), 0.001, (18.161))
10
11 while (n <= 15):
12     viz.vertexcolor([0.28,0.40,0.50]) #[0.28,0.40,0.60]
13     viz.vertex(p1[0]+n,0.01,p1[2])
14     viz.vertex(p2[0]+n,0.01,p2[2])
15
16     viz.linewidth(2)
17     viz.vertexcolor([0.28,0.40,0.50]) #[0.28,0.40,0.60]
18     viz.vertex(p1[0],0.01,p1[2]-n)
19     viz.vertex(p3[0],0.01,p3[2]-n)
20
21     viz.linewidth(2)
22     n = n+1
23
24 myGrid= viz.endlayer()
25 myGrid.alpha(0)
26
27
28 def makeGrid():
29     n = 0
30     viz.startlayer(viz.LINES)
31
32     p1 = ((-7.25), 0, (18.161))
33     p2 = ((-7.25), 0, (3.161))
34     p3 = ((7.75), 0, (18.161))
35
36 #   p1 = ((-7.813+0.563), 0, (7.813+10.348))
```

```
#   p2 = ((-7.813+0.563), 0, (-7.813+10.348))
#   p3 = ((7.813+0.563), 0, (7.813+10.348))

    while (n <= 15):
        viz.vertex(p1[0]+n,0,p1[2])
        viz.vertex(p2[0]+n,0,p2[2])
        viz.linewidth(2)
        viz.vertexcolor([0,1,0]) #was 0 1 0
        viz.vertex(p1[0],0,p1[2]-n)
        viz.vertex(p3[0],0,p3[2]-n)
        viz.linewidth(2)
        viz.vertexcolor([0,1,0]) #was 0 1 0
        n = n+1

        segment= viz.endlayer()
        segment.parent(myGrid)



#setting up the skybox
env = viz.add(viz.ENVIRONMENT_MAP,'sky\deepBlue.bmp')
sky = viz.add('skydome.dlc')
sky.texture(env)

#Loading the tunnels
theTunnels = viz.add('theTunnels.osg')
theTunnels.disable(viz.LIGHTING)

#Loading the sun and polarization pattern
theSunSkyPatterns = viz.add('theSunSkyPatterns.osg')
theSunSkyPatterns.disable(viz.LIGHTING)
theSunSkyPatterns.setEuler([0,60,00],viz.ABS_GLOBAL)
#Getting the child node to control the skylight pattern individually.
theSkyPatterns = theSunSkyPatterns.getChild('skyPattern')

theSkyPatterns.alpha(0) #during start up, the polarization pattern must
    be invisible
theSun = theSunSkyPatterns.getChild('sun') #this object (the sun) must
   be visible all the times
theSun.alpha(1)
#For Debugging Purpose Only
print theSkyPatterns.getEuler()

#setting up light
myLight = viz.addLight()
myLight.setEuler(theSun.getEuler())

#Loading the 3D desert model
desert1 = viz.add('brownDesert.osg')
desert2 = viz.add('brownDesertShadow.osg')

#Add main texture
tex1 = viz.add('brownSand.jpg')
tex2 = viz.add('brownSandShadow.jpg')

```

```
90 #Add detail texture in repeat mode
91 texDetail = viz.add('detail_sandsBW.jpg',wrap=viz.REPEAT)
92
93 #Apply textures to quad
94 desert1.texture(tex1)
95 desert1.texture(texDetail,unit=1)
96
97 desert2.texture(tex2)
98 desert2.texture(texDetail,unit=1)
99
100 ##Repeat detail texture 20 times
101 desert1.texmat(viz.Transform.scale(8,8,1),unit=1) #was 200 with v1
    texture
102 desert2.texmat(viz.Transform.scale(8,8,1),unit=1) #was 200 with v1
    texture
103
104 ##Apply detail shader to quad
105 desert1.apply( viz.addShader( frag = 'detail.frag' ) )
106 desert2.apply( viz.addShader( frag = 'detail.frag' ) )
107
108 ##Apply shader uniforms
109 desert1.apply( viz.addUniformInt( 'BaseTexture', 0 ) )
110 desert1.apply( viz.addUniformInt( 'DetailTexture', 1 ) )
111
112 desert2.apply( viz.addUniformInt( 'BaseTexture', 0 ) )
113 desert2.apply( viz.addUniformInt( 'DetailTexture', 1 ) )
114
115 #viz.go()
```

Listing F.4: narrations.py

```
1  import viz
2
3  narration1_1 = viz.add('narrations/narration1-1.mp3')
4  narration1_2 = viz.add('narrations/narration1-2.mp3')
5  narration1_3 = viz.add('narrations/narration1-3.mp3')
6  #
7  narration2_1 = viz.add('narrations/narration2-1.mp3')
8  narration2_2 = viz.add('narrations/narration2-2.mp3')
9  narration2_3 = viz.add('narrations/narration2-3.mp3')
10 narration2_4 = viz.add('narrations/narration2-4.mp3')
11 #
12 narration3_1 = viz.add('narrations/narration3-1.mp3')
13 narration3_2 = viz.add('narrations/narration3-2.mp3')
14 #
15 narration4_1 = viz.add('narrations/narration4-1.mp3')
16 narration4_2 = viz.add('narrations/narration4-2.mp3')
17 narration4_3 = viz.add('narrations/narration4-3.mp3')
18 narration4_4 = viz.add('narrations/narration4-4.mp3')
19 narration4_5 = viz.add('narrations/narration4-5.mp3')
20 narration4_6 = viz.add('narrations/narration4-6.mp3')
21 narration4_7 = viz.add('narrations/narration4-7.mp3')
22 narration4_8 = viz.add('narrations/narration4-8.mp3')
23
24
```

```
25 #
26 narration5_1 = viz.add('narrations/narration5 -1.mp3')
27 narration5_2 = viz.add('narrations/narration5 -2.mp3')
28 narration5_3 = viz.add('narrations/narration5 -3.mp3')
```

Listing F.5: actors.py

```
1  ''' John Porte developed the base scripts for loading and reading
2      multiple objects, retreiving their object centres for animation
3      purpose
4  '''
5
6  import viz
7  import vizact
8  import vizmat
9
10 FINALSCENE = 0
11
12 #SCALE OF THE ANT##################################################
13 SCALE = (0.015,0.015,0.015)
14 ANIMATIONSPEED = 1
15 OFFSET1 = 0.004; OFFSET2 = 0.0065; STUMPS_OFFSET = 0.001; STILTS_OFFSET
       = 0.001
16
17 #Loading .OSG files that contains movement points of the ants####
18 movementPaths = viz.add('antMovementPoints.osg')
19 movementPaths.disable(viz.RENDERING)
20
21 path_actor1_000 = movementPaths.getChild('path.actor1.000')
22 path_actor1_000_box = path_actor1_000.getBoundingBox()
23 ant1StartPos = path_actor1_000_box.getCenter()
24
25 path_actor2_000 = movementPaths.getChild('path.actor2.000')
26 path_actor2_000_box = path_actor2_000.getBoundingBox()
27 ant2StartPos = path_actor2_000_box.getCenter()
28
29 path_actor3_000 = movementPaths.getChild('path.actor3.000')
30 path_actor3_000_box = path_actor3_000.getBoundingBox()
31 ant3StartPos = path_actor3_000_box.getCenter()
32
33 path_actor4_000 = movementPaths.getChild('path.actor4.000')
34 path_actor4_000_box = path_actor4_000.getBoundingBox()
35 ant4StartPos = path_actor4_000_box.getCenter()
36
37 path_actor5_000 = movementPaths.getChild('path.actor5.000')
38 path_actor5_000_box = path_actor5_000.getBoundingBox()
39 ant5StartPos = path_actor5_000_box.getCenter()
40
41 path_actor6_000 = movementPaths.getChild('general.path.000')
42 path_actor6_000_box = path_actor6_000.getBoundingBox()
43 ant6StartPos = path_actor6_000_box.getCenter()
44
45 path_actor7_000 = movementPaths.getChild('control.path.000')
46 path_actor7_000_box = path_actor7_000.getBoundingBox()
47 ant7StartPos = path_actor7_000_box.getCenter()
```

```
48
49 path_actor8_000 = movementPaths.getChild('stilts.path.000')
50 path_actor8_000_box = path_actor8_000.getBoundingBox()
51 ant8StartPos = path_actor8_000_box.getCenter()
52
53 path_actor9_000 = movementPaths.getChild('stumps.path.000')
54 path_actor9_000_box = path_actor9_000.getBoundingBox()
55 ant9StartPos = path_actor9_000_box.getCenter()
56
57
58 #Pre-loading the ant##############################################
59 ant = viz.add('antique/cal3dantique.cfg')
60 ant.visible(0)
61 ################################################################
62
63 # Ant 1, 2, 3, 4 and 5 are for scene 1
64 ant1 = ant.copy()
65 ant1.runAction(vizact.fadeTo(0, time = 0, node = 'stilts.cmf')) #Hiding
       the stilts
66 ant1.setScale(SCALE)
67
68 normal = vizact.fadeTo(0, time = 0, node = 'stilts.cmf') #Hiding the
     stilts
69 stumps = vizact.fadeTo(0, time = 7, node = 'antique_normal.cmf')
70 stilts = vizact.fadeTo(1, time = 7, node = 'stilts.cmf')
71
72
73 ant1.idlepose(2)
74 ant1.disable(viz.RENDERING)
75 ant1.setPosition(ant1StartPos[0], -ant1StartPos[2]+OFFSET1,
     ant1StartPos[1])
76 ant1.setAnimationSpeed(0,ANIMATIONSPEED)
77
78 ant2 = ant.copy()
79 ant2.runAction(normal)
80 ant2.setScale(SCALE)
81 ant2.idlepose(2)
82 ant2.disable(viz.RENDERING)
83 ant2.setPosition(ant2StartPos[0], -ant2StartPos[2]+OFFSET1,
     ant2StartPos[1])
84 ant2.setAnimationSpeed(0,ANIMATIONSPEED)
85
86 ant3 = ant.copy()
87 ant3.runAction(normal)
88 ant3.setScale(SCALE)
89 ant3.idlepose(2)
90 ant3.disable(viz.RENDERING)
91 ant3.setPosition(ant3StartPos[0], -ant3StartPos[2]+OFFSET1,
     ant3StartPos[1])
92 ant3.setAnimationSpeed(0,ANIMATIONSPEED)
93
94 ant4 = ant.copy()
95 ant4.runAction(normal)
96 ant4.setScale(SCALE)
97 ant4.idlepose(2)
```

```
98  ant4.disable(viz.RENDERING)
99  ant4.setPosition(ant4StartPos[0], -ant4StartPos[2]+OFFSET1,
        ant4StartPos[1])
100 ant4.setAnimationSpeed(0,ANIMATIONSPEED)
101
102 ant5 = ant.copy()
103 ant5.runAction(normal)
104 ant5.setScale(SCALE)
105 ant5.idlepose(2)
106 ant5.disable(viz.RENDERING)
107 ant5.setPosition(ant5StartPos[0], -ant5StartPos[2]+OFFSET1,
        ant5StartPos[1])
108 ant5.setAnimationSpeed(0,ANIMATIONSPEED)
109
110 #Ant 6 for scene
        2#####################################################################
111 ant6 = ant.copy()
112 ant6.runAction(normal)
113 ant6.setScale(SCALE)
114 ant6.idlepose(2)
115 ant6.disable(viz.RENDERING)
116 ant6.setPosition(ant6StartPos[0], -ant6StartPos[2]+OFFSET1,
        ant6StartPos[1])
117 ant6.setAnimationSpeed(0,ANIMATIONSPEED)
118
119 #Ant 7(normal), 8 (stilts)and 9(stumps) are for scene
        2##############################
120 ant7 = ant.copy()
121 ant7.runAction(normal)
122 ant7.setScale(SCALE)
123 ant7.idlepose(2)
124 ant7.disable(viz.RENDERING)
125 ant7.setPosition(ant7StartPos[0], -ant7StartPos[2]+OFFSET1,
        ant7StartPos[1])
126 ant7.setEuler([65,0,0])
127 ant7.setAnimationSpeed(0,ANIMATIONSPEED)
128
129 ant8 = ant.copy()
130 ant8.runAction(normal)
131 ant8.setScale(SCALE)
132 ant8.idlepose(2)
133 ant8.disable(viz.RENDERING)
134 ant8.setPosition(ant8StartPos[0], -ant8StartPos[2]+OFFSET1,
        ant8StartPos[1])
135 ant8.setEuler([90,0,0])
136 ant8.setAnimationSpeed(0,ANIMATIONSPEED)
137
138 ant9 = ant.copy()
139 ant9.setScale(SCALE)
140 ant9.runAction(normal)
141 ant9.idlepose(2)
142 ant9.disable(viz.RENDERING)
143 ant9.setPosition(ant9StartPos[0], -ant9StartPos[2]+OFFSET1,
        ant9StartPos[1])
144 ant9.setEuler([100,0,0])
```

```
145 ant9.setAnimationSpeed(0,ANIMATIONSPEED)
146
147
148 ##########################################
149 #Extra ants scene 5
150
151 ant10 = ant.copy()
152 ant10.setScale(SCALE)
153 ant10.runAction(normal)
154 ant10.idlepose(2)
155 ant10.disable(viz.RENDERING)
156 ant10.setPosition(3.985, OFFSET1, 8.031)
157 ant10.setEuler([0,0,0])
158 ant10.setAnimationSpeed(0,ANIMATIONSPEED)
159
160 ##########################################
161
162 #Setting up movement path actor 1 scene 1
163 CP_ACTOR_1 = []
164
165 CP_COUNT_ACTOR_1 = 23
166 CONTROLPOINT_ACTOR_1_PREFEX = 'path.actor1.' #name of boxes = layera1
        ... layera17
167 CONTROLPOINT_ACTOR_1_SUFFEX = '' #use '_Cube' for .obj and '' for.osg
168
169 for cpNumber in range( 0,CP_COUNT_ACTOR_1):
170     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_1_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_1_SUFFEX)
171     pointNode = movementPaths.getChild(nodename)
172     nodebox = pointNode.getBoundingBox()
173     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]] #
            this is the xyz setting ONLY for .OSG file
174     print 'path.actor.1',cpNumber, cntr
175     CP_ACTOR_1.append(cntr)
176
177
178 #Setting up movement path actor 2 scene 1
179 CP_ACTOR_2 = []
180
181 CP_COUNT_ACTOR_2 = 19
182 CONTROLPOINT_ACTOR_2_PREFEX = 'path.actor2.' #name of boxes = layera1
        ... layera17
183 CONTROLPOINT_ACTOR_2_SUFFEX = '' #use '_Cube' for .obj and '' for.osg
184
185 for cpNumber in range( 0, CP_COUNT_ACTOR_2):
186     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_2_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_2_SUFFEX)
187     pointNode = movementPaths.getChild(nodename)
188     nodebox = pointNode.getBoundingBox()
189     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]] #
            this is the xyz setting ONLY for .OSG file
190     print 'path.actor.2',cpNumber, cntr
191     CP_ACTOR_2.append(cntr)
192
193 #Setting up movement path actor 3 scene 1
```

```
194 CP_ACTOR_3 = []
195
196 CP_COUNT_ACTOR_3 = 18
197 CONTROLPOINT_ACTOR_3_PREFEX = 'path.actor3.'  #name of boxes = layera1
        ... layera17
198 CONTROLPOINT_ACTOR_3_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
199
200 for cpNumber in range( 0,CP_COUNT_ACTOR_3):
201     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_3_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_3_SUFFEX)
202     pointNode = movementPaths.getChild(nodename)
203     nodebox = pointNode.getBoundingBox()
204     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
205     print 'path.actor.3',cpNumber, cntr
206     CP_ACTOR_3.append(cntr)
207
208
209 #Setting up movement path actor 4 scene 1
210 CP_ACTOR_4 = []
211
212 CP_COUNT_ACTOR_4 = 11
213 CONTROLPOINT_ACTOR_4_PREFEX = 'path.actor4.'  #name of boxes = layera1
        ... layera17
214 CONTROLPOINT_ACTOR_4_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
215
216 for cpNumber in range( 0,CP_COUNT_ACTOR_4):
217     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_4_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_4_SUFFEX)
218     pointNode = movementPaths.getChild(nodename)
219     nodebox = pointNode.getBoundingBox()
220     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
221     print 'path.actor.4',cpNumber, cntr
222     CP_ACTOR_4.append(cntr)
223
224 #Setting up movement path actor 5 scene 1
225 CP_ACTOR_5 = []
226
227 CP_COUNT_ACTOR_5 = 16
228 CONTROLPOINT_ACTOR_5_PREFEX = 'path.actor5.'  #name of boxes = layera1
        ... layera17
229 CONTROLPOINT_ACTOR_5_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
230
231 for cpNumber in range( 0,CP_COUNT_ACTOR_5):
232     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_5_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_5_SUFFEX)
233     pointNode = movementPaths.getChild(nodename)
234     nodebox = pointNode.getBoundingBox()
235     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
236     print 'path.actor.5',cpNumber, cntr
237     CP_ACTOR_5.append(cntr)
238
239 #Setting up foraging path actor in scene 2
```

```
240 CP_ACTOR_GENERAL = []
241
242 CP_COUNT_ACTOR_GENERAL = 22
243 CONTROLPOINT_ACTOR_GENERAL_PREFEX = 'general.path.'  #name of boxes =
        layera1 ... layera17
244 CONTROLPOINT_ACTOR_GENERAL_SUFFEX = ''  #use '_Cube' for .obj and ''
        for.osg
245
246 for cpNumber in range( 0,CP_COUNT_ACTOR_GENERAL):
247     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_GENERAL_PREFEX, cpNumber
            ,CONTROLPOINT_ACTOR_GENERAL_SUFFEX)
248     pointNode = movementPaths.getChild(nodename)
249     nodebox = pointNode.getBoundingBox()
250     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
251     print 'general.path.',cpNumber, cntr
252     CP_ACTOR_GENERAL.append(cntr)
253
254 #Setting up homebound path actor in scene 2
255 CP_ACTOR_RETURN = []
256
257 CP_COUNT_ACTOR_RETURN = 6
258 CONTROLPOINT_ACTOR_RETURN_PREFEX = 'return.path.'  #name of boxes =
        layera1 ... layera17
259 CONTROLPOINT_ACTOR_RETURN_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
260
261 for cpNumber in range( 0,CP_COUNT_ACTOR_RETURN):
262     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_RETURN_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_RETURN_SUFFEX)
263     pointNode = movementPaths.getChild(nodename)
264     nodebox = pointNode.getBoundingBox()
265     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
266     print 'return.path.',cpNumber, cntr
267     CP_ACTOR_RETURN.append(cntr)
268
269 #Setting up foraging path of normal condition in scene 4
270 CP_ACTOR_CONTROL = []
271
272 CP_COUNT_ACTOR_CONTROL = 13
273 CONTROLPOINT_ACTOR_CONTROL_PREFEX = 'control.path.'  #name of boxes =
        layera1 ... layera17
274 CONTROLPOINT_ACTOR_CONTROL_SUFFEX = ''  #use '_Cube' for .obj and ''
        for.osg
275
276 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL):
277     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_PREFEX, cpNumber
            ,CONTROLPOINT_ACTOR_CONTROL_SUFFEX)
278     pointNode = movementPaths.getChild(nodename)
279     nodebox = pointNode.getBoundingBox()
280     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
281     print 'control.path.',cpNumber, cntr
282     CP_ACTOR_CONTROL.append(cntr)
```

```
283
284 #Setting up foraging path of stilts condition in scene 4
285 CP_ACTOR_STILTS = []
286
287 CP_COUNT_ACTOR_STILTS = 13
288 CONTROLPOINT_ACTOR_STILTS_PREFEX = 'stilts.path.'  #name of boxes =
        layera1 ... layera17
289 CONTROLPOINT_ACTOR_STILTS_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
290
291 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS):
292     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_PREFEX, cpNumber,
          CONTROLPOINT_ACTOR_STILTS_SUFFEX)
293     pointNode = movementPaths.getChild(nodename)
294     nodebox = pointNode.getBoundingBox()
295     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
296     print 'stilts.path.',cpNumber, cntr
297     CP_ACTOR_STILTS.append(cntr)
298
299
300 #Setting up foraging path of stumps condition in scene 4
301 CP_ACTOR_STUMPS = []
302
303 CP_COUNT_ACTOR_STUMPS = 14
304 CONTROLPOINT_ACTOR_STUMPS_PREFEX = 'stumps.path.'  #name of boxes =
        layera1 ... layera17
305 CONTROLPOINT_ACTOR_STUMPS_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
306
307 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS):
308     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_PREFEX, cpNumber,
          CONTROLPOINT_ACTOR_STUMPS_SUFFEX)
309     pointNode = movementPaths.getChild(nodename)
310     nodebox = pointNode.getBoundingBox()
311     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
312     print 'stumps.path.',cpNumber, cntr
313     CP_ACTOR_STUMPS.append(cntr)
314
315
316
317 #Setting up homebound path of normal condition in scene 4
318 CP_ACTOR_CONTROL_RETURN = []
319
320 CP_COUNT_ACTOR_CONTROL_RETURN = 29
321 CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX = 'control.Rpath.'  #name of
        boxes = layera1 ... layera17
322 CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX = ''  #use '_Cube' for .obj
        and '' for.osg
323
324 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL_RETURN):
325     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX)
326     pointNode = movementPaths.getChild(nodename)
```

```
327     nodebox = pointNode.getBoundingBox()
328     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
329     print 'control.Rpath.',cpNumber, cntr
330     CP_ACTOR_CONTROL_RETURN.append(cntr)
331
332
333 #Setting up homebound path of stilts condition in scene 4
334 CP_ACTOR_STILTS_RETURN = []
335
336 CP_COUNT_ACTOR_STILTS_RETURN = 51
337 CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX = 'stilts.Rpath.'  #name of
        boxes = layera1 ... layera17
338 CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
        '' for.osg
339
340 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS_RETURN):
341     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX,
            cpNumber,CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX)
342     pointNode = movementPaths.getChild(nodename)
343     nodebox = pointNode.getBoundingBox()
344     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
345     print 'stilts.Rpath.',cpNumber, cntr
346     CP_ACTOR_STILTS_RETURN.append(cntr)
347
348
349 #Setting up homebound path of stumps condition in scene 4
350 CP_ACTOR_STUMPS_RETURN = []
351
352 CP_COUNT_ACTOR_STUMPS_RETURN = 37
353 CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX = 'stumps.Rpath.'  #name of
        boxes = layera1 ... layera17
354 CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
        '' for.osg
355
356 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS_RETURN):
357     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX,
            cpNumber,CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX)
358     pointNode = movementPaths.getChild(nodename)
359     nodebox = pointNode.getBoundingBox()
360     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
361     print 'stumps.Rpath.',cpNumber, cntr
362     CP_ACTOR_STUMPS_RETURN.append(cntr)
363
364
365
366
367
368 #Actors Activation
        ############################################################
369 def actorsScene1Enable():
370     ant1.enable(viz.RENDERING)
371     ant2.enable(viz.RENDERING)
```

```
372     ant3.enable(viz.RENDERING)
373     ant4.enable(viz.RENDERING)
374     ant5.enable(viz.RENDERING)
375
376 def actorsScene1Action():
377     for each in CP_ACTOR_1:
378         print "ant1.addAction(vizact.walkto(",each,"))"
379         ant1.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 80, 1))
380
381     for each in CP_ACTOR_2:
382         print "ant2.addAction(vizact.walkto(",each,"))"
383         ant2.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.28, 70, 1))
384
385     for each in CP_ACTOR_3:
386         print "ant3.addAction(vizact.walkto(",each,"))"
387         ant3.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.29, 90, 1))
388
389     for each in CP_ACTOR_4:
390         print "ant4.addAction(vizact.walkto(",each,"))"
391         ant4.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 75, 1))
392
393     for each in CP_ACTOR_5:
394         print "ant5.addAction(vizact.walkto(",each,"))"
395         ant5.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.28, 80, 1))
396
397 def actorsScene1Disable():
398     ant1.enable(viz.RENDERING)
399     ant2.enable(viz.RENDERING)
400     ant3.enable(viz.RENDERING)
401     ant4.enable(viz.RENDERING)
402     ant5.enable(viz.RENDERING)
403
404
405 def actorsScene2Enable():
406     ant6.enable(viz.RENDERING)
407
408 def actorsScene2Action():
409     for each in CP_ACTOR_GENERAL:
410         print "ant6.addAction(vizact.walkto(",each,"))"
411         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.292, 110, 1))
412
413 def actorsScene2Return():
414     ant6.addAction(vizact.spinto(0,1,0,200,0.15,viz.TIME))
415     for each in CP_ACTOR_RETURN:
416         print "ant6.addAction(vizact.walkto(",each,"))"
417         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.29, 35, 1))
418
419 def actorsScene2Disable():
```

```
420     #ant6.disable(viz.RENDERING)
421     ant6.add(vizact.fade(1,0,2))
422
423
424 def actorsScene4Enable():
425     ant7.enable(viz.RENDERING)
426     ant8.enable(viz.RENDERING)
427     ant9.enable(viz.RENDERING)
428
429 def controlAction():
430     for each in CP_ACTOR_CONTROL:
431         print "ant7.addAction(vizact.walkto(",each,"))"
432         ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
              each[2], 0.3, 25, 1))
433
434 def stiltsAction():
435     for each in CP_ACTOR_STILTS:
436         print "ant8.addAction(vizact.walkto(",each,"))"
437         ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
              each[2], 0.3, 25, 1))
438
439 def stumpsAction():
440     for each in CP_ACTOR_STUMPS:
441         print "ant9.addAction(vizact.walkto(",each,"))"
442         ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
              each[2], 0.3, 25, 1))
443
444 def controlReturn():
445     prevVertAnt7 = ant7.getPosition()
446     vertexListAnt7.append(ant7.getPosition())
447
448     count = 0
449     for each in CP_ACTOR_CONTROL_RETURN:
450         count = count + 1
451         if count < 14:
452             print "ant7.addAction(vizact.walkto(",each,"))"
453             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                  each[2], 0.30, 35, 1))
454         else:
455             print "ant7.addAction(vizact.walkto(",each,"))"
456             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                  each[2], 0.24, 900, 1))
457
458 def stiltsReturn():
459     prevVertAnt8 = ant8.getPosition()
460     vertexListAnt8.append(ant8.getPosition())
461
462     count = 0
463     for each in CP_ACTOR_STILTS_RETURN:
464         count = count + 1
465         if count < 20:
466             print "ant8.addAction(vizact.walkto(",each,"))"
467             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                  STILTS_OFFSET, each[2], 0.29, 35, 1))
468
```

```
469         else:
470             print "ant8.addAction(vizact.walkto(",each,"))"
471             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                    STILTS_OFFSET, each[2], 0.24, 900, 1))
472
473 def stumpsReturn():
474     prevVertAnt9 = ant9.getPosition()
475     vertexListAnt9.append(ant9.getPosition())
476
477     count = 0
478     for each in CP_ACTOR_STUMPS_RETURN:
479         count = count + 1
480         if count < 9:
481             print "ant9.addAction(vizact.walkto(",each,"))"
482             ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2-
                    STUMPS_OFFSET, each[2], 0.285, 35, 1))
483
484         else:
485             print "ant9.addAction(vizact.walkto(",each,"))"
486             ant9.addAction(vizact.walkto(each[0], each[1]++OFFSET1+OFFSET2
                    -STUMPS_OFFSET, each[2], 0.24, 900, 1))
487
488 def pickupAnts():
489     ant7.addAction(vizact.goto(9.478, 0.202, 0.3,0.05))
490     ant7.idlepose(3)
491     ant8.addAction(vizact.goto(9.437, 0.202,0.325,0.05))
492     ant8.idlepose(3)
493     ant9.addAction(vizact.goto(9.46,0.202,0.262,0.05))
494     ant9.idlepose(3)
495
496 def distributeAnts():
497     ant9.addAction(vizact.goto(9.453,0.202,-0.4,0.09))
498     ant8.addAction(vizact.goto(9.453,0.202,-0.2,0.07))
499     ant7.addAction(vizact.goto(9.453,0.202,0,0.05))
500
501 def applyStilts():
502     ant8.runAction(stilts)
503
504 def applyStumps():
505     ant9.runAction(stumps)
506
507 def dropAnts():
508     global prevVertAnt7, prevVertAnt8, prevVertAnt9
509     global vertexListAnt8, vertexListAnt8, vertexListAnt9
510
511     ant7.addAction(vizact.goto(9.453, 0.008,0,0.2))
512     ant7.addAction(vizact.spinto(0,1,0,270,0.68,viz.TIME))
513     ant7.idlepose(2)
514
515
516     ant8.addAction(vizact.goto(9.453,0.008,-0.2,0.2))
517     ant8.addAction(vizact.spinto(0,1,0,265,0.8,viz.TIME))
518     ant8.idlepose(2)
519
520
```

```
521     ant9.addAction(vizact.goto(9.453,0.008,-0.4,0.2))
522     ant9.addAction(vizact.spinto(0,1,0,272,0.58,viz.TIME))
523     ant9.idlepose(2)
524 
525     FINALSCENE = 1
526 
527 def actorsScene4Disable():
528     ant7.add(vizact.fade(1,0,2))
529     ant8.add(vizact.fade(1,0,2))
530     ant9.add(vizact.fade(1,0,2))
531 
532 def extraAntEnable():
533     ant10.enable(viz.RENDERING)
534 
535 def extraAntGo():
536     ant10.addAction(vizact.walkto(3.985, OFFSET1, 8.031, 0.295, 45, 1))
537     ant10.addAction(vizact.walkto(0.04, OFFSET1, 6.55, 0.295, 45, 1))
538     ant10.addAction(vizact.walkto(-31, OFFSET1, 19.969, 0.295, 45, 1))
539 
540 bread = viz.add('bread/bread.osg')
541 bread.setScale(0.0025,0.0025,0.0025)
542 bread.alpha(0)
543 
544 bread6 = bread.copy()
545 bread6.setScale(0.0025,0.0025,0.0025)
546 bread6.alpha(0)
547 
548 bread7 = bread.copy()
549 bread7.setScale(0.0025,0.0025,0.0025)
550 bread7.alpha(0)
551 
552 bread8 = bread.copy()
553 bread8.setScale(0.0025,0.0025,0.0025)
554 bread8.alpha(0)
555 
556 bread9 = bread.copy()
557 bread9.setScale(0.0025,0.0025,0.0025)
558 bread9.alpha(0)
559 
560 def giveFoodScene2():
561     bread6.add(vizact.fade(0,1,2))
562     ant6mount = ant6.getBone('mount0')
563     ant6mountLink = viz.link(ant6mount,bread6)
564     ant6mountLink.preTrans([0.0,0.0,0.02])
565 
566 def removeFoodScene2():
567     bread6.add(vizact.fade(1,0,2))
568 
569 def giveFoodControl():
570     bread7.add(vizact.fade(0,1,2))
571     ant7mount = ant7.getBone('mount0')
572     ant7mountLink = viz.link(ant7mount,bread7)
573     ant7mountLink.preTrans([0.0,0.0,0.02])
574 
575 def giveFoodStilts():
```

```
576     bread8.add(vizact.fade(0,1,2))
577     ant8mount = ant8.getBone('mount0')
578     ant8mountLink = viz.link(ant8mount,bread8)
579     ant8mountLink.preTrans([0.0,0.0,0.02])
580
581 def giveFoodStumps():
582     bread9.add(vizact.fade(0,1,2))
583     ant9mount = ant9.getBone('mount0')
584     ant9mountLink = viz.link(ant9mount,bread9)
585     ant9mountLink.preTrans([0.0,0.0,0.02])
586
587 def removeFoodControl():
588     bread7.add(vizact.fade(1,0,2))
589
590 def removeFoodStilts():
591     bread8.add(vizact.fade(1,0,2))
592
593 def removeFoodStumps():
594     bread9.add(vizact.fade(1,0,2))
595
596 def animationMultiplier(num):
597     global FINALSCENE
598
599     if ant1.getCycleRunning(1) or ant1.getCycleRunning(4):
600         ant1.setAnimationSpeed(0,1)
601     else:
602         ant1.setAnimationSpeed(0,1)
603
604     if ant10.getCycleRunning(1) or ant10.getCycleRunning(4):
605         ant10.setAnimationSpeed(0,1)
606     else:
607         ant10.setAnimationSpeed(0,1)
608
609     if ant2.getCycleRunning(1) or ant2.getCycleRunning(4):
610         ant2.setAnimationSpeed(0,1)
611     else:
612         ant2.setAnimationSpeed(0,1)
613
614     if ant3.getCycleRunning(1) or ant3.getCycleRunning(4):
615         ant3.setAnimationSpeed(0,1)
616     else:
617         ant3.setAnimationSpeed(0,1)
618
619     if ant4.getCycleRunning(1) or ant4.getCycleRunning(4):
620         ant4.setAnimationSpeed(0,1)
621     else:
622         ant4.setAnimationSpeed(0,1)
623
624     if ant5.getCycleRunning(1) or ant5.getCycleRunning(4):
625         ant5.setAnimationSpeed(0,1)
626     else:
627         ant5.setAnimationSpeed(0,1)
628
629     if ant6.getCycleRunning(1) or ant6.getCycleRunning(4):
630         ant6.setAnimationSpeed(0,1)
```

```
631         drawFreehandPreviewAnt6()
632     else:
633         ant6.setAnimationSpeed(0,1.5)
634 
635 
636     if (ant7.getCycleRunning(1) or ant7.getCycleRunning(4)) and
        FINALSCENE == 1:
637         ant7.setAnimationSpeed(0,1)
638         drawFreehandPreviewAnt7()
639     elif ant7.getCycleRunning(1) or ant7.getCycleRunning(4):
640         ant7.setAnimationSpeed(0,1)
641     else:
642         ant7.setAnimationSpeed(0,1.35)
643 
644     if (ant8.getCycleRunning(1) or ant8.getCycleRunning(4)) and
        FINALSCENE == 1:
645         ant8.setAnimationSpeed(0,1)
646         drawFreehandPreviewAnt8()
647     elif ant8.getCycleRunning(1) or ant8.getCycleRunning(4):
648         ant8.setAnimationSpeed(0,1)
649     else:
650         ant8.setAnimationSpeed(0,1)
651 
652     if (ant9.getCycleRunning(1) or ant9.getCycleRunning(4)) and
        FINALSCENE == 1:
653         ant9.setAnimationSpeed(0,1)
654         drawFreehandPreviewAnt9()
655     elif ant9.getCycleRunning(1) or ant9.getCycleRunning(4):
656         ant9.setAnimationSpeed(0,1)
657     else:
658         ant9.setAnimationSpeed(0,1)
659 
660 
661 
662 vertexListAnt6 = []
663 vertexListAnt7 = []
664 vertexListAnt8 = []
665 vertexListAnt9 = []
666 
667 prevVertAnt7 = [0,0,0]
668 prevVertAnt8 = [0,0,0]
669 prevVertAnt9 = [0,0,0]
670 
671 #viz.startlayer(viz.LINES)
672 viz.startlayer(viz.POINTS)
673 previewLine = viz.endlayer()
674 
675 prevVertAnt6 = ant6.getPosition()
676 vertexListAnt6.append(ant6.getPosition())
677 
678 #temp segments that will get discarded each frame
679 ant6segment = 0
680 ant7segment = 0
681 ant8segment = 0
682 ant9segment = 0
```

```
683
684 def drawFreehandPreviewAnt6():
685     global previewLine, vertexListAnt6
686     global ant6segment
687
688     prevVertAnt6 = vertexListAnt6[-1]
689     nextVertAnt6 = ant6.getPosition()
690     # We only want to draw a line segment if it is visible
691     if vizmat.Distance(prevVertAnt6, nextVertAnt6) > 0.040:
692         viz.startlayer(viz.LINES)
693         viz.linewidth(2)
694         viz.vertexcolor(1,0,0)
695         viz.vertex(prevVertAnt6)
696         viz.vertex(nextVertAnt6)
697         segment = viz.endlayer()
698         vertexListAnt6.append(nextVertAnt6)
699         # Attaching the segment to the preview, so that we may be able to
                access
700         # it later for deletion
701         segment.parent(previewLine)
702     else:
703         if ant6segment:
704             ant6segment.remove()
705         viz.startlayer(viz.LINES)
706         viz.linewidth(2)
707         viz.vertexcolor(1,0,0)
708         viz.vertex(prevVertAnt6)
709         viz.vertex(nextVertAnt6)
710         ant6segment = viz.endlayer()
711
712 def drawFreehandPreviewAnt7():
713     global previewLine, vertexListAnt7
714     global ant7segment
715
716     prevVertAnt7 = vertexListAnt7[-1]
717     nextVertAnt7 = ant7.getPosition()
718     # We only want to draw a line segment if it is visible
719     if vizmat.Distance(prevVertAnt7, nextVertAnt7) > 0.04:
720         viz.startlayer(viz.LINES)
721         viz.linewidth(2)
722         viz.vertexcolor(1,0,0)
723         viz.vertex(prevVertAnt7)
724         viz.vertex(nextVertAnt7)
725         segment = viz.endlayer()
726         vertexListAnt7.append(nextVertAnt7)
727         # Attaching the segment to the preview, so that we may be able to
                access
728         # it later for deletion
729         segment.parent(previewLine)
730     else:
731         if ant7segment:
732             ant7segment.remove()
733         viz.startlayer(viz.LINES)
734         viz.linewidth(2)
735         viz.vertexcolor(1,0,0)
```

```
736         viz.vertex(prevVertAnt7)
737         viz.vertex(nextVertAnt7)
738         ant7segment = viz.endlayer()
739
740
741 def drawFreehandPreviewAnt8():
742     global previewLine, vertexListAnt8
743     global ant8segment
744
745     prevVertAnt8 = vertexListAnt8[-1]
746     nextVertAnt8 = ant8.getPosition()
747     # We only want to draw a line segment if it is visible
748     if vizmat.Distance(prevVertAnt8, nextVertAnt8) > 0.04:
749         viz.startlayer(viz.LINES)
750         viz.linewidth(2)
751         viz.vertexcolor(1,0,0)
752         viz.vertex(prevVertAnt8)
753         viz.vertex(nextVertAnt8)
754         segment = viz.endlayer()
755         vertexListAnt8.append(nextVertAnt8)
756         # Attaching the segment to the preview, so that we may be able to
                access
757         # it later for deletion
758         segment.parent(previewLine)
759     else:
760         if ant8segment:
761             ant8segment.remove()
762         viz.startlayer(viz.LINES)
763         viz.linewidth(2)
764         viz.vertexcolor(1,0,0)
765         viz.vertex(prevVertAnt8)
766         viz.vertex(nextVertAnt8)
767         ant8segment = viz.endlayer()
768
769 def drawFreehandPreviewAnt9():
770     global previewLine, vertexListAnt9
771     global ant9segment
772
773     prevVertAnt9 = vertexListAnt9[-1]
774     nextVertAnt9 = ant9.getPosition()
775     # We only want to draw a line segment if it is visible
776     if vizmat.Distance(prevVertAnt9, nextVertAnt9) > 0.04:
777         viz.startlayer(viz.LINES)
778         viz.linewidth(2)
779         viz.vertexcolor(1,0,0)
780         viz.vertex(prevVertAnt9)
781         viz.vertex(nextVertAnt9)
782         segment = viz.endlayer()
783         vertexListAnt9.append(nextVertAnt9)
784         # Attaching the segment to the preview, so that we may be able to
                access
785         # it later for deletion
786         segment.parent(previewLine)
787     else:
788         if ant9segment:
```

```
789             ant9segment.remove()
790         viz.startlayer(viz.LINES)
791         viz.linewidth(2)
792         viz.vertexcolor(1,0,0)
793         viz.vertex(prevVertAnt9)
794         viz.vertex(nextVertAnt9)
795         ant9segment = viz.endlayer()
796
797 viz.callback(viz.TIMER_EVENT, animationMultiplier)
798 viz.starttimer(0,0.01,viz.FOREVER)
```

Listing F.6: actors2.py

```
1  ''' John Porte developed the base scripts for loading and reading
2      multiple objects, retreiving their object centres for animation
3      purpose
4  '''
5
6  import viz
7  import vizact
8  import vizmat
9
10 FINALSCENE = 0
11
12 #SCALE OF THE ANT#####################################################
13 SCALE = (0.008, 0.008, 0.008)
14 ANIMATIONSPEED = 1
15 OFFSET1 = 0.013; OFFSET2 = 0.006; STUMPS_OFFSET = 0.0015; STILTS_OFFSET
       = 0.001
16
17 #Loading .OSG files that contains movement points of the ants####
18 movementPaths = viz.add('antMovementPoints.osg')
19 movementPaths.disable(viz.RENDERING)
20
21 path_actor1_000 = movementPaths.getChild('path.actor1.000')
22 path_actor1_000_box = path_actor1_000.getBoundingBox()
23 ant1StartPos = path_actor1_000_box.getCenter()
24
25 path_actor2_000 = movementPaths.getChild('path.actor2.000')
26 path_actor2_000_box = path_actor2_000.getBoundingBox()
27 ant2StartPos = path_actor2_000_box.getCenter()
28
29 path_actor3_000 = movementPaths.getChild('path.actor3.000')
30 path_actor3_000_box = path_actor3_000.getBoundingBox()
31 ant3StartPos = path_actor3_000_box.getCenter()
32
33 path_actor4_000 = movementPaths.getChild('path.actor4.000')
34 path_actor4_000_box = path_actor4_000.getBoundingBox()
35 ant4StartPos = path_actor4_000_box.getCenter()
36
37 path_actor5_000 = movementPaths.getChild('path.actor5.000')
38 path_actor5_000_box = path_actor5_000.getBoundingBox()
39 ant5StartPos = path_actor5_000_box.getCenter()
40
41 path_actor6_000 = movementPaths.getChild('general.path.000')
```

```
42 path_actor6_000_box = path_actor6_000.getBoundingBox()
43 ant6StartPos = path_actor6_000_box.getCenter()
44
45 path_actor7_000 = movementPaths.getChild('control.path.000')
46 path_actor7_000_box = path_actor7_000.getBoundingBox()
47 ant7StartPos = path_actor7_000_box.getCenter()
48
49 path_actor8_000 = movementPaths.getChild('stilts.path.000')
50 path_actor8_000_box = path_actor8_000.getBoundingBox()
51 ant8StartPos = path_actor8_000_box.getCenter()
52
53 path_actor9_000 = movementPaths.getChild('stumps.path.000')
54 path_actor9_000_box = path_actor9_000.getBoundingBox()
55 ant9StartPos = path_actor9_000_box.getCenter()
56
57
58 #Pre-loading the ant##############################################
59 ant = viz.add('anthony3d/anthony3d.cfg')
60 ant.visible(0)
61 ################################################################
62
63 # Ant 1, 2, 3, 4 and 5 are for scene 1
64 ant1 = ant.copy()
65 ant1.runAction(vizact.fadeTo(0, time = 0, node = 'stilts.cmf')) #Hiding
       the stilts
66 ant1.setScale(SCALE)
67
68 normal = vizact.fadeTo(0, time = 0, node = 'stilts.cmf') #Hiding the
      stilts
69 stumps = vizact.fadeTo(0, time = 7, node = 'normal.cmf')
70 stilts = vizact.fadeTo(1, time = 7, node = 'stilts.cmf')
71
72 #antActions = [normal,stumps,stilts]
73
74 antFarm = []  #:)
75
76 def childAnt(srcAnt):  #make a copy of each ant and store it in the
      antFarm list
77     global antFarm
78     destAnt = srcAnt.copy()
79     destAnt.runAction(normal)
80     destAnt.setScale(SCALE)
81     viz.link(srcAnt,destAnt)
82     #destAnt.idlepose(2)
83     srcAnt.visible(0)
84     antFarm.append([srcAnt,destAnt])
85     return destAnt
86
87 def syncAnts():
88     global antFarm, antActions
89     for antpair in antFarm:
90         src = antpair[0]
91         dst = antpair[1]
92         for cycle in range(1,6):
93             if src.getCycleRunning(cycle): #if cycle running
```

```
94              if not dst.getCycleRunning(cycle): #make sure dst ant cycle
                    is also running
95                  dst.state(cycle)
96          if src.getAnimationSpeed(0) != dst.getAnimationSpeed(0): #if
                animation speed is different sync it too
97              dst.setAnimationSpeed(0,src.getAnimationSpeed(0))
98          if src.getVisible(): #this is all good for enabling ants, but not
                 for disabling
99              dst.visible(viz.ON)
100             src.visible(viz.OFF)
101         #if src.getAction() != dst.getAction():
102               #dst.runAction(src.getAction())
103
104
105 ant1.idlepose(2)
106 ant1.disable(viz.RENDERING)
107 ant1.setPosition(ant1StartPos[0], -ant1StartPos[2]+OFFSET1,
        ant1StartPos[1])
108 ant1.setAnimationSpeed(0,ANIMATIONSPEED)
109 ant1b = childAnt(ant1)
110 ant1b.disable(viz.LIGHTING)
111
112 ant2 = ant.copy()
113 ant2.runAction(normal)
114 ant2.setScale(SCALE)
115 ant2.idlepose(2)
116 ant2.disable(viz.RENDERING)
117 ant2.setPosition(ant2StartPos[0], -ant2StartPos[2]+OFFSET1,
        ant2StartPos[1])
118 ant2.setAnimationSpeed(0,ANIMATIONSPEED)
119 ant2b = childAnt(ant2)
120 ant2b.disable(viz.LIGHTING)
121
122 ant3 = ant.copy()
123 ant3.runAction(normal)
124 ant3.setScale(SCALE)
125 ant3.idlepose(2)
126 ant3.disable(viz.RENDERING)
127 ant3.setPosition(ant3StartPos[0], -ant3StartPos[2]+OFFSET1,
        ant3StartPos[1])
128 ant3.setAnimationSpeed(0,ANIMATIONSPEED)
129 ant3b = childAnt(ant3)
130 ant3b.disable(viz.LIGHTING)
131
132 ant4 = ant.copy()
133 ant4.runAction(normal)
134 ant4.setScale(SCALE)
135 ant4.idlepose(2)
136 ant4.disable(viz.RENDERING)
137 ant4.setPosition(ant4StartPos[0], -ant4StartPos[2]+OFFSET1,
        ant4StartPos[1])
138 ant4.setAnimationSpeed(0,ANIMATIONSPEED)
139 ant4b = childAnt(ant4)
140 ant4b.disable(viz.LIGHTING)
141
```

```
142 ant5 = ant.copy()
143 ant5.runAction(normal)
144 ant5.setScale(SCALE)
145 ant5.idlepose(2)
146 ant5.disable(viz.RENDERING)
147 ant5.setPosition(ant5StartPos[0], -ant5StartPos[2]+OFFSET1,
        ant5StartPos[1])
148 ant5.setAnimationSpeed(0,ANIMATIONSPEED)
149 ant5b = childAnt(ant5)
150 ant5b.disable(viz.LIGHTING)
151
152 #Ant 6 for scene
        2#####################################################################
153 ant6 = ant.copy()
154 ant6.runAction(normal)
155 ant6.setScale(SCALE)
156 ant6.idlepose(2)
157 ant6.disable(viz.RENDERING)
158 ant6.setPosition(ant6StartPos[0], -ant6StartPos[2]+OFFSET1,
        ant6StartPos[1])
159 ant6.setAnimationSpeed(0,ANIMATIONSPEED)
160 ant6b = childAnt(ant6)
161 ant6b.disable(viz.LIGHTING)
162
163 #Ant 7(normal), 8 (stilts)and 9(stumps) are for scene
        2#############################
164 ant7 = ant.copy()
165 ant7.runAction(normal)
166 ant7.setScale(SCALE)
167 ant7.idlepose(2)
168 ant7.disable(viz.RENDERING)
169 ant7.setPosition(ant7StartPos[0], -ant7StartPos[2]+OFFSET1,
        ant7StartPos[1])
170 ant7.setEuler([65,0,0])
171 ant7.setAnimationSpeed(0,ANIMATIONSPEED)
172 ant7b = childAnt(ant7)
173 ant7b.disable(viz.LIGHTING)
174
175 ant8 = ant.copy()
176 ant8.runAction(normal)
177 ant8.setScale(SCALE)
178 ant8.idlepose(2)
179 ant8.disable(viz.RENDERING)
180 ant8.setPosition(ant8StartPos[0], -ant8StartPos[2]+OFFSET1,
        ant8StartPos[1])
181 ant8.setEuler([90,0,0])
182 ant8.setAnimationSpeed(0,ANIMATIONSPEED)
183 ant8b = childAnt(ant8)
184 ant8b.disable(viz.LIGHTING)
185
186 ant9 = ant.copy()
187 ant9.setScale(SCALE)
188 ant9.runAction(normal)
189 ant9.idlepose(2)
190 ant9.disable(viz.RENDERING)
```

```
191 ant9.setPosition(ant9StartPos[0], -ant9StartPos[2]+OFFSET1,
        ant9StartPos[1])
192 ant9.setEuler([100,0,0])
193 ant9.setAnimationSpeed(0,ANIMATIONSPEED)
194 ant9b = childAnt(ant9)
195 ant9b.disable(viz.LIGHTING)
196
197 ##############################################
198 #Extra ants scene 5
199
200 ant10 = ant.copy()
201 ant10.setScale(SCALE)
202 ant10.runAction(normal)
203 ant10.idlepose(2)
204 ant10.disable(viz.RENDERING)
205 ant10.setPosition(3.985, OFFSET1, 8.031)
206 ant10.setEuler([0,0,0])
207 ant10.setAnimationSpeed(0,ANIMATIONSPEED)
208 childAnt(ant10)
209
210 ##############################################
211
212 #Setting up movement path actor 1 scene 1
213 CP_ACTOR_1 = []
214
215 CP_COUNT_ACTOR_1 = 23
216 CONTROLPOINT_ACTOR_1_PREFEX = 'path.actor1.'  #name of boxes = layera1
        ... layera17
217 CONTROLPOINT_ACTOR_1_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
218
219 for cpNumber in range( 0,CP_COUNT_ACTOR_1):
220     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_1_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_1_SUFFEX)
221     pointNode = movementPaths.getChild(nodename)
222     nodebox = pointNode.getBoundingBox()
223     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
224     print 'path.actor.1',cpNumber, cntr
225     CP_ACTOR_1.append(cntr)
226
227
228 #Setting up movement path actor 2 scene 1
229 CP_ACTOR_2 = []
230
231 CP_COUNT_ACTOR_2 = 19
232 CONTROLPOINT_ACTOR_2_PREFEX = 'path.actor2.'  #name of boxes = layera1
        ... layera17
233 CONTROLPOINT_ACTOR_2_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
234
235 for cpNumber in range( 0, CP_COUNT_ACTOR_2):
236     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_2_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_2_SUFFEX)
237     pointNode = movementPaths.getChild(nodename)
238     nodebox = pointNode.getBoundingBox()
```

```
239     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
240     print 'path.actor.2',cpNumber, cntr
241     CP_ACTOR_2.append(cntr)
242
243 #Setting up movement path actor 3 scene 1
244 CP_ACTOR_3 = []
245
246 CP_COUNT_ACTOR_3 = 18
247 CONTROLPOINT_ACTOR_3_PREFEX = 'path.actor3.'  #name of boxes = layera1
        ... layera17
248 CONTROLPOINT_ACTOR_3_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
249
250 for cpNumber in range( 0,CP_COUNT_ACTOR_3):
251     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_3_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_3_SUFFEX)
252     pointNode = movementPaths.getChild(nodename)
253     nodebox = pointNode.getBoundingBox()
254     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
255     print 'path.actor.3',cpNumber, cntr
256     CP_ACTOR_3.append(cntr)
257
258
259 #Setting up movement path actor 4 scene 1
260 CP_ACTOR_4 = []
261
262 CP_COUNT_ACTOR_4 = 11
263 CONTROLPOINT_ACTOR_4_PREFEX = 'path.actor4.'  #name of boxes = layera1
        ... layera17
264 CONTROLPOINT_ACTOR_4_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
265
266 for cpNumber in range( 0,CP_COUNT_ACTOR_4):
267     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_4_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_4_SUFFEX)
268     pointNode = movementPaths.getChild(nodename)
269     nodebox = pointNode.getBoundingBox()
270     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
271     print 'path.actor.4',cpNumber, cntr
272     CP_ACTOR_4.append(cntr)
273
274 #Setting up movement path actor 5 scene 1
275 CP_ACTOR_5 = []
276
277 CP_COUNT_ACTOR_5 = 16
278 CONTROLPOINT_ACTOR_5_PREFEX = 'path.actor5.'  #name of boxes = layera1
        ... layera17
279 CONTROLPOINT_ACTOR_5_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
280
281 for cpNumber in range( 0,CP_COUNT_ACTOR_5):
282     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_5_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_5_SUFFEX)
283     pointNode = movementPaths.getChild(nodename)
284     nodebox = pointNode.getBoundingBox()
```

```
285     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
286     print 'path.actor.5',cpNumber, cntr
287     CP_ACTOR_5.append(cntr)
288
289 #Setting up foraging path actor in scene 2
290 CP_ACTOR_GENERAL = []
291
292 CP_COUNT_ACTOR_GENERAL = 22
293 CONTROLPOINT_ACTOR_GENERAL_PREFEX = 'general.path.'  #name of boxes =
        layera1 ... layera17
294 CONTROLPOINT_ACTOR_GENERAL_SUFFEX = ''  #use '_Cube' for .obj and ''
        for.osg
295
296 for cpNumber in range( 0,CP_COUNT_ACTOR_GENERAL):
297     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_GENERAL_PREFEX, cpNumber
            ,CONTROLPOINT_ACTOR_GENERAL_SUFFEX)
298     pointNode = movementPaths.getChild(nodename)
299     nodebox = pointNode.getBoundingBox()
300     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
301     print 'general.path.',cpNumber, cntr
302     CP_ACTOR_GENERAL.append(cntr)
303
304 #Setting up homebound path actor in scene 2
305 CP_ACTOR_RETURN = []
306
307 CP_COUNT_ACTOR_RETURN = 6
308 CONTROLPOINT_ACTOR_RETURN_PREFEX = 'return.path.'  #name of boxes =
        layera1 ... layera17
309 CONTROLPOINT_ACTOR_RETURN_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
310
311 for cpNumber in range( 0,CP_COUNT_ACTOR_RETURN):
312     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_RETURN_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_RETURN_SUFFEX)
313     pointNode = movementPaths.getChild(nodename)
314     nodebox = pointNode.getBoundingBox()
315     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
316     print 'return.path.',cpNumber, cntr
317     CP_ACTOR_RETURN.append(cntr)
318
319 #Setting up foraging path of normal condition in scene 4
320 CP_ACTOR_CONTROL = []
321
322 CP_COUNT_ACTOR_CONTROL = 13
323 CONTROLPOINT_ACTOR_CONTROL_PREFEX = 'control.path.'  #name of boxes =
        layera1 ... layera17
324 CONTROLPOINT_ACTOR_CONTROL_SUFFEX = ''  #use '_Cube' for .obj and ''
        for.osg
325
326 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL):
327     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_PREFEX, cpNumber
            ,CONTROLPOINT_ACTOR_CONTROL_SUFFEX)
```

```
328     pointNode = movementPaths.getChild(nodename)
329     nodebox = pointNode.getBoundingBox()
330     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
331     print 'control.path.',cpNumber, cntr
332     CP_ACTOR_CONTROL.append(cntr)
333
334 #Setting up foraging path of stilts condition in scene 4
335 CP_ACTOR_STILTS = []
336
337 CP_COUNT_ACTOR_STILTS = 13
338 CONTROLPOINT_ACTOR_STILTS_PREFEX = 'stilts.path.'  #name of boxes =
        layera1 ... layera17
339 CONTROLPOINT_ACTOR_STILTS_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
340
341 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS):
342     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_STILTS_SUFFEX)
343     pointNode = movementPaths.getChild(nodename)
344     nodebox = pointNode.getBoundingBox()
345     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
346     print 'stilts.path.',cpNumber, cntr
347     CP_ACTOR_STILTS.append(cntr)
348
349
350 #Setting up foraging path of stumps condition in scene 4
351 CP_ACTOR_STUMPS = []
352
353 CP_COUNT_ACTOR_STUMPS = 14
354 CONTROLPOINT_ACTOR_STUMPS_PREFEX = 'stumps.path.'  #name of boxes =
        layera1 ... layera17
355 CONTROLPOINT_ACTOR_STUMPS_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
356
357 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS):
358     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_STUMPS_SUFFEX)
359     pointNode = movementPaths.getChild(nodename)
360     nodebox = pointNode.getBoundingBox()
361     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
362     print 'stumps.path.',cpNumber, cntr
363     CP_ACTOR_STUMPS.append(cntr)
364
365
366
367 #Setting up homebound path of normal condition in scene 4
368 CP_ACTOR_CONTROL_RETURN = []
369
370 CP_COUNT_ACTOR_CONTROL_RETURN = 29
371 CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX = 'control.Rpath.'  #name of
        boxes = layera1 ... layera17
```

```
372 CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX = ''  #use '_Cube' for .obj
    and '' for.osg
373
374 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL_RETURN):
375     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX)
376     pointNode = movementPaths.getChild(nodename)
377     nodebox = pointNode.getBoundingBox()
378     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
379     print 'control.Rpath.',cpNumber, cntr
380     CP_ACTOR_CONTROL_RETURN.append(cntr)
381
382
383 #Setting up homebound path of stilts condition in scene 4
384 CP_ACTOR_STILTS_RETURN = []
385
386 CP_COUNT_ACTOR_STILTS_RETURN = 51
387 CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX = 'stilts.Rpath.'  #name of
    boxes = layera1 ... layera17
388 CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
      '' for.osg
389
390 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS_RETURN):
391     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX)
392     pointNode = movementPaths.getChild(nodename)
393     nodebox = pointNode.getBoundingBox()
394     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
395     print 'stilts.Rpath.',cpNumber, cntr
396     CP_ACTOR_STILTS_RETURN.append(cntr)
397
398
399 #Setting up homebound path of stumps condition in scene 4
400 CP_ACTOR_STUMPS_RETURN = []
401
402 CP_COUNT_ACTOR_STUMPS_RETURN = 37
403 CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX = 'stumps.Rpath.'  #name of
    boxes = layera1 ... layera17
404 CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
      '' for.osg
405
406 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS_RETURN):
407     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX)
408     pointNode = movementPaths.getChild(nodename)
409     nodebox = pointNode.getBoundingBox()
410     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
411     print 'stumps.Rpath.',cpNumber, cntr
412     CP_ACTOR_STUMPS_RETURN.append(cntr)
413
414
415
```

```
416
417
418 #Actors Activation
    ############################################################
419 def actorsScene1Enable():
420     ant1.enable(viz.RENDERING)
421     ant2.enable(viz.RENDERING)
422     ant3.enable(viz.RENDERING)
423     ant4.enable(viz.RENDERING)
424     ant5.enable(viz.RENDERING)
425
426 def actorsScene1Action():
427     for each in CP_ACTOR_1:
428         print "ant1.addAction(vizact.walkto(",each,"))"
429         ant1.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 80, 1))
430
431     for each in CP_ACTOR_2:
432         print "ant2.addAction(vizact.walkto(",each,"))"
433         ant2.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.28, 70, 1))
434
435     for each in CP_ACTOR_3:
436         print "ant3.addAction(vizact.walkto(",each,"))"
437         ant3.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.29, 90, 1))
438
439     for each in CP_ACTOR_4:
440         print "ant4.addAction(vizact.walkto(",each,"))"
441         ant4.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 75, 1))
442
443     for each in CP_ACTOR_5:
444         print "ant5.addAction(vizact.walkto(",each,"))"
445         ant5.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.28, 80, 1))
446
447 def actorsScene1Disable():
448     ant1.enable(viz.RENDERING)
449     ant2.enable(viz.RENDERING)
450     ant3.enable(viz.RENDERING)
451     ant4.enable(viz.RENDERING)
452     ant5.enable(viz.RENDERING)
453
454
455 def actorsScene2Enable():
456     ant6.enable(viz.RENDERING)
457
458 def actorsScene2Action():
459     for each in CP_ACTOR_GENERAL:
460         print "ant6.addAction(vizact.walkto(",each,"))"
461         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.292, 110, 1))
462
463 def actorsScene2Return():
```

```
464     ant6.addAction(vizact.spinto(0,1,0,200,0.15,viz.TIME))
465     for each in CP_ACTOR_RETURN:
466         print "ant6.addAction(vizact.walkto(",each,"))"
467         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
                0.29, 35, 4))
468
469 def actorsScene2Disable():
470     #ant6.disable(viz.RENDERING)
471     ant6.add(vizact.fade(1,0,2))
472
473
474 def actorsScene4Enable():
475     ant7.enable(viz.RENDERING)
476     ant8.enable(viz.RENDERING)
477     ant9.enable(viz.RENDERING)
478
479 def controlAction():
480     for each in CP_ACTOR_CONTROL:
481         print "ant7.addAction(vizact.walkto(",each,"))"
482         ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
483
484 def stiltsAction():
485     for each in CP_ACTOR_STILTS:
486         print "ant8.addAction(vizact.walkto(",each,"))"
487         ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
488
489 def stumpsAction():
490     for each in CP_ACTOR_STUMPS:
491         print "ant9.addAction(vizact.walkto(",each,"))"
492         ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
493
494 def controlReturn():
495     prevVertAnt7 = ant7.getPosition()
496     vertexListAnt7.append(ant7.getPosition())
497
498     count = 0
499     for each in CP_ACTOR_CONTROL_RETURN:
500         count = count + 1
501         if count < 14:
502             print "ant7.addAction(vizact.walkto(",each,"))"
503             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                    each[2], 0.30, 35, 4))
504         else:
505             print "ant7.addAction(vizact.walkto(",each,"))"
506             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                    each[2], 0.24, 900, 4))
507
508 def stiltsReturn():
509     prevVertAnt8 = ant8.getPosition()
510     vertexListAnt8.append(ant8.getPosition())
511
512     count = 0
```

```
513     for each in CP_ACTOR_STILTS_RETURN:
514         count = count + 1
515         if count < 20:
516             print "ant8.addAction(vizact.walkto(",each,"))"
517             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                    STILTS_OFFSET, each[2], 0.29, 35, 4))
518
519         else:
520             print "ant8.addAction(vizact.walkto(",each,"))"
521             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                    STILTS_OFFSET, each[2], 0.24, 900, 4))
522
523 def stumpsReturn():
524     prevVertAnt9 = ant9.getPosition()
525     vertexListAnt9.append(ant9.getPosition())
526
527     count = 0
528     for each in CP_ACTOR_STUMPS_RETURN:
529         count = count + 1
530         if count < 9:
531             print "ant9.addAction(vizact.walkto(",each,"))"
532             ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2-
                    STUMPS_OFFSET, each[2], 0.285, 35, 4))
533
534         else:
535             print "ant9.addAction(vizact.walkto(",each,"))"
536             ant9.addAction(vizact.walkto(each[0], each[1]++OFFSET1+OFFSET2
                    -STUMPS_OFFSET, each[2], 0.24, 900, 4))
537
538 def pickupAnts():
539     ant7.addAction(vizact.goto(9.478, 0.202, 0.3,0.05))
540     ant7.idlepose(5)
541     ant8.addAction(vizact.goto(9.437, 0.202,0.325,0.05))
542     ant8.idlepose(5)
543     ant9.addAction(vizact.goto(9.46,0.202,0.262,0.05))
544     ant9.idlepose(5)
545
546 def distributeAnts():
547     ant9.addAction(vizact.goto(9.453,0.202,-0.4,0.09))
548     ant8.addAction(vizact.goto(9.453,0.202,-0.2,0.07))
549     ant7.addAction(vizact.goto(9.453,0.202,0,0.05))
550
551 def applyStilts():
552     ant8.idlepose(5)
553     ant8b.idlepose(5)
554     ant8b.runAction(stilts)
555     ant8.idlepose(5)
556     ant8b.idlepose(5)
557
558 def applyStumps():
559     ant9.idlepose(5)
560     ant9b.idlepose(5)
561     ant9b.runAction(stumps)
562     ant9.idlepose(5)
563     ant9b.idlepose(5)
```

```
564
565 def dropAnts():
566     global prevVertAnt7, prevVertAnt8, prevVertAnt9
567     global vertexListAnt8, vertexListAnt8, vertexListAnt9
568
569     ant7.addAction(vizact.goto(9.453, 0.008,0,0.2))
570     ant7.addAction(vizact.spinto(0,1,0,270,0.68,viz.TIME))
571     ant7.idlepose(5)
572
573
574     ant8.addAction(vizact.goto(9.453,0.008,-0.2,0.2))
575     ant8.addAction(vizact.spinto(0,1,0,265,0.8,viz.TIME))
576     ant8.idlepose(5)
577
578
579     ant9.addAction(vizact.goto(9.453,0.008,-0.4,0.2))
580     ant9.addAction(vizact.spinto(0,1,0,272,0.58,viz.TIME))
581     ant9.idlepose(5)
582
583     FINALSCENE = 1
584
585 def actorsScene4Disable():
586     ant7.add(vizact.fade(1,0,2))
587     ant8.add(vizact.fade(1,0,2))
588     ant9.add(vizact.fade(1,0,2))
589
590 def extraAntEnable():
591     ant10.enable(viz.RENDERING)
592
593 def extraAntGo():
594     ant10.addAction(vizact.walkto(3.985, OFFSET1, 8.031, 0.295, 45, 1))
595     ant10.addAction(vizact.walkto(0.04, OFFSET1, 6.55, 0.295, 45, 1))
596     ant10.addAction(vizact.walkto(-31, OFFSET1, 19.969, 0.295, 45, 1))
597
598 bread = viz.add('bread/bread.osg')
599 #bread.setScale(0.0059,0.0059,0.0059)
600 bread.setScale(0.0052,0.0052,0.0052)
601 bread.alpha(0)
602
603 bread6 = bread.copy()
604 bread6.setScale(0.0055,0.0068,0.0045)
605 bread6.alpha(0)
606
607 bread7 = bread.copy()
608 bread7.setScale(0.0055,0.0068,0.0045)
609 bread7.alpha(0)
610
611 bread8 = bread.copy()
612 bread8.setScale(0.0055,0.0068,0.0045)
613 bread8.alpha(0)
614
615 bread9 = bread.copy()
616 bread9.setScale(0.0055,0.0068,0.0045)
617 bread9.alpha(0)
618
```

```
619 def giveFoodScene2():
620     bread6.add(vizact.fade(0,1,2))
621     ant6mount = ant6b.getBone('mount0')
622     ant6mountLink = viz.link(ant6mount,bread6)
623     #ant6mountLink.preTrans([0.0009,0.0,0.0])
624     ant6mountLink.preTrans([0.0009,-0.0008,0])
625 
626 def removeFoodScene2():
627     bread6.add(vizact.fade(1,0,2))
628 
629 def giveFoodControl():
630     bread7.add(vizact.fade(0,1,2))
631     ant7mount = ant7b.getBone('mount0')
632     ant7mountLink = viz.link(ant7mount,bread7)
633     #ant7mountLink.preTrans([0.0009,0.0,0.0])
634     ant7mountLink.preTrans([0.0009,-0.0008,0])
635     ant7.idlepose(5)
636     ant7b.idlepose(5)
637 
638 
639 def giveFoodStilts():
640     bread8.add(vizact.fade(0,1,2))
641     ant8mount = ant8b.getBone('mount0')
642     ant8mountLink = viz.link(ant8mount,bread8)
643     #ant8mountLink.preTrans([0.0009,0.0,0.0])
644     ant8mountLink.preTrans([0.0009,-0.0008,0])
645     ant8.idlepose(5)
646     ant8b.idlepose(5)
647 
648 def giveFoodStumps():
649     bread9.add(vizact.fade(0,1,2))
650     ant9mount = ant9b.getBone('mount0')
651     ant9mountLink = viz.link(ant9mount,bread9)
652     #ant9mountLink.preTrans([0.0009,0.0,0.0])
653     ant9mountLink.preTrans([0.0009,-0.0008,0])
654     ant9.idlepose(5)
655     ant9b.idlepose(5)
656 
657 def removeFoodControl():
658     bread7.add(vizact.fade(1,0,2))
659     ant7.idlepose(2)
660     ant7b.idlepose(2)
661 
662 def removeFoodStilts():
663     bread8.add(vizact.fade(1,0,2))
664     ant8.idlepose(2)
665     ant8b.idlepose(2)
666 
667 def removeFoodStumps():
668     bread9.add(vizact.fade(1,0,2))
669     ant9.idlepose(2)
670     ant9b.idlepose(2)
671 
672 def animationMultiplier(num):
673     global FINALSCENE
```

```
674
675     if ant1.getCycleRunning(1) or ant1.getCycleRunning(4):
676         ant1.setAnimationSpeed(0,5.0)
677     else:
678         ant1.setAnimationSpeed(0,1)
679
680     if ant10.getCycleRunning(1) or ant10.getCycleRunning(4):
681         ant10.setAnimationSpeed(0,5.0)
682     else:
683         ant10.setAnimationSpeed(0,1)
684
685     if ant2.getCycleRunning(1) or ant2.getCycleRunning(4):
686         ant2.setAnimationSpeed(0,5.0)
687     else:
688         ant2.setAnimationSpeed(0,1)
689
690     if ant3.getCycleRunning(1) or ant3.getCycleRunning(4):
691         ant3.setAnimationSpeed(0,5.0)
692     else:
693         ant3.setAnimationSpeed(0,1)
694
695     if ant4.getCycleRunning(1) or ant4.getCycleRunning(4):
696         ant4.setAnimationSpeed(0,5.0)
697     else:
698         ant4.setAnimationSpeed(0,1)
699
700     if ant5.getCycleRunning(1) or ant5.getCycleRunning(4):
701         ant5.setAnimationSpeed(0,5.0)
702     else:
703         ant5.setAnimationSpeed(0,1)
704
705     if ant6.getCycleRunning(1) or ant6.getCycleRunning(4):
706         ant6.setAnimationSpeed(0,5.0)
707         drawFreehandPreviewAnt6()
708     else:
709         ant6.setAnimationSpeed(0,0.8)
710
711     if (ant7.getCycleRunning(1) or ant7.getCycleRunning(4)) and
        FINALSCENE == 1:
712         ant7.setAnimationSpeed(0,5.0)
713         drawFreehandPreviewAnt7()
714     elif ant7.getCycleRunning(1) or ant7.getCycleRunning(4):
715         ant7.setAnimationSpeed(0,5.0)
716     else:
717         ant7.setAnimationSpeed(0,0.85)
718
719     if (ant8.getCycleRunning(1) or ant8.getCycleRunning(4)) and
        FINALSCENE == 1:
720         ant8.setAnimationSpeed(0,5.0)
721         drawFreehandPreviewAnt8()
722     elif ant8.getCycleRunning(1) or ant8.getCycleRunning(4):
723         ant8.setAnimationSpeed(0,5.0)
724     else:
725         ant8.setAnimationSpeed(0,0.5)
726
```

```
727     if (ant9.getCycleRunning(1) or ant9.getCycleRunning(4)) and
            FINALSCENE == 1:
728         ant9.setAnimationSpeed(0,5.0)
729         drawFreehandPreviewAnt9()
730     elif ant9.getCycleRunning(1) or ant9.getCycleRunning(4):
731         ant9.setAnimationSpeed(0,5.0)
732     else:
733         ant9.setAnimationSpeed(0,1)
734 
735     syncAnts()
736 
737 
738 vertexListAnt6 = []
739 vertexListAnt7 = []
740 vertexListAnt8 = []
741 vertexListAnt9 = []
742 
743 prevVertAnt7 = [0,0,0]
744 prevVertAnt8 = [0,0,0]
745 prevVertAnt9 = [0,0,0]
746 
747 #viz.startlayer(viz.LINES)
748 viz.startlayer(viz.POINTS)
749 previewLine = viz.endlayer()
750 
751 prevVertAnt6 = ant6.getPosition()
752 vertexListAnt6.append(ant6.getPosition())
753 
754 #temp segments that will get discarded each frame
755 ant6segment = 0
756 ant7segment = 0
757 ant8segment = 0
758 ant9segment = 0
759 
760 def drawFreehandPreviewAnt6():
761     global previewLine, vertexListAnt6
762     global ant6segment
763 
764     prevVertAnt6 = vertexListAnt6[-1]
765     nextVertAnt6 = ant6.getPosition()
766     # We only want to draw a line segment if it is visible
767     if vizmat.Distance(prevVertAnt6, nextVertAnt6) > 0.040:
768         viz.startlayer(viz.LINES)
769         viz.linewidth(2)
770         viz.vertexcolor(1,0,0)
771         viz.vertex(prevVertAnt6)
772         viz.vertex(nextVertAnt6)
773         segment = viz.endlayer()
774         vertexListAnt6.append(nextVertAnt6)
775         # Attaching the segment to the preview, so that we may be able to
                access
776         # it later for deletion
777         segment.parent(previewLine)
778     else:
779         if ant6segment:
```

```
780             ant6segment.remove()
781         viz.startlayer(viz.LINES)
782         viz.linewidth(2)
783         viz.vertexcolor(1,0,0)
784         viz.vertex(prevVertAnt6)
785         viz.vertex(nextVertAnt6)
786         ant6segment = viz.endlayer()
787
788 def drawFreehandPreviewAnt7():
789     global previewLine, vertexListAnt7
790     global ant7segment
791
792     prevVertAnt7 = vertexListAnt7[-1]
793     nextVertAnt7 = ant7.getPosition()
794     # We only want to draw a line segment if it is visible
795     if vizmat.Distance(prevVertAnt7, nextVertAnt7) > 0.04:
796         viz.startlayer(viz.LINES)
797         viz.linewidth(2)
798         viz.vertexcolor(1,0,0)
799         viz.vertex(prevVertAnt7)
800         viz.vertex(nextVertAnt7)
801         segment = viz.endlayer()
802         vertexListAnt7.append(nextVertAnt7)
803         # Attaching the segment to the preview, so that we may be able to
                access
804         # it later for deletion
805         segment.parent(previewLine)
806     else:
807         if ant7segment:
808             ant7segment.remove()
809         viz.startlayer(viz.LINES)
810         viz.linewidth(2)
811         viz.vertexcolor(1,0,0)
812         viz.vertex(prevVertAnt7)
813         viz.vertex(nextVertAnt7)
814         ant7segment = viz.endlayer()
815
816
817 def drawFreehandPreviewAnt8():
818     global previewLine, vertexListAnt8
819     global ant8segment
820
821     prevVertAnt8 = vertexListAnt8[-1]
822     nextVertAnt8 = ant8.getPosition()
823     # We only want to draw a line segment if it is visible
824     if vizmat.Distance(prevVertAnt8, nextVertAnt8) > 0.04:
825         viz.startlayer(viz.LINES)
826         viz.linewidth(2)
827         viz.vertexcolor(1,0,0)
828         viz.vertex(prevVertAnt8)
829         viz.vertex(nextVertAnt8)
830         segment = viz.endlayer()
831         vertexListAnt8.append(nextVertAnt8)
832         # Attaching the segment to the preview, so that we may be able to
                access
```

```
833         # it later for deletion
834         segment.parent(previewLine)
835     else:
836         if ant8segment:
837             ant8segment.remove()
838         viz.startlayer(viz.LINES)
839         viz.linewidth(2)
840         viz.vertexcolor(1,0,0)
841         viz.vertex(prevVertAnt8)
842         viz.vertex(nextVertAnt8)
843         ant8segment = viz.endlayer()
844 
845 def drawFreehandPreviewAnt9():
846     global previewLine, vertexListAnt9
847     global ant9segment
848 
849     prevVertAnt9 = vertexListAnt9[-1]
850     nextVertAnt9 = ant9.getPosition()
851     # We only want to draw a line segment if it is visible
852     if vizmat.Distance(prevVertAnt9, nextVertAnt9) > 0.04:
853         viz.startlayer(viz.LINES)
854         viz.linewidth(2)
855         viz.vertexcolor(1,0,0)
856         viz.vertex(prevVertAnt9)
857         viz.vertex(nextVertAnt9)
858         segment = viz.endlayer()
859         vertexListAnt9.append(nextVertAnt9)
860         # Attaching the segment to the preview, so that we may be able to
                access
861         # it later for deletion
862         segment.parent(previewLine)
863     else:
864         if ant9segment:
865             ant9segment.remove()
866         viz.startlayer(viz.LINES)
867         viz.linewidth(2)
868         viz.vertexcolor(1,0,0)
869         viz.vertex(prevVertAnt9)
870         viz.vertex(nextVertAnt9)
871         ant9segment = viz.endlayer()
872 
873 viz.callback(viz.TIMER_EVENT,animationMultiplier)
874 viz.starttimer(0,0.01,viz.FOREVER)
```

Listing F.7: actors3.py

```
1 ''' John Porte developed the base scripts for loading and reading
2     multiple objects, retreiving their object centres for animation
3     purpose
4 '''
5 
6 import viz
7 import vizact
8 import vizmat
9 
```

```
10 FINALSCENE = 0
11
12 #SCALE OF THE ANT#####################################################
13 SCALE = (0.0065,0.0065,0.0065)
14 ANIMATIONSPEED = 1
15 OFFSET1 = 0.01; OFFSET2 = 0.007; STUMPS_OFFSET = 0.004; STILTS_OFFSET =
       0.004
16
17 #Loading .OSG files that contains movement points of the ants####
18 movementPaths = viz.add('antMovementPoints.osg')
19 movementPaths.disable(viz.RENDERING)
20
21 path_actor1_000 = movementPaths.getChild('path.actor1.000')
22 path_actor1_000_box = path_actor1_000.getBoundingBox()
23 ant1StartPos = path_actor1_000_box.getCenter()
24
25 path_actor2_000 = movementPaths.getChild('path.actor2.000')
26 path_actor2_000_box = path_actor2_000.getBoundingBox()
27 ant2StartPos = path_actor2_000_box.getCenter()
28
29 path_actor3_000 = movementPaths.getChild('path.actor3.000')
30 path_actor3_000_box = path_actor3_000.getBoundingBox()
31 ant3StartPos = path_actor3_000_box.getCenter()
32
33 path_actor4_000 = movementPaths.getChild('path.actor4.000')
34 path_actor4_000_box = path_actor4_000.getBoundingBox()
35 ant4StartPos = path_actor4_000_box.getCenter()
36
37 path_actor5_000 = movementPaths.getChild('path.actor5.000')
38 path_actor5_000_box = path_actor5_000.getBoundingBox()
39 ant5StartPos = path_actor5_000_box.getCenter()
40
41 path_actor6_000 = movementPaths.getChild('general.path.000')
42 path_actor6_000_box = path_actor6_000.getBoundingBox()
43 ant6StartPos = path_actor6_000_box.getCenter()
44
45 path_actor7_000 = movementPaths.getChild('control.path.000')
46 path_actor7_000_box = path_actor7_000.getBoundingBox()
47 ant7StartPos = path_actor7_000_box.getCenter()
48
49 path_actor8_000 = movementPaths.getChild('stilts.path.000')
50 path_actor8_000_box = path_actor8_000.getBoundingBox()
51 ant8StartPos = path_actor8_000_box.getCenter()
52
53 path_actor9_000 = movementPaths.getChild('stumps.path.000')
54 path_actor9_000_box = path_actor9_000.getBoundingBox()
55 ant9StartPos = path_actor9_000_box.getCenter()
56
57
58 #Pre-loading the ant#################################################
59 ant = viz.add('cfortis/cfortis.cfg')
60 ant.visible(0)
61 ############################################################
62
63 # Ant 1, 2, 3, 4 and 5 are for scene 1
```

```
64 ant1 = ant.copy()
65 ant1.runAction(vizact.fadeTo(0, time = 0, node = 'stilts.cmf')) #Hiding
       the stilts
66 ant1.setScale(SCALE)
67
68 normal = vizact.fadeTo(0, time = 0, node = 'stilts.cmf') #Hiding the
       stilts
69 stumps = vizact.fadeTo(0, time = 7, node = 'normal.cmf')
70 stilts = vizact.fadeTo(1, time = 7, node = 'stilts.cmf')
71
72 #antActions = [normal,stumps,stilts]
73
74 antFarm = []  #:)
75
76 def childAnt(srcAnt):  #make a copy of each ant and store it in the
       antFarm list
77     global antFarm
78     destAnt = srcAnt.copy()
79     destAnt.runAction(normal)
80     destAnt.setScale(SCALE)
81     viz.link(srcAnt,destAnt)
82     destAnt.state(2)
83     srcAnt.visible(0)
84     antFarm.append([srcAnt,destAnt])
85     return destAnt
86
87 def syncAnts():
88     global antFarm, antActions
89     for antpair in antFarm:
90         src = antpair[0]
91         dst = antpair[1]
92         for cycle in range(1,6):
93             if src.getCycleRunning(cycle): #if cycle running
94                 if not dst.getCycleRunning(cycle): #make sure dst ant cycle
                       is also running
95                     dst.state(cycle)
96         if src.getAnimationSpeed(0) != dst.getAnimationSpeed(0): #if
               animation speed is different sync it too
97             dst.setAnimationSpeed(0,src.getAnimationSpeed(0))
98         if src.getVisible(): #this is all good for enabling ants, but not
                for disabling
99             dst.visible(viz.ON)
100            src.visible(viz.OFF)
101        #if src.getAction() != dst.getAction():
102                #dst.runAction(src.getAction())
103
104
105 ant1.idlepose(2)
106 ant1.disable(viz.RENDERING)
107 ant1.setPosition(ant1StartPos[0], -ant1StartPos[2]+OFFSET1,
       ant1StartPos[1])
108 ant1.setAnimationSpeed(0,ANIMATIONSPEED)
109 childAnt(ant1)
110
111 ant2 = ant.copy()
```

```
112 ant2.runAction(normal)
113 ant2.setScale(SCALE)
114 ant2.idlepose(2)
115 ant2.disable(viz.RENDERING)
116 ant2.setPosition(ant2StartPos[0], -ant2StartPos[2]+OFFSET1,
        ant2StartPos[1])
117 ant2.setAnimationSpeed(0,ANIMATIONSPEED)
118 childAnt(ant2)
119
120 ant3 = ant.copy()
121 ant3.runAction(normal)
122 ant3.setScale(SCALE)
123 ant3.idlepose(2)
124 ant3.disable(viz.RENDERING)
125 ant3.setPosition(ant3StartPos[0], -ant3StartPos[2]+OFFSET1,
        ant3StartPos[1])
126 ant3.setAnimationSpeed(0,ANIMATIONSPEED)
127 childAnt(ant3)
128
129 ant4 = ant.copy()
130 ant4.runAction(normal)
131 ant4.setScale(SCALE)
132 ant4.idlepose(2)
133 ant4.disable(viz.RENDERING)
134 ant4.setPosition(ant4StartPos[0], -ant4StartPos[2]+OFFSET1,
        ant4StartPos[1])
135 ant4.setAnimationSpeed(0,ANIMATIONSPEED)
136 childAnt(ant4)
137
138 ant5 = ant.copy()
139 ant5.runAction(normal)
140 ant5.setScale(SCALE)
141 ant5.idlepose(2)
142 ant5.disable(viz.RENDERING)
143 ant5.setPosition(ant5StartPos[0], -ant5StartPos[2]+OFFSET1,
        ant5StartPos[1])
144 ant5.setAnimationSpeed(0,ANIMATIONSPEED)
145 childAnt(ant5)
146
147 #Ant 6 for scene
        2#####################################################################
148 ant6 = ant.copy()
149 ant6.runAction(normal)
150 ant6.setScale(SCALE)
151 ant6.idlepose(2)
152 ant6.disable(viz.RENDERING)
153 ant6.setPosition(ant6StartPos[0], -ant6StartPos[2]+OFFSET1,
        ant6StartPos[1])
154 ant6.setAnimationSpeed(0,ANIMATIONSPEED)
155 ant6b = childAnt(ant6)
156
157 #Ant 7(normal), 8 (stilts)and 9(stumps) are for scene
        2#############################
158 ant7 = ant.copy()
159 ant7.runAction(normal)
```

```
160 ant7.setScale(SCALE)
161 ant7.idlepose(2)
162 ant7.disable(viz.RENDERING)
163 ant7.setPosition(ant7StartPos[0], -ant7StartPos[2]+OFFSET1,
        ant7StartPos[1])
164 ant7.setEuler([65,0,0]) #was 45
165 ant7.setAnimationSpeed(0,ANIMATIONSPEED)
166 ant7b = childAnt(ant7)
167
168 ant8 = ant.copy()
169 ant8.runAction(normal)
170 ant8.setScale(SCALE)
171 ant8.idlepose(2)
172 ant8.disable(viz.RENDERING)
173 ant8.setPosition(ant8StartPos[0], -ant8StartPos[2]+OFFSET1,
        ant8StartPos[1])
174 ant8.setEuler([90,0,0])
175 ant8.setAnimationSpeed(0,ANIMATIONSPEED)
176 ant8b = childAnt(ant8)
177
178 ant9 = ant.copy()
179 ant9.setScale(SCALE)
180 ant9.runAction(normal)
181 ant9.idlepose(2)
182 ant9.disable(viz.RENDERING)
183 ant9.setPosition(ant9StartPos[0], -ant9StartPos[2]+OFFSET1,
        ant9StartPos[1])
184 ant9.setEuler([100,0,0])
185 ant9.setAnimationSpeed(0,ANIMATIONSPEED)
186 ant9b = childAnt(ant9)
187
188 ##############################################
189 #Extra ants scene 5
190
191 ant10 = ant.copy()
192 ant10.setScale(SCALE)
193 ant10.runAction(normal)
194 ant10.idlepose(2)
195 ant10.disable(viz.RENDERING)
196 ant10.setPosition(3.985, OFFSET1, 8.031)
197 ant10.setEuler([0,0,0])
198 ant10.setAnimationSpeed(0,ANIMATIONSPEED)
199 childAnt(ant10)
200
201 ##############################################
202
203 #Setting up movement path actor 1 scene 1
204 CP_ACTOR_1 = []
205
206 CP_COUNT_ACTOR_1 = 23
207 CONTROLPOINT_ACTOR_1_PREFEX = 'path.actor1.'  #name of boxes = layera1
        ... layera17
208 CONTROLPOINT_ACTOR_1_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
209
210 for cpNumber in range( 0,CP_COUNT_ACTOR_1):
```

```
211     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_1_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_1_SUFFEX)
212     pointNode = movementPaths.getChild(nodename)
213     nodebox = pointNode.getBoundingBox()
214     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
215     print 'path.actor.1',cpNumber, cntr
216     CP_ACTOR_1.append(cntr)
217
218
219 #Setting up movement path actor 2 scene 1
220 CP_ACTOR_2 = []
221
222 CP_COUNT_ACTOR_2 = 19
223 CONTROLPOINT_ACTOR_2_PREFEX = 'path.actor2.'  #name of boxes = layera1
        ... layera17
224 CONTROLPOINT_ACTOR_2_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
225
226 for cpNumber in range( 0, CP_COUNT_ACTOR_2):
227     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_2_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_2_SUFFEX)
228     pointNode = movementPaths.getChild(nodename)
229     nodebox = pointNode.getBoundingBox()
230     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
231     print 'path.actor.2',cpNumber, cntr
232     CP_ACTOR_2.append(cntr)
233
234 #Setting up movement path actor 3 scene 1
235 CP_ACTOR_3 = []
236
237 CP_COUNT_ACTOR_3 = 18
238 CONTROLPOINT_ACTOR_3_PREFEX = 'path.actor3.'  #name of boxes = layera1
        ... layera17
239 CONTROLPOINT_ACTOR_3_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
240
241 for cpNumber in range( 0,CP_COUNT_ACTOR_3):
242     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_3_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_3_SUFFEX)
243     pointNode = movementPaths.getChild(nodename)
244     nodebox = pointNode.getBoundingBox()
245     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
246     print 'path.actor.3',cpNumber, cntr
247     CP_ACTOR_3.append(cntr)
248
249
250 #Setting up movement path actor 4 scene 1
251 CP_ACTOR_4 = []
252
253 CP_COUNT_ACTOR_4 = 11
254 CONTROLPOINT_ACTOR_4_PREFEX = 'path.actor4.'  #name of boxes = layera1
        ... layera17
255 CONTROLPOINT_ACTOR_4_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
256
```

```
257 for cpNumber in range( 0,CP_COUNT_ACTOR_4):
258     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_4_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_4_SUFFEX)
259     pointNode = movementPaths.getChild(nodename)
260     nodebox = pointNode.getBoundingBox()
261     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
262     print 'path.actor.4',cpNumber, cntr
263     CP_ACTOR_4.append(cntr)
264
265 #Setting up movement path actor 5 scene 1
266 CP_ACTOR_5 = []
267
268 CP_COUNT_ACTOR_5 = 16
269 CONTROLPOINT_ACTOR_5_PREFEX = 'path.actor5.'  #name of boxes = layera1
        ... layera17
270 CONTROLPOINT_ACTOR_5_SUFFEX = ''  #use '_Cube' for .obj and '' for.osg
271
272 for cpNumber in range( 0,CP_COUNT_ACTOR_5):
273     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_5_PREFEX, cpNumber,
            CONTROLPOINT_ACTOR_5_SUFFEX)
274     pointNode = movementPaths.getChild(nodename)
275     nodebox = pointNode.getBoundingBox()
276     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
277     print 'path.actor.5',cpNumber, cntr
278     CP_ACTOR_5.append(cntr)
279
280 #Setting up foraging path actor in scene 2
281 CP_ACTOR_GENERAL = []
282
283 CP_COUNT_ACTOR_GENERAL = 22
284 CONTROLPOINT_ACTOR_GENERAL_PREFEX = 'general.path.'  #name of boxes =
        layera1 ... layera17
285 CONTROLPOINT_ACTOR_GENERAL_SUFFEX = ''  #use '_Cube' for .obj and ''
        for.osg
286
287 for cpNumber in range( 0,CP_COUNT_ACTOR_GENERAL):
288     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_GENERAL_PREFEX, cpNumber
            ,CONTROLPOINT_ACTOR_GENERAL_SUFFEX)
289     pointNode = movementPaths.getChild(nodename)
290     nodebox = pointNode.getBoundingBox()
291     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
            this is the xyz setting ONLY for .OSG file
292     print 'general.path.',cpNumber, cntr
293     CP_ACTOR_GENERAL.append(cntr)
294
295 #Setting up homebound path actor in scene 2
296 CP_ACTOR_RETURN = []
297
298 CP_COUNT_ACTOR_RETURN = 6
299 CONTROLPOINT_ACTOR_RETURN_PREFEX = 'return.path.'  #name of boxes =
        layera1 ... layera17
300 CONTROLPOINT_ACTOR_RETURN_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
```

```
301
302 for cpNumber in range( 0,CP_COUNT_ACTOR_RETURN):
303     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_RETURN_PREFEX, cpNumber,
        CONTROLPOINT_ACTOR_RETURN_SUFFEX)
304     pointNode = movementPaths.getChild(nodename)
305     nodebox = pointNode.getBoundingBox()
306     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
        this is the xyz setting ONLY for .OSG file
307     print 'return.path.',cpNumber, cntr
308     CP_ACTOR_RETURN.append(cntr)
309
310 #Setting up foraging path of normal condition in scene 4
311 CP_ACTOR_CONTROL = []
312
313 CP_COUNT_ACTOR_CONTROL = 13
314 CONTROLPOINT_ACTOR_CONTROL_PREFEX = 'control.path.'  #name of boxes =
    layera1 ... layera17
315 CONTROLPOINT_ACTOR_CONTROL_SUFFEX = ''  #use '_Cube' for .obj and ''
    for.osg
316
317 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL):
318     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_PREFEX, cpNumber
        ,CONTROLPOINT_ACTOR_CONTROL_SUFFEX)
319     pointNode = movementPaths.getChild(nodename)
320     nodebox = pointNode.getBoundingBox()
321     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
        this is the xyz setting ONLY for .OSG file
322     print 'control.path.',cpNumber, cntr
323     CP_ACTOR_CONTROL.append(cntr)
324
325 #Setting up foraging path of stilts condition in scene 4
326 CP_ACTOR_STILTS = []
327
328 CP_COUNT_ACTOR_STILTS = 13
329 CONTROLPOINT_ACTOR_STILTS_PREFEX = 'stilts.path.'  #name of boxes =
    layera1 ... layera17
330 CONTROLPOINT_ACTOR_STILTS_SUFFEX = ''  #use '_Cube' for .obj and '' for
    .osg
331
332 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS):
333     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_PREFEX, cpNumber,
        CONTROLPOINT_ACTOR_STILTS_SUFFEX)
334     pointNode = movementPaths.getChild(nodename)
335     nodebox = pointNode.getBoundingBox()
336     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
        this is the xyz setting ONLY for .OSG file
337     print 'stilts.path.',cpNumber, cntr
338     CP_ACTOR_STILTS.append(cntr)
339
340
341 #Setting up foraging path of stumps condition in scene 4
342 CP_ACTOR_STUMPS = []
343
344 CP_COUNT_ACTOR_STUMPS = 14
```

```
345 CONTROLPOINT_ACTOR_STUMPS_PREFEX = 'stumps.path.'  #name of boxes =
        layera1 ... layera17
346 CONTROLPOINT_ACTOR_STUMPS_SUFFEX = ''  #use '_Cube' for .obj and '' for
        .osg
347
348 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS):
349     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_PREFEX, cpNumber,
          CONTROLPOINT_ACTOR_STUMPS_SUFFEX)
350     pointNode = movementPaths.getChild(nodename)
351     nodebox = pointNode.getBoundingBox()
352     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
353     print 'stumps.path.',cpNumber, cntr
354     CP_ACTOR_STUMPS.append(cntr)
355
356
357
358 #Setting up homebound path of normal condition in scene 4
359 CP_ACTOR_CONTROL_RETURN = []
360
361 CP_COUNT_ACTOR_CONTROL_RETURN = 29
362 CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX = 'control.Rpath.'  #name of
        boxes = layera1 ... layera17
363 CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX = ''  #use '_Cube' for .obj
        and '' for.osg
364
365 for cpNumber in range( 0,CP_COUNT_ACTOR_CONTROL_RETURN):
366     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_CONTROL_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_CONTROL_RETURN_SUFFEX)
367     pointNode = movementPaths.getChild(nodename)
368     nodebox = pointNode.getBoundingBox()
369     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
370     print 'control.Rpath.',cpNumber, cntr
371     CP_ACTOR_CONTROL_RETURN.append(cntr)
372
373
374 #Setting up homebound path of stilts condition in scene 4
375 CP_ACTOR_STILTS_RETURN = []
376
377 CP_COUNT_ACTOR_STILTS_RETURN = 51
378 CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX = 'stilts.Rpath.'  #name of
        boxes = layera1 ... layera17
379 CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
        '' for.osg
380
381 for cpNumber in range( 0,CP_COUNT_ACTOR_STILTS_RETURN):
382     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STILTS_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_STILTS_RETURN_SUFFEX)
383     pointNode = movementPaths.getChild(nodename)
384     nodebox = pointNode.getBoundingBox()
385     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
386     print 'stilts.Rpath.',cpNumber, cntr
387     CP_ACTOR_STILTS_RETURN.append(cntr)
```

```
388
389
390 #Setting up homebound path of stumps condition in scene 4
391 CP_ACTOR_STUMPS_RETURN = []
392
393 CP_COUNT_ACTOR_STUMPS_RETURN = 37
394 CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX = 'stumps.Rpath.'  #name of
      boxes = layera1 ... layera17
395 CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX = ''  #use '_Cube' for .obj and
      '' for .osg
396
397 for cpNumber in range( 0,CP_COUNT_ACTOR_STUMPS_RETURN):
398     nodename = '%s%03d%s' % (CONTROLPOINT_ACTOR_STUMPS_RETURN_PREFEX,
          cpNumber,CONTROLPOINT_ACTOR_STUMPS_RETURN_SUFFEX)
399     pointNode = movementPaths.getChild(nodename)
400     nodebox = pointNode.getBoundingBox()
401     cntr = [nodebox.center[0],-nodebox.center[2],nodebox.center[1]]  #
          this is the xyz setting ONLY for .OSG file
402     print 'stumps.Rpath.',cpNumber, cntr
403     CP_ACTOR_STUMPS_RETURN.append(cntr)
404
405
406
407
408
409 #Actors Activation
      ############################################################
410 def actorsScene1Enable():
411     ant1.enable(viz.RENDERING)
412     ant2.enable(viz.RENDERING)
413     ant3.enable(viz.RENDERING)
414     ant4.enable(viz.RENDERING)
415     ant5.enable(viz.RENDERING)
416
417 def actorsScene1Action():
418     for each in CP_ACTOR_1:
419         print "ant1.addAction(vizact.walkto(",each,"))"
420         ant1.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 80, 1))
421
422     for each in CP_ACTOR_2:
423         print "ant2.addAction(vizact.walkto(",each,"))"
424         ant2.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.28, 70, 1))
425
426     for each in CP_ACTOR_3:
427         print "ant3.addAction(vizact.walkto(",each,"))"
428         ant3.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.29, 90, 1))
429
430     for each in CP_ACTOR_4:
431         print "ant4.addAction(vizact.walkto(",each,"))"
432         ant4.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
              0.30, 75, 1))
433
```

```
434     for each in CP_ACTOR_5:
435         print "ant5.addAction(vizact.walkto(",each,"))"
436         ant5.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
                0.28, 80, 1))
437
438 def actorsScene1Disable():
439     ant1.enable(viz.RENDERING)
440     ant2.enable(viz.RENDERING)
441     ant3.enable(viz.RENDERING)
442     ant4.enable(viz.RENDERING)
443     ant5.enable(viz.RENDERING)
444
445
446 def actorsScene2Enable():
447     ant6.enable(viz.RENDERING)
448
449 def actorsScene2Action():
450     for each in CP_ACTOR_GENERAL:
451         print "ant6.addAction(vizact.walkto(",each,"))"
452         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
                0.292, 110, 1))
453
454 def actorsScene2Return():
455     ant6.addAction(vizact.spinto(0,1,0,200,0.15,viz.TIME))
456     for each in CP_ACTOR_RETURN:
457         print "ant6.addAction(vizact.walkto(",each,"))"
458         ant6.addAction(vizact.walkto(each[0], each[1]+OFFSET1, each[2],
                0.29, 35, 4))
459
460 def actorsScene2Disable():
461     #ant6.disable(viz.RENDERING)
462     ant6.add(vizact.fade(1,0,2))
463
464
465 def actorsScene4Enable():
466     ant7.enable(viz.RENDERING)
467     ant8.enable(viz.RENDERING)
468     ant9.enable(viz.RENDERING)
469
470 def controlAction():
471     for each in CP_ACTOR_CONTROL:
472         print "ant7.addAction(vizact.walkto(",each,"))"
473         ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
474
475 def stiltsAction():
476     for each in CP_ACTOR_STILTS:
477         print "ant8.addAction(vizact.walkto(",each,"))"
478         ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
479
480 def stumpsAction():
481     for each in CP_ACTOR_STUMPS:
482         print "ant9.addAction(vizact.walkto(",each,"))"
```

```
483         ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                each[2], 0.3, 35, 1))
484
485 def controlReturn():
486     prevVertAnt7 = ant7.getPosition()
487     vertexListAnt7.append(ant7.getPosition())
488
489     count = 0
490     for each in CP_ACTOR_CONTROL_RETURN:
491         count = count + 1
492         if count < 14:
493             print "ant7.addAction(vizact.walkto(",each,"))"
494             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                    each[2], 0.30, 35, 4))
495         else:
496             print "ant7.addAction(vizact.walkto(",each,"))"
497             ant7.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2,
                    each[2], 0.24, 900, 4))
498
499 def stiltsReturn():
500     prevVertAnt8 = ant8.getPosition()
501     vertexListAnt8.append(ant8.getPosition())
502
503     count = 0
504     for each in CP_ACTOR_STILTS_RETURN:
505         count = count + 1
506         if count < 20:
507             print "ant8.addAction(vizact.walkto(",each,"))"
508             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                    STILTS_OFFSET, each[2], 0.29, 35, 4))
509
510         else:
511             print "ant8.addAction(vizact.walkto(",each,"))"
512             ant8.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2+
                    STILTS_OFFSET, each[2], 0.24, 900, 4))
513
514 def stumpsReturn():
515     prevVertAnt9 = ant9.getPosition()
516     vertexListAnt9.append(ant9.getPosition())
517
518     count = 0
519     for each in CP_ACTOR_STUMPS_RETURN:
520         count = count + 1
521         if count < 9:
522             print "ant9.addAction(vizact.walkto(",each,"))"
523             ant9.addAction(vizact.walkto(each[0], each[1]+OFFSET1+OFFSET2-
                    STUMPS_OFFSET, each[2], 0.285, 35, 4))
524
525         else:
526             print "ant9.addAction(vizact.walkto(",each,"))"
527             ant9.addAction(vizact.walkto(each[0], each[1]++OFFSET1+OFFSET2
                    -STUMPS_OFFSET, each[2], 0.24, 900, 4))
528
529 def pickupAnts():
530     ant7.addAction(vizact.goto(9.478, 0.202, 0.3,0.05))
```

```
531     ant7.idlepose(3)
532     ant8.addAction(vizact.goto(9.437, 0.202,0.325,0.05))
533     ant8.idlepose(3)
534     ant9.addAction(vizact.goto(9.46,0.202,0.262,0.05))
535     ant9.idlepose(3)
536
537 def distributeAnts():
538     ant9.addAction(vizact.goto(9.453,0.202,-0.4,0.09))
539     ant8.addAction(vizact.goto(9.453,0.202,-0.2,0.07))
540     ant7.addAction(vizact.goto(9.453,0.202,0,0.05))
541
542 def applyStilts():
543     ant8.idlepose(3)
544     ant8b.idlepose(3)
545     ant8b.runAction(stilts)
546     ant8.idlepose(3)
547     ant8b.idlepose(3)
548
549 def applyStumps():
550     ant9.idlepose(3)
551     ant9b.idlepose(3)
552     ant9b.runAction(stumps)
553     ant9.idlepose(3)
554     ant9b.idlepose(3)
555
556 def dropAnts():
557     global prevVertAnt7, prevVertAnt8, prevVertAnt9
558     global vertexListAnt8, vertexListAnt8, vertexListAnt9
559
560     ant7.addAction(vizact.goto(9.453, 0.008,0,0.2))
561     ant7.addAction(vizact.spinto(0,1,0,270,0.68,viz.TIME))
562     ant7.idlepose(2)
563
564
565     ant8.addAction(vizact.goto(9.453,0.008,-0.2,0.2))
566     ant8.addAction(vizact.spinto(0,1,0,265,0.8,viz.TIME))
567     ant8.idlepose(2)
568
569
570     ant9.addAction(vizact.goto(9.453,0.008,-0.4,0.2))
571     ant9.addAction(vizact.spinto(0,1,0,272,0.58,viz.TIME))
572     ant9.idlepose(2)
573
574     FINALSCENE = 1
575
576 def actorsScene4Disable():
577     ant7.add(vizact.fade(1,0,2))
578     ant8.add(vizact.fade(1,0,2))
579     ant9.add(vizact.fade(1,0,2))
580
581 def extraAntEnable():
582     ant10.enable(viz.RENDERING)
583
584 def extraAntGo():
585     ant10.addAction(vizact.walkto(3.985, OFFSET1, 8.031, 0.295, 45, 1))
```

```
586     ant10.addAction(vizact.walkto(0.04, OFFSET1, 6.55, 0.295, 45, 1))
587     ant10.addAction(vizact.walkto(-31, OFFSET1, 19.969, 0.295, 45, 1))
588
589 bread = viz.add('bread/bread.osg')
590 bread.setScale(0.0045,0.0045,0.0045)
591 bread.alpha(0)
592
593 bread6 = bread.copy()
594 bread6.setScale(0.0045,0.0045,0.0045)
595 bread6.alpha(0)
596
597 bread7 = bread.copy()
598 bread7.setScale(0.0045,0.0045,0.0045)
599 bread7.alpha(0)
600
601 bread8 = bread.copy()
602 bread8.setScale(0.0045,0.0045,0.0045)
603 bread8.alpha(0)
604
605 bread9 = bread.copy()
606 bread9.setScale(0.0045,0.0045,0.0045)
607 bread9.alpha(0)
608
609 def giveFoodScene2():
610     bread6.add(vizact.fade(0,1,2))
611     ant6mount = ant6b.getBone('mount0')
612     ant6mountLink = viz.link(ant6mount,bread6)
613     ant6mountLink.preTrans([0.0,-0.0002,-0.002])
614
615 def removeFoodScene2():
616     bread6.add(vizact.fade(1,0,2))
617
618 def giveFoodControl():
619     bread7.add(vizact.fade(0,1,2))
620     ant7mount = ant7b.getBone('mount0')
621     ant7mountLink = viz.link(ant7mount,bread7)
622     ant7mountLink.preTrans([0.0,-0.0002,-0.002])
623
624 def giveFoodStilts():
625     bread8.add(vizact.fade(0,1,2))
626     ant8mount = ant8b.getBone('mount0')
627     ant8mountLink = viz.link(ant8mount,bread8)
628     ant8mountLink.preTrans([0.0,-0.0002,-0.002])
629
630 def giveFoodStumps():
631     bread9.add(vizact.fade(0,1,2))
632     ant9mount = ant9b.getBone('mount0')
633     ant9mountLink = viz.link(ant9mount,bread9)
634     ant9mountLink.preTrans([0.0,-0.0002,-0.002])
635
636 def removeFoodControl():
637     bread7.add(vizact.fade(1,0,2))
638
639 def removeFoodStilts():
640     bread8.add(vizact.fade(1,0,2))
```

```
641
642 def removeFoodStumps():
643     bread9.add(vizact.fade(1,0,2))
644
645 def animationMultiplier(num):
646     global FINALSCENE
647
648     if ant1.getCycleRunning(1) or ant1.getCycleRunning(4):
649         ant1.setAnimationSpeed(0,7.7)
650     else:
651         ant1.setAnimationSpeed(0,1)
652
653     if ant10.getCycleRunning(1) or ant10.getCycleRunning(4):
654         ant10.setAnimationSpeed(0,7.7)
655     else:
656         ant10.setAnimationSpeed(0,1)
657
658     if ant2.getCycleRunning(1) or ant2.getCycleRunning(4):
659         ant2.setAnimationSpeed(0,7.7)
660     else:
661         ant2.setAnimationSpeed(0,1)
662
663     if ant3.getCycleRunning(1) or ant3.getCycleRunning(4):
664         ant3.setAnimationSpeed(0,7.7)
665     else:
666         ant3.setAnimationSpeed(0,1)
667
668     if ant4.getCycleRunning(1) or ant4.getCycleRunning(4):
669         ant4.setAnimationSpeed(0,7.7)
670     else:
671         ant4.setAnimationSpeed(0,1)
672
673     if ant5.getCycleRunning(1) or ant5.getCycleRunning(4):
674         ant5.setAnimationSpeed(0,7.7)
675     else:
676         ant5.setAnimationSpeed(0,1)
677
678     if ant6.getCycleRunning(1) or ant6.getCycleRunning(4):
679         ant6.setAnimationSpeed(0,7.7)
680         drawFreehandPreviewAnt6()
681     else:
682         ant6.setAnimationSpeed(0,1.5)
683
684
685     if (ant7.getCycleRunning(1) or ant7.getCycleRunning(4)) and
        FINALSCENE == 1:
686         ant7.setAnimationSpeed(0,7.7)
687         drawFreehandPreviewAnt7()
688     elif ant7.getCycleRunning(1) or ant7.getCycleRunning(4):
689         ant7.setAnimationSpeed(0,7.7)
690     else:
691         ant7.setAnimationSpeed(0,1.35)
692
693     if (ant8.getCycleRunning(1) or ant8.getCycleRunning(4)) and
        FINALSCENE == 1:
```

```
694         ant8.setAnimationSpeed(0,7.7)
695         drawFreehandPreviewAnt8()
696     elif ant8.getCycleRunning(1) or ant8.getCycleRunning(4):
697         ant8.setAnimationSpeed(0,7.7)
698     else:
699         ant8.setAnimationSpeed(0,1.5)
700 
701     if (ant9.getCycleRunning(1) or ant9.getCycleRunning(4)) and
        FINALSCENE == 1:
702         ant9.setAnimationSpeed(0,7.7)
703         drawFreehandPreviewAnt9()
704     elif ant9.getCycleRunning(1) or ant9.getCycleRunning(4):
705         ant9.setAnimationSpeed(0,7.7)
706     else:
707         ant9.setAnimationSpeed(0,1.25)
708 
709     syncAnts()
710 
711 
712 
713 vertexListAnt6 = []
714 vertexListAnt7 = []
715 vertexListAnt8 = []
716 vertexListAnt9 = []
717 
718 prevVertAnt7 = [0,0,0]
719 prevVertAnt8 = [0,0,0]
720 prevVertAnt9 = [0,0,0]
721 
722 #viz.startlayer(viz.LINES)
723 viz.startlayer(viz.POINTS)
724 previewLine = viz.endlayer()
725 
726 prevVertAnt6 = ant6.getPosition()
727 vertexListAnt6.append(ant6.getPosition())
728 
729 #temp segments that will get discarded each frame
730 ant6segment = 0
731 ant7segment = 0
732 ant8segment = 0
733 ant9segment = 0
734 
735 def drawFreehandPreviewAnt6():
736     global previewLine, vertexListAnt6
737     global ant6segment
738 
739     prevVertAnt6 = vertexListAnt6[-1]
740     nextVertAnt6 = ant6.getPosition()
741     # We only want to draw a line segment if it is visible
742     if vizmat.Distance(prevVertAnt6, nextVertAnt6) > 0.040:
743         viz.startlayer(viz.LINES)
744         viz.linewidth(2)
745         viz.vertexcolor(1,0,0)
746         viz.vertex(prevVertAnt6)
747         viz.vertex(nextVertAnt6)
```

```
748         segment = viz.endlayer()
749         vertexListAnt6.append(nextVertAnt6)
750         # Attaching the segment to the preview, so that we may be able to
                access
751         # it later for deletion
752         segment.parent(previewLine)
753     else:
754         if ant6segment:
755             ant6segment.remove()
756         viz.startlayer(viz.LINES)
757         viz.linewidth(2)
758         viz.vertexcolor(1,0,0)
759         viz.vertex(prevVertAnt6)
760         viz.vertex(nextVertAnt6)
761         ant6segment = viz.endlayer()
762
763 def drawFreehandPreviewAnt7():
764     global previewLine, vertexListAnt7
765     global ant7segment
766
767     prevVertAnt7 = vertexListAnt7[-1]
768     nextVertAnt7 = ant7.getPosition()
769     # We only want to draw a line segment if it is visible
770     if vizmat.Distance(prevVertAnt7, nextVertAnt7) > 0.04:
771         viz.startlayer(viz.LINES)
772         viz.linewidth(2)
773         viz.vertexcolor(1,0,0)
774         viz.vertex(prevVertAnt7)
775         viz.vertex(nextVertAnt7)
776         segment = viz.endlayer()
777         vertexListAnt7.append(nextVertAnt7)
778         # Attaching the segment to the preview, so that we may be able to
                access
779         # it later for deletion
780         segment.parent(previewLine)
781     else:
782         if ant7segment:
783             ant7segment.remove()
784         viz.startlayer(viz.LINES)
785         viz.linewidth(2)
786         viz.vertexcolor(1,0,0)
787         viz.vertex(prevVertAnt7)
788         viz.vertex(nextVertAnt7)
789         ant7segment = viz.endlayer()
790
791
792 def drawFreehandPreviewAnt8():
793     global previewLine, vertexListAnt8
794     global ant8segment
795
796     prevVertAnt8 = vertexListAnt8[-1]
797     nextVertAnt8 = ant8.getPosition()
798     # We only want to draw a line segment if it is visible
799     if vizmat.Distance(prevVertAnt8, nextVertAnt8) > 0.04:
800         viz.startlayer(viz.LINES)
```

```
801         viz.linewidth(2)
802         viz.vertexcolor(1,0,0)
803         viz.vertex(prevVertAnt8)
804         viz.vertex(nextVertAnt8)
805         segment = viz.endlayer()
806         vertexListAnt8.append(nextVertAnt8)
807         # Attaching the segment to the preview, so that we may be able to
                access
808         # it later for deletion
809         segment.parent(previewLine)
810     else:
811         if ant8segment:
812             ant8segment.remove()
813         viz.startlayer(viz.LINES)
814         viz.linewidth(2)
815         viz.vertexcolor(1,0,0)
816         viz.vertex(prevVertAnt8)
817         viz.vertex(nextVertAnt8)
818         ant8segment = viz.endlayer()
819
820 def drawFreehandPreviewAnt9():
821     global previewLine, vertexListAnt9
822     global ant9segment
823
824     prevVertAnt9 = vertexListAnt9[-1]
825     nextVertAnt9 = ant9.getPosition()
826     # We only want to draw a line segment if it is visible
827     if vizmat.Distance(prevVertAnt9, nextVertAnt9) > 0.04:
828         viz.startlayer(viz.LINES)
829         viz.linewidth(2)
830         viz.vertexcolor(1,0,0)
831         viz.vertex(prevVertAnt9)
832         viz.vertex(nextVertAnt9)
833         segment = viz.endlayer()
834         vertexListAnt9.append(nextVertAnt9)
835         # Attaching the segment to the preview, so that we may be able to
                access
836         # it later for deletion
837         segment.parent(previewLine)
838     else:
839         if ant9segment:
840             ant9segment.remove()
841         viz.startlayer(viz.LINES)
842         viz.linewidth(2)
843         viz.vertexcolor(1,0,0)
844         viz.vertex(prevVertAnt9)
845         viz.vertex(nextVertAnt9)
846         ant9segment = viz.endlayer()
847
848 viz.callback(viz.TIMER_EVENT, animationMultiplier)
849 viz.starttimer(0,0.01,viz.FOREVER)
```

Appendix G

R codes

This chapter lists the code used for the statistical analyses in R (Team, 2010). The statistical analysis for Simulator Sickness (SS) was separated from the learning outcomes analysis as this was the first priority in assessing the suitability of the simulations for future use. If an educational or a training simulation causes a severe SS, then the benefit would be greatly diminished or even none. The statistical analysis for VEAR used the same the code from the study with ANTECATALYST, only differing in input files, output files and comments that describe the results.

G.1 R codes of ANTECATALYST

G.1.1 Codes for analysing Simulator Sickness

Listing: AVAvsSS[2010-08-01][0023].r

```
1  #Clearing workspace (and variables too)################################
2  rm(list=ls())
3
4  #Setting up the working directory#######################################
5  setwd("/home/ephraim/e1_thesis/results/caseStudy1")
6
7  #loading required library###############################################
8  #library(Rcmdr)
9  library(psychometric)
10 library(psych)
```

```
11 library(Cairo)
12 library(lattice)
13 library(rgl)
14 library(mgcv)
15 library(pwr)
16 require(gplots)
17 library(Hmisc)
18 library(xtable)
19
20 #nonparametic analysis packages
21 #library(npmc)
22 #library(vegan) #use adonis(...) for nonparametric manova
23
24 # Loading the results###################################################
25 results <- read.table("resultsFromExperiment1-Odometric behaviour of
       Cataglyphis Ants [2009-03-14][1238].csv",
26     header=TRUE,
27     sep="\t",
28     na.strings="NA",
29     dec=".",
30     strip.white=TRUE)
31 #removing cases with NA entries into a new dataframe
32 results <- na.omit(results)
33
34 #Creating a dataframe
35 results.df <- data.frame(results)
36 attach(results.df)
37 names(results.df)
38
39 ######################################################################
40 #Research questions#####################################################
41 ######################################################################
42
43 ###Does additional visual information of virtual actors affect
       simulator sickness?
44 AVAvsSS <- data.frame(subset(results.df,
45     Hw.failure == "N",
46     select=c(Conditions,
47         Gender,
48         Age,
49         Participant,
50         PreD.Score,
51         PreN.Score,
52         PreO.Score,
53         PreSSQ.Score,
54         PostD.Score,
55         PostN.Score,
56         PostO.Score,
57         PostSSQ.Score,
58         D.Incr,
59         N.Incr,
60         O.Incr,
61         SSQ.Incr
62         )
63     )
```

```
64      )
65
66  AVAvsSS_A <- subset(AVAvsSS, Conditions == "A")
67  AVAvsSS_B <- subset(AVAvsSS, Conditions == "B")
68  AVAvsSS_C <- subset(AVAvsSS, Conditions == "C")
69  AVAvsSS_M <- subset(AVAvsSS, Gender == "m")
70  AVAvsSS_F <- subset(AVAvsSS, Gender == "f")
71
72  ########################################################################
73  #Displaying the summary of the dataframe
74  ########################################################################
75  #Summary in overall
76  summary(AVAvsSS)
77  #Summary by groups
78  summary(subset(AVAvsSS, Conditions == 'A', digits = 2))
79  summary(subset(AVAvsSS, Conditions == 'B', digits = 2))
80  summary(subset(AVAvsSS, Conditions == 'C', digits = 2))
81
82  ##Describing the data
83  describe(AVAvsSS,
84      digits = 2,
85      na.rm = TRUE,
86      interp=FALSE,
87      skew = TRUE,
88      ranges = TRUE,
89      trim=.1
90      )
91  ##Describing the data by groups
92  describe.by(AVAvsSS, AVAvsSS$Conditions,
93      digits = 2,
94      na.rm = TRUE,
95      interp=FALSE,
96      skew = TRUE,
97      ranges = TRUE,
98      trim=.1
99      )
100
101 ########################################################################
102 #Plotting the graphs and saving them on file
103 ########################################################################
104 ###Boxplot of SSQ
105 ##Save it on file for publication...
106 Cairo(2560, 1600, file="AVAvsSS_boxplots.pdf", type="pdf", bg="white",
        dpi = 200, units="px")
107 par(mfrow = c(3, 4))
108 boxplot(PreN.Score ~ Conditions, data = AVAvsSS,
109         main="Pre-SSQ: Nausea",
110         xlab="Experimental Conditions",
111         ylab="Level of severity"
112     )
113
114 boxplot(PreD.Score ~ Conditions, data = AVAvsSS,
115     main="Pre-SSQ: Disorientation",
116         xlab="Experimental Conditions",
117         ylab="Level of severity"
```

```
118     )
119
120 boxplot(PreO.Score ~ Conditions, data = AVAvsSS,
121     main="Pre-SSQ: Occulomotor",
122           xlab="Experimental Conditions",
123           ylab="Level of severity"
124     )
125
126 boxplot(PreSSQ.Score ~ Conditions, data = AVAvsSS,
127     main="Pre-SSQ: Total severity score",
128           xlab="Experimental Conditions",
129           ylab="Level of severity"
130     )
131
132 boxplot(PostN.Score ~ Conditions, data = AVAvsSS,
133     main="Post-SSQ: Nausea",
134           xlab="Experimental Conditions",
135           ylab="Level of severity"
136     )
137
138 boxplot(PostD.Score ~ Conditions, data = AVAvsSS,
139     main="Post-SSQ: Disorientation",
140           xlab="Experimental Conditions",
141           ylab="Level of severity"
142     )
143
144 boxplot(PostO.Score ~ Conditions, data = AVAvsSS,
145     main="Post-SSQ: Occulomotor",
146           xlab="Experimental Conditions",
147           ylab="Level of severity"
148     )
149
150 boxplot(PostSSQ.Score ~ Conditions, data = AVAvsSS,
151     main="Post-SSQ: Total severity score",
152           xlab="Experimental Conditions",
153           ylab="Level of severity"
154     )
155
156 boxplot(D.Incr ~ Conditions, data = AVAvsSS,
157     main="Increase in Disorientation",
158           xlab="Experimental Condition",
159           ylab="Level of severity"
160     )
161
162 boxplot(N.Incr ~ Conditions, data = AVAvsSS,
163     main="Increase in Nausea",
164           xlab="Experimental Condition",
165           ylab="Level of severity"
166     )
167
168 boxplot(O.Incr ~ Conditions, data = AVAvsSS,
169     main="Increase in Occulomotor",
170           xlab="Experimental Condition",
171           ylab="Level of severity"
172     )
```

```
173
174 boxplot(SSQ.Incr ~ Conditions, data = AVAvsSS,
175    main="Increase in Total Severity Score",
176         xlab="Experimental Condition",
177         ylab="Level of severity"
178    )
179
180 dev.off()
181
182 ###Plotting histogram of each dependant variables
183 ##Save on file...dev.new()
184 Cairo(2560, 1600, file="plot-AVAvsSS-histograms.pdf", type="pdf", bg="
       white", dpi = 200, units="px")
185 par(mfrow = c(3, 4))
186
187 hist(AVAvsSS$PreD.Score,
188    #scale="frequency",
189    breaks="Sturges",
190    col="darkgray",
191         xlab="Level of severity",
192    main ="Pre-SSQ: Disorientation")
193
194
195 hist(AVAvsSS$PreN.Score,
196    #scale="frequency",
197    breaks="Sturges",
198    col="darkgray",
199         xlab="Level of severity",
200    main ="Pre-SSQ: Nausea")
201
202 hist(AVAvsSS$PreO.Score,
203    #scale="frequency",
204    breaks="Sturges",
205    col="darkgray",
206         xlab="Level of severity",
207    main ="Pre-SSQ: Oculomotor")
208
209 hist(AVAvsSS$PreSSQ.Score,
210    #scale="frequency",
211    breaks="Sturges",
212    col="darkgray",
213         xlab="Level of severity",
214    main ="Pre-SSQ: Total severity score")
215
216 hist(AVAvsSS$PostD.Score,
217    #scale="frequency",
218    breaks="Sturges",
219    col="darkgray",
220         xlab="Level of severity",
221    main ="Post-SSQ: Disorientation")
222
223 hist(AVAvsSS$PostN.Score,
224    #scale="frequency",
225    breaks="Sturges",
226    col="darkgray",
```

```
227         xlab="Level of severity",
228    main ="Post-SSQ: Nausea")
229
230 hist(AVAvsSS$PostO.Score,
231    #scale="frequency",
232    breaks="Sturges",
233    col="darkgray",
234         xlab="Level of severity",
235    main ="Post-SSQ: Oculomotor")
236
237 hist(AVAvsSS$PostSSQ.Score,
238    #scale="frequency",
239    breaks="Sturges",
240    col="darkgray",
241         xlab="Level of severity",
242    main ="Post-SSQ: Total severity score")
243
244 hist(AVAvsSS$N.Incr,
245    #scale="frequency",
246    breaks="Sturges",
247    col="darkgray",
248         xlab="Level of severity",
249    main ="Increase in Nausea")
250
251 hist(AVAvsSS$D.Incr,
252    #scale="frequency",
253    breaks="Sturges",
254    col="darkgray",
255         xlab="Level of severity",
256    main ="Increase in Disorientation")
257
258 hist(AVAvsSS$O.Incr,
259    #scale="frequency",
260    breaks="Sturges",
261    col="darkgray",
262         xlab="Level of severity",
263    main ="Increase in Occulomotor")
264
265 hist(AVAvsSS$SSQ.Incr,
266    #scale="frequency",
267    breaks="Sturges",
268    col="darkgray",
269         xlab="Level of severity",
270    main ="Increase in Total Severity Score")
271
272 dev.off()
273
274
275
276 ##########################################################################
277 #Normality and Variance check
278 ##########################################################################
279 ###Normality test###
280 #Shapiro-Wilk Normality Test. When the p-value is higher than
       significance levels
```

```
#the sample forms a gaussian (bell-shaped) distribution.
#Thode(2002) reviewed over forty procedures for assesing goodness-of-fit and concludes
#that Shapiro and Wilk (1965, 1968) and test statistics derived from moments
#are generally the best methods for assessing normality.
#Thode, H. C. (2002). Testing for normality. New York: Marcel Dekker.

shapiro.test(AVAvsSS$PreD.Score) #Not normal

#  Shapiro-Wilk normality test

#data:  AVAvsSS$PreD.Score
#W = 0.6939, p-value < 2.2e-16


shapiro.test(AVAvsSS$PreN.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsSS$PreN.Score
# W = 0.8483, p-value = 3.569e-13


shapiro.test(AVAvsSS$PreO.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsSS$PreO.Score
# W = 0.895, p-value = 1.216e-10


shapiro.test(AVAvsSS$PreSSQ.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsSS$PreSSQ.Score
# W = 0.8499, p-value = 4.272e-13


shapiro.test(AVAvsSS$PostD.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsSS$PostD.Score
# W = 0.6908, p-value < 2.2e-16


shapiro.test(AVAvsSS$PostN.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsSS$PostN.Score
# W = 0.8043, p-value = 4.137e-15

```

```
334
335 shapiro.test(AVAvsSS$PostO.Score) #Not normal
336
337     # Shapiro-Wilk normality test
338
339 # data:  AVAvsSS$PostO.Score
340 # W = 0.8636, p-value = 2.071e-12
341
342
343 shapiro.test(AVAvsSS$PostSSQ.Score) #Not normal
344
345     # Shapiro-Wilk normality test
346
347 # data:  AVAvsSS$PostSSQ.Score
348 # W = 0.8435, p-value = 2.093e-13
349
350
351 ###Checking wether dependent variables have the same variances between
        groups##
352 ###If Bartlett test's p-value is less than 0.05
353 ###then the assumption of equal variances would be rejected.
354 tapply(AVAvsSS$PreD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
355         # A          B          C
356  # 978.4656  626.9380 1596.3544
357
358 bartlett.test(PreD.Score ~ Conditions, data=AVAvsSS)  #not equal
359
360     # Bartlett test of homogeneity of variances
361
362 # data:  PreD.Score by Conditions
363 # Bartlett's K-squared = 14.4709, df = 2, p-value = 0.0007206
364
365
366 tapply(AVAvsSS$PreN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
367         # A          B          C
368 # 652.8925  332.2180  599.7870
369
370 bartlett.test(PreN.Score ~ Conditions, data=AVAvsSS) #not equal
371
372     # Bartlett test of homogeneity of variances
373
374 # data:  PreN.Score by Conditions
375 # Bartlett's K-squared = 7.7517, df = 2, p-value = 0.02074
376
377
378 tapply(AVAvsSS$PreO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
379         # A          B          C
380 # 654.9867  343.1792  549.0955
381
382 bartlett.test(PreO.Score ~ Conditions, data=AVAvsSS)  #not equal
383
384     # Bartlett test of homogeneity of variances
385
386 # data:  PreO.Score by Conditions
387 # Bartlett's K-squared = 6.4473, df = 2, p-value = 0.03981
```

```
388
389
390 tapply(AVAvsSS$PreSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
391         # A         B         C
392 # 837.0275  418.2744  875.5540
393
394 bartlett.test(PreSSQ.Score ~ Conditions, data=AVAvsSS) #not equal
395
396    # Bartlett test of homogeneity of variances
397
398 # data:  PreSSQ.Score by Conditions
399 # Bartlett's K-squared = 9.8809, df = 2, p-value = 0.007151
400
401
402 tapply(AVAvsSS$PostD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
403         # A         B         C
404 # 539.6996  470.6748  899.4973
405
406 bartlett.test(PostD.Score ~ Conditions, data=AVAvsSS) #not equal
407
408    # Bartlett test of homogeneity of variances
409
410 # data:  PostD.Score by Conditions
411 # Bartlett's K-squared = 8.2837, df = 2, p-value = 0.01589
412
413
414 tapply(AVAvsSS$PostN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
415         # A         B         C
416 # 229.2001  408.6901  383.3623
417
418 bartlett.test(PostN.Score ~ Conditions, data=AVAvsSS) #equal
419
420    # Bartlett test of homogeneity of variances
421
422 # data:  PostN.Score by Conditions
423 # Bartlett's K-squared = 5.6655, df = 2, p-value = 0.05885
424
425
426 tapply(AVAvsSS$PostO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
427         # A         B         C
428 # 458.2716  385.8081  544.9325
429
430 bartlett.test(PostO.Score ~ Conditions, data=AVAvsSS) #equal
431
432    # Bartlett test of homogeneity of variances
433
434 # data:  PostO.Score by Conditions
435 # Bartlett's K-squared = 2.0106, df = 2, p-value = 0.3659
436
437
438 tapply(AVAvsSS$PostSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
439         # A         B         C
440 # 411.3303  456.2264  621.1001
441
442 bartlett.test(PostSSQ.Score ~ Conditions, data=AVAvsSS) #equal
```

```
443
444     # Bartlett test of homogeneity of variances
445
446 # data:  PostSSQ.Score by Conditions
447 # Bartlett's K-squared = 3.1931, df = 2, p-value = 0.2026
448
449
450 ###Checking wether dependent variables have the same variances between
        groups##
451 ###If the resulting p-value of Levene's test is less than some critical
         value (typically .05),
452 ###the obtained differences in sample variances are unlikely to have
        occurred based on random sampling.
453 ###Thus, the null hypothesis of equal variances is rejected and it is
        concluded that there is a difference
454 ###between the variances in the population. --Wikipedia
455
456 tapply(AVAvsSS$PreD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
457         # A         B         C
458  # 978.4656  626.9380 1596.3544
459
460 levene.test(PreD.Score ~ Conditions, data=AVAvsSS)  #equal
461 # Levene's Test for Homogeneity of Variance (center = median)
462         # Df F value Pr(>F)
463 # group   2  0.6534 0.5214
464       # 197
465
466 tapply(AVAvsSS$PreN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
467       # A        B        C
468 # 652.8925 332.2180 599.7870
469
470 levene.test(PreN.Score ~ Conditions, data=AVAvsSS) #equal
471 # Levene's Test for Homogeneity of Variance (center = median)
472       # Df F value Pr(>F)
473 # group   2  0.7466 0.4753
474       # 197
475
476 tapply(AVAvsSS$PreO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
477       # A        B        C
478 # 654.9867 343.1792 549.0955
479
480 levene.test(PreO.Score ~ Conditions, data=AVAvsSS)  #equal
481 # Levene's Test for Homogeneity of Variance (center = median)
482       # Df F value Pr(>F)
483 # group   2  1.5492   0.215
484       # 197
485
486 tapply(AVAvsSS$PreSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
487       # A        B        C
488 # 837.0275 418.2744 875.5540
489
490 levene.test(PreSSQ.Score ~ Conditions, data=AVAvsSS) #equal
491 # Levene's Test for Homogeneity of Variance (center = median)
492       # Df F value Pr(>F)
493 # group   2  0.9624 0.3838
```

```
494       # 197
495
496 tapply(AVAvsSS$PostD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
497        # A         B         C
498 # 539.6996 470.6748 899.4973
499
500 levene.test(PostD.Score ~ Conditions, data=AVAvsSS) #equal
501 # Levene's Test for Homogeneity of Variance (center = median)
502        # Df F value Pr(>F)
503 # group   2  0.1081 0.8976
504       # 197
505
506 tapply(AVAvsSS$PostN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
507        # A         B         C
508 # 229.2001 408.6901 383.3623
509
510 levene.test(PostN.Score ~ Conditions, data=AVAvsSS) #equal
511 # Levene's Test for Homogeneity of Variance (center = median)
512        # Df F value Pr(>F)
513 # group   2   1.737 0.1787
514       # 197
515
516 tapply(AVAvsSS$PostO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
517        # A         B         C
518 # 458.2716 385.8081 544.9325
519
520 levene.test(PostO.Score ~ Conditions, data=AVAvsSS) #equal
521 # Levene's Test for Homogeneity of Variance (center = median)
522        # Df F value Pr(>F)
523 # group   2  0.0948 0.9096
524       # 197
525
526 tapply(AVAvsSS$PostSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
527        # A         B         C
528 # 411.3303 456.2264 621.1001
529
530 > levene.test(PostSSQ.Score ~ Conditions, data=AVAvsSS) #equal
531 # Levene's Test for Homogeneity of Variance (center = median)
532        # Df F value Pr(>F)
533 # group   2  0.2286 0.7958
534       # 197
535
536
537
538 ##And do the statistical analysis, ref: http://tolstoy.newcastle.edu.au
    /R/e2/help/07/04/14503.html
539 ##Does visual complexity correlate to SS?
540
541 tapply(AVAvsSS$SSQ.Incr, AVAvsSS$Conditions, median, na.rm=TRUE)
542 kruskal.test(SSQ.Incr ~ Conditions, data=AVAvsSS)
543
544 #Kruskal-Wallis rank sum test
545 #data: SSQ.Incr by Conditions
546 #Kruskal-Wallis chi-squared = 0.5535, df = 2, p-value = 0.7583 #No, it
    does not.
```

```
547
548 #General results
549 median(AVAvsSS_A$PostSSQ.Score - AVAvsSS_A$PreSSQ.Score, na.rm=TRUE) #
    median difference
550 wilcox.test(AVAvsSS_A$PreSSQ.Score, AVAvsSS_A$PostSSQ.Score,
    alternative='two.sided', paired=TRUE) #V = 827.5, p-value = 0.03269
551 mean(rowMeans(sign(outer(AVAvsSS_A$PreSSQ.Score, AVAvsSS_A$PostSSQ.
    Score, FUN="-")))) #0.12 Cliff's d
552
553 median(AVAvsSS_B$PostSSQ.Score - AVAvsSS_B$PreSSQ.Score, na.rm=TRUE) #
    median difference
554 wilcox.test(AVAvsSS_B$PreSSQ.Score, AVAvsSS_B$PostSSQ.Score,
    alternative='two.sided', paired=TRUE) #V = 887.5, p-value = 0.01596
555 mean(rowMeans(sign(outer(AVAvsSS_B$PreSSQ.Score, AVAvsSS_B$PostSSQ.
    Score, FUN="-")))) #0.18 Cliff's d
556
557 median(AVAvsSS_C$PostSSQ.Score - AVAvsSS_C$PreSSQ.Score, na.rm=TRUE) #
    median difference
558 wilcox.test(AVAvsSS_C$PreSSQ.Score, AVAvsSS_C$PostSSQ.Score,
    alternative='two.sided', paired=TRUE) #V = 1449.5, p-value =
    0.002523
559 mean(rowMeans(sign(outer(AVAvsSS_C$PreSSQ.Score, AVAvsSS_C$PostSSQ.
    Score, FUN="-")))) #0.15 Cliff's d
560
561 #Gender Difference
562 tapply(AVAvsSS$SSQ.Incr, AVAvsSS$Gender, median, na.rm=TRUE)
563 wilcox.test(SSQ.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #
    W = 3262.5, p-value = 0.016
564 mean(rowMeans(sign(outer(AVAvsSS_F$SSQ.Incr, AVAvsSS_M$SSQ.Incr, FUN="-
    "))))) #-0.22 Cliff's d
565
566 wilcox.test(N.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 3688.5, p-value = 0.19
567 mean(rowMeans(sign(outer(AVAvsSS_F$N.Incr, AVAvsSS_M$N.Incr, FUN="-")))
    ) #-0.11 Cliff's d
568
569 wilcox.test(D.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 3558.5, p-value = 0.09
570 mean(rowMeans(sign(outer(AVAvsSS_F$D.Incr, AVAvsSS_M$D.Incr, FUN="-")))
    ) #-0.14 Cliff's d
571
572 wilcox.test(O.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 3395, p-value = 0.04
573 mean(rowMeans(sign(outer(AVAvsSS_F$O.Incr, AVAvsSS_M$O.Incr, FUN="-")))
    ) #-0.18 Cliff's d
```

G.1.2 Codes for analysing Learning Outcomes

Listing: AVAvsCAL[2010-04-09][1436].r

```
1 #Clearing workspace (and variables too)#################################
```

```
rm(list=ls())

#Setting up the working directory#####################################
setwd("/home/ephraim/[e1]thesis/results/caseStudy1")

#loading required library#############################################

library(car)
library(psychometric)
library(Hmisc)
library(psych)
library(Cairo)
library(lattice)
library(mgcv)
require(gplots)
library(xtable)

#nonparametic analysis packages
#library(npmc)
#library(vegan) #use adonis(...) for nonparametric manova

# Loading the results#################################################
results <-
  read.table("resultsFromExperiment1-Odometric behaviour of Cataglyphis
      Ants [2009-03-14][1238].csv",
    header=TRUE,
    sep="\t",
    na.strings="NA",
    dec=".",
    strip.white=TRUE)

#removing cases with NA entries into a new dataframe
results <- na.omit(results)

#Creating a dataframe
results.df <- data.frame(results)
attach(results.df)
names(results.df)

AVAvsCAL <- data.frame(subset(results.df,
   Hw.failure == "N",
    select=c(Conditions,
        Gender,
        Age,
        Participant,
        PreTR.Score,
        PreTT.Score,
        Presence.Score,
        PAQ.PU.x.,
        PAQ.AS.y.,
        Affective.Score,
        CognitiveLoad.Score,
        PostTR.Score,
        PostTT.Score,
        TR.Incr,
```

```
56          TT.Incr
57          )
58      )
59      )
60
61  AVAvsCAL_A <- subset(AVAvsCAL, Conditions == "A")
62  AVAvsCAL_B <- subset(AVAvsCAL, Conditions == "B")
63  AVAvsCAL_C <- subset(AVAvsCAL, Conditions == "C")
64  AVAvsCAL_M <- subset(AVAvsCAL, Gender == "m")
65  AVAvsCAL_F <- subset(AVAvsCAL, Gender == "f")
66
67  ######################################################################
68  #Displaying the summary of the dataframe
69  ######################################################################
70  #Summary in overall
71  summary(AVAvsCAL)
72  #Summary by groups
73  summary(subset(AVAvsCAL, Conditions == 'A', digits = 2))
74  summary(subset(AVAvsCAL, Conditions == 'B', digits = 2))
75  summary(subset(AVAvsCAL, Conditions == 'C', digits = 2))
76
77  ##Describing the data
78  describe(AVAvsCAL,
79  #   digits = 2,
80      na.rm = TRUE,
81      interp=FALSE,
82      skew = TRUE,
83      ranges = TRUE,
84      trim=.1
85      )
86  ##Describing the data by groups
87  describe.by(AVAvsCAL, AVAvsCAL$Conditions,
88  #   digits = 2,
89      na.rm = TRUE,
90      interp=FALSE,
91      skew = TRUE,
92      ranges = TRUE,
93      trim=.1
94      )
95
96  ######################################################################
97  #Plotting the graphs and saving them on file
98  ######################################################################
99  ###Boxplot of SSQ
100 ##Save it on file for publication...
101 Cairo(2000, 900, file="AVAvsCAL_boxplots.pdf", type="pdf", bg="white",
        dpi = 150, units="px")
102 par(mfrow = c(2, 6))
103 boxplot(PreTR.Score ~ Conditions, data = AVAvsCAL,
104         main="Pretest retention",
105         xlab="Types of AVA",
106         ylab="Pretest retention"
107     )
108 boxplot(PreTT.Score ~ Conditions, data = AVAvsCAL,
109         main="Pretest transfer",
```

```
110         xlab="Types of AVA",
111         ylab="Pretest transfer"
112     )
113 boxplot(Presence.Score ~ Conditions, data = AVAvsCAL,
114     main="Presence",
115         xlab="Types of AVA",
116         ylab="Presence"
117     )
118 boxplot(PAQ.PU.x. ~ Conditions, data = AVAvsCAL,
119     main="Pleasant-unpleasant",
120         xlab="Types of AVA",
121         ylab="Pleasant-unpleasant"
122     )
123 boxplot(PAQ.AS.y. ~ Conditions, data = AVAvsCAL,
124     main="Arousal-sleepy",
125         xlab="Types of AVA",
126         ylab="Arousal-sleepy"
127     )
128 boxplot(Affective.Score ~ Conditions, data = AVAvsCAL,
129     main="Affective",
130         xlab="Types of AVA",
131         ylab="Affective score"
132     )
133 boxplot(CognitiveLoad.Score ~ Conditions, data = AVAvsCAL,
134     main="Cognitive load",
135         xlab="Types of AVA",
136         ylab="Cognitive load score"
137     )
138 boxplot(PostTR.Score ~ Conditions, data = AVAvsCAL,
139         main="Posttest retention",
140         xlab="Types of AVA",
141         ylab="Posttest retention"
142     )
143 boxplot(PostTT.Score ~ Conditions, data = AVAvsCAL,
144         main="Posttest transfer",
145         xlab="Types of AVA",
146         ylab="Posttest transfer"
147     )
148 boxplot(TR.Incr ~ Conditions, data = AVAvsCAL,
149     main="Improvement in retention",
150         xlab="Types of AVA",
151         ylab="Improvement in retention"
152     )
153 boxplot(TT.Incr ~ Conditions, data = AVAvsCAL,
154     main="Improvement in transfer",
155         xlab="Types of AVA",
156         ylab="Improvement in transfer"
157     )
158 dev.off()
159
160
161 ###Plotting histogram of each dependant variables
162 ##Save on file...dev.new
163 ##Save it on file for publication...
```

```
Cairo(1280, 800, file="plot-AVAvsCAL-histograms.png", type="png", bg="
    white")
par(mfrow = c(3, 4))
hist(AVAvsCAL$PreTR.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Pretest retention")
hist(AVAvsCAL$PreTT.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Pretest transfer")
hist(AVAvsCAL$Presence.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Pre-SSQ: Oculomotor")
hist(AVAvsCAL$PAQ.PU.x.,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Pleasant-unpleasant scale")
hist(AVAvsCAL$PAQ.AS.y.,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Arousal-sleepy")
hist(AVAvsCAL$Affective.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Affective score")
hist(AVAvsCAL$CognitiveLoad.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Cognitive Load")
hist(AVAvsCAL$PostTR.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Posttest retention")
hist(AVAvsCAL$PostTT.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Posttest transfer")
hist(AVAvsCAL$TR.Incr,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
    xlab ="Improvement in retention")
hist(AVAvsCAL$TT.Incr,
    #scale="frequency",
```

```
    breaks="Sturges",
    col="darkgray",
    xlab ="Improvement in transfer")
dev.off()

#########################################################################
#Normality and Variance check
#########################################################################
###Normality test###
#Shapiro-Wilk Normality Test. When the p-value is higher than
    significance levels
#the sample forms a gaussian (bell-shaped) distribution.
#Thode(2002) reviewed over forty procedures for assesing goodness-of-
    fit and concludes
#that Shapiro and Wilk (1965, 1968) and test statistics derived from
    moments
#are generally the best methods for assessing normality
# Thode, H. C. (2002). Testing for normality. New York: Marcel Dekker.
shapiro.test(AVAvsCAL$PreTR.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PreTR.Score
# W = 0.9504, p-value = 2.046e-06


shapiro.test(AVAvsCAL$PreTT.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PreTT.Score
# W = 0.7661, p-value < 2.2e-16


shapiro.test(AVAvsCAL$Presence.Score) #Normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$Presence.Score
# W = 0.9919, p-value = 0.3296


shapiro.test(AVAvsCAL$PAQ.PU.x.) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PAQ.PU.x.
# W = 0.9804, p-value = 0.006694


shapiro.test(AVAvsCAL$PAQ.AS.y.) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PAQ.AS.y.
```

```
# W = 0.9844, p-value = 0.02562


shapiro.test(AVAvsCAL$Affective.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$Affective.Score
# W = 0.9392, p-value = 1.948e-07


shapiro.test(AVAvsCAL$CognitiveLoad.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$CognitiveLoad.Score
# W = 0.905, p-value = 5.303e-10


shapiro.test(AVAvsCAL$PostTR.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PostTR.Score
# W = 0.7054, p-value < 2.2e-16


shapiro.test(AVAvsCAL$PostTT.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PostTT.Score
# W = 0.8734, p-value = 6.881e-12


shapiro.test(AVAvsCAL$TR.Incr) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$TR.Incr
# W = 0.9589, p-value = 1.505e-05


shapiro.test(AVAvsCAL$TT.Incr) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$TT.Incr
# W = 0.9284, p-value = 2.516e-08

###Checking wether dependent variables have the same variances between
    groups##
###If Bartlett test s p-value is less than 0.05
###then the assumption of equal variances would be rejected.
tapply(AVAvsCAL$PreTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
```

```
        # A         B         C
 # 94.8976  157.6501  108.0026

bartlett.test(PreTR.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PreTR.Score by Conditions
# Bartlett's K-squared = 4.4148, df = 2, p-value = 0.11


tapply(AVAvsCAL$PreTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
        # A         B         C
# 127.6042  169.0508  163.3523

bartlett.test(PreTT.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PreTT.Score by Conditions
# Bartlett's K-squared = 1.383, df = 2, p-value = 0.5008


tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
        # A         B         C
# 41.86751  63.54429  53.93712

bartlett.test(Presence.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  Presence.Score by Conditions
# Bartlett's K-squared = 2.5934, df = 2, p-value = 0.2734


tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Conditions, var, na.rm=TRUE)
         # A         B         C
 # 91.30395  89.80645  107.85748

bartlett.test(PAQ.PU.x. ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PAQ.PU.x. by Conditions
# Bartlett's K-squared = 0.7232, df = 2, p-value = 0.6966


tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Conditions, var, na.rm=TRUE)
         # A         B         C
# 104.21582  101.94060  88.40602

bartlett.test(PAQ.AS.y. ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

```

```
# data:  PAQ.AS.y. by Conditions
# Bartlett's K-squared = 0.5451, df = 2, p-value = 0.7614


tapply(AVAvsCAL$Affective.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
        # A         B         C
# 0.5973275 0.6720730 0.5446373

bartlett.test(Affective.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  Affective.Score by Conditions
# Bartlett's K-squared = 0.7547, df = 2, p-value = 0.6857


tapply(AVAvsCAL$CognitiveLoad.Score, AVAvsCAL$Conditions, var, na.rm=
    TRUE)
        # A         B         C
# 0.4160311 0.4613415 0.3929426

bartlett.test(CognitiveLoad.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  CognitiveLoad.Score by Conditions
# Bartlett's K-squared = 0.4454, df = 2, p-value = 0.8004


tapply(AVAvsCAL$PostTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
# 33.58934 27.14174 18.07609

bartlett.test(PostTR.Score ~ Conditions, data=AVAvsCAL) #not equal

   # Bartlett test of homogeneity of variances

# data:  PostTR.Score by Conditions
# Bartlett's K-squared = 6.5621, df = 2, p-value = 0.03759


tapply(AVAvsCAL$PostTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
# 329.2726 326.9809 251.8904

bartlett.test(PostTT.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PostTT.Score by Conditions
# Bartlett's K-squared = 1.5839, df = 2, p-value = 0.453


tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
```

```
# 114.3626 163.6505 101.8081

bartlett.test(TR.Incr ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  TR.Incr by Conditions
# Bartlett's K-squared = 4.161, df = 2, p-value = 0.1249


tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
# 335.9375 377.1441 387.9550

bartlett.test(TT.Incr ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  TT.Incr by Conditions
# Bartlett's K-squared = 0.3614, df = 2, p-value = 0.8347

###Checking wether dependent variables have the same variances between
    groups##
###If the resulting p-value of Levene's test is less than some critical
     value (typically .05),
###the obtained differences in sample variances are unlikely to have
    occurred based on random sampling.
###Thus, the null hypothesis of equal variances is rejected and it is
    concluded that there is a difference
###between the variances in the population. --Wikipedia

tapply(AVAvsCAL$PreTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
 # 94.8976 157.6501 108.0026

levene.test(PreTR.Score ~ Conditions, data=AVAvsCAL) #equal
# Levene's Test for Homogeneity of Variance (center = median)
       # Df F value Pr(>F)
# group   2  1.6938 0.1865
      # 197

tapply(AVAvsCAL$PreTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
# 127.6042 169.0508 163.3523

levene.test(PreTT.Score ~ Conditions, data=AVAvsCAL) #equal
# Levene's Test for Homogeneity of Variance (center = median)
       # Df F value Pr(>F)
# group   2  0.2391 0.7875
      # 197

tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
       # A        B        C
# 41.86751 63.54429 53.93712

```

```
484 levene.test(Presence.Score ~ Conditions, data=AVAvsCAL)  #equal
485 # Levene's Test for Homogeneity of Variance (center = median)
486         # Df F value Pr(>F)
487 # group   2  1.2782 0.2808
488       # 197
489
490 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Conditions, var, na.rm=TRUE)
491         # A         B         C
492  # 91.30395  89.80645 107.85748
493
494 levene.test(PAQ.PU.x. ~ Conditions, data=AVAvsCAL) #equal
495 # Levene's Test for Homogeneity of Variance (center = median)
496         # Df F value Pr(>F)
497 # group   2  0.3847 0.6811
498       # 197
499
500 tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Conditions, var, na.rm=TRUE)
501         # A         B         C
502 # 104.21582 101.94060  88.40602
503
504 levene.test(PAQ.AS.y. ~ Conditions, data=AVAvsCAL) #equal
505 # Levene's Test for Homogeneity of Variance (center = median)
506         # Df F value Pr(>F)
507 # group   2  0.0795 0.9236
508       # 197
509
510 tapply(AVAvsCAL$Affective.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
511         # A         B         C
512 # 0.5973275 0.6720730 0.5446373
513
514 > levene.test(Affective.Score ~ Conditions, data=AVAvsCAL) #equal
515 # Levene's Test for Homogeneity of Variance (center = median)
516         # Df F value Pr(>F)
517 # group   2  0.1057 0.8997
518       # 197
519
520 tapply(AVAvsCAL$CognitiveLoad.Score, AVAvsCAL$Conditions, var, na.rm=
    TRUE)
521         # A         B         C
522 # 0.4160311 0.4613415 0.3929426
523
524 levene.test(CognitiveLoad.Score ~ Conditions, data=AVAvsCAL) #equal
525 # Levene's Test for Homogeneity of Variance (center = median)
526         # Df F value Pr(>F)
527 # group   2  0.0428 0.9581
528       # 197
529
530 tapply(AVAvsCAL$PostTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
531         # A         B         C
532 # 33.58934 27.14174 18.07609
533
534 > levene.test(PostTR.Score ~ Conditions, data=AVAvsCAL) #equal
535 # Levene's Test for Homogeneity of Variance (center = median)
536         # Df F value Pr(>F)
537 # group   2  0.1325   0.876
```

```
538       # 197
539
540 tapply(AVAvsCAL$PostTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
541       # A         B         C
542 # 329.2726 326.9809 251.8904
543
544 levene.test(PostTT.Score ~ Conditions, data=AVAvsCAL) #equal
545 # Levene's Test for Homogeneity of Variance (center = median)
546       # Df F value Pr(>F)
547 # group   2  0.7029 0.4964
548       # 197
549
550 tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
551       # A         B         C
552 # 114.3626 163.6505 101.8081
553
554 levene.test(TR.Incr ~ Conditions, data=AVAvsCAL) #equal
555 # Levene's Test for Homogeneity of Variance (center = median)
556       # Df F value Pr(>F)
557 # group   2  1.2064 0.3015
558       # 197
559
560 tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
561       # A         B         C
562 # 335.9375 377.1441 387.9550
563
564 levene.test(TT.Incr ~ Conditions, data=AVAvsCAL) #equal
565 # Levene's Test for Homogeneity of Variance (center = median)
566       # Df F value Pr(>F)
567 # group   2  0.4385 0.6456
568       # 197
569
570 ##And do the statistical analysis
571 ##Ref http://tolstoy.newcastle.edu.au/R/e2/help/07/04/14503.html
572 ######################################################################
573 #Research questions#####################################################
574 ######################################################################
575
576 ###Does visual complexity of AVA deteriorate cognitive learning?
577 ##Kruskal-Wallis rank sum test - retention
578 kruskal.test(TR.Incr ~ Conditions, data = AVAvsCAL) #chisq = 0.4195, df
      = 2, p=0.81 [NO]
579
580 ##There is a significant improvement in Retention
581 #Overall
582 median(AVAvsCAL$PostTR.Score - AVAvsCAL$PreTR.Score, na.rm=TRUE) #
      median difference
583 wilcox.test(AVAvsCAL$PostTR.Score, AVAvsCAL$PreTR.Score, alternative='
      two.sided', paired=TRUE)
584
585 #Individual group
586 median(AVAvsCAL_A$PostTR.Score - AVAvsCAL_A$PreTR.Score, na.rm=TRUE) #
      median difference, 31.25
587 wilcox.test(AVAvsCAL_A$PostTR.Score, AVAvsCAL_A$PreTR.Score,
      alternative='two.sided', paired=TRUE) #V = 1830, p-value = 1.378e
```

```
    -11
mean(rowMeans(sign(outer(AVAvsCAL_A$PostTR.Score, AVAvsCAL_A$PreTR.
    Score, FUN="-")))) #d=0.98

median(AVAvsCAL_B$PostTR.Score - AVAvsCAL_B$PreTR.Score, na.rm=TRUE) #
    median difference, 31.25
wilcox.test(AVAvsCAL_B$PostTR.Score, AVAvsCAL_B$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 1891, p-value = 9.562e
    -12
mean(rowMeans(sign(outer(AVAvsCAL_B$PostTR.Score, AVAvsCAL_B$PreTR.
    Score, FUN="-")))) #d=0.94

median(AVAvsCAL_C$PostTR.Score - AVAvsCAL_C$PreTR.Score, na.rm=TRUE) #
    median difference, 31.25
wilcox.test(AVAvsCAL_C$PostTR.Score, AVAvsCAL_C$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 2926, p-value = 2.776e
    -14
mean(rowMeans(sign(outer(AVAvsCAL_C$PostTR.Score, AVAvsCAL_C$PreTR.
    Score, FUN="-")))) #d=0.99

##Kruskal-Wallis rank sum test - transfer
kruskal.test(TT.Incr ~ Conditions, data = AVAvsCAL) #chisq = 0.8535, df
    = 2, p=0.65 [NO]

##There is a significant improvement in Transfer
#Overall
median(AVAvsCAL$PostTT.Score - AVAvsCAL$PreTT.Score, na.rm=TRUE) #
    median difference
wilcox.test(AVAvsCAL$PostTT.Score, AVAvsCAL$PreTT.Score, alternative='
    two.sided', paired=TRUE) #V = 11878.5, p-value < 2.2e-16

#Individual group
median(AVAvsCAL_A$PostTT.Score - AVAvsCAL_A$PreTT.Score, na.rm=TRUE) #
    median difference, 12.5
wilcox.test(AVAvsCAL_A$PostTT.Score, AVAvsCAL_A$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 960, p-value = 4.048e-08
mean(rowMeans(sign(outer(AVAvsCAL_A$PostTT.Score, AVAvsCAL_A$PreTT.
    Score, FUN="-")))) #d=0.56

median(AVAvsCAL_B$PostTT.Score - AVAvsCAL_B$PreTT.Score, na.rm=TRUE) #
    median difference, 25
wilcox.test(AVAvsCAL_B$PostTT.Score, AVAvsCAL_B$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 1232.5, p-value = 6.897e
    -09
mean(rowMeans(sign(outer(AVAvsCAL_B$PostTT.Score, AVAvsCAL_B$PreTT.
    Score, FUN="-")))) #d=0.61

median(AVAvsCAL_C$PostTT.Score - AVAvsCAL_C$PreTT.Score, na.rm=TRUE) #
    median difference, 25
wilcox.test(AVAvsCAL_C$PostTT.Score, AVAvsCAL_C$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 1900, p-value = 6.759e
    -10
mean(rowMeans(sign(outer(AVAvsCAL_C$PostTT.Score, AVAvsCAL_C$PreTT.
    Score, FUN="-")))) #d=0.65

```

```
619
620 ###Does visual complexity of AVA alters affective component in learning
        ?
621 ##Kruskal-Wallis rank sum test - presence
622 kruskal.test(Presence.Score ~ Conditions, data = AVAvsCAL) #Kruskal-
        Wallis chi-squared = 2.241, df = 2, p-value = 0.3261 [no]
623 kruskal.test(PAQ.PU.x.~ Conditions, data = AVAvsCAL) #Kruskal-Wallis
        chi-squared = 2.6766, df = 2, p-value = 0.2623 [no]
624 kruskal.test(PAQ.AS.y.~ Conditions, data = AVAvsCAL) #Kruskal-Wallis
        chi-squared = 1.0801, df = 2, p-value = 0.5827 [no]
625 kruskal.test(Affective.Score ~ Conditions, data = AVAvsCAL) #Kruskal-
        Wallis chi-squared = 0.118, df = 2, p-value = 0.9427 [no]
626 kruskal.test(CognitiveLoad.Score ~ Conditions, data = AVAvsCAL) #
        Kruskal-Wallis chi-squared = 0.2716, df = 2, p-value = 0.873 [no]
627
628 ##Gender effect
629 tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Gender, median, na.rm=TRUE)
630 wilcox.test(Presence.Score ~ Gender, alternative="two.sided", data=
        AVAvsCAL)
631 mean(rowMeans(sign(outer(AVAvsCAL_F$Presence.Score, AVAvsCAL_M$Presence
        .Score, FUN="-")))) #0.15 Cliff's d
632
633 tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Gender, median, na.rm=TRUE)
634 wilcox.test(PAQ.AS.y. ~ Gender, alternative="two.sided", data=AVAvsCAL)
        #W = 5146.5, p-value = 0.008181
635 mean(rowMeans(sign(outer(AVAvsCAL_F$PAQ.AS.y., AVAvsCAL_M$PAQ.AS.y.,
        FUN="-")))) #0.24 Cliff's d
636
637 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Gender, median, na.rm=TRUE)
638 wilcox.test(PAQ.PU.x. ~ Gender, alternative="two.sided", data=AVAvsCAL)
639 mean(rowMeans(sign(outer(AVAvsCAL_F$PAQ.PU.x., AVAvsCAL_M$PAQ.PU.x.,
        FUN="-")))) #0.02 Cliff's d
640
641 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Gender, median, na.rm=TRUE)
642 wilcox.test(PAQ.PU.x. ~ Gender, alternative="two.sided", data=AVAvsCAL)
643 mean(rowMeans(sign(outer(AVAvsCAL_F$PAQ.PU.x., AVAvsCAL_M$PAQ.PU.x.,
        FUN="-"))))
644
645 tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Gender, median, na.rm=TRUE)
646 wilcox.test(TR.Incr ~ Gender, alternative="two.sided", data=AVAvsCAL)
647
648 tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Gender, median, na.rm=TRUE)
649 wilcox.test(TT.Incr ~ Gender, alternative="two.sided", data=AVAvsCAL)
650
651 ###Does affective component of simulation correlate to cognitive
        learning?
652 #Ref http://myowelt.blogspot.com/2008/04/beautiful-correlation-tables-
        in-r.html
653 corstarsl <- function(x){
654     require(Hmisc)
655     x <- as.matrix(x)
656     R <- rcorr(x, type="spearman")$r
657     p <- rcorr(x, type="spearman")$P
658
659     ## define notions for significance levels; spacing is important.
```

```
660     mystars <- ifelse(p < .001, "***", ifelse(p < .01, "** ", ifelse(p <
            .05, "* ", " ")))
661
662     ## trunctuate the matrix that holds the correlations to two decimal
663     R <- format(round(cbind(rep(-1.11, ncol(x)), R), 2))[,-1]
664
665     ## build a new matrix that includes the correlations with their
            apropriate stars
666     Rnew <- matrix(paste(R, mystars, sep=""), ncol=ncol(x))
667     diag(Rnew) <- paste(diag(R), " ", sep="")
668     rownames(Rnew) <- colnames(x)
669     colnames(Rnew) <- paste(colnames(x), "", sep="")
670
671     ## remove upper triangle
672     Rnew <- as.matrix(Rnew)
673     Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
674     Rnew <- as.data.frame(Rnew)
675
676     ## remove last column and return the matrix (which is now a data
            frame)
677     Rnew <- cbind(Rnew[1:length(Rnew)-1])
678     return(Rnew)
679     }
680
681 corstarsl(AVAvsCAL[,c("Presence.Score",
682         "PAQ.AS.y.",
683         "PAQ.PU.x.",
684         "Affective.Score",
685         "CognitiveLoad.Score",
686         "TR.Incr",
687         "TT.Incr")])
688
689 #Prints out the laTex code
690 xtable(corstarsl(AVAvsCAL[,c("Presence.Score",
691         "PAQ.AS.y.",
692         "PAQ.PU.x.",
693         "Affective.Score",
694         "CognitiveLoad.Score",
695         "TR.Incr",
696         "TT.Incr")]))
697
698 #          Presence.Score     PAQ.AS.y.  PAQ.PU.x.  Affective.Score
        CognitiveLoad.Score  TR.Incr
699 #Presence.Score
700 #PAQ.AS.y.                  0.50***
701 #PAQ.PU.x.                  0.55***    0.44***
702 #Affective.Score           0.59***    0.40***    0.49***
703 #CognitiveLoad.Score       -0.16*      -0.12      -0.14*              -0.07
704 #TR.Incr                    -0.13      -0.07      -0.14               -0.14
                    0.07
705 #TT.Incr                     0.01      -0.01      -0.02                0.01
                   -0.04      0.04
706
707 #Scatterplot Matrix#####################################################
708 #Saving it in a file first, because it is very big
```

```
709 Cairo(1280, 1280, file="AVAvsCAL-scatterPlotMatrix", type="png", bg="
    white")
710 par(mfrow = c(1, 1))
711
712 scatterplot.matrix(~Presence.Score+PAQ.AS.y.+PAQ.PU.x.+Affective.Score+
    CognitiveLoad.Score+TR.Incr+TT.Incr | Conditions,
713     reg.line=lm,
714     smooth=TRUE,
715     span=0.5,
716     diagonal= 'density',
717     by.groups=TRUE,
718     data=AVAvsCAL)
719
720 #Writing it in a file -> AVAvsCAL-scatterPlotMatrix
721 dev.off()
722
723 #Correlation Matrix########################################################
724 #These pairs scatter plots are awfully complicated, use the previous
725 #one instead. Don't use this one unless necessary.
726
727 #Saving it in a file first, because it is very big
728 Cairo(1280, 1280, file="AVAvsCAL-correlationMatrix.png", type="png", bg
    ="white")
729
730 panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
731 {
732     usr <- par("usr"); on.exit(par(usr))
733     par(usr = c(0, 1, 0, 1))
734     r <- abs(cor(x, y, method = "spearman"))
735     txt <- format(c(r, 0.123456789), digits=digits)[1]
736     txt <- paste(prefix, txt, sep="")
737     if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
738
739     test <- cor.test(x,y, method = "spearman")
740     # borrowed from printCoefmat
741     Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
742                      cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
743                      symbols = c("***", "**", "*", ".", " "))
744
745     text(0.5, 0.5, txt, cex = cex * r)
746     text(.8, .8, Signif, cex=cex, col=2)
747 }
748
749 pairs(AVAvsCAL[,c("Presence.Score",
750       "PAQ.AS.y.",
751       "PAQ.PU.x.",
752       "Affective.Score",
753       "CognitiveLoad.Score",
754       "TR.Incr",
755       "TT.Incr")],
756     lower.panel=panel.smooth,
757     upper.panel=panel.cor)
758
759 #Writing it in a file -> AVAvsCAL-correlationMatrix.png
760 dev.off()
```

G.2 R codes of VEAR

G.2.1 Codes for analysing Simulator Sickness

Listing: AVAvsSS[2010-10-05][0917].r

```
#Clearing workspace (and variables too)#################################
rm(list=ls())

#Setting up the working directory#######################################
setwd("/home/ephraim/e1_thesis/results/caseStudy2")

#loading required library###############################################
#library(Rcmdr)
library(psychometric)
library(psy)
library(Cairo)
library(lattice)
library(rgl)
library(mgcv)
library(pwr)
require(gplots)
library(Hmisc)
library(xtable)

#nonparametic analysis packages
#library(npmc)
#library(vegan) #use adonis(...) for nonparametric manova

# Loading the results###################################################
results <- read.table("resultsFromExperiment2-VillageFirefighting.csv",
    header=TRUE,
    sep="\t",
    na.strings="NA",
    dec=".",
    strip.white=TRUE)
#removing cases with NA entries into a new dataframe
results <- na.omit(results)

#Creating a dataframe
results.df <- data.frame(results)
attach(results.df)
names(results.df)

########################################################################
#Research questions#####################################################
########################################################################
```

```
42
43 ###Does additional visual information of virtual actors affect
       simulator sickness?
44 AVAvsSS <- data.frame(subset(results.df,
45     Usable == "Y",
46     select=c(Conditions,
47         Gender,
48         Age,
49         Participant,
50         PreD.Score,
51         PreN.Score,
52         PreO.Score,
53         PreSSQ.Score,
54         PostD.Score,
55         PostN.Score,
56         PostO.Score,
57         PostSSQ.Score,
58         D.Incr,
59         N.Incr,
60         O.Incr,
61         SSQ.Incr
62         )
63     )
64     )
65
66 AVAvsSS_A <- subset(AVAvsSS, Conditions == "A")
67 AVAvsSS_B <- subset(AVAvsSS, Conditions == "B")
68 AVAvsSS_C <- subset(AVAvsSS, Conditions == "C")
69 AVAvsSS_D <- subset(AVAvsSS, Conditions == "D")
70 AVAvsSS_M <- subset(AVAvsSS, Gender == "m")
71 AVAvsSS_F <- subset(AVAvsSS, Gender == "f")
72
73 ######################################################################
74 #Displaying the summary of the dataframe
75 ######################################################################
76 #Summary in overall
77 summary(AVAvsSS)
78 #Summary by groups
79 summary(subset(AVAvsSS, Conditions == 'A', digits = 2))
80 summary(subset(AVAvsSS, Conditions == 'B', digits = 2))
81 summary(subset(AVAvsSS, Conditions == 'C', digits = 2))
82 summary(subset(AVAvsSS, Conditions == 'D', digits = 2))
83
84 ##Describing the data
85 describe(AVAvsSS,
86 #   digits = 2,
87     na.rm = TRUE,
88     interp=FALSE,
89     skew = TRUE,
90     ranges = TRUE,
91     trim=.1
92     )
93 ##Describing the data by groups
94 describe.by(AVAvsSS, AVAvsSS$Conditions,
95 #   digits = 2,
```

```
    na.rm = TRUE,
    interp=FALSE,
    skew = TRUE,
    ranges = TRUE,
    trim=.1
    )

#########################################################################
#Plotting the graphs and saving them on file
#########################################################################
###Boxplot of SSQ
##Save it on file for publication...
Cairo(2560, 1600, file="AVAvsSS_boxplots.pdf", type="pdf", bg="white",
    dpi = 200, units="px")
par(mfrow = c(3, 4))
boxplot(PreN.Score ~ Conditions, data = AVAvsSS,
        main="Pre-SSQ: Nausea",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PreD.Score ~ Conditions, data = AVAvsSS,
    main="Pre-SSQ: Disorientation",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PreO.Score ~ Conditions, data = AVAvsSS,
    main="Pre-SSQ: Occulomotor",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PreSSQ.Score ~ Conditions, data = AVAvsSS,
    main="Pre-SSQ: Total severity score",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PostN.Score ~ Conditions, data = AVAvsSS,
    main="Post-SSQ: Nausea",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PostD.Score ~ Conditions, data = AVAvsSS,
    main="Post-SSQ: Disorientation",
        xlab="Experimental conditions",
        ylab="Level of severity"
    )

boxplot(PostO.Score ~ Conditions, data = AVAvsSS,
    main="Post-SSQ: Occulomotor",
        xlab="Experimental conditions",
        ylab="Level of severity"
```

```
150     )
151
152 boxplot(PostSSQ.Score ~ Conditions, data = AVAvsSS,
153     main="Post-SSQ: Total severity score",
154           xlab="Experimental Conditions",
155           ylab="Level of severity"
156     )
157
158 boxplot(D.Incr ~ Conditions, data = AVAvsSS,
159     main="Increase in Disorientation",
160           xlab="Experimental conditions",
161           ylab="Level of severity"
162     )
163
164 boxplot(N.Incr ~ Conditions, data = AVAvsSS,
165     main="Increase in Nausea",
166           xlab="Experimental conditions",
167           ylab="Level of severity"
168     )
169
170 boxplot(O.Incr ~ Conditions, data = AVAvsSS,
171     main="Increase in Occulomotor",
172           xlab="Experimental Conditions",
173           ylab="Level of severity"
174     )
175
176 boxplot(SSQ.Incr ~ Conditions, data = AVAvsSS,
177     main="Increase in Total Severity Score",
178           xlab="Experimental Conditions",
179           ylab="Level of severity"
180     )
181
182 dev.off()
183
184 ###Plotting histogram of each dependant variables
185 ##Save on file...dev.new()
186 Cairo(2560, 1600, file="plot-AVAvsSS-histograms.pdf", type="pdf", bg="
        white", dpi = 200, units="px")
187 par(mfrow = c(3, 4))
188
189 hist(AVAvsSS$PreD.Score,
190     #scale="frequency",
191     breaks="Sturges",
192     col="darkgray",
193           xlab="Level of severity",
194     main ="Pre-SSQ: Disorientation")
195
196
197 hist(AVAvsSS$PreN.Score,
198     #scale="frequency",
199     breaks="Sturges",
200     col="darkgray",
201           xlab="Level of severity",
202     main ="Pre-SSQ: Nausea")
203
```

```
hist(AVAvsSS$PreO.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Pre-SSQ: Oculomotor")

hist(AVAvsSS$PreSSQ.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Pre-SSQ: Total severity score")

hist(AVAvsSS$PostD.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Post-SSQ: Disorientation")

hist(AVAvsSS$PostN.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Post-SSQ: Nausea")

hist(AVAvsSS$PostO.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Post-SSQ: Oculomotor")

hist(AVAvsSS$PostSSQ.Score,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Post-SSQ: Total severity score")

hist(AVAvsSS$N.Incr,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Increase in Nausea")

hist(AVAvsSS$D.Incr,
    #scale="frequency",
    breaks="Sturges",
    col="darkgray",
         xlab="Level of severity",
    main ="Increase in Disorientation")
```

```
259
260 hist(AVAvsSS$O.Incr,
261     #scale="frequency",
262     breaks="Sturges",
263     col="darkgray",
264          xlab="Level of severity",
265     main ="Increase in Occulomotor")
266
267 hist(AVAvsSS$SSQ.Incr,
268     #scale="frequency",
269     breaks="Sturges",
270     col="darkgray",
271          xlab="Level of severity",
272     main ="Increase in Total Severity Score")
273
274 dev.off()
275
276
277 ######################################################################
278 #Normality and Variance check
279 ######################################################################
280 ###Normality test###
281 #Shapiro-Wilk Normality Test. When the p-value is higher than
        significance levels
282 #the sample forms a gaussian (bell-shaped) distribution.
283 #Thode(2002) reviewed over forty procedures for assesing goodness-of-
        fit and concludes
284 #that Shapiro and Wilk (1965, 1968) and test statistics derived from
        moments
285 #are generally the best methods for assessing normality.
286 #Thode, H. C. (2002). Testing for normality. New York: Marcel Dekker.
287
288 shapiro.test(AVAvsSS$PreD.Score) #Not normal
289
290     # Shapiro-Wilk normality test
291
292 # data:  AVAvsSS$PreD.Score
293 # W = 0.5626, p-value = 8.81e-15
294
295
296 shapiro.test(AVAvsSS$PreN.Score) #Not normal
297
298     # Shapiro-Wilk normality test
299
300 # data:  AVAvsSS$PreN.Score
301 # W = 0.5465, p-value = 4.726e-15
302
303
304 shapiro.test(AVAvsSS$PreO.Score) #Not normal
305
306     # Shapiro-Wilk normality test
307
308 # data:  AVAvsSS$PreO.Score
309 # W = 0.6643, p-value = 7.001e-13
310
```

```
311
312 shapiro.test(AVAvsSS$PreSSQ.Score) #Not normal
313
314     # Shapiro-Wilk normality test
315
316 # data:  AVAvsSS$PreSSQ.Score
317 # W = 0.6062, p-value = 5.201e-14
318
319
320 shapiro.test(AVAvsSS$PostD.Score) #Not normal
321
322     # Shapiro-Wilk normality test
323
324 # data:  AVAvsSS$PostD.Score
325 # W = 0.5983, p-value = 3.734e-14
326
327
328 shapiro.test(AVAvsSS$PostN.Score) #Not normal
329
330     # Shapiro-Wilk normality test
331
332 # data:  AVAvsSS$PostN.Score
333 # W = 0.571, p-value = 1.23e-14
334
335
336 shapiro.test(AVAvsSS$PostO.Score) #Not normal
337
338     # Shapiro-Wilk normality test
339
340 # data:  AVAvsSS$PostO.Score
341 # W = 0.6779, p-value = 1.347e-12
342
343
344 shapiro.test(AVAvsSS$PostSSQ.Score) #Not normal
345
346     # Shapiro-Wilk normality test
347
348 # data:  AVAvsSS$PostSSQ.Score
349 # W = 0.6482, p-value = 3.312e-13
350
351 ###Checking wether dependent variables have the same variances between
        groups##
352 ###If Bartlett test s p-value is less than 0.05
353 ###then the assumption of equal variances would be rejected.
354 tapply(AVAvsSS$PreD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
355         # A          B          C          D
356 # 1201.6580   399.5279  2182.1787   173.8536
357
358 bartlett.test(PreD.Score ~ Conditions, data=AVAvsSS)  #not equal
359
360     # Bartlett test of homogeneity of variances
361
362 # data:  PreD.Score by Conditions
363 # Bartlett's K-squared = 34.3763, df = 3, p-value = 1.65e-07
364
```

```
365
366 tapply(AVAvsSS$PreN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
367          A           B           C           D
368 1223.80025   281.26918   611.07789    98.20619
369
370 bartlett.test(PreN.Score ~ Conditions, data=AVAvsSS) #not equal
371
372    # Bartlett test of homogeneity of variances
373
374 # data:  PreN.Score by Conditions
375 # Bartlett's K-squared = 32.2897, df = 3, p-value = 4.547e-07
376
377
378 tapply(AVAvsSS$PreO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
379        A         B         C         D
380 652.2323 145.2826 924.5008 155.3367
381
382 bartlett.test(PreO.Score ~ Conditions, data=AVAvsSS)  #not equal
383
384    # Bartlett test of homogeneity of variances
385
386 # data:  PreO.Score by Conditions
387 # Bartlett's K-squared = 25.9048, df = 3, p-value = 9.985e-06
388
389
390 tapply(AVAvsSS$PreSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
391          A           B           C           D
392 1213.6593   278.7529 1328.0893   154.8035
393
394 bartlett.test(PreSSQ.Score ~ Conditions, data=AVAvsSS) #not equal
395
396    # Bartlett test of homogeneity of variances
397
398 # data:  PreSSQ.Score by Conditions
399 # Bartlett's K-squared = 30.8623, df = 3, p-value = 9.087e-07
400
401
402 tapply(AVAvsSS$PostD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
403        A         B         C         D
404 792.6807 424.4407 452.1216 173.8536
405
406 bartlett.test(PostD.Score ~ Conditions, data=AVAvsSS) #not equal
407
408    # Bartlett test of homogeneity of variances
409
410 # data:  PostD.Score by Conditions
411 # Bartlett's K-squared = 11.4962, df = 3, p-value = 0.009324
412
413
414 tapply(AVAvsSS$PostN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
415          A           B           C           D
416 417.64612 137.81757  96.21226  72.30566
417
418 bartlett.test(PostN.Score ~ Conditions, data=AVAvsSS) #not equal
419
```

```
420    # Bartlett test of homogeneity of variances
421
422 # data:  PostN.Score by Conditions
423 # Bartlett's K-squared = 20.8672, df = 3, p-value = 0.0001122
424
425
426 tapply(AVAvsSS$PostO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
427         # A          B          C          D
428 # 327.02457  121.47925   55.26758  104.01198
429
430 bartlett.test(PostO.Score ~ Conditions, data=AVAvsSS) #not equal
431
432    # Bartlett test of homogeneity of variances
433
434 # data:  PostO.Score by Conditions
435 # Bartlett's K-squared = 17.4868, df = 3, p-value = 0.0005611
436
437
438 tapply(AVAvsSS$PostSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
439        # A         B         C         D
440 # 574.0445  183.1043  136.3458  102.9443
441
442 bartlett.test(PostSSQ.Score ~ Conditions, data=AVAvsSS) #not equal
443
444    # Bartlett test of homogeneity of variances
445
446 # data:  PostSSQ.Score by Conditions
447 # Bartlett's K-squared = 20.3056, df = 3, p-value = 0.0001467
448
449 ###Checking wether dependent variables have the same variances between
        groups##
450 ###If the resulting p-value of Levene's test is less than some critical
         value (typically .05),
451 ###the obtained differences in sample variances are unlikely to have
        occurred based on random sampling.
452 ###Thus, the null hypothesis of equal variances is rejected and it is
        concluded that there is a difference
453 ###between the variances in the population. --Wikipedia
454
455 tapply(AVAvsSS$PreD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
456         # A          B          C          D
457 # 1201.6580   399.5279 2182.1787   173.8536
458
459 levene.test(PreD.Score ~ Conditions, data=AVAvsSS)  #equal
460 # Levene's Test for Homogeneity of Variance (center = median)
461       # Df F value Pr(>F)
462 # group  3  1.5193 0.2154
463       # 84
464
465 tapply(AVAvsSS$PreN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
466          # A          B          C          D
467 # 1223.80025   281.26918   611.07789    98.20619
468
469 levene.test(PreN.Score ~ Conditions, data=AVAvsSS) #equal
470 # Levene's Test for Homogeneity of Variance (center = median)
```

```
471         # Df F value Pr(>F)
472 # group   3  1.6647 0.1808
473         # 84
474
475 tapply(AVAvsSS$PreO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
476          # A         B         C         D
477 # 652.2323 145.2826 924.5008 155.3367
478
479 levene.test(PreO.Score ~ Conditions, data=AVAvsSS)  #equal
480 # Levene's Test for Homogeneity of Variance (center = median)
481         # Df F value Pr(>F)
482 # group   3  1.1997  0.315
483         # 84
484
485 tapply(AVAvsSS$PreSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
486           # A          B          C          D
487 # 1213.6593  278.7529 1328.0893  154.8035
488
489 levene.test(PreSSQ.Score ~ Conditions, data=AVAvsSS) #equal
490 # Levene's Test for Homogeneity of Variance (center = median)
491         # Df F value Pr(>F)
492 # group   3  1.3916  0.251
493         # 84
494
495 tapply(AVAvsSS$PostD.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
496          # A         B         C         D
497 # 792.6807 424.4407 452.1216 173.8536
498
499 levene.test(PostD.Score ~ Conditions, data=AVAvsSS) #equal
500 # Levene's Test for Homogeneity of Variance (center = median)
501         # Df F value Pr(>F)
502 # group   3  0.6818 0.5656
503         # 84
504
505 tapply(AVAvsSS$PostN.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
506           # A          B          C          D
507 # 417.64612 137.81757  96.21226  72.30566
508
509 levene.test(PostN.Score ~ Conditions, data=AVAvsSS) #equal
510 # Levene's Test for Homogeneity of Variance (center = median)
511         # Df F value Pr(>F)
512 # group   3  0.3353 0.7998
513         # 84
514
515 tapply(AVAvsSS$PostO.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
516           # A          B          C          D
517 # 327.02457 121.47925  55.26758 104.01198
518
519 levene.test(PostO.Score ~ Conditions, data=AVAvsSS) #equal
520 # Levene's Test for Homogeneity of Variance (center = median)
521         # Df F value Pr(>F)
522 # group   3  0.5956 0.6196
523         # 84
524
525 tapply(AVAvsSS$PostSSQ.Score, AVAvsSS$Conditions, var, na.rm=TRUE)
```

```
526         # A         B         C         D
527 # 574.0445 183.1043 136.3458 102.9443
528
529 levene.test(PostSSQ.Score ~ Conditions, data=AVAvsSS) #equal
530 # Levene's Test for Homogeneity of Variance (center = median)
531       # Df F value Pr(>F)
532 # group  3  0.5392 0.6567
533       # 84
534
535
536
537
538
539
540
541 ##And do the statistical analysis, ref: http://tolstoy.newcastle.edu.au
      /R/e2/help/07/04/14503.html
542 ##Does visual complexity correlate to SS?
543
544 tapply(AVAvsSS$SSQ.Incr, AVAvsSS$Conditions, median, na.rm=TRUE)
545 kruskal.test(SSQ.Incr ~ Conditions, data=AVAvsSS)
546
547 #Kruskal-Wallis rank sum test
548 #data:  SSQ.Incr by Conditions
549 #Kruskal-Wallis chi-squared = 3.1141, df = 3, p-value = 0.3744 #No, it
      does not.
550
551 #General results
552 median(AVAvsSS_A$PostSSQ.Score - AVAvsSS_A$PreSSQ.Score, na.rm=TRUE) #
      median difference
553 wilcox.test(AVAvsSS_A$PreSSQ.Score, AVAvsSS_A$PostSSQ.Score,
      alternative='two.sided', paired=TRUE) #V = 71, p-value = 0.08015
554 mean(rowMeans(sign(outer(AVAvsSS_A$PreSSQ.Score, AVAvsSS_A$PostSSQ.
      Score, FUN="-")))) #0.21 Cliff's d
555
556 median(AVAvsSS_B$PostSSQ.Score - AVAvsSS_B$PreSSQ.Score, na.rm=TRUE) #
      median difference
557 wilcox.test(AVAvsSS_B$PreSSQ.Score, AVAvsSS_B$PostSSQ.Score,
      alternative='two.sided', paired=TRUE) #V = 22.5, p-value = 0.3728
558 mean(rowMeans(sign(outer(AVAvsSS_B$PreSSQ.Score, AVAvsSS_B$PostSSQ.
      Score, FUN="-")))) #-0.055 Cliff's d
559
560 median(AVAvsSS_C$PostSSQ.Score - AVAvsSS_C$PreSSQ.Score, na.rm=TRUE) #
      median difference
561 wilcox.test(AVAvsSS_C$PreSSQ.Score, AVAvsSS_C$PostSSQ.Score,
      alternative='two.sided', paired=TRUE) #V = 38.5, p-value = 0.2829
562 mean(rowMeans(sign(outer(AVAvsSS_C$PreSSQ.Score, AVAvsSS_C$PostSSQ.
      Score, FUN="-")))) #0.08 Cliff's d
563
564 median(AVAvsSS_D$PostSSQ.Score - AVAvsSS_D$PreSSQ.Score, na.rm=TRUE) #
      median difference
565 wilcox.test(AVAvsSS_D$PreSSQ.Score, AVAvsSS_D$PostSSQ.Score,
      alternative='two.sided', paired=TRUE) #V = 46, p-value = 0.6083
566 mean(rowMeans(sign(outer(AVAvsSS_D$PreSSQ.Score, AVAvsSS_D$PostSSQ.
      Score, FUN="-")))) #0.015 Cliff's d
```

```
567
568 #Gender Difference
569 tapply(AVAvsSS$SSQ.Incr, AVAvsSS$Gender, median, na.rm=TRUE)
570 wilcox.test(SSQ.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #
    W = 944.5, p-value = 0.8422
571 mean(rowMeans(sign(outer(AVAvsSS_F$SSQ.Incr, AVAvsSS_M$SSQ.Incr, FUN="-
    "))))  #NaN Cliff's d
572
573 wilcox.test(N.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 921, p-value = 0.6525
574 mean(rowMeans(sign(outer(AVAvsSS_F$N.Incr, AVAvsSS_M$N.Incr, FUN="-")))
    )  #NaN Cliff's d
575
576 wilcox.test(D.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 966, p-value = 0.9925
577 mean(rowMeans(sign(outer(AVAvsSS_F$D.Incr, AVAvsSS_M$D.Incr, FUN="-")))
    )  #NaN Cliff's d
578
579 wilcox.test(O.Incr ~ Gender, alternative="two.sided", data=AVAvsSS) #W
    = 949, p-value = 0.8727
580 mean(rowMeans(sign(outer(AVAvsSS_F$O.Incr, AVAvsSS_M$O.Incr, FUN="-")))
    )  #NaN Cliff's d
```

G.2.2 Codes for analysing Learning Outcomes

Listing: AVAvsCAL[2010-10-05][0917].r

```
1  #Clearing workspace (and variables too)#################################
2  rm(list=ls())
3
4  #Setting up the working directory#######################################
5  setwd("/home/ephraim/el_thesis/results/caseStudy2")
6
7  #loading required library###############################################
8
9  library(car)
10 library(psychometric)
11 library(Hmisc)
12 library(psych)
13 library(Cairo)
14 library(lattice)
15 library(mgcv)
16 require(gplots)
17 library(xtable)
18
19 #nonparametic analysis packages
20 library(npmc)
21 library(vegan) #use adonis(...) for nonparametric manova
22
23 # Loading the results###################################################
24 results <-
```

```
  read.table("resultsFromExperiment2-VillageFirefighting.csv",
   header=TRUE,
   sep="\t",
   na.strings="NA",
   dec=".",
   strip.white=TRUE)

#removing cases with NA entries into a new dataframe
results <- na.omit(results)

#Creating a dataframe
results.df <- data.frame(results)
attach(results.df)
names(results.df)

AVAvsCAL <- data.frame(subset(results.df,
   Usable == "Y",
   select=c(Conditions,
       Gender,
       Age,
       Participant,
       PreTR.Score,
       PreTT.Score,
       Presence.Score,
       PAQ.PU.x.,
       PAQ.AS.y.,
       Affective.Score,
       CognitiveLoad.Score,
       PostTR.Score,
       PostTT.Score,
       TR.Incr,
       TT.Incr
       )
   )
   )

AVAvsCAL_A <- subset(AVAvsCAL, Conditions == "A")
AVAvsCAL_B <- subset(AVAvsCAL, Conditions == "B")
AVAvsCAL_C <- subset(AVAvsCAL, Conditions == "C")
AVAvsCAL_D <- subset(AVAvsCAL, Conditions == "D")
AVAvsCAL_M <- subset(AVAvsCAL, Gender == "m")
AVAvsCAL_F <- subset(AVAvsCAL, Gender == "f")

#####################################################################
#Displaying the summary of the dataframe
#####################################################################
#Summary in overall
summary(AVAvsCAL)
#Summary by groups
summary(subset(AVAvsCAL, Conditions == 'A', digits = 2))
summary(subset(AVAvsCAL, Conditions == 'B', digits = 2))
summary(subset(AVAvsCAL, Conditions == 'C', digits = 2))
summary(subset(AVAvsCAL, Conditions == 'D', digits = 2))

##Describing the data
```

```
describe(AVAvsCAL,
#   digits = 2,
    na.rm = TRUE,
    interp=FALSE,
    skew = TRUE,
    ranges = TRUE,
    trim=.1
    )
##Describing the data by groups
describe.by(AVAvsCAL, AVAvsCAL$Conditions,
#   digits = 2,
    na.rm = TRUE,
    interp=FALSE,
    skew = TRUE,
    ranges = TRUE,
    trim=.1
    )

########################################################################
#Plotting the graphs and saving them on file
########################################################################
###Boxplot of CAL
##Save it on file for publication...
Cairo(2000, 900, file="AVAvsCAL_boxplots.pdf", type="pdf", bg="white",
    dpi = 150, units="px")
par(mfrow = c(2, 6))
boxplot(PreTR.Score ~ Conditions, data = AVAvsCAL,
        main="Pretest retention",
        xlab="Experimental conditions",
        ylab="Pretest retention"
    )
boxplot(PreTT.Score ~ Conditions, data = AVAvsCAL,
        main="Pretest transfer",
        xlab="Experimental conditions",
        ylab="Pretest transfer"
    )
boxplot(Presence.Score ~ Conditions, data = AVAvsCAL,
    main="Presence",
        xlab="Experimental conditions",
        ylab="Presence"
    )
boxplot(PAQ.PU.x. ~ Conditions, data = AVAvsCAL,
    main="Pleasant-unpleasant",
        xlab="Experimental conditions",
        ylab="Pleasant-unpleasant"
    )
boxplot(PAQ.AS.y. ~ Conditions, data = AVAvsCAL,
    main="Arousal-sleepy",
        xlab="Experimental conditions",
        ylab="Arousal-sleepy"
    )
boxplot(Affective.Score ~ Conditions, data = AVAvsCAL,
    main="Affective",
        xlab="Experimental conditions",
        ylab="Affective score"
```

```
134     )
135 boxplot(CognitiveLoad.Score ~ Conditions, data = AVAvsCAL,
136     main="Cognitive load",
137           xlab="Experimental conditions",
138           ylab="Cognitive load score"
139     )
140 boxplot(PostTR.Score ~ Conditions, data = AVAvsCAL,
141           main="Posttest retention",
142           xlab="Experimental conditions",
143           ylab="Posttest retention"
144     )
145 boxplot(PostTT.Score ~ Conditions, data = AVAvsCAL,
146           main="Posttest transfer",
147           xlab="Experimental conditions",
148           ylab="Posttest transfer"
149     )
150 boxplot(TR.Incr ~ Conditions, data = AVAvsCAL,
151     main="Improvement in retention",
152           xlab="Experimental conditions",
153           ylab="Improvement in retention"
154     )
155 boxplot(TT.Incr ~ Conditions, data = AVAvsCAL,
156     main="Improvement in transfer",
157           xlab="Experimental conditions",
158           ylab="Improvement in transfer"
159     )
160 dev.off()
161
162
163 ###Plotting histogram of each dependant variables
164 ##Save on file...dev.new
165 ##Save it on file for publication...
166 Cairo(1280, 800, file="plot-AVAvsCAL-histograms.png", type="png", bg="
        white")
167 par(mfrow = c(3, 4))
168 hist(AVAvsCAL$PreTR.Score,
169     #scale="frequency",
170     breaks="Sturges",
171     col="darkgray",
172     xlab ="Pretest retention")
173 hist(AVAvsCAL$PreTT.Score,
174     #scale="frequency",
175     breaks="Sturges",
176     col="darkgray",
177     xlab ="Pretest transfer")
178 hist(AVAvsCAL$Presence.Score,
179     #scale="frequency",
180     breaks="Sturges",
181     col="darkgray",
182     xlab ="Pre-SSQ: Oculomotor")
183 hist(AVAvsCAL$PAQ.PU.x.,
184     #scale="frequency",
185     breaks="Sturges",
186     col="darkgray",
187     xlab ="Pleasant-unpleasant scale")
```

```
188 hist(AVAvsCAL$PAQ.AS.y.,
189     #scale="frequency",
190     breaks="Sturges",
191     col="darkgray",
192     xlab ="Arousal-sleepy")
193 hist(AVAvsCAL$Affective.Score,
194     #scale="frequency",
195     breaks="Sturges",
196     col="darkgray",
197     xlab ="Affective score")
198 hist(AVAvsCAL$CognitiveLoad.Score,
199     #scale="frequency",
200     breaks="Sturges",
201     col="darkgray",
202     xlab ="Cognitive Load")
203 hist(AVAvsCAL$PostTR.Score,
204     #scale="frequency",
205     breaks="Sturges",
206     col="darkgray",
207     xlab ="Posttest retention")
208 hist(AVAvsCAL$PostTT.Score,
209     #scale="frequency",
210     breaks="Sturges",
211     col="darkgray",
212     xlab ="Posttest transfer")
213 hist(AVAvsCAL$TR.Incr,
214     #scale="frequency",
215     breaks="Sturges",
216     col="darkgray",
217     xlab ="Improvement in retention")
218 hist(AVAvsCAL$TT.Incr,
219     #scale="frequency",
220     breaks="Sturges",
221     col="darkgray",
222     xlab ="Improvement in transfer")
223 dev.off()
224
225 ######################################################################
226 #Normality and Variance check
227 ######################################################################
228 ###Normality test###
229 #Shapiro-Wilk Normality Test. When the p-value is higher than
        significance levels
230 #the sample forms a gaussian (bell-shaped) distribution.
231 #Thode(2002) reviewed over forty procedures for assesing goodness-of-
        fit and concludes
232 #that Shapiro and Wilk (1965, 1968) and test statistics derived from
        moments
233 #are generally the best methods for assessing normality
234 # Thode, H. C. (2002). Testing for normality. New York: Marcel Dekker.
235 shapiro.test(AVAvsCAL$PreTR.Score) #Not normal
236
237     # Shapiro-Wilk normality test
238
239 # data:  AVAvsCAL$PreTR.Score
```

```
# W = 0.923, p-value = 6.152e-05


shapiro.test(AVAvsCAL$PreTT.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PreTT.Score
# W = 0.8667, p-value = 2.219e-07


shapiro.test(AVAvsCAL$Presence.Score) #Normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$Presence.Score
# W = 0.9866, p-value = 0.5045


shapiro.test(AVAvsCAL$PAQ.PU.x.) #Normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PAQ.PU.x.
# W = 0.9855, p-value = 0.4366


shapiro.test(AVAvsCAL$PAQ.AS.y.) #Normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$PAQ.AS.y.
# W = 0.9893, p-value = 0.6971


shapiro.test(AVAvsCAL$Affective.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$Affective.Score
# W = 0.9644, p-value = 0.0162


shapiro.test(AVAvsCAL$CognitiveLoad.Score) #Not normal

    # Shapiro-Wilk normality test

# data:  AVAvsCAL$CognitiveLoad.Score
# W = 0.9454, p-value = 0.00103


shapiro.test(AVAvsCAL$PostTR.Score) #Not normal

    # Shapiro-Wilk normality test

```

```
# data:  AVAvsCAL$PostTR.Score
# W = 0.9505, p-value = 0.002081


shapiro.test(AVAvsCAL$PostTT.Score) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsCAL$PostTT.Score
# W = 0.9288, p-value = 0.0001218


shapiro.test(AVAvsCAL$TR.Incr) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsCAL$TR.Incr
# W = 0.9713, p-value = 0.04744


shapiro.test(AVAvsCAL$TT.Incr) #Not normal

   # Shapiro-Wilk normality test

# data:  AVAvsCAL$TT.Incr
# W = 0.9372, p-value = 0.0003489



###Checking wether dependent variables have the same variances between
    groups##
###If Bartlett test s p-value is less than 0.05
###then the assumption of equal variances would be rejected.
tapply(AVAvsCAL$PreTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
        # A         B         C         D
# 47.11687  52.80429  50.75079  55.51755

bartlett.test(PreTR.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PreTR.Score by Conditions
# Bartlett's K-squared = 0.1535, df = 3, p-value = 0.9847


tapply(AVAvsCAL$PreTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
         # A         B         C         D
 # 92.97564 103.42250  51.87646 133.84104

bartlett.test(PreTT.Score ~ Conditions, data=AVAvsCAL) #equal

   # Bartlett test of homogeneity of variances

# data:  PreTT.Score by Conditions
# Bartlett's K-squared = 4.4572, df = 3, p-value = 0.2161
```

```
349
350
351 tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
352         # A         B         C         D
353 # 38.96838  54.71429  33.52857  48.03953
354
355 bartlett.test(Presence.Score ~ Conditions, data=AVAvsCAL) #equal
356
357    # Bartlett test of homogeneity of variances
358
359 # data:  Presence.Score by Conditions
360 # Bartlett's K-squared = 1.402, df = 3, p-value = 0.7051
361
362
363 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Conditions, var, na.rm=TRUE)
364         # A         B         C         D
365 # 101.2490  119.8905  184.0905  142.7194
366
367 bartlett.test(PAQ.PU.x. ~ Conditions, data=AVAvsCAL) #equal
368
369    # Bartlett test of homogeneity of variances
370
371 # data:  PAQ.PU.x. by Conditions
372 # Bartlett's K-squared = 2.0248, df = 3, p-value = 0.5673
373
374
375 tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Conditions, var, na.rm=TRUE)
376         # A         B         C         D
377  # 87.67984  103.24762  108.41429   71.38340
378
379 bartlett.test(PAQ.AS.y. ~ Conditions, data=AVAvsCAL) #equal
380
381    # Bartlett test of homogeneity of variances
382
383 # data:  PAQ.AS.y. by Conditions
384 # Bartlett's K-squared = 1.0786, df = 3, p-value = 0.7822
385
386
387 tapply(AVAvsCAL$Affective.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
388         # A         B         C         D
389 # 0.5551625  0.3889262  0.6930362  0.3636628
390
391 bartlett.test(Affective.Score ~ Conditions, data=AVAvsCAL) #equal
392
393    # Bartlett test of homogeneity of variances
394
395 # data:  Affective.Score by Conditions
396 # Bartlett's K-squared = 2.8466, df = 3, p-value = 0.4159
397
398
399 tapply(AVAvsCAL$CognitiveLoad.Score, AVAvsCAL$Conditions, var, na.rm=
    TRUE)
400         # A         B         C         D
401 # 0.4644269  0.7619048  0.6619048  0.6511858
402
```

```
403 bartlett.test(CognitiveLoad.Score ~ Conditions, data=AVAvsCAL) #equal
404
405    # Bartlett test of homogeneity of variances
406
407 # data:  CognitiveLoad.Score by Conditions
408 # Bartlett's K-squared = 1.3222, df = 3, p-value = 0.7239
409
410
411 tapply(AVAvsCAL$PostTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
412         # A         B         C         D
413 # 116.51380 120.56979  75.83282  89.24203
414
415 bartlett.test(PostTR.Score ~ Conditions, data=AVAvsCAL) #not equal
416
417    # Bartlett test of homogeneity of variances
418
419 # data:  PostTR.Score by Conditions
420 # Bartlett's K-squared = 1.4493, df = 3, p-value = 0.694
421
422
423 tapply(AVAvsCAL$PostTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
424         # A         B         C         D
425  # 77.61683 121.92621  82.60583  92.97564
426
427 bartlett.test(PostTT.Score ~ Conditions, data=AVAvsCAL) #equal
428
429    # Bartlett test of homogeneity of variances
430
431 # data:  PostTT.Score by Conditions
432 # Bartlett's K-squared = 1.2527, df = 3, p-value = 0.7404
433
434
435 tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
436         # A         B         C         D
437 # 181.77127 118.80964  72.75257  60.14401
438
439 bartlett.test(TR.Incr ~ Conditions, data=AVAvsCAL) #not equal
440
441    # Bartlett test of homogeneity of variances
442
443 # data:  TR.Incr by Conditions
444 # Bartlett's K-squared = 8.0093, df = 3, p-value = 0.04582
445
446
447 tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
448         # A         B         C         D
449 # 175.80350 143.07330  56.50239 193.08216
450
451 bartlett.test(TT.Incr ~ Conditions, data=AVAvsCAL) #not equal
452
453    # Bartlett test of homogeneity of variances
454
455 # data:  TT.Incr by Conditions
456 # Bartlett's K-squared = 7.9158, df = 3, p-value = 0.04778
457
```

```
458
459
460 ###Checking wether dependent variables have the same variances between
        groups##
461 ###If the resulting p-value of Levene's test is less than some critical
        value (typically .05),
462 ###the obtained differences in sample variances are unlikely to have
        occurred based on random sampling.
463 ###Thus, the null hypothesis of equal variances is rejected and it is
        concluded that there is a difference
464 ###between the variances in the population. --Wikipedia
465
466 tapply(AVAvsCAL$PreTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
467         # A         B         C         D
468 # 47.11687  52.80429  50.75079  55.51755
469
470 levene.test(PreTR.Score ~ Conditions, data=AVAvsCAL) #equal
471 # Levene's Test for Homogeneity of Variance (center = median)
472         # Df F value Pr(>F)
473 # group  3  0.1505 0.9291
474         # 84
475
476 tapply(AVAvsCAL$PreTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
477          # A          B          C          D
478  # 92.97564  103.42250   51.87646  133.84104
479
480 levene.test(PreTT.Score ~ Conditions, data=AVAvsCAL) #equal
481 # Levene's Test for Homogeneity of Variance (center = median)
482         # Df F value Pr(>F)
483 # group  3  0.3284 0.8048
484         # 84
485
486 tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
487         # A         B         C         D
488 # 38.96838  54.71429  33.52857  48.03953
489
490 levene.test(Presence.Score ~ Conditions, data=AVAvsCAL)  #equal
491 # Levene's Test for Homogeneity of Variance (center = median)
492         # Df F value Pr(>F)
493 # group  3  0.2939 0.8297
494         # 84
495
496 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Conditions, var, na.rm=TRUE)
497         # A         B         C         D
498 # 101.2490  119.8905  184.0905  142.7194
499
500 levene.test(PAQ.PU.x. ~ Conditions, data=AVAvsCAL) #equal
501 # Levene's Test for Homogeneity of Variance (center = median)
502         # Df F value Pr(>F)
503 # group  3  0.3943 0.7574
504         # 84
505
506 tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Conditions, var, na.rm=TRUE)
507          # A          B          C          D
508  # 87.67984  103.24762  108.41429   71.38340
```

```
509
510 levene.test(PAQ.AS.y. ~ Conditions, data=AVAvsCAL) #equal
511 # Levene's Test for Homogeneity of Variance (center = median)
512       # Df F value Pr(>F)
513 # group  3  0.1892 0.9035
514       # 84
515
516 tapply(AVAvsCAL$Affective.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
517         # A         B         C         D
518 # 0.5551625 0.3889262 0.6930362 0.3636628
519
520 levene.test(Affective.Score ~ Conditions, data=AVAvsCAL) #equal
521 # Levene's Test for Homogeneity of Variance (center = median)
522       # Df F value Pr(>F)
523 # group  3  0.5609 0.6423
524       # 84
525
526 tapply(AVAvsCAL$CognitiveLoad.Score, AVAvsCAL$Conditions, var, na.rm=
    TRUE)
527         # A         B         C         D
528 # 0.4644269 0.7619048 0.6619048 0.6511858
529
530 levene.test(CognitiveLoad.Score ~ Conditions, data=AVAvsCAL) #equal
531 # Levene's Test for Homogeneity of Variance (center = median)
532       # Df F value Pr(>F)
533 # group  3  0.4577 0.7125
534       # 84
535
536 tapply(AVAvsCAL$PostTR.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
537         # A         B         C         D
538 # 116.51380 120.56979  75.83282  89.24203
539
540 levene.test(PostTR.Score ~ Conditions, data=AVAvsCAL) #equal
541 # Levene's Test for Homogeneity of Variance (center = median)
542       # Df F value Pr(>F)
543 # group  3  0.1545 0.9265
544       # 84
545
546 tapply(AVAvsCAL$PostTT.Score, AVAvsCAL$Conditions, var, na.rm=TRUE)
547         # A         B         C         D
548  # 77.61683 121.92621  82.60583  92.97564
549
550 levene.test(PostTT.Score ~ Conditions, data=AVAvsCAL) #equal
551 # Levene's Test for Homogeneity of Variance (center = median)
552       # Df F value Pr(>F)
553 # group  3  0.2342 0.8723
554       # 84
555
556 tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
557         # A         B         C         D
558 # 181.77127 118.80964  72.75257  60.14401
559
560 levene.test(TR.Incr ~ Conditions, data=AVAvsCAL) #not equal
561 # Levene's Test for Homogeneity of Variance (center = median)
562       # Df F value  Pr(>F)
```

```
563 # group   3   2.8011 0.04486 *
564       # 84
565 # ---
566 # Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
567
568 tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Conditions, var, na.rm=TRUE)
569         # A          B          C          D
570 # 175.80350 143.07330  56.50239 193.08216
571
572 levene.test(TT.Incr ~ Conditions, data=AVAvsCAL) #equal
573 # Levene's Test for Homogeneity of Variance (center = median)
574       # Df F value Pr(>F)
575 # group   3   1.416 0.2438
576       # 84
577
578 ##And do the statistical analysis
579 ##Ref http://tolstoy.newcastle.edu.au/R/e2/help/07/04/14503.html
580 ######################################################################
581 #Research questions#####################################################
582 ######################################################################
583
584 ###Does visual complexity of AVA deteriorate cognitive learning?
585 ##Kruskal-Wallis rank sum test - retention
586 kruskal.test(TR.Incr ~ Conditions, data = AVAvsCAL) #Kruskal-Wallis chi
    -squared = 8.2494, df = 3, p-value = 0.04113 [Yes]
587
588 TR.Incr_comp <- data.frame(subset(results.df,
589     Usable == "Y",
590     select=c(Conditions,
591         TR.Incr
592         )
593     )
594     )
595 #must rename the heading to class and var, then perform npcm
596 names(TR.Incr_comp)[1] <- 'class'
597 names(TR.Incr_comp)[2] <- 'var'
598 summary(npmc(TR.Incr_comp), type="BF")
599
600 #$`Results of the multiple Behrens-Fisher-Test`
601 #  cmp    effect    lower.cl  upper.cl p.value.1s  p.value.2s
602 #1 1-2 0.4151139 0.17809985 0.6521279  0.9998919 0.775584136
603 #2 1-3 0.5341615 0.28871514 0.7796078  0.8436452 0.988161659
604 #3 1-4 0.3497164 0.11924027 0.5801926  0.9999999 0.306003939
605 #4 2-3 0.6598639 0.42088742 0.8988405  0.1677704 0.285068974
606 #5 2-4 0.4482402 0.20534526 0.6911351  0.9980456 0.948428163
607 #6 3-4 0.2422360 0.05181377 0.4326583  1.0000000 0.004767127
608
609 ##There is a significant improvement in Retention
610 #Overall
611 median(AVAvsCAL$PostTR.Score - AVAvsCAL$PreTR.Score, na.rm=TRUE) #
    median difference 22.2
612 wilcox.test(AVAvsCAL$PostTR.Score, AVAvsCAL$PreTR.Score, alternative='
    two.sided', paired=TRUE) #V = 3727, p-value = 1.194e-15
613
614 #Individual group
```

```
615 median(AVAvsCAL_A$PostTR.Score - AVAvsCAL_A$PreTR.Score, na.rm=TRUE) #
    median difference, 22.2
616 wilcox.test(AVAvsCAL_A$PostTR.Score, AVAvsCAL_A$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 249, p-value = 7.292e-05
617 mean(rowMeans(sign(outer(AVAvsCAL_A$PostTR.Score, AVAvsCAL_A$PreTR.
    Score, FUN="-")))) #d=0.86
618
619 median(AVAvsCAL_B$PostTR.Score - AVAvsCAL_B$PreTR.Score, na.rm=TRUE) #
    median difference, 16.65
620 wilcox.test(AVAvsCAL_B$PostTR.Score, AVAvsCAL_B$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 210, p-value = 9.449e-05
621 mean(rowMeans(sign(outer(AVAvsCAL_B$PostTR.Score, AVAvsCAL_B$PreTR.
    Score, FUN="-")))) #d=0.84
622
623 median(AVAvsCAL_C$PostTR.Score - AVAvsCAL_C$PreTR.Score, na.rm=TRUE) #
    median difference, 22.2
624 wilcox.test(AVAvsCAL_C$PostTR.Score, AVAvsCAL_C$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 231, p-value = 5.663e-05
625 mean(rowMeans(sign(outer(AVAvsCAL_C$PostTR.Score, AVAvsCAL_C$PreTR.
    Score, FUN="-")))) #d=0.96
626
627 median(AVAvsCAL_D$PostTR.Score - AVAvsCAL_D$PreTR.Score, na.rm=TRUE) #
    median difference, 16.65
628 wilcox.test(AVAvsCAL_D$PostTR.Score, AVAvsCAL_D$PreTR.Score,
    alternative='two.sided', paired=TRUE) #V = 276, p-value = 2.753e-05
629 mean(rowMeans(sign(outer(AVAvsCAL_D$PostTR.Score, AVAvsCAL_D$PreTR.
    Score, FUN="-")))) #d=0.84
630
631 ##Kruskal-Wallis rank sum test - transfer
632 kruskal.test(TT.Incr ~ Conditions, data = AVAvsCAL) #Kruskal-Wallis chi
    -squared = 2.4934, df = 3, p-value = 0.4765 [No]
633
634 ##There is a significant improvement in Transfer
635 #Overall
636 median(AVAvsCAL$PostTT.Score - AVAvsCAL$PreTT.Score, na.rm=TRUE) #
    median difference 8.33
637 wilcox.test(AVAvsCAL$PostTT.Score, AVAvsCAL$PreTT.Score, alternative='
    two.sided', paired=TRUE) #V = 2290.5, p-value = 3.573e-08 [There is
     a significance, but which one?]
638
639 #Individual group
640 median(AVAvsCAL_A$PostTT.Score - AVAvsCAL_A$PreTT.Score, na.rm=TRUE) #
    median difference, 16.66
641 wilcox.test(AVAvsCAL_A$PostTT.Score, AVAvsCAL_A$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 226, p-value = 0.001233
642 mean(rowMeans(sign(outer(AVAvsCAL_A$PostTT.Score, AVAvsCAL_A$PreTT.
    Score, FUN="-")))) #d=0.63
643
644 median(AVAvsCAL_B$PostTT.Score - AVAvsCAL_B$PreTT.Score, na.rm=TRUE) #
    median difference, 8.33
645 wilcox.test(AVAvsCAL_B$PostTT.Score, AVAvsCAL_B$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 120.5, p-value =
    0.006724
646 mean(rowMeans(sign(outer(AVAvsCAL_B$PostTT.Score, AVAvsCAL_B$PreTT.
    Score, FUN="-")))) #d=0.51
```

```
647
648 median(AVAvsCAL_C$PostTT.Score - AVAvsCAL_C$PreTT.Score, na.rm=TRUE) #
    median difference, 8.33
649 wilcox.test(AVAvsCAL_C$PostTT.Score, AVAvsCAL_C$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 136, p-value = 0.0004173
650 mean(rowMeans(sign(outer(AVAvsCAL_C$PostTT.Score, AVAvsCAL_C$PreTT.
    Score, FUN="-")))) #d=0.63
651
652 median(AVAvsCAL_D$PostTT.Score - AVAvsCAL_D$PreTT.Score, na.rm=TRUE) #
    median difference, 8.33 [BUT NOT THIS ONE]
653 wilcox.test(AVAvsCAL_D$PostTT.Score, AVAvsCAL_D$PreTT.Score,
    alternative='two.sided', paired=TRUE) #V = 122.5, p-value = 0.1094
654 mean(rowMeans(sign(outer(AVAvsCAL_D$PostTT.Score, AVAvsCAL_D$PreTT.
    Score, FUN="-")))) #d=0.37
655
656 ###Does visual complexity of AVA alters affective component in learning
    ?
657 kruskal.test(Presence.Score ~ Conditions, data = AVAvsCAL) #Kruskal-
    Wallis chi-squared = 16.1833, df = 3, p-value = 0.00104 [YES]
658 Presence_comp <- data.frame(subset(results.df,
659    Usable == "Y",
660    select=c(Conditions,
661       Presence.Score
662       )
663    )
664    )
665 #must rename the heading to class and var, then perform npcm
666 names(Presence_comp)[1] <- 'class'
667 names(Presence_comp)[2] <- 'var'
668 summary(npmc(Presence_comp), type="BF")
669 #$`Results of the multiple Behrens-Fisher-Test`
670 #  cmp     effect  lower.cl  upper.cl   p.value.1s  p.value.2s
671 #1 1-2 0.4813665 0.2380648 0.7246681 0.9917919746 0.999109731
672 #2 1-3 0.7867495 0.5927613 0.9807377 0.0007114924 0.001673160
673 #3 1-4 0.6984877 0.4855002 0.9114752 0.0425652486 0.075340416
674 #4 2-3 0.7823129 0.5859585 0.9786674 0.0015081190 0.002222835
675 #5 2-4 0.6821946 0.4566288 0.9077604 0.0886847714 0.149366495
676 #6 3-4 0.3685300 0.1285810 0.6084791 0.9999999679 0.468761761
677
678
679
680 kruskal.test(PAQ.PU.x.~ Conditions, data = AVAvsCAL) #Kruskal-Wallis
    chi-squared = 5.5827, df = 3, p-value = 0.1338 [no]
681
682
683 kruskal.test(PAQ.AS.y.~ Conditions, data = AVAvsCAL) #Kruskal-Wallis
    chi-squared = 13.3227, df = 3, p-value = 0.003988 [Yes]
684 PAQ.AS.y._comp <- data.frame(subset(results.df,
685    Usable == "Y",
686    select=c(Conditions,
687       PAQ.AS.y.
688       )
689    )
690    )
691 #must rename the heading to class and var, then perform npcm
```

```
692 names(PAQ.AS.y._comp)[1] <- 'class'
693 names(PAQ.AS.y._comp)[2] <- 'var'
694 summary(npmc(PAQ.AS.y._comp), type="BF")
695 #$`Results of the multiple Behrens-Fisher-Test`
696 #  cmp     effect  lower.cl  upper.cl  p.value.1s  p.value.2s
697 #1 1-2 0.5538302 0.3151496 0.7925109 0.772044751 0.936072530
698 #2 1-3 0.7556936 0.5517502 0.9596370 0.005223274 0.009264696
699 #3 1-4 0.7438563 0.5488230 0.9388897 0.005032619 0.009748830
700 #4 2-3 0.6927438 0.4618120 0.9236755 0.078660120 0.130305001
701 #5 2-4 0.6873706 0.4633412 0.9114000 0.076502191 0.130485787
702 #6 3-4 0.4472050 0.2056141 0.6887958 0.999541331 0.941275565
703
704 kruskal.test(Affective.Score ~ Conditions, data = AVAvsCAL) #Kruskal-
        Wallis chi-squared = 6.2946, df = 3, p-value = 0.09813 [no]
705
706
707 kruskal.test(CognitiveLoad.Score ~ Conditions, data = AVAvsCAL) #
        Kruskal-Wallis chi-squared = 9.8473, df = 3, p-value = 0.01991 [YES
        ]
708 CognitiveLoad.Score_comp <- data.frame(subset(results.df,
709     Usable == "Y",
710     select=c(Conditions,
711         CognitiveLoad.Score
712         )
713     )
714     )
715 #must rename the heading to class and var, then perform npcm
716 names(CognitiveLoad.Score_comp)[1] <- 'class'
717 names(CognitiveLoad.Score_comp)[2] <- 'var'
718 summary(npmc(CognitiveLoad.Score_comp), type="BF")
719 #$`Results of the multiple Behrens-Fisher-Test`
720 #  cmp     effect    lower.cl  upper.cl p.value.1s p.value.2s
721 #1 1-2 0.6884058 0.46951896 0.9072926  0.0657492 0.11234214
722 #2 1-3 0.3664596 0.13093734 0.6019819  1.0000000 0.43143172
723 #3 1-4 0.5311909 0.29984539 0.7625365  0.8866689 0.98724615
724 #4 2-3 0.2551020 0.04465888 0.4655452  1.0000000 0.01737742
725 #5 2-4 0.3592133 0.13345931 0.5849672  1.0000000 0.34685079
726 #6 3-4 0.6376812 0.40882704 0.8665353  0.2480676 0.37864838
727
728 ##Gender effect
729 tapply(AVAvsCAL$Presence.Score, AVAvsCAL$Gender, median, na.rm=TRUE)
730 wilcox.test(Presence.Score ~ Gender, alternative="two.sided", data=
        AVAvsCAL) #W = 832, p-value = 0.2591 [no]
731 mean(rowMeans(sign(outer(AVAvsCAL_F$Presence.Score, AVAvsCAL_M$Presence
        .Score, FUN="-"))))
732
733 tapply(AVAvsCAL$PAQ.AS.y., AVAvsCAL$Gender, median, na.rm=TRUE)
734 wilcox.test(PAQ.AS.y. ~ Gender, alternative="two.sided", data=AVAvsCAL)
        #W = 806.5, p-value = 0.1799 [no]
735 mean(rowMeans(sign(outer(AVAvsCAL_F$PAQ.AS.y., AVAvsCAL_M$PAQ.AS.y.,
        FUN="-"))))
736
737 tapply(AVAvsCAL$PAQ.PU.x., AVAvsCAL$Gender, median, na.rm=TRUE)
738 wilcox.test(PAQ.PU.x. ~ Gender, alternative="two.sided", data=AVAvsCAL)
```

```
739 mean(rowMeans(sign(outer(AVAvsCAL_F$PAQ.PU.x., AVAvsCAL_M$PAQ.PU.x.,
        FUN="-")))) #W = 765.5, p-value = 0.09235 [no]
740
741 tapply(AVAvsCAL$TR.Incr, AVAvsCAL$Gender, median, na.rm=TRUE)
742 wilcox.test(TR.Incr ~ Gender, alternative="two.sided", data=AVAvsCAL) #
        W = 906.5, p-value = 0.6086 [no]
743
744 tapply(AVAvsCAL$TT.Incr, AVAvsCAL$Gender, median, na.rm=TRUE)
745 wilcox.test(TT.Incr ~ Gender, alternative="two.sided", data=AVAvsCAL) #
        W = 900.5, p-value = 0.5691 [no]
746
747 ###Does affective component of simulation correlate to cognitive
        learning?
748 #Ref http://myowelt.blogspot.com/2008/04/beautiful-correlation-tables-
        in-r.html
749 corstarsl <- function(x){
750     require(Hmisc)
751     x <- as.matrix(x)
752     R <- rcorr(x, type="spearman")$r
753     p <- rcorr(x, type="spearman")$P
754
755     ## define notions for significance levels; spacing is important.
756     mystars <- ifelse(p < .001, "***", ifelse(p < .01, "** ", ifelse(p <
            .05, "* ", " ")))
757
758     ## trunctuate the matrix that holds the correlations to two decimal
759     R <- format(round(cbind(rep(-1.11, ncol(x)), R), 2))[,-1]
760
761     ## build a new matrix that includes the correlations with their
            apropriate stars
762     Rnew <- matrix(paste(R, mystars, sep=""), ncol=ncol(x))
763     diag(Rnew) <- paste(diag(R), " ", sep="")
764     rownames(Rnew) <- colnames(x)
765     colnames(Rnew) <- paste(colnames(x), "", sep="")
766
767     ## remove upper triangle
768     Rnew <- as.matrix(Rnew)
769     Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
770     Rnew <- as.data.frame(Rnew)
771
772     ## remove last column and return the matrix (which is now a data
            frame)
773     Rnew <- cbind(Rnew[1:length(Rnew)-1])
774     return(Rnew)
775     }
776
777 corstarsl(AVAvsCAL[,c("Presence.Score",
778         "PAQ.AS.y.",
779         "PAQ.PU.x.",
780         "Affective.Score",
781         "CognitiveLoad.Score",
782         "TR.Incr",
783         "TT.Incr")])
784
```

```
785 #                          Presence.Score PAQ.AS.y. PAQ.PU.x. Affective.
        Score CognitiveLoad.Score TR.Incr
786 #Presence.Score
787 #PAQ.AS.y.                  0.52***
788 #PAQ.PU.x.                  0.47***    0.30**
789 #Affective.Score            0.52***    0.60***    0.38***
790 #CognitiveLoad.Score       -0.45***   -0.08      -0.33**              -0.18
791 #TR.Incr                     0.12       0.01       0.14                0.11
                  -0.28**
792 #TT.Incr                    -0.08      -0.31**    -0.10               -0.14
                     -0.20     0.26*
793
794 #Prints out the laTex code
795 xtable(corstarsl(AVAvsCAL[,c("Presence.Score",
796        "PAQ.AS.y.",
797        "PAQ.PU.x.",
798        "Affective.Score",
799        "CognitiveLoad.Score",
800        "TR.Incr",
801        "TT.Incr")]))
802
803
804
805
806
807 #Scatterplot Matrix#########################################################
808 #Saving it in a file first, because it is very big
809 Cairo(1280, 1280, file="AVAvsCAL-scatterPlotMatrix", type="png", bg="
        white")
810 par(mfrow = c(1, 1))
811
812 scatterplot.matrix(~Presence.Score+PAQ.AS.y.+PAQ.PU.x.+Affective.Score+
        CognitiveLoad.Score+TR.Incr+TT.Incr | Conditions,
813     reg.line=lm,
814     smooth=TRUE,
815     span=0.5,
816     diagonal= 'density',
817     by.groups=TRUE,
818     data=AVAvsCAL)
819
820 #Writing it in a file -> AVAvsCAL-scatterPlotMatrix
821 dev.off()
822
823 #Correlation Matrix#########################################################
824 #These pairs scatter plots are awfully complicated, use the previous
825 #one instead. Don't use this one unless necessary.
826
827 #Saving it in a file first, because it is very big
828 Cairo(1280, 1280, file="AVAvsCAL-correlationMatrix.png", type="png", bg
        ="white")
829
830 panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
831 {
832      usr <- par("usr"); on.exit(par(usr))
833      par(usr = c(0, 1, 0, 1))
```

```
834     r <- abs(cor(x, y, method = "spearman"))
835     txt <- format(c(r, 0.123456789), digits=digits)[1]
836     txt <- paste(prefix, txt, sep="")
837     if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
838
839     test <- cor.test(x,y, method = "spearman")
840     # borrowed from printCoefmat
841     Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
842                      cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
843                      symbols = c("***", "**", "*", ".", " "))
844
845     text(0.5, 0.5, txt, cex = cex * r)
846     text(.8, .8, Signif, cex=cex, col=2)
847 }
848
849 pairs(AVAvsCAL[,c("Presence.Score",
850         "PAQ.AS.y.",
851         "PAQ.PU.x.",
852         "Affective.Score",
853         "CognitiveLoad.Score",
854         "TR.Incr",
855         "TT.Incr")],
856     lower.panel=panel.smooth,
857     upper.panel=panel.cor)
858
859 #Writing it in a file -> AVAvsCAL-correlationMatrix.png
860 dev.off()
```

Appendix H

Final Ethics Approval

Appendix H (page 440) removed from Open Access version as it may contain sensitive/confidential content.

References

Avid. (2007). *Softimage—XSI.* Available from http://www.softimage.com/ (Avid's Softimage—XSI official website. Last accessed August 4th, 2007)

Ayres, P., & Paas, A. (2007). Can cognitive load approach make instructional animation more effective? *Applied Cognitive Psychology*, *21*(6), 811–820.

Baddeley, A. D. (1998). *Human memory: Theory and practice* (Revised ed.). Allyn and Bacon, Needham Heights, MA. Allyn and Bacon, Boston. (ISBN 0-205-27948-1)

Begleitter, M. (2001). *From word to image: Storyboarding and the filmmaking process*. Michael Wiese Production.

Bigoin, N., Porte, J., Kartiko, I., & Kavakli, M. (2007). Effects of depth cues on simulator sickness. In *Proceedings of the 1st international conference on immersive telecommunications* (pp. 1–4). (ACM SIGMM Technically sponsored by Create-Net and EURASIP)

Bisch-Knaden, S., & Wehner, R. (2003). Landmark memories are more robust when acquired at the nest site than en route: experiments in desert ants. *Naturwissenschaften*, *90*(3), 127–130.

Blender Foundation. (2011). *Blender.* Available from http://www.blender.org (Last accessed on September 2011)

Bruck, S., & Watters, P. (2009a). The annual review of cybertherapy and cybermedicine 2009 - studies in health technology and informatics series. In B. K. Wiederhold & G. Riva (Eds.), (chap. Cybersickness and anxiety during simulated motion: Implications for VRET). IOS Press, Amsterdam. Proc Cybertherapy 2009. Available from http://www.booksonline.iospress.nl/

Bruck, S., & Watters, P. A. (2009b). Estimating cybersickness of simulated motion using the simulator sickness questionnaire (ssq): A controlled study. *Computer Graphics, Imaging and Visualization, International Conference on*, 486–488.

Bruno, F., Bruno, S., Sensi, G. D., Luchi, M.-L., Mancusoc, S., & Muzzupappaa, M. (2010). From 3d reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage*, *11*(1), 42–49.

Buisine, S., & Martin, J.-C. (2007). The effects of speech-gesture cooperation in animated agents' behavior in multimedia presentations. *Interacting with Computers*, *19*, 484–493.

Burke, S. (2001). Missing values, outliers, robust statistics and non-parametric methods. *LCGC Online Supplement*, 19–24. Available from http://

www.lcgceurope.com/lcgceurope/data/articlestandard/lcgceurope/502001/4509/article.pdf

Chen, C. J. (2006). The design, development and evaluation of a virtual reality based learning environment. *Australasian Journal of Educational Technology*, *22*(1), 39–63.

Chen, G., Li, X. cheng, Wu, G. qing, Wang, Y., Fang, B., Xiong, X. feng, et al. (2010). The use of virtual reality for the functional simulation of hepatic tumors (case control study). *International Journal of Surgery*, *8*(1), 72–78.

Cozens, P. M., Saville, G., & Hillier, D. (2005). Crime prevention through environmental design (cpted): a review and modern bibliography. *Property Management*, *23*(5), 328-356. Available from www.emeraldinsight.com/0263-7472.htm

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. Harper Collins, New York.

Csikszentmihalyi, M. (1997). *Finding flow: The psychology of engagement with everyday life*. Basic Books.

Dede, C., & Ketelhut, D. (2003). *Designing for motivation and usability in a museum-based multi-user virtual environment.* In Proceedings of AERA 2003. Available from http://muve.gse.harvard.edu/rivercityproject/documents/AELppr.pdf

Donderi, D. C. (2006). Visual complexity: A review. *Psychological Bulletin*, *132*(1), 73-97.

Dras, M., Richards, D., Taylor, M., & Gardiner, M. (2010). Generating and detecting deceptive language in virtual agents. In *The 9th International Conference on Autonomous Agents and Multiagent Systems: Workshop 19 – International Workshop on Interacting with ECAs as Virtual Characters. Toronto, Canada.*

e-on software. (2006). *Vue 6 esprit: The art of natural 3d!* Available from http://www.e-onsoftware.com/

Fisher, B. L. (2010). *Antweb.* http://www.antweb.org/. (Last accessed: July 2010)

Gaddis, T. (1998). *Virtual reality in the school.* Virtual Reality and Education Laboratory: East Carolina University.

Gardner, H. J., & Martin, M. A. (2007). Analyzing Ordinal Scales in Studies of Virtual Environments: Likert or Lump It! *Presence: Teleoperators and Virtual Environments*, *16*(4), 439 – 446.

Ghani, J. A., & Deshpande, S. P. (1994). Task characteristics and the experience of optimal flow in human – computer interaction. *The Journal of Psychology*, *128*(4), 381–391. Available from http://www.tandfonline.com/doi/abs/10.1080/00223980.1994.9712742

Gimp Development Team. (2011). *Gimp.* Available from http://www.gimp.org (Last accessed on September 2011)

Grantcharov, T. (2006). Virtual reality simulation in training and assessment of laparoscopic skills. *European Clinics in Obstetrics and Gynaecology*, *2*(4), 197 – 200.

Hettinger, L. J., Berbaum, K. S., Kennedy, R. S., Dunlap, W. P., & Nolan, M. D. (1990).

Vection and simulator sickness. *Military Psychology*, *2:3*, 171–181.

Hongpaisanwiwat, C. (2006, July). The comparison study of attention drawing techniques for multimedia learning. *Information Technology Based Higher Education and Training, 2006. ITHET '06. 7th International Conference on*, 652-653.

Inkscape Development Team. (2011). *Inkscape.* Available from `http://www.inkscape.org` (Last accessed on September 2011)

Ito, H., & Takano, H. (2004). Controlling visually induced self-motion perception: Effect of overlapping dynamic visual noise. *Journal of Physiological Anthropology and Applied Human Science*, *23*(6), 307–311. Available from `http://www.jstage.jst.go.jp/browse/jpa`

Johnsen, K., Raij, A., Stevens, A., Lind, D. S., & Lok, B. (2007). The validity of a virtual human experience for interpersonal skills education. In *Chi '07: Proceedings of the sigchi conference on human factors in computing systems* (pp. 1049–1058). New York, NY, USA: ACM.

Kartiko, I., & Kavakli, M. (2008). Virtual reality police training: How much visual information is too much for knowledge transfer? In *Proceedings of the 2nd international conference on cybercrime forensics education and training* (pp. 1–13). (ISBN 1899253-19x)

Kartiko, I., Kavakli, M., & Cheng. (2011a). The affective and cognitive impacts of visual complexity of animated-virtual actors and the virtual environment in virtual reality-based instructional animation. *IEEE Transactions on Affective Computing*, 1–14. (Submitted on Oct 24, 2011)

Kartiko, I., Kavakli, M., & Cheng, K. (2009a). *Does the visual complexity of animated-virtual actors in virtual reality applications affect learning?* Presented at the 9th Meeting of the Australasian Society for Cognitive Science. Sydney.

Kartiko, I., Kavakli, M., & Cheng, K. (2009b). The impacts of animated - virtual actors' visual complexity and simulator sickness in virtual reality applications. In B. Werner (Ed.), *Proceedings of the 6th international conference computer graphics, imaging and visualization* (pp. 147–152). IEEE Computer Society. (ISBN 978-0-7695-3789-4)

Kartiko, I., Kavakli, M., & Cheng, K. (2010). Learning science in a virtual reality application: The impacts of animated-virtual actors visual complexity. *Computers & Education*, *55*(2), 881–891.

Kartiko, I., Kavakli, M., & Cheng, K. (2011b). Correlations between simulator sickness and visual complexity of animated-virtual actors and virtual environments. *Journal of Computer Animation and Virtual World*, 1–25. (Submitted on Sept 30, 2011)

Kavakli, M. (2006). Training simulations for crime risk assessment. In *7th international conference on information technology based higher education and training, 2006.* (pp. 203–210).

Kavakli, M., & Gero, J. S. (2003). Strategic knowledge differences between an expert and a novice designer. In U. Lindemann (Ed.), *Human behaviour in design* (pp. 42–51). Springer. (ISBN: 3-540-40632-8)

Kavakli, M., & Kartiko, I. (2007). Avatar creation for believable agents. In *Proceedings*

of the 10th international conference on computer graphics and artificial intelligence (pp. 1–6).

Kavakli, M., Kartiko, I., Porte, J., & Bigoin, N. (2007). Effects of digital content on motion sickness in immersive virtual environments. In *Proceedings of the 3rd international conference on computer science and information systems* (pp. 1–12).

Kavakli, M., & Lloyd, M. (2005). Lightspeed: Fusing the virtual space with telepresence. *INISTA 2005, Innovations in Intelligent Systems and Applications, IEEE Computational Intelligence Society Turkey Chapter, YILDIZ Technical University, Istanbul, Turkey*, 231–234. Available from `http://www..inista.org/`

Kavakli, M., Szilas, N., Porte, J., & Kartiko, I. (2007). Riskman: A multi-agent system for risk management. In H. Lin (Ed.), *Architectural design of multi-agent systems: Technologies and techniques* (pp. 356–376). IGI Global.

Keil, D., & Goldin, D. (2006). Indirect Interaction in Environments for Multiagent Systems. In D. Weyns, V. Parunak, & F. Michel (Eds.), *Environments for Multiagent Systems II* (Vol. 3830, pp. 68 – 87).

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology*, *3*(3), 203–220.

Knaden, M., Lange, C., & Wehner, R. (2006). The importance of procedural knowledge in desert-ant navigation. *Current Biology*, *16*(21), R916–R917.

Knaden, M., & Wehner, R. (2005). Nest mark orientation in desert ants cataglyphis: what does it do to the path integrator? *Animal Behaviour*, *70*(6), 1349–1354.

Korakakis, G., Pavlatou, E., Palyvos, J., & Spyrellis, N. (2009). 3D visualization types in multimedia applications for science learning: A case study for 8th grade students in Greece. *Computers & Education*, *52*, 390401.

Lee, E. A.-L., & Wong, K. W. (2008). A review of using virtual reality for learning. In Z. P. et al. (Ed.), (pp. 231–241). Springer-Verlag Berlin Heidelberg 2008.

Lin, J. J.-W., Duh, H. B., Parker, D. E., Abi-Rached, H., & Furness, T. A. (2002). *Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment.* Proceedings of the IEEE Virtual Reality 2002 (VR'02).

Lowe, R. K. (2004, 5-8 December). Animation and learning: Value for money? In R. Atkinson, C. McBeath, D. Jonas-Dwyer, & R. Phillips (Eds.), *Beyond the comfort zone: Proceedings of the 21st ascilite conference* (pp. 558 – 561). Available from `http://www.ascilite.org.au/conferences/perth04/procs/lowe-r.html`

Lusk, M. M., & Atkinson, R. K. (2007). Animated pedagogical agents: Does their degree of embodiment impact learning from static or animated worked examples? *Applied Cognitive Psychology*, *21*(6), 747-764.

Mantovani, F. (2003). Vr learning: Potential and challenges for the use of 3d environments in education and training. In G. Riva & C. Galimberti (Eds.), (pp. 208–223). Amsterdam, IOS Press. Available from `http://citeseer.ist.psu.edu/mantovani03vr.html`

Mayer, R. E. (2001). *Multimedia learning* (R. E. Mayer, Ed.). Cambridge Press. (ISBN 9780521787499)

Mayer, R. E. (2003). The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction*, *13*(2), 125-139.

Mikropoulos, T. (2006, dec). Presence: a unique characteristic in educational virtual environments. *Virtual Reality*, *10*(3), 197–206.

Mikropoulos, T. A., & Strouboulis, V. (2004). Factors that influence presence in educational virtual environments. *CyberPsychology & Behavior*, *7*(5), 582 – 591.

Milner, R. (1993, January). Elements of interaction: Turing award lecture. *Commun. ACM*, *36*, 78–89. Available from `http://doi.acm.org/10.1145/151233.151240`

Moon, D. C., Moon, J., & Keagy, A. (2010). Direct and indirect interactions. *Nature Education Knowledge*, *1*(11), 9.

Moreno, R. (2006). Learning in high-tech and multimedia environments. *Current Directions in Psychological Science*, *15*(2), 63-67.

Moreno, R. (2007). Optimising learning from animations by minimising cognitive load: Cognitive and affective consequences of signalling and segmentation methods. *Applied Cognitive Psychology*, *21*(6), 765-781.

Moreno, R., & Mayer, R. (2007). Interactive multimodal learning environments. *Educational Psychology Review*, *19*(3), 309–326.

Moreno, R., & Mayer, R. E. (2001). Virtual reality and learning: Cognitive and motivational effects of students' sense of presence. In *the ninth international conference on human-computer interaction. 2001* (pp. 65–67).

Moreno, R., & Mayer, R. E. (2002). Learning science in virtual reality multimedia environments: Role of methods and media. *Journal of Educational Psychology*, *94*(3), 598 – 610.

Moreno, R., & Mayer, R. E. (2004). Personalized messages that promote science learning in virtual environments. *Journal of Educational Psychology*, *96*(1), 165-173.

Mourant, R. R., Ahmad, N., Jaeger, B. K., & Lin, Y. (2007). Optic flow and geometric field of view in a driving simulator display. *Displays*, *28*(3), 145–149.

Mourant, R. R., Rengarajan, P., Cox, D., Lin, Y., & Yaeger, B. K. (2007, October). *The effect of driving environments on simulator sickness.* Proceedings of the 51st Annual Meeting of the Human Factors and Ergonomics Society, Baltimore, MD. Available from `http://www1.coe.neu.edu/~mourant/mourant/Publications.html` (1232-1236)

Narendra, A., Cheng, K., & Wehner, R. (2007). Acquiring, retaining and integrating memories of the outbound distance in the Australian desert ant Melophorus bagoti. *J Exp Biol*, *210*(4), 570-577. Available from `http://jeb.biologists.org/cgi/content/abstract/210/4/570`

Neuhäuser, M. (2010). A nonparametric two-sample comparison for skewed data with unequal variances. *Journal of Clinical Epidemiology*, *63*(6), 691–693.

Oliva, A., Mack, M., Shrestha, M., & Peeper, A. (2004). *Identifying the perceptual dimensions of visual complexity of scenes.* Proc. 26th Annual Meeting of the Cognitive Science Society. Available from http://cvcl.mit.edu/Papers/Oliva_CogSc04.pdf

Powers, M. B., & Emmelkamp, P. M. (2008). Virtual reality exposure therapy for anxiety disorders: A meta-analysis. *Journal of Anxiety Disorders*, *22*(3), 561 – 569.

Raij, A., Johnsen, K., Dickerson, R., Lok, B., Cohen, M., Bernard, T., et al. (2006). Interpersonal scenarios: Virtual approx real? In *Virtual reality conference, 2006* (pp. 59–66).

Richards, D., & Barles, J. (2005). *Actors vs. animation for adult learning ?* ACM International Conference Proceeding Series; Vol. 123. Proceedings of the second Australasian conference on Interactive entertainment table of contents Sydney, Australia. (p.163 - 166. ISBN:0-9751533-2-3)

Richards, D., Dras, M., Porte, J., & Taylor, M. (2009, 15 - 18 June). *The role, measurement and delivery of immersion in training environments.* SimTecT 2009 Simulation Conference: Simulation - Concepts, Capability and Technology. Adelaide, Australia.

Richards, D., Fassbender, E., Bilgin, A., & Thompson, W. F. (2008, September). An investigation of the role of background music in ivws for learning. *ALT-J*, *16*(3), 231 – 244.

Riva, G., Mantovani, F., & Gaggioli, A. (2004). Presence and rehabilitation: toward second-generation virtual reality applications in neuropsychology. *Journal of NeuroEngineering and Rehabilitation*, *1*(1), 9. Available from http://www.jneuroengrehab.com/content/1/1/9

Roberts, M. N. (2007). *Complexity and aesthetic preference for diverse visual stimuli*. Unpublished doctoral dissertation, Departament de Psicologia, Universitat de les Illes Balears.

Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological Review*, *110*(1), 145–172.

Russell, J. A., & Pratt, G. (1980). A description of the affective quality attributed to environments. *Journal of Personality and Social Psychology*, *38*(2), 311–322.

Schatz, B., Chameron, S., Beugnon, G., & Collett, T. S. (1999). The use of path integration to guide route learning in ants. *Nature*, *399*(6738), 769 – 772.

Service, N. R. F. (1999). *VF/1-6 (1999). Village Firefighter (VF) Manual.* (ISBN 0731097303x)

Service, N. R. F. (2001a). *Compressed Air Breathing Apparatus Standard Operating Procedures.*

Service, N. R. F. (2001b). *SP/3(1995). Specialist Training Manual: Compressed Air Breathing Apparatus (CABA).* (ISBN 0731064321)

Shigemasu, H., Morita, T., Matsuzaki, N., Sato, T., Harasawa, M., & Aizawa, K. (2006). Effects of physical display size and amplitude of oscillation on visually induced motion sickness. In *Vrst '06: Proceedings of the acm symposium on virtual reality software*

and technology (pp. 372–375). New York, NY, USA: ACM Press.

Simon, M. (2000). *Storyboards: Motion in art*. Focal Press.

Skovlund, E., & Fenstad, G. U. (2001). Should we always choose a nonparametric test when comparing two apparently nonnormal distributions? *Journal of Clinical Epidemiology*, *54*(1), 86–92.

Slater, M. (1999). Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, *8*(5), 560–565.

Slater, M., & Garau, M. (2007). The Use of Questionnaire Data in Presence Studies: Do Not Seriously Likert. *Presence: Teleoperators and Virtual Environments*, *16*(4), 447–456.

Stanney, K. M., & Zyda, M. (2002). Handbook of virtual environment: Design, implementation and applications. In K. M. Stanney (Ed.), (pp. 1–14). Lawrence Erlbaum Associates.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257 – 285.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, *4*(4), 295 - 312.

Takatalo, J., Nyman, G., & Laaksonen, L. (2008). Components of human experience in virtual environments. *Computers in Human Behaviour*, *24*(1), 1–15.

Team, R. D. C. (2010). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Available from `http://www.R-project.org` (ISBN 3-900051-07-0)

Torpy, A. (2011). *L3dt.* Available from `http://www.bundysoft.com` (Last accessed on September 2011)

Trenholme, D., & Smith, S. (2008, September). Computer game engines for developing first-person virtual environments. *Virtual Reality*, *12*(3), 181–187.

Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: Can it facilitate? *International Journal of Human-Computer Studies*, *57*(4), 247-262.

Veletsianos, G. (2007). Cognitive and affective benefits of an animated pedagogical agent: Considering contextual relevance and aesthetics. *Journal of Educational Computing Research*, *36*(4), 373–377.

Verdaasdonk, E., Stassen, L., Monteny, L., & Dankelman, J. (2006). Validation of a new basic virtual reality simulator for training of basic endoscopic skills. *Surgical Endoscopy*, *20*(3), 511–518.

Wagner, D., Billinghurst, M., & Schmalstieg, D. (2006). How real should virtual characters be? In *Ace '06: Proceedings of the 2006 acm sigchi international conference on advances in computer entertainment technology* (p. 57). California, USA: ACM Press.

Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shawe, E., & Collins, H. (2008). The politeness effect: Pedagogical agents and learning outcomes. *Int. J. Human-Computer Studies*, *66*, 98–112.

Wehner, R. (1999, August). Large-scale navigation: The insect case. In (Vol.

1661, p. 753). Springer Berlin - Heidelberg. Available from http://www.springerlink.com/content/waacr6t1y99mhb96

Wehner, R. (2003, August). Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural and Behavioral Physiology*, *189*, 579-588. Available from http://www.springerlink.com/content/2ewcymk1f6y44heh/

Whitelock, D., Romano, D., Jelfs, A., & Brna, P. (2000). Perfect presence: What does this mean for the design of virtual learning environments? *Education and Information Technologies*, *5*(4), 277 – 289. Available from http://dx.doi.org/10.1023/A:1012001523715

Wiederhold, M. D. (2005). *Physiological monitoring during simulation training and testing, final report* (Tech. Rep.). Virtual Reality Medical Centre. Available from http://stinet.dtic.mil (contract DAAH01-03-C-R-301)

Williams, R. (2002). *The animator's survival kit: A manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators*. (ISBN 978-0571202287)

Winn, W. (1996). Learning virtually. In *Northcon/96* (pp. 237–242).

Witmer, B. G., & Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper. Virtual Environ.*, *7*(3), 225–240.

Wittlinger, M., Wehner, R., & Wolf, H. (2006). The ant odometer: Stepping on stilts and stumps. *Science Magazine*, *312*, 1965–1967.

Wittlinger, M., Wehner, R., & Wolf, H. (2007). The desert ant odometer: a stride integrator that accounts for stride length and walking speed. *Journal of experimental Biology*, *210*, 198–207.

Wolf, H., & Wehner, R. (2000). Pinpointing food sources: olfactory and anemotactic orientation in desert ants, cataglyphis fortis. *J Exp Biol*, *203*(5), 857-868. Available from http://jeb.biologists.org/cgi/content/abstract/203/5/857

Wolf, H., & Wehner, R. (2005). Desert ants compensate for navigation uncertainty. *J Exp Biol*, *208*(22), 4223-4230. Available from http://jeb.biologists.org/cgi/content/abstract/208/22/4223

Worldviz. (2010). *Vizard: Virtual reality toolkit.* Available from http://www.worldviz.com (Last accessed on October 2010)

Young, P., & Munro, M. (1998). Visualising software in virtual reality. In *Proceedings of the ieee 6 th international workshop on program comprehension* (pp. 19–26). Available from http://www.sdml.info/library/Young98.pdf

Zollikofer, C. P. E. (1994). Stepping patterns in ants: Influence of speed and curvature. *Journal of experimental biology*, *192*, 95-106. Available from http://jeb.biologists.org/cgi/content/abstract/192/1/95

Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, *38*(9), 25–32.