Appendix A – Design-Based Research Summary of Data

Introduction

The notes below aim to characterize the lessons conducted during the three iterations (semesters) of the ITEC100 course:

- Iteration 1 – Semester 2 of 2005
- Iteration 2 – Semester 1 of 2006
- Iteration 3 – Semester 2 0f 2006.

The approaches to description and analysis have been outlined in the Methodology chapter (Chapter 3). The summaries containing observations, key incidents and reflective notes relating to each lesson have been distilled as far as possible (while still preserving pertinent aspects of the lesson) to provide a condensed account of the dataset. Screenshots have been included to illustrate interfaces or approaches in order to more accurately represent how the web-conferencing environment was used to facilitate learning under different circumstances.

Note once again that no redesigns occurred in the first lesson of any of the three semesters. Students needed some time to orient themselves with the technology and also the course without being overloaded by more complex interface or activity designs. Using standard interfaces and teacher-centred or teacher-led activity designs allowed students who had never used the web-conferencing environment to become comfortable with its use. Adopting a similar approach in the first lesson of all three semesters also provided a means of calibration, allowing the extent to which applying alike approaches to instruction resulted in similar collaborations to be gauged.

Design-Based Research – Summaries of Iterations

Summaries of Iteration 1 Lessons – Semester 2 of 2005

Iteration 1 Topic 1

Topic: Topic: Introduction to writing, compiling and running java programs, as well as providing a general introduction to the course.

Attendees: 9

Summary

OB1: In the first 45 minutes of this lesson the teacher provides students with a tour of the tools in the virtual classroom that will mediate collaboration and learning throughout the coming semester. This involves activities such as a check to see if students can hear the teacher's audio, asking students to introduce themselves using the text-chat pod, asking them to download solutions from the file-share pod, how to adjust their preferences for different modem speeds, complete a poll about modem connection speeds, and use the whiteboard. Note that these activities attempt to simultaneously community build, understand the profile of the class members, as well as familiarize students with the virtual classroom affordances. Throughout this lesson about 15 minutes was also spent discussing and demonstrating aspects of the course (such as the learning management system, the approach to assessment, and so on).

OB2: The preliminary tutorial questions are then covered. The standard sharing layout allows the teacher to quickly gather responses from many students simultaneously in the chat pod (see Figure 63). For instance when asking for the difference between a syntax error and a logical error in question two the students provide in quick succession (within 14 seconds) the following responses:

> NK: wording error
> KC: Bad Java grammar ...
> LI: violation of language
> SP: error in programming language
> NK: misspelling
> JR: The compiler will not accept it

OB3: Then when asked for the meaning of the term "logical error" students provide the following responses to the teachers' questions within 24 seconds:

> JR: errors are not detected by the compiler as they are syntactically correct
> AB: when you get a result you dont expect
> KC: That is where the computer did not know what I wanted it to do ...
> LI: bad programmer
> NK: not easy to find
> SP: code works but result is wrong
> JR: programmer must test the program to find logical errors
> SA: The syntax is ok but the results are not in accordance

Due to the small number of comments visible in the chat pod at any one time the teacher chooses to enlarge the chat pod and place it along the bottom section of the browser window (see Figure 64).

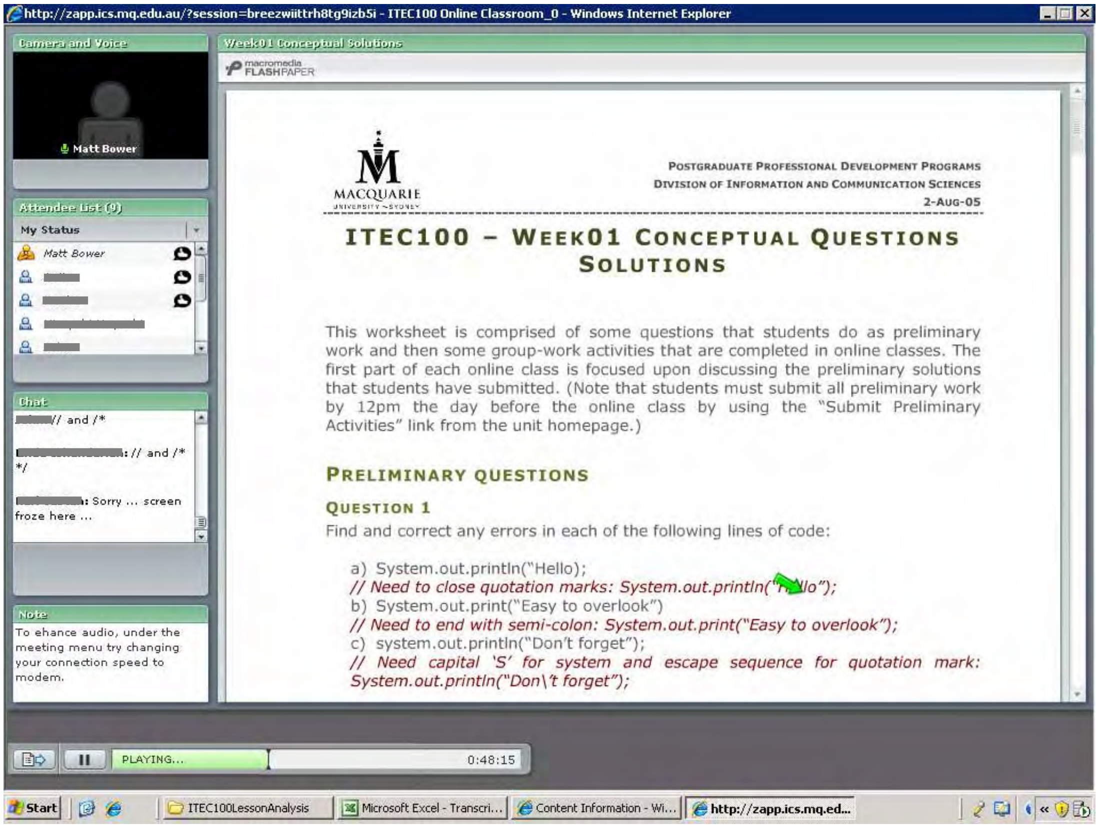

Figure 63 – Iteration 1 Topic 1 Using the standard 'Sharing' layout to cover tutorial questions

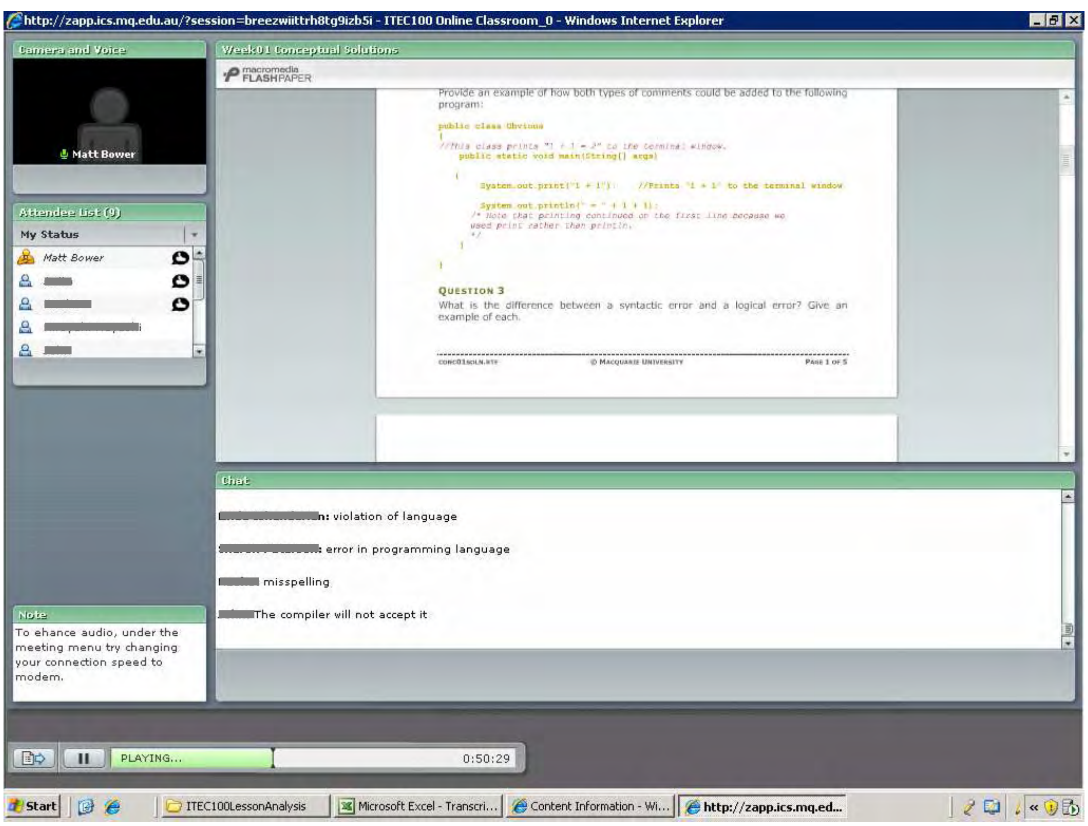

Figure 64 – Iteration 1 Topic 1 Changing the size and placement of chat-pod

OB4: The chat-pod question-response approach being adopted works well for the questions asking about escape sequences and about the difference between byte code and source code. However the teacher answers the last question "Describe the process of compiling and then running a program behind the scenes" because it involves more elaborate descriptions and diagrams (which have been pre-prepared in the solutions). That is to say the text-chat medium which has been used to elicit student responses is not an appropriate modality to present more complex, conceptual models. The preliminary tutorial questions have taken approximately 21 minutes to be covered.

OB5: To cover the practical activities the switch is made to the default sharing layout which contains the large share pod on the right hand side and a small chat pod on the left hand side. This allows students to observe programming processes, thus providing an appropriate modality for the type of information being share, in accordance with Symbol System Theory (Salomon, 1994). The teacher covers the practical activities using a predominantly transmissive approach, broadcasting model solutions in the IDE and describing them. For the Cube exercise the students provide suggestions about how to debug the program while the teacher implements their suggestions in the IDE (see Figure 65).

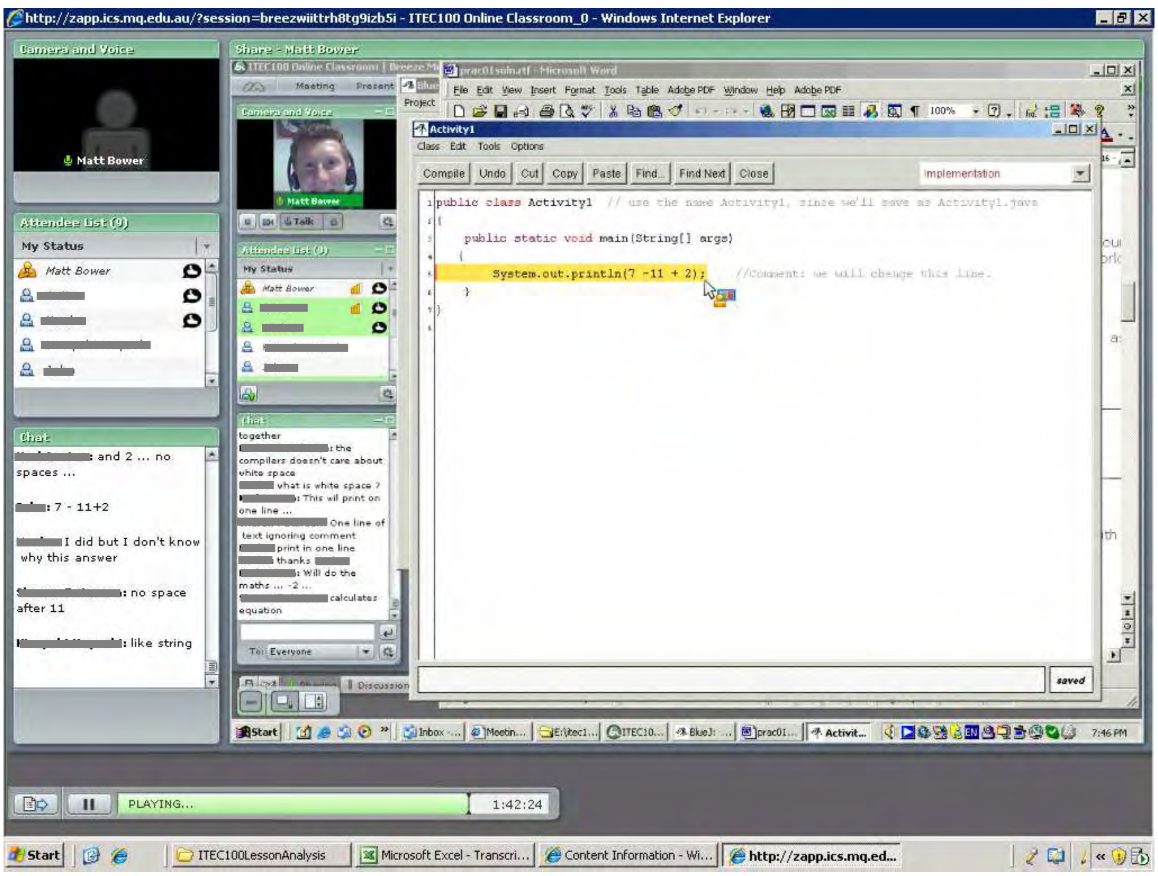

Figure 65 – Iteration 1 Topic 1 Using screen-sharing to complete a programming activity

OB6: When asked for feedback regarding the lesson students expressed generally positive sentiments. Technology suggestions posed by students include making the text-chat window as large as possible and the need for them to have broadband internet access in order to receive adequate broadcast of the lesson.

Key Incidents:

- KI1: While demonstrating the facilities of the virtual classroom the teacher forgets to increase student privileges so they can use whiteboard tools.
- KI2: During the 5 minute mid-class break the teacher forgets to turn off microphone. This means students can still hear his mumblings both live and on the recording.
- KI3: The attempt to use green arrow pointer tool to highlight the tutorial question solution being broadcast is ineffective due to "jumpiness of the presentation".
- KI4: A few students note that their computer froze for a minute or so during the lesson, in some way due to the university network.
- KI5: A student (JB) keeps sending private messages to the teacher, so the teacher encourages him to send public messages and explains to everyone how to select between sending public and private messages in the web-conferencing system.
- KI6: A student (LI) asks that the attendee list be made larger during the practical exercises so the class could see all people present in the room (a useful and important suggestion that was adopted).
- KI7: At times the teacher audio repeats some of the text-chat comments students make in order to emphasize them. For instance, in response to SPs contribution regarding the multi-line comment markers the teacher replies "SP's pointed out that need to pay attention to the order of the comment markers for multiple lines".
- KI8: During preliminary question 2 the teacher chooses to change the layout so that the text-chat pod is enlarged and placed along the bottom of the screen, allowing more of the students' comments can be viewed at one time. A student comments that this is better than the previous layout.
- KI9: Several general teaching strategies were observed, including:
  - In the introduction to the subject the teacher insists (using both audio and text-chat) that students must ask questions whenever they have them in this course. The teacher also encourages them to provide feedback if they have technical problems.
  - General teacher strategies such as positivity, acknowledgement, enthusiasm and encouragement (especially of questions) are evident throughout the discourse of the lesson.
- KI10: The webcam was paused part way through the lesson because it did not contribute in any way to learning. The webcam broadcast provides a degree of personalization but consumes bandwidth and can be distracting to both the teacher and students.

Reflective Notes:

- RN1: During the screen-share component of the lesson the teacher often has to ask students whether they can see the current state of the desktop (due to lags in the broadcast they indicate they sometimes experience). The technology imposes a communicative overhead and potential layer of interference (i.e., a distributed process loss, Neale, Carroll, & Rosson, 2004).
- RN2: The text-chat allowed students to ask questions while the teacher was speaking and other students to answer questions without interrupting the teacher's audio. This is an example of leveraging the technological affordances of the web-conferencing environment to support different and potentially more effective modes of interaction than possible in face-to-face contexts.

Iteration 1 Topic 2

Topic: Fundamentals of Objects and Classes as they relate to object oriented programming.

Attendees: 8

Summary

OB1: After spending 10 minutes addressing matters of housekeeping, 45 minutes is spent covering the tutorial questions for the week. The approach of questioning students using audio and having students responding using text-chat is once again applied, using a standard sharing interface. It appears to be an effective approach to covering this tutorial question requiring identification of program variables because several students can simultaneously provide the short answers required for the declarative knowledge task. This is observed to work successfully for other questions with short, declarative answers. For instance, when the teacher asks "what's the difference between a constructor and a method, just succinctly?" the following stream of text-chat follows:

> AB: constructos initialize objects, methods do things to objects
> JB: contructor initialises the object, method defined object behaviour
> SP: Constructor creates initial instance of objectmethod performs actions on object
> MiHa: Constructor uses the same name with different parameters
> LI: a constructor can only be called once, a method can be called multiple times for an object
> MH: a constructor is used to initialize the object
> SA: A constructor is a method used in the initialisation of an object whenever that object comes into being. Method is a sequence of instructions that can be invoked many times and which may or may not have parameters or return values

The text-chat responses allow student answers to be compared and contrasted, and further discussions to be held around the concepts. The approach allows a multistructural understanding to be demonstrated by virtue of having required pieces of information relating to the concept, but not a relational understanding since students do not inter-relate all items of knowledge in the same way that they would if they were writing constructors and methods.

OB2: The teacher presents some of the students' solutions to the preliminary tutorial questions through the web-conferencing environment, which shows students that their answers are being reviewed in detail (with the intention of increasing the perceived importance of completing these pre-class tasks to a high standard). This is an example of applying a general teaching strategy (demonstrating students' accountability) in the web-conferencing environment.

OB3: Diagrams are required for the last two tutorial questions in order to represent more complex concepts in context (see Figure 66). The visual representation allows more information to be presented and interrelated at once without requiring students to hold the concepts in working memory, reducing cognitive load (van Merriënboer & Ayres, 2005). This reduces cognitive load as compared to using verbal explanations where students would need to hold the items in memory. This is an example of offloading cognitive effort to the environment (Hollan, Hutchins, & Kirsh, 2000). Due to the amount of information the teacher wishes to interrelate pre-prepared diagrams are broadcast instead of using time in class drawing the diagrams or having students attempt to draw the diagrams. While this is more time efficient, it does not allow the formedness of students' mental models to be revealed.

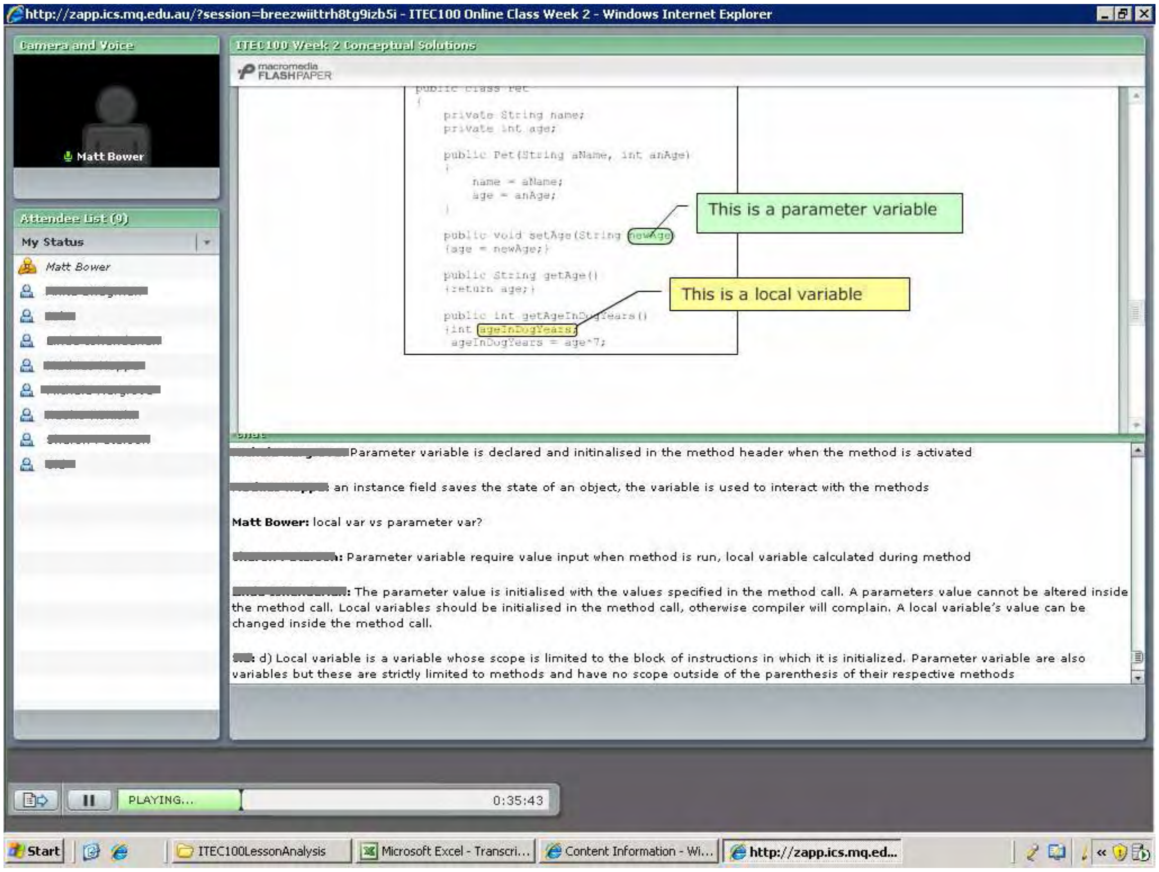

Figure 66 – Iteration 1 Topic 2 Pre-prepared diagrams to demonstrate concepts

OB4: Following the mid-class break, approximately 30 minutes is spent broadcasting student solutions to the practical exercises (Figure 67). The presentational (teacher-centred) approach adopted increases the rate of teacher discourse with a corresponding decrease in the rate of student contribution.

OB5: The last section of the lesson relates to a group programming activity where the students instruct the teacher on how to debug a program. The teacher shares his screen and the students make suggestions about what to do next. This appeared to be an effective approach to engage students and elicit contributions from them. It also allowed programming 'process' knowledge to be shared, where the teacher could provide a form of cognitive apprenticeship (Brown, Collins, & Duguid, 1989).

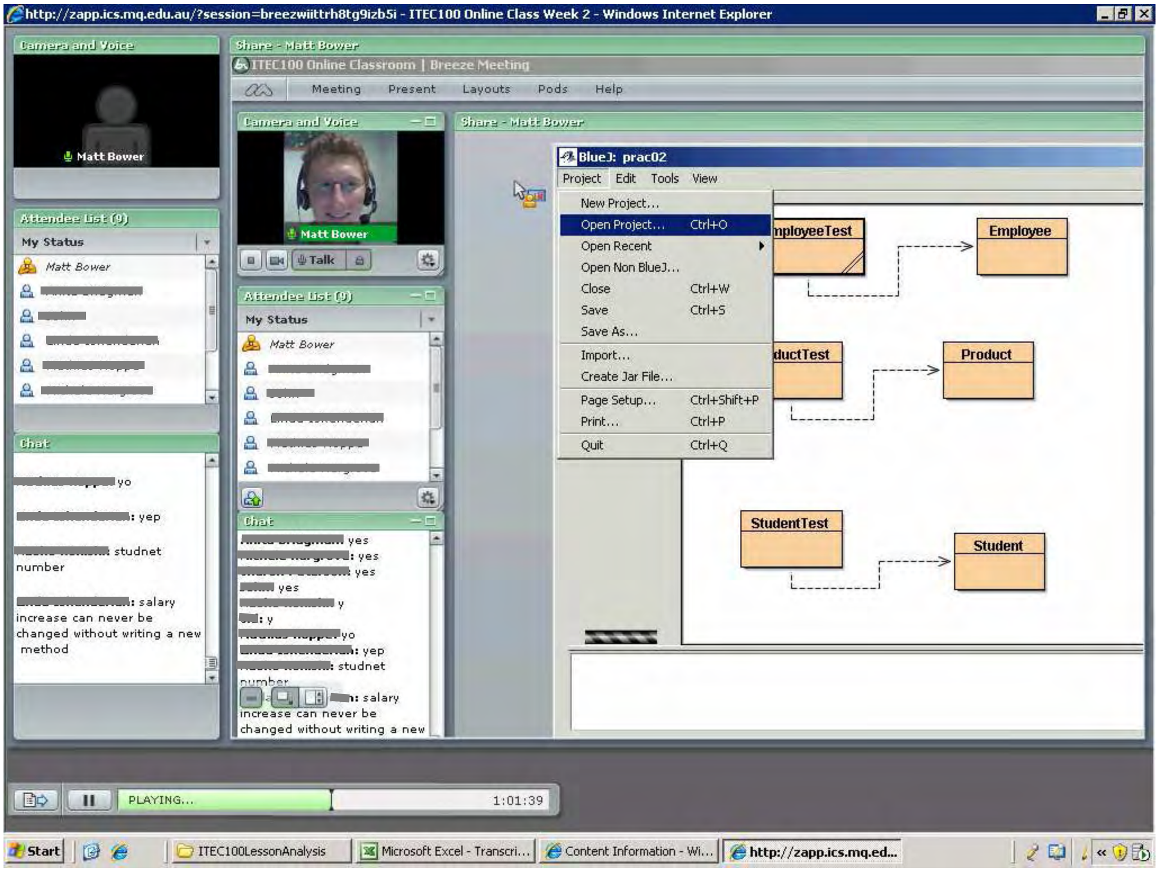

Figure 67 – Iteration 1 Topic 2 Teacher-centred approach to covering practical activities using screen-sharing of the IDE

Key Incidents:

- KI1: When one of the solutions being broadcast is omitting one variable the teacher uses both audio and text-chat to instruct students to "Please add aCompany to the list of parameters in this program." This is an example of a techno-pedagogic tactic to emphasize a point of information.
- KI2: The teacher uses the defining variable types activity to hold discussions regarding concepts and more clearly define student conceptions by asking questions like "what is a local variable?" (going beyond identification to comprehension). This discourse encouraging elaborative comments leads to a balance of teacher and student contribution.
- KI3: The teacher encourages students to copy and paste solutions into the note-pod as a more efficient way of sharing responses to the preliminary tutorial questions than re-typing. This allows faster contributions of information to the text-chat pod.
- KI4: The model solution to tutorial question 4 has some mistakes in it, which most likely confuses some students. However, the elaborate solution providing examples in-situ allows students to actually see the difference between different types of variables. This could not be done via text-chat. One student spontaneously comments "that was helpful, thanks".
- KI5: The teacher once again leaves the microphone switched on during the break.
- KI6: When teacher is using screen-share to broadcast student practical solutions he has the chat window in view so that he can see student comments, which means it

appears twice for students. He does not realize that if the web-conferencing browser window is minimized then student comments will appear in mini pop-up windows.
- KI7: Minutes of time are lost while the teacher tries to load the relevant student practical files to share with the class.
- KI8: Students have not been formally introduced to the scroll or full screen toggle buttons, which may have detracted from their viewing quality.

Reflective Notes:
- RN1: The teacher once again uses the tactic of repeating student comments to emphasize them and "give them voice". This "highlighting" (Mayer, 2005b) of the student text-chat using audio represents a useful technology-based pedagogic tactic that can be pervasively applied.
- RN2: There is a trade-off between covering the concept efficiently and having students contribute. For instance, it may have been more engaging to have students draw diagrams but this may have taken several times longer. The educational designer needs to decide which tasks are the most worthwhile to apply a student-centred (collaborative) approach in order to utilize time efficiently.
- RN3: At times a conversational environment (Waite, Jackson, & Diwan, 2003) arose while the conceptual material was being covered with student involvement being evidenced through several independently initiated student questions. However, when the teacher presents the solutions to the practical exercises the student contributions halt.

Iteration 1 Topic 3

Topic: Introduction to types and numbers.

Attendees: 9

Summary

OB1: The first 20 minutes are spent covering the solutions to the tutorial questions. As a pilot experiment the teacher attempts using a group-work approach to cover the material. Students are given 20 minutes to strive for consensus regarding answers in designated group-work rooms. They are encouraged to upload their solutions to the rooms and harvest questions that remain unresolved in the group. During the group-work trial students are not aware of how to use the technology to collaborate (upload files to broadcast documents) and the coordination of activity in the group-work room is stalled by virtue of no clear leader or direction on how to interact.

OB2: The teacher then spends 25 minutes covering the tutorial questions in an attempt to clarify conceptions. A transmissive approach was used where the teacher broadcast the document solutions and discussed them. There was correspondingly little input from students. Occasionally the teacher would ask questions like "are there any questions?". This results in several sequences of purely "yes" and "no" responses by students. At times some students seemed disengaged, indicated (for instance) by behaviours such as making irrelevant contributions to a note-pod. There is no opportunity to gauge or develop the formedness of students' mental models because they are not provided with the opportunity to discuss or represent their understanding.

OB3: After the mid-lesson break the teacher covers the practical tasks using by broadcasting and discussing student solutions using screen-share (see Figure 68). A predominantly transmissive approach to instruction is adopted. Student contributions during this time indicate engagement, potentially due to the way in which the teacher is effectively covering areas of the practical tasks where students indicated weakly formed (multistructural at best) understanding. However their contributions are restricted to short comments regarding their sentiments or whether or not they feel they understand. There is limited evidence of them developing an understanding because there is little opportunity for them to share their mental models.

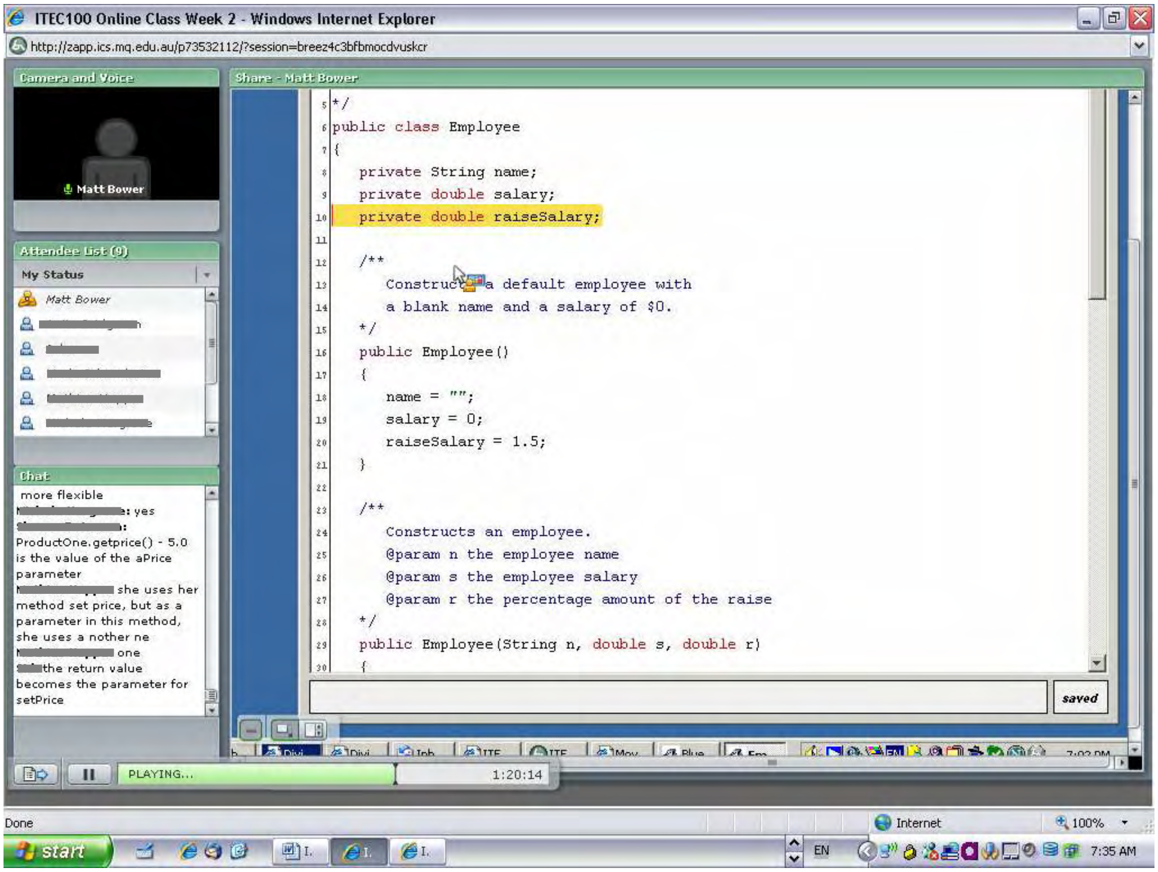

Figure 68 – Iteration 1 Topic 3 Using screen-sharing to model programming processes

Key Incidents:

- KI1: During the group-work trial the teacher is unsure of how the technology operates when using multiple rooms, commenting "You may have to shut down this room to access the other rooms. I'm not sure."
- KI2: When students were asked "How did you find the group-work?" responses included "good, but the chat window was small and difficult to read", and a student with a dial-up connection responded "It was too slow to use". The teacher shows them how to adjust the size of pods, and students practise this for a few moments.
- KI3: The teacher struggled with the green arrow pointer tool during the presentation of the false-swap tutorial solution. However several students commented that the pre-prepared diagrams and audio explanation led to a "much clearer" understanding.
- KI4: Even though a transmissive, teacher-centred approach to sharing student practical solutions is used, students seem engaged (based on their feedback). As well as due to an appropriately pitched presentation, this engagement may be also be because the task is "authentic" and "meaningful" (Herrington, Oliver, & Reeves, 2002) – they are comparing and contrasting approaches to questions that they have previously attempted. The engagement may also be because they are gaining insight into the abilities and thinking of their peers through sharing of a student solution.

Reflective Notes:

- RN1: The strategy of not only have an increasing level of difficulty of questions in preliminary tasks, but also to have an increasing level of collaborative difficulty as the lesson progresses allows students to graduate into collaborative modes of interaction. This approach may also apply throughout the semester – having students attempt student-centred group-work in their own virtual classroom before they had developed a familiarity with the tools or an appreciation of efficient patterns of collaboration compromised the effectiveness of the group-work activity that was conducted at the beginning of the lesson.

Iteration 1 Topic 4

Topic: Introduction to Applets and Graphics

Attendees: 10

Summary of Lesson

OB1: After introductory comments 36 minutes is spent covering the tutorial questions. Once again a teacher-led approach is adopted whereby students use text-chat to post responses to the teacher's audio questions. After students have made their suggestions the solution document is broadcast to summarize the answer to each question (see Figure 69).

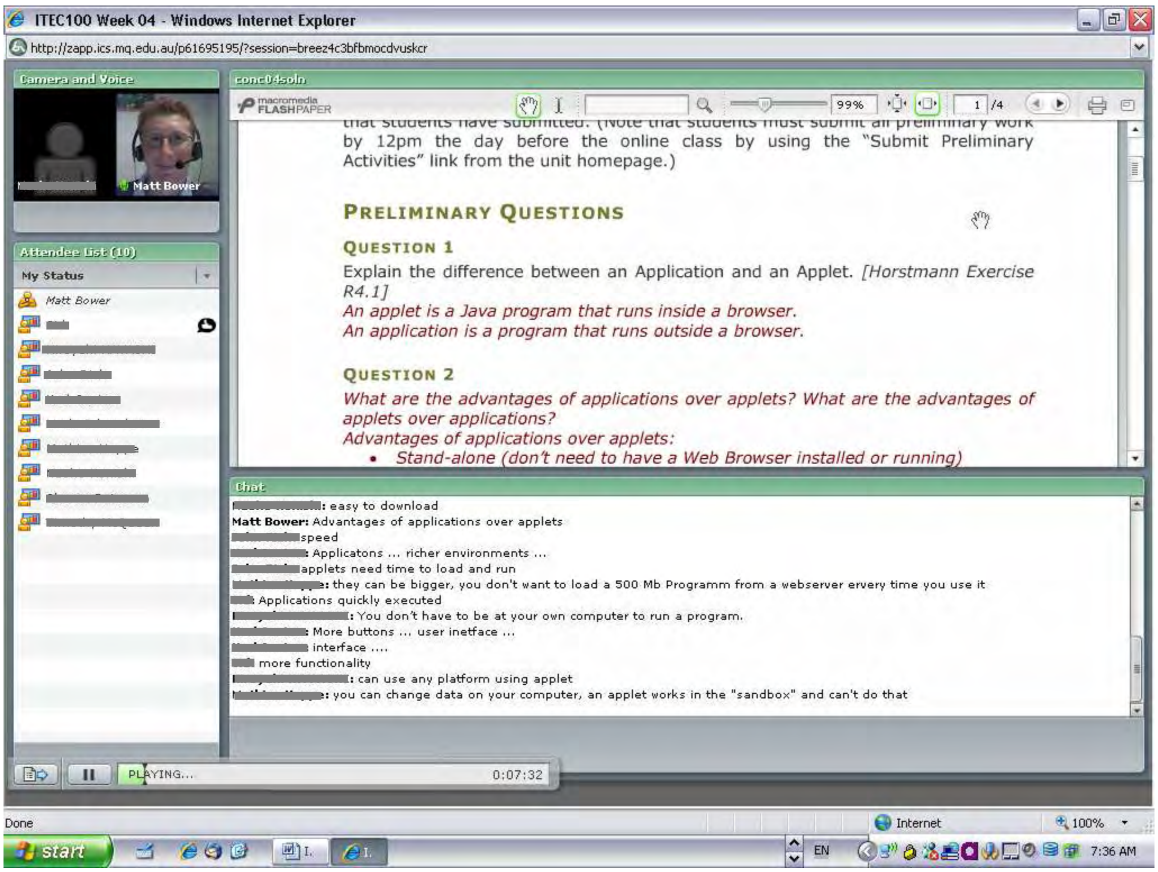

Figure 69 – Iteration 1 Topic 4 Teacher-led approach to covering tutorial questions

This results in moderate levels of student contribution and discussion, however the types of contributions are predominantly responsive.

OB2: Students are then provided with the task of downloading a zip file containing the class' drawings they had created in the practical preliminary tasks and combining them onto one applet canvas. The teacher uses the standard screen-sharing layout to demonstrate how to read the java documentation files that they have created in order to perform this task. Students are allocated fifteen minutes (and the ten minute break if required) to spend on this task.

OB3: After the break the teacher presents particular student solutions to the practical task and in some cases comments on the approach to coding they have used. The teacher is providing the commentary and thus is unable to assess the evolving level of student understanding. This is followed by the presentation of a debugging task that arose from the student's attempts at the preliminary practical work. The teacher demonstrates a program that is not rendering correctly because the required information is not collected in the constructor. During the teacher's demonstration the students ask several independently initiated questions using text-chat. This appears to be because the task is more "meaningful" (Herrington, Oliver, & Reeves, 2002) and has been based on a pre-identified area of student conceptual weakness.

OB4: Finally the teacher attempts to lead a 'draw a clock' programming task relevant to their assignment. However the task specification is unclear and confuses students. As well, the web-conferencing environment appears to be malfunctioning (for instance, it is not allowing pods to be moved or certain pods to broadcast). The teacher is flustered and suggests that due to time restrictions students download the solution to the task and review it (which the teacher uploads).

Key Incidents:

- KI1: During the preliminary tutorial question coverage the teacher has not turned screen "synch" mode on so as he is scrolling through the model solutions students cannot see the part of the notes to which he is referring. This critical error is only detected after 27 minutes into the lesson, and requires the teacher to quickly backtrack over the work that was covered while broadcasting the solutions.
- KI2: The teacher responds to a student's question of "MB, if you are talking I can't hear you" with the audio comment that the student will need to fiddle with their sound settings. Obviously using audio to respond was a poor choice if the student could not hear what the teacher was saying.
- KI3: During the coverage of the preliminary conceptual work students ask several independent questions. When responding to questions such as "what is casting?" and "what is meant by 'recovering'?" the teacher compromises deep conceptual understanding for more (time-effective) functional descriptions. The teacher explains that these concepts will make more sense later in the course.
- KI4: Several technical difficulties are reported during the lesson, such as computers freezing and lag in students being able to see the screen-share.
- KI5: During the screen-sharing session that takes place during the practical exercises several students comment that they cannot see the code clearly. The teacher suggests they click on the full-screen mode button but a student comments that this means they cannot type in the text-chat pod. The teacher should have advised them to click on the scroll button which broadcasts the screen in the sharing pod using an aspect ratio of one-to-one.
- KI6: When the teacher places the interface in "full screen" mode students cannot type responses to questions (and the teacher has not understood that students will automatically be placed in "full screen" mode if the "presenter's changes affect everybody" mode is switched on).
- KI7: The problems with the task and web-conferencing system for the 'draw a clock' activity at the end of the lesson destroyed the chance of conducting an effective learning experience.

Iteration 1 Topic 5

Topic: Introduction to conditional statements ('if' statements)

Attendees: 10

Summary

OB1: Once again after initial housekeeping the tutorial questions are covered. Initially the scrolling of solutions is not synchronized but this is quickly detected this time (as opposed to the previous week). The teacher dominates collaborations, often just presenting and describing the solutions. For instance in question 3 regarding the outputs for different combinations of "if" statements the solutions are broadcast and described by the teacher, with very little student input resulting (refer to Figure 70).

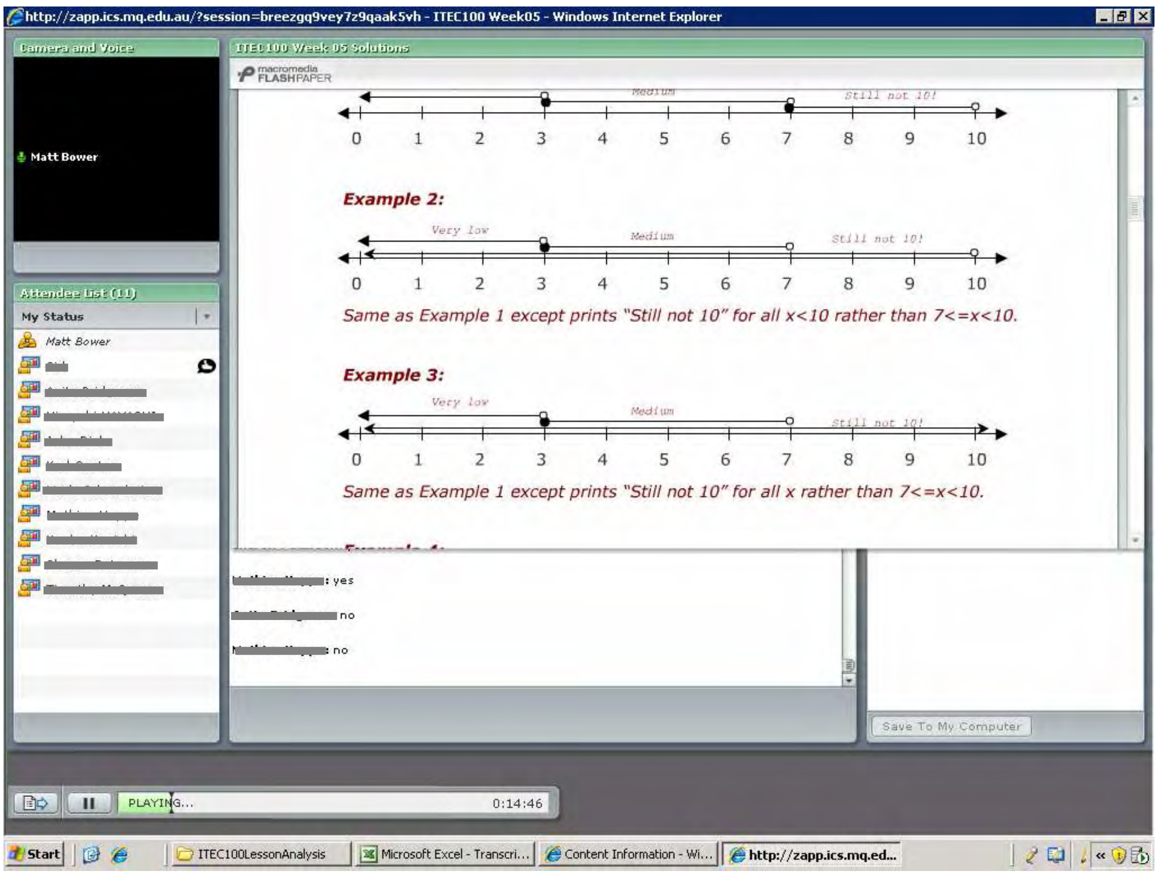

Figure 70 – Iteration 1 Topic 5 Presenting conceptual answers using pre-prepared diagrams

On some questions (such as those relating to the meaning of lazy evaluation, dangling else and the difference between the 'equals' method and 'equals' operator) the teacher asks students to describe their understanding. This provides the teacher with the opportunity to some extent assess the formedness of student mental models, as well as providing a stimulus for further discussion. The large number (thirteen) of tutorial questions were covered in a relatively small amount of time (47 minutes) using the more transmissive and text-chat question-response approach.

OB2: After the break the practical activities (for instance that requiring students to write a tax-calculator program) are covered using screen-share mode. The teacher again takes the approach of presenting and discussing good or novel student solutions, but again the more teacher-centred approach appears to result in limited amounts of student contribution.

OB3: Following this the students are given ten minutes to complete a task requiring them to combine the functionality of a "Month" program with a "Leap Year" to create a "Date" program. They do this on their own machines. Because this activity occurs in each individuals' private space there is no chance for students to negotiate understandings, receive feedback or learn "vicariously" (Bandura, 1977). None of the components of Laurillard's (2002) Conversational Framework are engaged because there is no discourse for students to apprehend.

OB4: Once ten minutes has elapsed the teacher asks students to leave the previous task (due to time restrictions) and listen to an explanation about the clock task from the previous week (which is relevant to their assignment). Eleven minutes is spent on this transmissive presentation, in which time students make no content related contributions.

Key Incidents:

- KI1: At one point a student is asked to explain his approach to using a 'switch' statement to grade student scores. He copies his segment of program code into the text-chat pod which removes all line-breaks and thus destroys the formatting of the code. Text-chat was an inefficient modality to represent this information – a note-pod would have been more useful because of the ease of contribution it afforded and its capacity to preserve the formatting of the text.
- KI2: There were occasions where the teacher was describing phenomena, with the artefact to which he was referring being shown at the end of or after the description. In reviewing the recording it was apparent how this detracted from the clarity of explanation in accordance with both the multimedia principle (Fletcher & Tobias, 2005) and the split attention effect (Ayres & Sweller, 2005).
- KI3: Two students express that there is problems with the broadcast of the screen-share, either causing their computer to freeze or being out of synch with the audio.
- KI4: Explaining the logical concept of De Morgan's law using only audio was sub-optimal because students were required hold pieces of information in working memory, unnecessarily increasing cognitive load. Diagrams and associated example cases should have been used to support the formation of students' mental models.

Reflective Notes:

- RN1: Using a program and/or the debugger to demonstrate programming concepts such as control flow and the dangling-else situation would allow students to acquire an understanding of the concepts in-situ rather than requiring them to work with abstractions. It is conjectured that showing code in action is the "process" phase of the abstraction cycle (Aharoni, 2000) which allows students to more confidently and accurately form their programming mental models. However such approaches take more time than presenting a textual-audio explanation.
- RN2: Having students setup with audio would be useful when students need to make more elaborate contributions or take a more central roll in presenting ideas.
- RN3: It would be useful to set approaches that make students demonstrate their mental models, allowing better exchange and remediation from the teacher and

between students. It is conjectured that certain tasks and certain room designs allow student's mental models to be more accurately revealed.

- RN4: Students might be quiet because they're disengaged, or because they're intensely engaged and concentrating. However without any feedback from students this distinction can only be based on teacher intuition, as indicated by Laurillard's (2002) Conversational Framework. With interactive tasks there is less need to rely on instinct or feel.
- RN5: Online teachers (just like classroom teachers) need to avoid the temptation and pressure to be instructing all of the time. While it is useful for some purposes, too much is unbalanced and does not allow teacher to gauge student progress or allow students to be adequately engaged to facilitate deep learning.
- RN6: Several (if not all) aspects of face-to-face teaching carry across to the virtual classroom teaching. For instance, issues relating to people management, scope and sequencing of instruction, use of tone and rhythm in speech and so on all apply.
- RN7: The process of reviewing lessons each week has allowed continual refinements to take place (for instance, through appreciating the functionality of the virtual classroom), which is a more natural approach to developing tactical adjustment than adopting a 4-weekly staged approach, for instance.

Iteration 1 Topic 6

Topic: Introduction to iteration ('loops')

Attendees: 9

Summary

OB1: Three minutes after starting the tutorial question coverage there were problems with the web-conferencing environment (some students were unable to see the solution document being broadcast and others were unable to scroll through it). After nearly three minutes trouble-shooting the teacher provides students with a link to a new room so they can continue. However uploading the solution document in the new room still takes much longer than normal (over 30 seconds in some cases, potentially due to network problems).

OB2: In part due to time pressure caused by the prior technological problems, the teacher adopts a transmissive approach to covering the solutions to the tutorial questions. For instance the question relating to debugging an erroneous factorial loop is covered by purely descriptive techniques, whereas modelling the debugging process on the debugger or at least using a diagram would be clearer. At one point when the teacher asks if there are any questions or if students would like to show some of the loops using the debugger there is no student response. When asked if they were happy to continue with the instructive approach five students responded affirmatively.

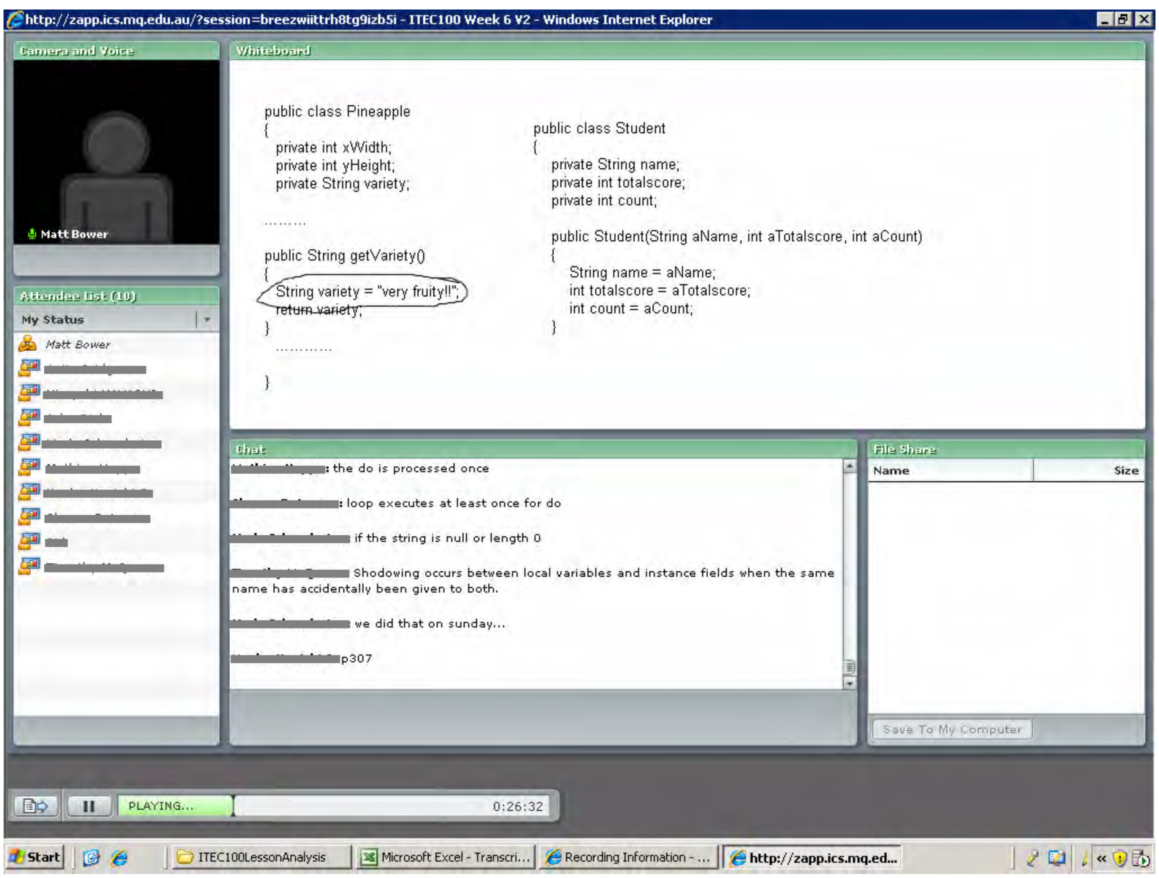

Figure 71 – Iteration 1 Topic 6 Using a whiteboard to share student answers

OB3: At one point the teacher stipulates “time for input from you. Can you copy and paste your examples of shadowing to the whiteboard.” Some students then copy and paste their solutions to the whiteboard (see Figure 71).

This provides students with the opportunity to compare and contrast their solutions in the one solution space, preventing attention from being split. However once the code is on the whiteboard the teacher contributes all the discourse, with the only student input being the responses of “no” to the question “do you have any questions?”. The teacher’s dominance of the episode meant that student mental models were not revealed despite the opportunity for conversational approaches.

OB4: After the break some in class conceptual activities are performed that the students had not previously attempted at home. The questions related to the number of times various loops would iterate. Students were asked to respond using text-chat how many times they thought each loop would iterate, providing an efficient means of eliciting several students’ ideas at once (see Figure 72).

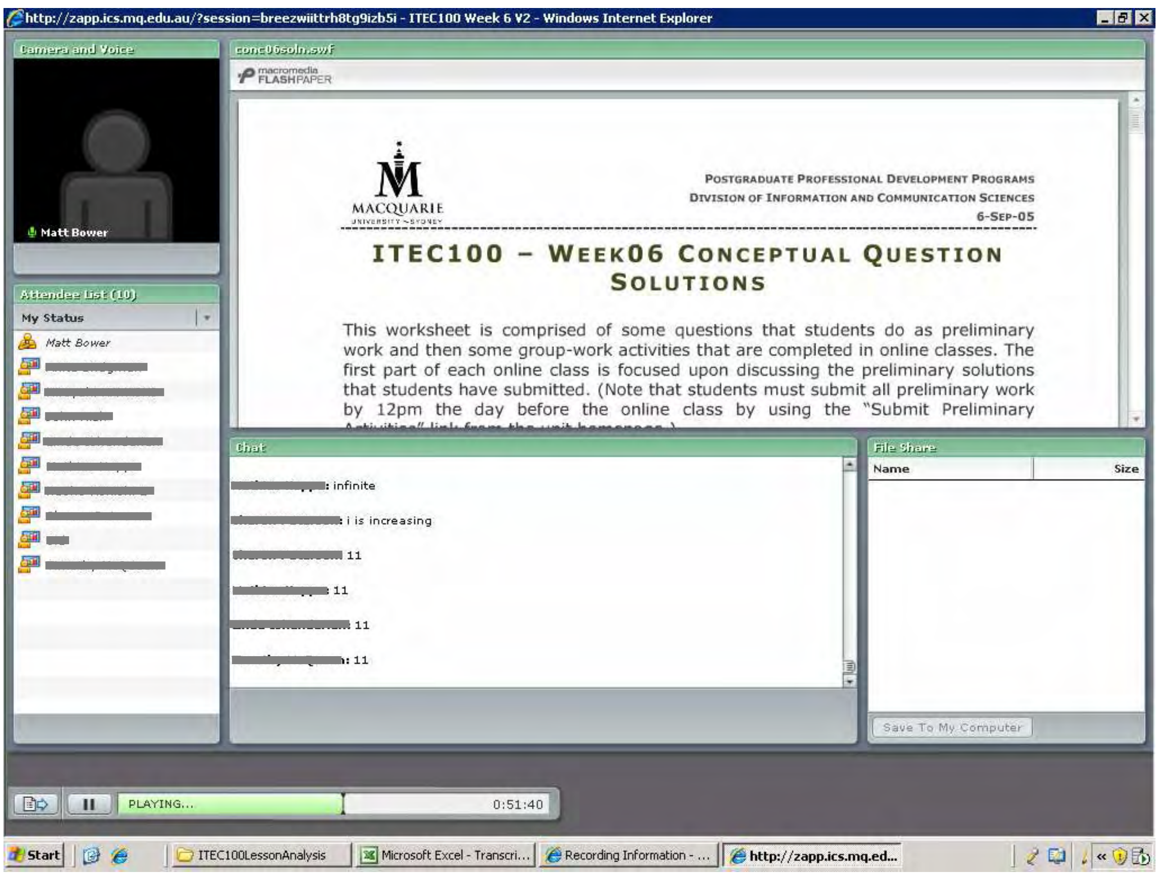

Figure 72 – Iteration 1 Topic 6 Soliciting declarative knowledge from several students at once using text-chat

OB5: The teacher then asks students what the output of the nested loop containing the modulus 10 function will print. Some students seem vague or unsure and so the teacher opts to run through the program using the debugger, which provides students with a visual representation of a “notional machine” (du Boulay, O'Shea, & Monk, 1989). Student feedback indicates this dynamic representation assists their learning.

OB6: The next section of the lesson covers the practical tasks. First a student's random colour circle applet is shown. One student asks if it would be possible to adjust the program to create entirely random colours (rather than select from three). The teacher attempts this but fails, which begins a debugging session. The problem is eventually rectified after referring to the Java API. This provides the valuable opportunity for students to see a situated debugging task being resolved. The teacher also broadcasts and discusses two other student solutions before ending the lesson.

Key Incidents:

- KI1: The teacher uses brief text-chat contributions to reiterate verbal indicators of which tutorial question is being attempted, which also act as a delineator in the chat history.
- KI2: One student (AB) cannot see the conceptual solutions until 28 minutes into their use.
- KI3: The audio explanations to the tutorial questions being provided by the teacher are often of sub-optimal clarity. They are unrehearsed, often circular and unaccompanied by visual aids. The lack of visual aids fails to utilize either the modality principle (Low & Sweller, 2005) or the multimedia principle (Fletcher & Tobias, 2005).
- KI4: The debugging broadcast of the 'modulus ten' loop makes the logic underpinning the output very clear. The debugger provides a visual representation of the machine that allows students to more clearly form their mental models (as opposed to having students attempt to construct a mental model on the basis of the teacher's audio descriptions).
- KI5: The ad hoc adjustment to the Random Circle applet provided students with an example of debugging in practice, allowing students to observe problem solving in a "situated cognition" (Brown, Collins, & Duguid, 1989) context.
- KI6: The teacher talked about the RandomCircle program before running it. This prevented students who possessed unclear models about how the program operated from relating the code to the output. Running the program before explaining how it worked would make the explanation of the code more relevant and understandable.
- KI7: When covering the in-class activities regarding the number of times loops operate the teacher intends to broadcast the questions and then reveal the solutions one by one after students have attempted them. However, the teacher does not have synchronized scrolling on so that students cannot see the questions to which he is referring (so they must reference them from their question sheets, resulting in split attention).

Iteration 1 Topic 7

Topic: Introduction to polymorphism

Attendees: 8

Summary

OB1: After housekeeping and a brief delay while the teacher sets up the first activity, the lesson begins with the coverage of the first pre-class practical exercise. Screen-sharing is used to show the program code for the polymorphism example (see Figure 73). The teacher gives a six minute audio explanation of the code during which time there is no student input because full screen mode is being used (which prevents them from typing in the chat-pod).

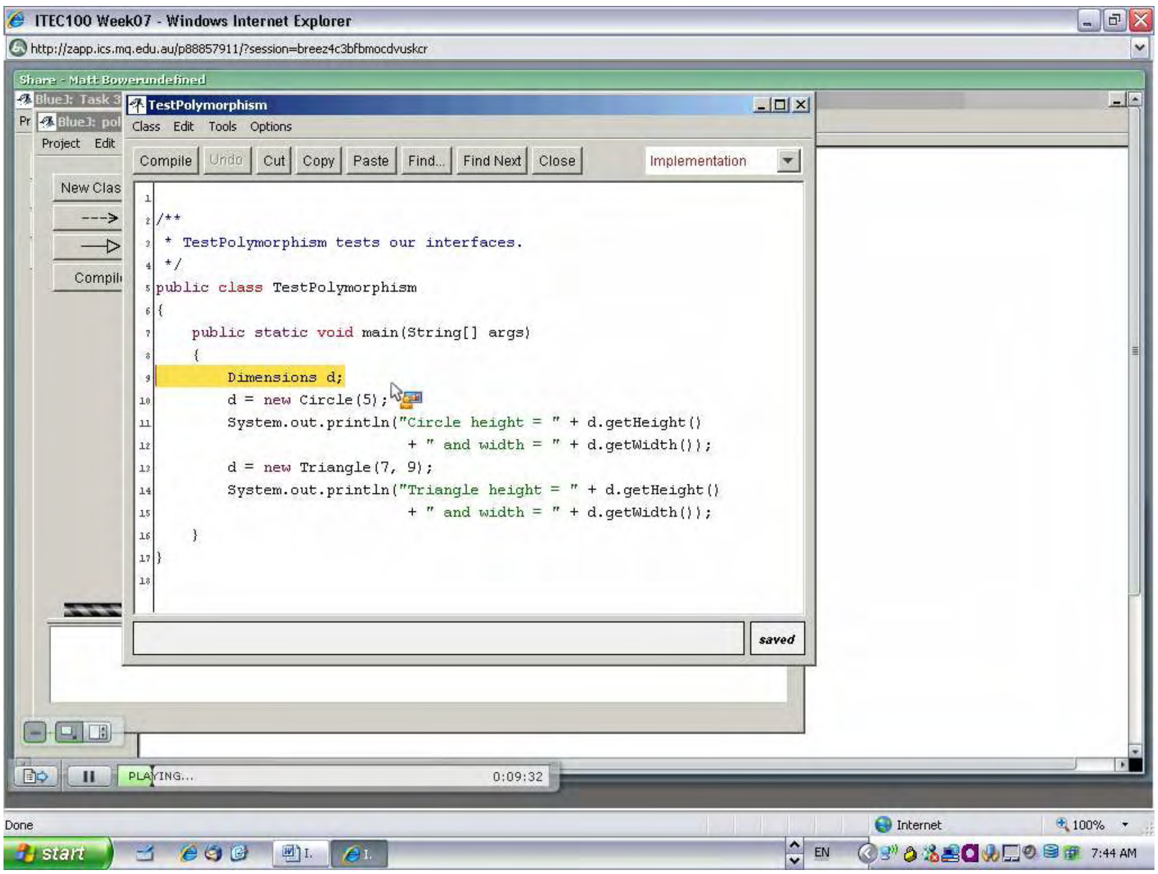

Figure 73 – Iteration 1 Topic 7 Using full screen mode to show source code files of the polymorphism example

After full screen mode is switched off students indicate that they did not have any questions anyway. The teacher then spends another four minutes explaining the polymorphism exercise, switching between the source code files of the example. After this time the students respond that they do not have any questions (when asked). Students were recipients of information and did not appear engaged. Switching between files made it difficult for students to interrelate program code contained in different files because they had to hold items of information in working memory. This resulted in split attention (Ayres & Sweller, 2005).

OB2: The tutorial questions are covered briefly with the students providing text-chat responses to the teacher's audio questions and prompts. The teacher provides further explanations rather than encouraging students to elaborate their conceptions. However given the amount of time spent on the first activity the approach adopted represents an efficient and potentially sufficient way to cover the material.

OB3: Once the conceptual solutions have been covered teacher directs students to attempt in-class practical task 2 involving creating a colour interface for the program used in the first activity. Students are given 15 minutes to try this at home. Once again there is no discussion between students which means that they cannot gain from the benefits of collaborative learning approaches. One student completes the exercise and uploads their solution to the file-share pod. The teacher then talks through the student's solution with the class. Screen-share is not used meaning students cannot see the specific pieces of code to which the teacher is referring, resulting in split attention (Ayres & Sweller, 2005), and compromised ability to signal which aspects of the code students should focus upon (Mayer, 2005b).

OB4: The teacher then uses a question-response approach to covering the three in-class activities relating to casting, in each case followed by a teacher explanation. It appeared that some students did not understand the concept of casting. The teacher's spontaneous attempts to explain this using only audio did not rectify all students' conceptions. Once again using a purely auditory mode of communication without visual aids increases students' cognitive load (van Merriënboer & Ayres, 2005) and reduces the capacity to provide a representation of an accurate mental model which describes how the items of knowledge associated with casting interrelate.

OB5: Next students were directed to the RandomShape program, and the teacher asks two students if he may broadcast their solutions (to which they express agreement). Twelve minutes is then spent trying to resolve with students the preferred way to broadcast (how to use full screen toggle). Finally the teacher broadcasts the programs, accompanied by transmission style explanations. During this time there is limited student input.

OB6: At the end of the lesson there is not enough time to cover the clock and population counter examples (which are quite involved) and as such the teacher refers students to the solutions which they can download.

Key Incidents:

- KI1: In the introductory polymorphism example the switching between the various layouts and java files was disconcerting and prevented students from being able to clearly draw relationships between the four classes of the polymorphism example. This is a classic example of split attention (Ayres & Sweller, 2005).
- KI2: While covering the preliminary tutorial questions the teacher may be revealing more of the solution document than expected (due to the different sizes of the sharing pod for different screens). This means while the teacher asks for student answers they are already being displayed.
- KI3: The teacher discussion surrounding the student's solution to the Colour interface problem is so much less clear because the actual code being described is not broadcast. This results in split attention (Ayres & Sweller, 2005) and failure to leverage the multimedia principle (Fletcher & Tobias, 2005).
- KI4: Before the last demonstration regarding the RandomShape program twelve minutes is spent while attempting to understand the best approach to screen-share.

Students want a clearer (larger) broadcast but using full screen mode however this prevents them from typing into the text-chat pod. While the teacher mentions the scroll button (which allows students to see the broadcast using an aspect ratio of one-to-one) he does not emphasize its use. Insisting students use this feature (or having students use audio) would have resolved this issue.

Reflective Notes:

- RN1: During this lesson the power of adopting a "conversational" approach (Waite, Jackson, & Diwan, 2003) with the student while doing the activities was observed. Asking "why" was often a successful strategy for revealing students' mental models. The explanations that students posit provide an indication of their level of understanding. In this lesson there were several periods where the teacher did not query students understanding, which corresponded with low levels of student contribution.
- RN2: Students appeared very confused with the concept of casting. The solely verbal approach appeared to be of limited help. It is important to develop student's mental model of why casting is needed – an explanation of its rationale and purpose can then contribute to students understanding of when it needs to be used. Without this they did not have a way of thinking about casting. While it is important for them to acquire rules of casting, such as "Casts are required whenever converting from an interface", it is critical to provide an explanation of "why" (ensuring a relational and not just procedural understanding). A diagrammatic approach may be useful to support mental model development.
- RN3: For practical task 2 it would be much more useful for students to try to work together to write a colour interface.
- RN4: There are times when a transmissive approach to learning is appropriate. However often throughout this lesson more student engagement was desirable but not achieved. Reasons for this need to be understood and effective approaches to achieving collaborative learning in web-conferencing environments need to be identified.

Iteration 1 Week 8

Topic: Introduction to events and control handling

Attendees: 7

Summary

OB1: This week instead of answering a series of tutorial questions, for the pre-class tutorial exercise students were required to write a two page explanation of event handling. Because the students who volunteer to present their explanations did not have audio set-up in advance the teacher opts to present their work for them. However, the teacher explanation of the students' work is not fluent because it was not anticipated that he would have to do this (so the monologue was not practised). After spending approximately 8 minutes broadcasting some of the students' solutions the teacher then spends 5 more minutes presenting the model solutions. In all this time the teacher does not ask for questions. However two students do ask a question, which are the only two comments made by any students during this approximately 15 minutes sequence. The teacher (rather than other students) provides the response. The teacher then presents one more example, after which a student asks a question that prompts a brief whole class discussion about the use of event handling in industry. In this episode the collaborative mode appears strongly determined by the teacher.

OB2: The teacher then broadcasts student solutions to the practical activities using a standard screen-sharing layout, though two minutes is required to find and setup their programs so that they are ready for presentation to the class. The first task relates to debugging a student's (SA's) drawCircle solution so that the mouse-click determines the circle centre rather than the top left corner of the bounding box. The code is presented without first running the program. One student is able to detect the problem just from inspecting the code, and on this basis other students suggest solutions.

OB3: Following this the teacher broadcasts another student's solution to the second practical question requiring a text field input to change the radius of the circle drawn on an applet canvas. While attempting to adjust the program so that the circle maintains the same centre when the radius is changed students ask questions and the teacher provides elaborate explanations. Even though there is an element of interactivity, the teacher assumes the role of knowledge provider.

OB4: After the mid lesson break students are provided with a zip file containing working programs from the two previous tasks and are presented with the exercise of combining them into one program. The teacher explains that he will lead a screen-sharing session where they provide him with instructions on the actions he should perform. This results in approximately 20 minutes of student-teacher interaction with 41 separate text-chat contributions by students. As well, a student question asks the question whether it is possible to change the radius of the circle upon clicking on the applet canvas, which results in a spontaneous discussion and demonstration that takes approximately 5 more minutes. Having in interface that provides a shared artefact to work upon and an authentic problem to solve coincides with raised levels of engagement, in accordance with conjectures by Jonassen (2000).

OB5: The teacher then broadcasts and explains the solution to task three requiring students to write a method that creates buttons. Finally a student program that changes the colour of a circle each time it is graphed is presented to the class, again using a transmissive approach to instruction. During the 28 minutes over which these presentations are made students only make seven contributions to the text-chat pod.

Key Incidents:

- KI1: When debugging SA's example the teacher once again starts by inspecting the code rather than running it, which makes it more difficult for students to comprehend how the program operates. Running the program first would allow students to see the problems in the output rather than identifying them directly from reading the code (which is much more difficult for them at this stage). Starting with running the program allows them to form a mental model to which sequences of code can be associated, rather than needing to elaborate the running program from the code.
- KI2: Two students once again request that the font size used in the IDE be increased so that they can more easily see the code being broadcast. They still require teacher prompting to use the scroll button which allows the aspect ratio to be one-to-one.
- KI3: While the teacher is presenting a solution to the circle colour changer program, SA asks the question "Is it possible to change the colours in a sequential pattern through button clicks ?" and LA responds with "you could put a counter variable in and when the modulus is a certain value when divided by 3 set a different colour". This conversation occurs while the teacher is talking, demonstrating the utility of text-chat for facilitating cross-collaboration that does not interfere with (teacher) audio broadcast. In this way the teacher audio, visual broadcast and student text-chat forms an effective "multimodal cluster" (Baldry & Thibault, 2006)
- KI4: At the end of the lesson a student indicates frustration about their poor connection (due to network problems they were experiencing).

Reflections:

- RN1: To the extent that the web-conferencing environment provides a communal thinking space, during the presentation of student conceptual solutions the teacher entirely dominates that space. Because no activity is required from students they appear more likely to disengage.
- RN2: The tasks that require students to contribute, to solve a problem, appear to lead to far more engagement, collaboration and questions. This in turn appears to correlate with student learning.

Iteration 1 Topic 9

Topic: Abstract classes

Attendees: 9

Summary

OB1: The tutorial questions were once again covered using a question-response approach, where the teacher asked a question like "What are abstract classes?" and the students used text-chat to provide responses. The teacher also challenges students with related and probing questions resulting in considerable student discourse.

OB2: At one point two students indicate that they do not understand why superclass variables can store references to subclass variables but not the other way around. The teacher spends approximately seven minutes drawing and discussing an example on the whiteboard to help clarify students understanding (ref. Figure 74).

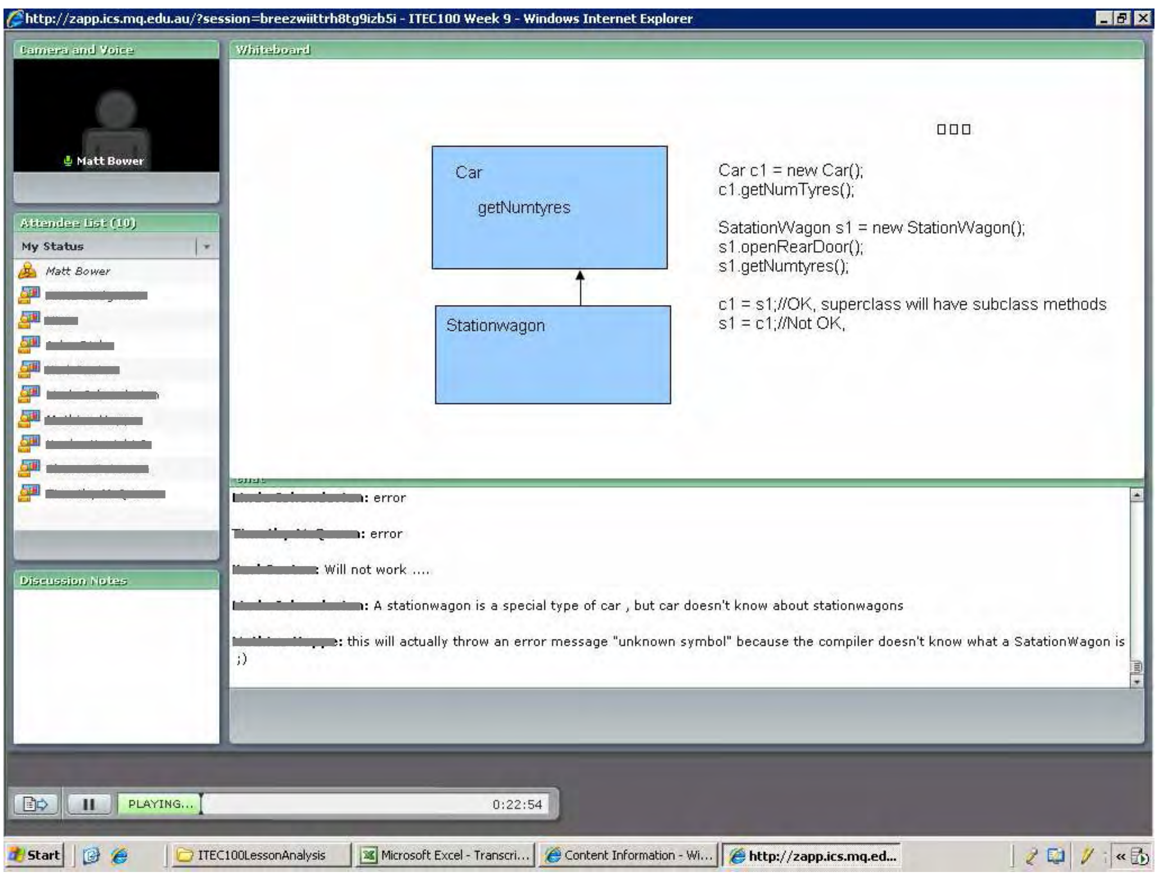

Figure 74 – Iteration 1 Topic 9 Drawing on the whiteboard to support a conceptual explanation

This allows the code to be related to a visual representation of how superclasses and subclasses operate, leveraging the multimedia principle (Fletcher & Tobias, 2005). This involved a lower level of cognitive load than would have been required if a purely verbal explanation was used because some of the cognitive effort was offloaded to the environment (Hollan, Hutchins, & Kirsh, 2000).

OB3: More tutorial questions are covered using a question-response approach. At one point the teacher spends another nine minutes using the whiteboard to draw a diagram supporting the explanation of the difference between a shallow and a deep copy (again clarifying the concept beyond that possible if only audio had been used). However, the time required to draw these diagrams is substantial considering they only cover two of the several concepts of which the lesson is comprised.

OB4: After the break the teacher broadcasts a screen-share of two solutions to a practical question regarding extending the Rectangle class to make a Square class. The teacher encourages students to compare and contrast approaches, and at one point reflects a student question back to the class by asking "which do you think is better?". This stimulates a class discussion. However broadcasting the two programs one after the other results in split attention (Ayres & Sweller, 2005) during this evaluative task.

OB5: The teacher then asks for volunteers to upload their solution to the MU administration system question. After some initial set-up time the teacher discusses student solutions. However the ad-hoc commentary is unprepared and undirected. Only in the final minutes of the 20 minute section relating to the MU administration system is there a learning conversation with students, and this follows the teacher's question regarding the utility of declaring the Person and Student class to be abstract. Students were divided as to whether this was the best approach and some discussion resulted as people explained the reasons for their position. Because the teacher had not set any goal, students performed no activity, which prevented the teacher from being able to gauge their level of understanding, provide any meaningful feedback or adjust the task in light of student's actions. Thus a lack of clear goal led to flow-on effects that prevented the stages of learning identified in Laurillard's (2002) Conversational Framework from being implemented. This was the last activity conducted in the lesson (apart from an explanation and guiding comments regarding students' major project assessment task).

Key Incidents:

- KI1: In this lesson student engagement and collaboration was increased by the teacher deliberately reflecting student questions back to class rather than answering them himself.
- KI2: The tactics of repeating student comments to emphasize them and reinforcing the question being addressed by typing it in the text-chat were used on several occasions in this lesson.
- KI3: Use of the whiteboard to clarify superclass/subclass referencing and deep/shallow copies was slow (because of the limitations of the whiteboard tool), but resulted in a clearer explanation than using audio alone (Multimedia Principle, Fletcher & Tobias, 2005).
- KI4: There were more examples of poor teacher web-conferencing skills in this lesson. During a section of the screen-share presentation the browser window of the virtual classroom had not been minimized so student messages didn't pop up. On another occasion the teacher had not selected share screen mode and so was discussing a program while unknowingly failing to present it. As well there was an instance where the scrolling feature of a document being shared was not put on 'synchronize' mode meaning the part of the document to which the teacher was referring was not visible to students. Finally, when the lesson started the teacher's audio levels were incorrectly set prompting student requests to increase the volume.

- KI5: The teacher provides another explanation of the scroll button following yet another student question to make the font larger.

Reflections:

- RN1: Reviewing the tutorial questions quite quickly with quite a few stimulating questions about "why" etc appeared to be an effective way to get students to reveal their mental models. Revelation of students' mental models requires that they be allowed to express those models in the way that they are represented in students' minds (Symbol System Theory, Salomon, 1994). Often this will be visual, implying students should be using the whiteboard.
- RN2: The teacher should aim to eliciting student feedback at least every two minutes (especially while delivering a long explanation) in order to gauge their comprehension and preferences regarding the direction for the learning episode.
- RN3: The extensive yet unfocused presentation of student solutions to the MU administration system practical task did not engage students. A compare / contrast / evaluate task would have required their interaction rather than teacher just discussing the code in an ad-hoc fashion.

Iteration 1 Topic 10

Topic: Graphical User Interfaces

Attendees: 9

Summary

OB1: The tutorial questions are covered using a question-response approach. For instance the first question asks "What's the difference between a label, panel, container, layout manager and frame?" and students provide (rapid) textual responses in the chat pod based upon prompts by the teacher. The last tutorial question requires students to explain how a simple Graphical User Interface program works, line by line. The teacher spends nine minutes explaining this while model solutions are broadcast. During this explanation to the last tutorial question students are listening and not contributing at all. All tutorial questions have been covered within 27 minutes.

OB2: The teacher then broadcasts the IDE to demonstrate the solution to the practical activity requiring the use of radio buttons to change the colour of a panel. During the 9 minute transmissive explanation there is once again no input from students.

OB3: Students are then asked to provide the teacher with directions on how to adjust the program from the previous activity so that it uses a drop-down menu rather than radio-buttons to change the background colour. The teacher broadcasts his screen and students contribute instructions on how to solve the problem. A great deal of discourse arises during the 21 minutes of the activity. The task requires students to use their problem solving skills and engage with the activity rather than be the passive recipients of information. The screen-share modality allows programming process knowledge to be communicated in a context that resembles that in which it would be expected to be applied, in accordance with Symbol System Theory (Salomon, 1994) and 'transfer appropriate processing' (Bransford, 1979).

OB4: After a five minute break the students are asked to find investment calculators on the Internet that might provide an idea of how to design the interface for their major assignment. Students work on this for eight minutes individually, with occasional comments and links posted through the text-chat.

OB5: The final part of the lesson relates to providing further guidance on their major assignment. After spending 10 minutes performing a screen-shared explanation of how they might design and use classes to form the back-end of their application, the teacher uses a whiteboard to draw a possible layout for their program (see Figure 75). The teacher then refers back to the code to demonstrate how this layout might be created using programming code. There is only occasional student discourse during the 23 minutes the teacher discusses the assignment, however based on the questions and comments the students do make they appear to be intently engaged. Student engagement may have been heightened as the assignment was worth 10% of their assessment for the subject.

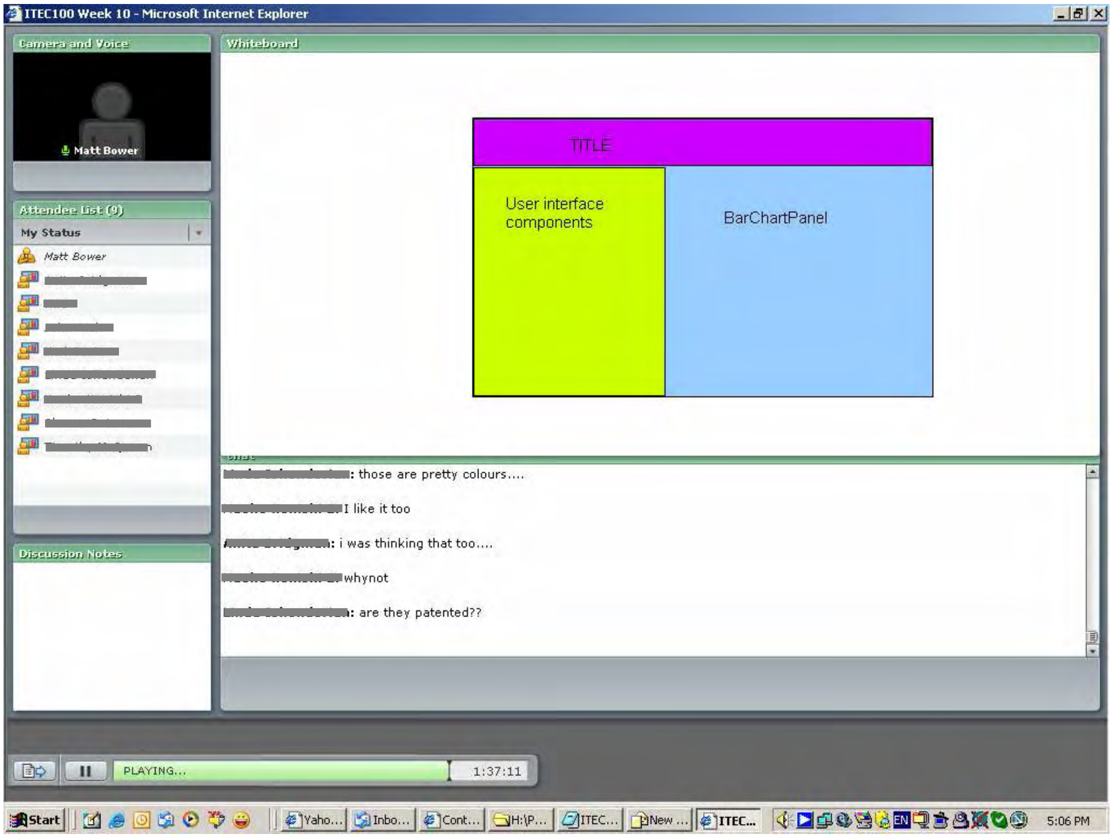

Figure 75 – Iteration 1 Topic 10 Using the whiteboard to represent visual information

Key Incidents:

- KI1: There were further examples this week of how the teacher's use of the web-conferencing system could be improved. The teacher's audio levels were once again incorrectly set at the beginning of the lesson. As well, the first text-chat message that the teacher posts is accidentally directed to one student.
- KI2: Again, a rapid question-response approach appeared to be an efficient way to cover short declarative questions in the conceptual exercises.

Iteration 1 Topic 11

Topic: Arrays and Streams

Attendees: 9

Summary

OB1: Again a question-response approach to covering many of the tutorial questions is adopted. For instance to the first question "What are the advantages of ArrayLists over arrays?" elicits the following discourse:

> SP: no fixed size
> AB: they(arrayList) are flexible in size
> TM: you can use methods
> LI: they can hold objects
> MH: you can just add items and make the list longer
> KC: ArrayLists ... dynamic ...
> HH: lager size can be created.

Short pieces of textual knowledge that do not need to be related to one-another can be efficiently elicited and represented using this approach.

OB2: For some of the tutorial questions requiring more elaborate responses the teacher chooses to present and discuss the model solution document (partially due to the difficulty in having students represent more elaborate responses using the text-chat and partially as a way to spend less time on the questions). Students make a total of 155 text-chat contributions during the 47 minutes spent covering the tutorial questions, the vast majority of which is directly focused on discussing the curriculum matter.

OB3: After the break the teacher spends 22 minutes broadcasting his screen using the standard sharing interface and tracing through a BankAccount example that demonstrates exception handling (see Figure 76). While discussing the example students make 30 comments, only 4 of which relate exclusively to curriculum matter. The other comments relate to matters such as whether people have looked at the question or problems one student experiences with the audio transmission. This very low rate of student discussion relating to curriculum matter coincides with a highly transmissive approach adopted by the teacher.

OB4: Due to lack of time the teacher requests that willing students upload their solutions to the tutorial questions into the file-share pod and compare and contrast their approaches for homework.

OB5: The last section of the class covers how the concepts they have learnt regarding arrays can be used to help them with the assignment. Students are confused by the logic underlying the valuation of payment series. After considerable discussion regarding this, a diagram is drawn on the whiteboard in an attempt to clarify the approach demonstrated using code. Responses from some students indicate that this was clarifying. The teacher has chosen to allow the class to run overtime based on the perception that the discussion is valuable for students.

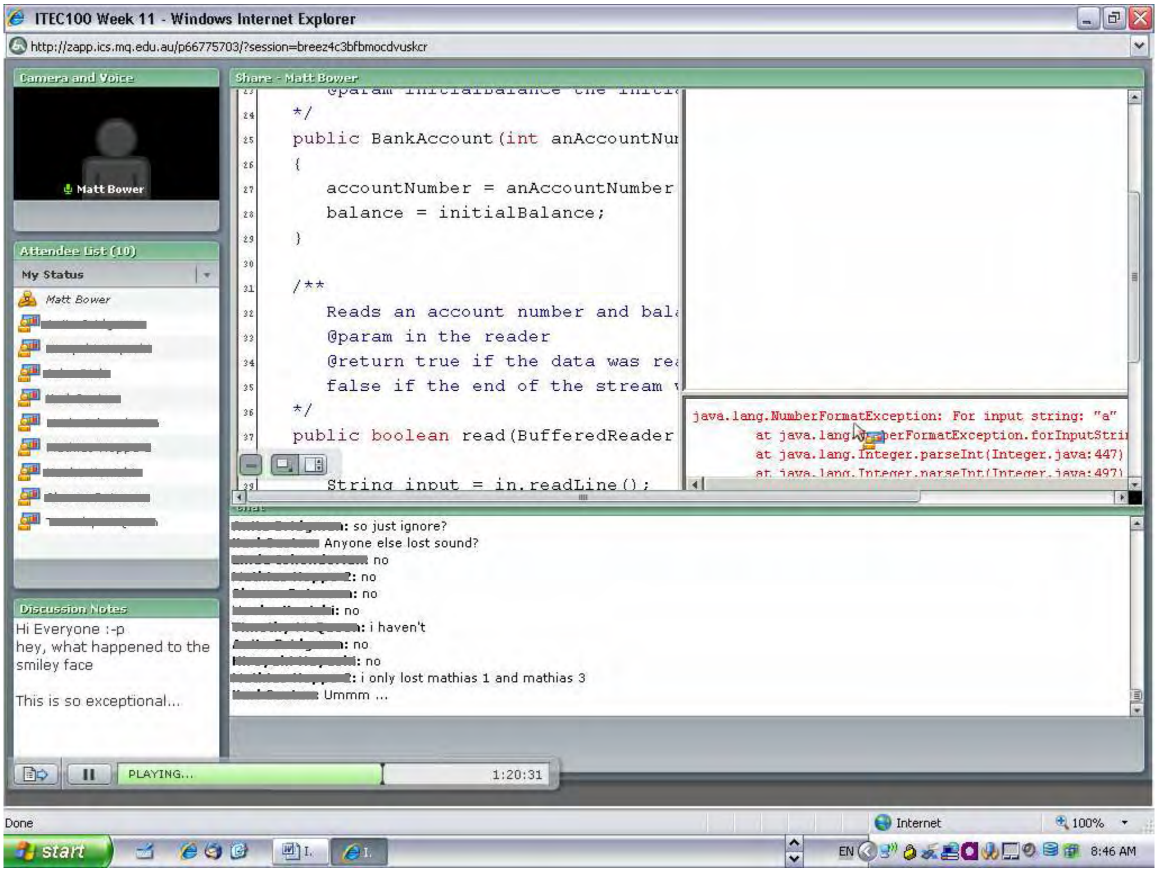

Figure 76 – Iteration 1 Topic 11 Transmissive explanation of programming using Exception Handling

Key Incidents:

- KI1: One student (KC) loses sound for some time during the demonstration of practical question 4.
- KI2: The teacher again applies the approach of repeating student text-chat questions using audio to emphasize them. This has become a pervasively applied teacher tactic.
- KI3: During the discussion regarding the assignment the teacher once again forgets to broadcast his desktop while talking about a program on his screen. A student has to turn their audio on to inform the teacher of this.
- KI4: During the discussion of the assignment a student (MH) only understands how the payment calculations work and the corresponding graphs can be produced after the teacher draws a diagram of 4 payments on the whiteboard.

Reflective Notes:

- RN1: In the web-conferencing environment there is a greater need to insist on explicit feedback from students about their level of understanding because there are less emotive cues available from students than in face-to-face environments.
- RN2: Whether text-chat is sufficient to facilitate student contributions depends on the form of response that the teacher anticipates. Text-chat is appropriate if small chunks of unrelated declarative knowledge are required but not if more elaborate and/or visual representations are required.

Iteration 1 Topic 12

Topic: Files and System Design

Attendees: 9

Summary

OB1: The teacher set the tone for the tutorial questions by commenting that students should be used to the approach to covering them by now and so the class should be able to complete them quickly. The teacher's rapid questioning style relating to the large number of declarative questions contained in this week's preliminary conceptual exercises resulted in periods of rapid and numerous student responses. As well, students ask several independent questions in an attempt to clarify their mental models. For instance, relating to the question regarding moving the file pointer to the middle of a file, a student (KC) asks "What hapens if the number [length of file in bytes] is uneven .... there is a remainder ?". This results in the teacher briefly reminding students how integer arithmetic discards remainders. Once again, multiple contributions through the text-chat pod allows students to represent a multistructural level of understanding but not a relational level since the capacity to interrelate the information is limited.

OB2: After covering the preliminary tutorial questions on both file access and system design, an in-class activity relating to association and inheritance is covered. The exercise provides students with a java file and requires them to identify inheritance, association and dependency relationships. The conversation that results supports the development of student's understanding of how the concepts (abstractions) of relationships are represented in program code (concrete examples). The key concepts are summarized verbally by the teacher and emphasized using text-chat. For instance the teacher types in the text-chat pod "instance fields types => association relationships".

OB3: After the break the practical questions are covered using a standard screen-share broadcast. The teacher uses a transmissive approach to present a model solution to the exercise requiring every full-stop in a file to be replaced by an exclamation mark. Student solutions to other file-reading practical tasks are presented, during which time there is no conceptual input from the students. The transmissive approach adopted by the teacher allows the practical questions to be covered in approximately 16 minutes.

OB4: The final 28 minutes of the lesson are spent on an in-class activity requiring the creation of a program to change all capitalized HTML tags in a file to lowercase. The students and teacher work together to incrementally change a basic file-reader program to achieve the specification. The teacher broadcast their screen and prompted the students for text-chat suggestions. A conversation is engaged. This results in thirty-four student contributions to the text-chat pod, including several student questions. These questions reveal weaknesses in student mental models that can then be remedied. Students expressed their interest in the task, with comments such as:

> LI: thats pretty cool and practical stuff
> NK: I agree
> KC: Nope ... very good ... I ould use that at work ...
> HH: Wow That's very cool!! That's very useful to me!

The authentic and relevant task (Herrington, Oliver, & Reeves, 2002) seemed to heighten the level of student engagement.

Key Incidents:

- KI1: The established collaborative pattern adopted for the preliminary conceptual activities appears to allow students to immediately address the curriculum material and progress through it quickly. No time needs to be spent discussing how they are supposed to use the technology to collaborate, or the role they are supposed to play in the episode.
- KI2: While covering the conceptual material, teacher questions followed by a pause for students to respond encouraged many student contributions. Alternatively when the teacher makes contiguous statements regarding content or even asked questions without providing wait time for their response, student contributions were few.
- KI3: By summarizing the key concepts from the in-class activity on association and inheritance relationships the teacher removes the possibility for students to perform and report on this abstraction. This is an example of the teacher performing work that the students should have been undertaking. If students described their abstractions it would have allowed their mental models to be revealed.

Reflective Notes:

- RN1: The best learning in this episode appears to result from "tightly coupled" (Neale, Carroll, & Rosson, 2004) collaborations between students and the teacher that are focused upon forming mental models.
- RN2: Student contribution appears to be encouraged when they are made responsible for solving a problem (often in conjunction with the teacher).
- RN3: Collaborations do not appear to be rules based in so far as doing a particular set of actions will automatically manifest high quality or at least large amounts of collaborations. Collaboration appears to be highly complex and dependent on many interrelated factors on many levels (teacher tactics, technology interface, task type, activity structure).

Summaries of Iteration 2 Lessons – Semester 1 of 2006

Iteration 2 Topic 1

Topic: Topic: Introduction to writing, compiling and running java programs, as well as providing a general introduction to the course.

Attendees: 11

Summary

OB1: The web-conferencing environment was introduced using a similar approach to Semester 2 of 2005. Students were asked to introduce themselves using text-chat, and provided with the opportunity to use the tools such as the polling tool, whiteboard, and so on. The features of the web-conferencing environment such as the bandwidth options and user status were explained. Students in this semester appeared equally able to learn and use the basic functionality of the web-conferencing system.

OB2: After an early break the tutorial questions were covered using the same question-response approach of Iteration 1. The teacher used audio to queried students regarding the preliminary tutorial questions and students provided text-chat replies and comments (see Figure 77).

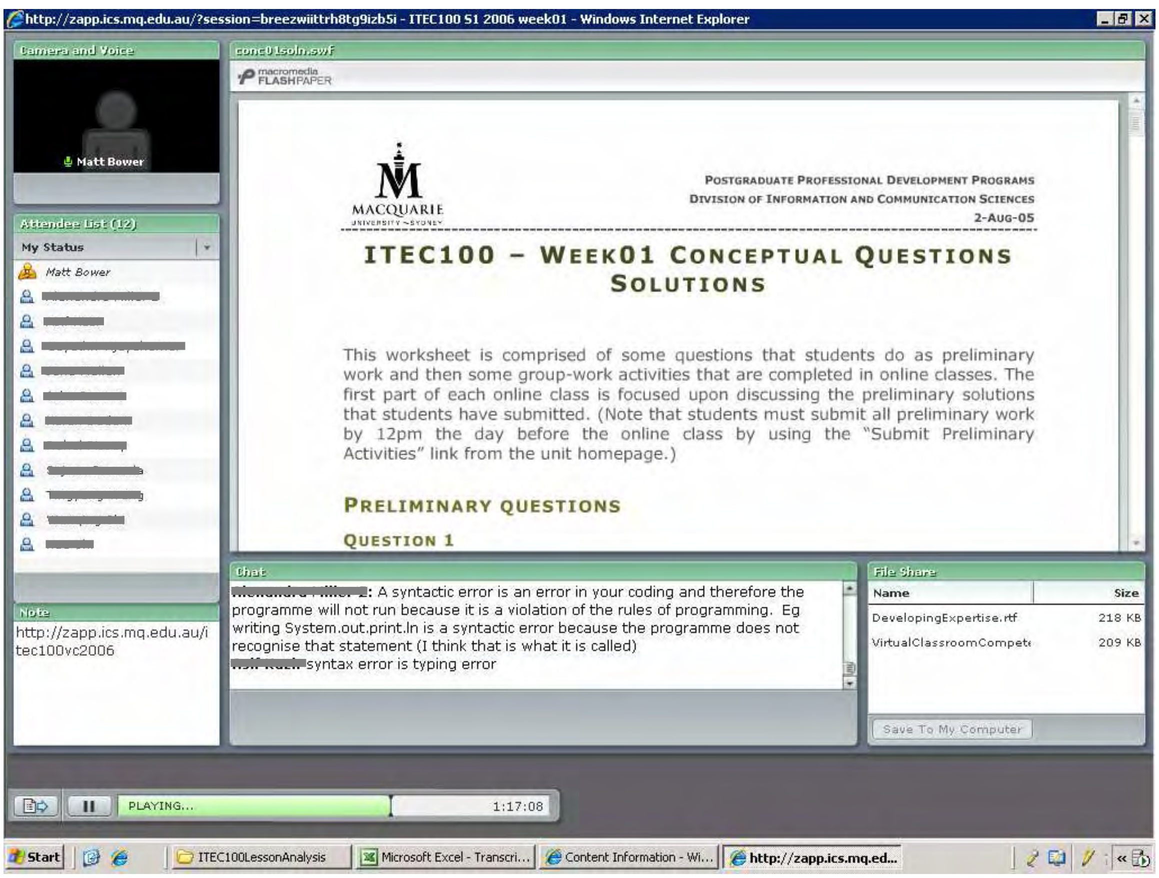

Figure 77 – Iteration 2 Topic 1 Covering tutorial questions using a similar learning design to Iteration 1

OB3: The discourse that resulted from applying similar approaches appeared similar to 2005 Semester 2, even though different students were participating. For instance, the teacher question regarding the meaning of "syntax error" results in the following student text-chat discourse:

> XS: syntactiv error will stop program from running
> JB: If the programmer has incorrectly used the programming language when writing their program, a syntactic (or compile-time) error will result. Each programming language has its own set of language (syntax) rules. If these rules are not strictly adhered to, the compiler will generate an error when it tries to convert the program to byte code. All syntactic errors must be rectified by the programmer before the compiler will successfully compile the program.
> AM: A syntactic error is an error in your coding and therefore the programme will not run because it is a violation of the rules of programming. E.g. writing System.out.print.ln is a syntactic error because the programme does not recognise that statement (I think that is what it is called)
> AK: syntax error is typing error
> SR: A syntax error is a violation of the rules of the programming language. Can be detected by the compiler.
> KD: something wrong wid format

Students had been encouraged to copy and paste answers into the text-chat pod if appropriate, which led to large sections of text being contributed in some cases by some students. When the teacher requests an explanation of the term "logical error" students contribute the following:

> AK: logical error is runtime error
> XS: logical error occurs while program could still running doing things unexpected
> WS: synta error like u spelling sth wrong

The teacher then elaborated on these responses.

OB4: The same modalities as Iteration 1 are used throughout the remainder of the tutorial questions. For instance in the last preliminary tutorial question regarding how a computer compiles and runs a Java program the solution document is broadcast, which overcomes the problem of providing an answer incorporating visual representations.

OB5: Students have similar technological questions to Iteration 1, for instance regarding the size of the screen-share broadcast. The teacher adopts a slightly more dominant approach to completing the class debugging task than was used in Iteration one (in part due to time constraints). Students appear to contribution correspondingly less to the episode.

Key Incidents:

- KI1: The large sections of text that students were contributing to the text-chat as a result of copying and pasting their preliminary tutorial question solutions at times overwhelmed the solution space and made the discourse in the text-chat pod more difficult to follow. A note-pod may have been more appropriate – it would have afforded the capacity to preserve formatting and would have prevented the text-chat channel from being consumed.
- KI2: The teacher employs the tactic of placing prompts in the text-chat to emphasize the next question is being addressed and to provide a delineator in the lesson transcript, continuing the tactic implemented in the previous Iteration.

Reflective Notes:

- RN1: There appeared to be a high degree of consistency in the type of collaboration that transpires in Iteration 1 and Iteration 2.
- RN2: Student technological questions and teacher technological errors characterize this lesson in a similar manner to the first lesson of Iteration 1.

Iteration 2 Topic 2

Topic: Fundamentals of Objects and Classes as they relate to object oriented programming.

Attendees: 9

Summary of Lesson

OB1: This was the first lesson in which group-work approaches were systematically designed and attempted. For tutorial question 1 students were divided into two rooms and asked to construct a group answer identifying the classes, objects, instance fields, methods and local variables in a program. The interface had been redesigned to provide the program in the middle column of the window, a communal solution space in the top right note-pod and a text-chat pod at the bottom right of the interface. A shared solution space was provided in order to allow activity to centre around students rather than the teacher (see Figure 78).

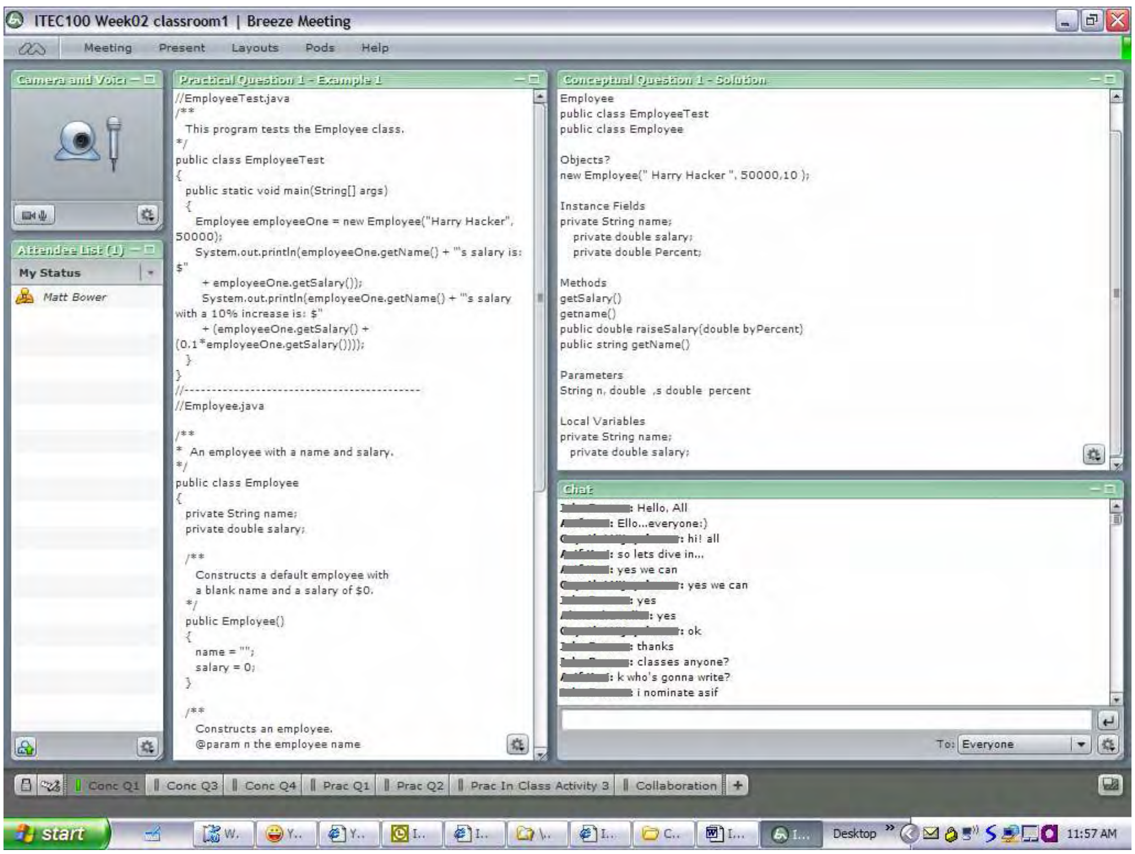

Figure 78 – Iteration 2 Topic 2 Purpose built interface to facilitate student-centred sharing of declarative knowledge

Students in both groups were able to complete this exercise in their groups, though the time required to do so was considerably longer than using the question-response approach of Iteration 1. The most capable student in each group (JB in group 1 and XS in group 2) spontaneously adopted the role of coordinator, assisting the flow of the discourse by questioning of peers and provision of explanations where required. The teacher was able to review the group-work room text-chat transcripts to identify any remaining misconception and then address them once students had returned to the main room. Students spent 13.5 minutes in their groups completing this exercise.

OB2: For the four "true or false" responses required in preliminary question 3, the default 'Discussion' layout was used including a chat-pod, a blank note-pod (for a solution space), voting tool and camera and voice pod (see Figure 79). Students were asked to agree on a group solution to the task. Students had been given only minimal instructions about how to use this particular "multimodal cluster" (Baldry & Thibault, 2006) to collaborate.

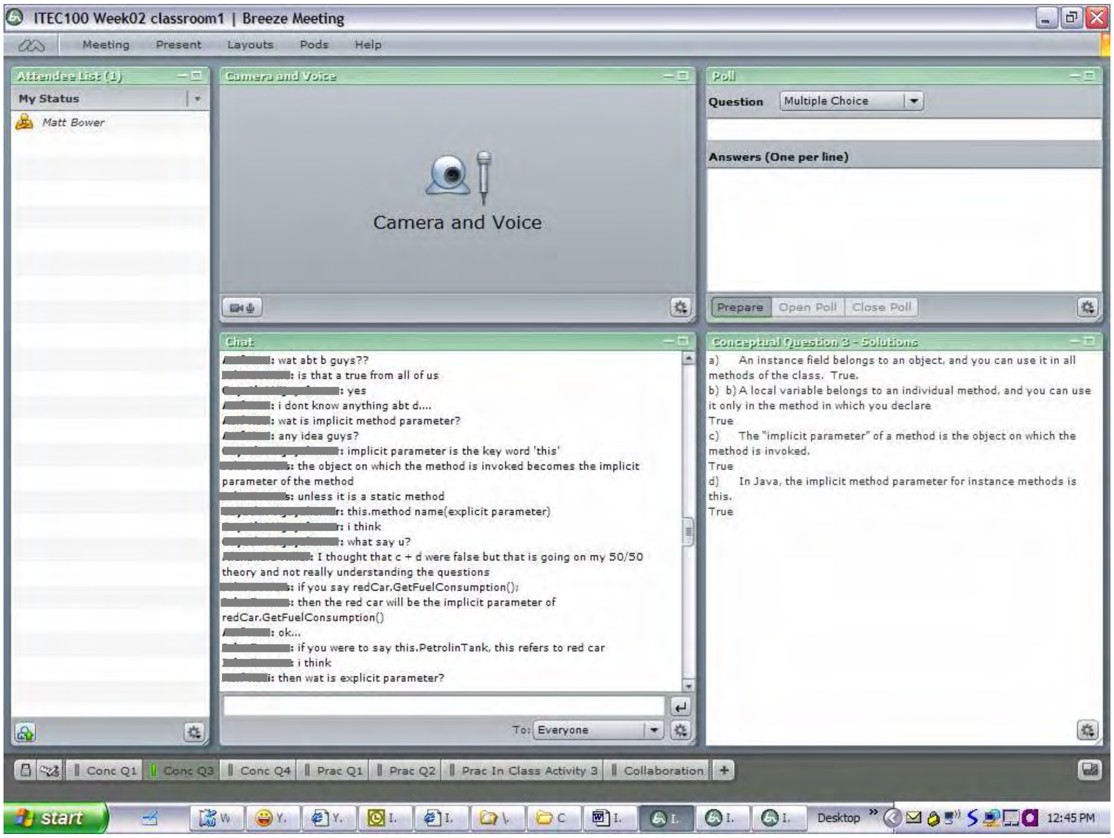

Figure 79 – Iteration 2 Topic 2 Default 'Discussion' layout used for student-centred collaboration

Groups one and two approached the task in very different ways. In Group 1 students discussed the task using the text-chat pod, placing emphasis on negotiating meanings and arriving at consensus. They did not use the voting tool. The questions were pasted to the note-pod solution space and the group answer was placed below each. The most capable student remedied misconceptions and poorly formed concepts of weaker students. In Group 2 the voting tool was used to determine the group answer. One student made the reflection "but majority will win". Solutions were not posted to the note-pod, but rather intermingled in the text-chat discourse.

OB3: The teacher adopted a standard transmissive approach to covering preliminary tutorial questions 2, 4 and 5, in order to make up for time lost using the group-work approaches on the other questions. This involved broadcasting the solution document and using audio to provide ancillary explanations.

OB4: In Iteration 2 a different approach was adopted for practical tasks one and two as compared to Iteration 1. Two (anonymized) student solutions were provided in each group-work room, and students were asked to identify which program they felt was superior and why. For practical task 1 the two programs were provided side-by side using note-pods in order to promote easier comparison and contrast and reducing the split attention (Ayres & Sweller, 2005) that occurred in the previous iteration (see Figure 80). The size of each program to be read meant that they could not entirely fit in the visible portion of the window and hence scrollbar use was required (which did not appear to compromise the activity).

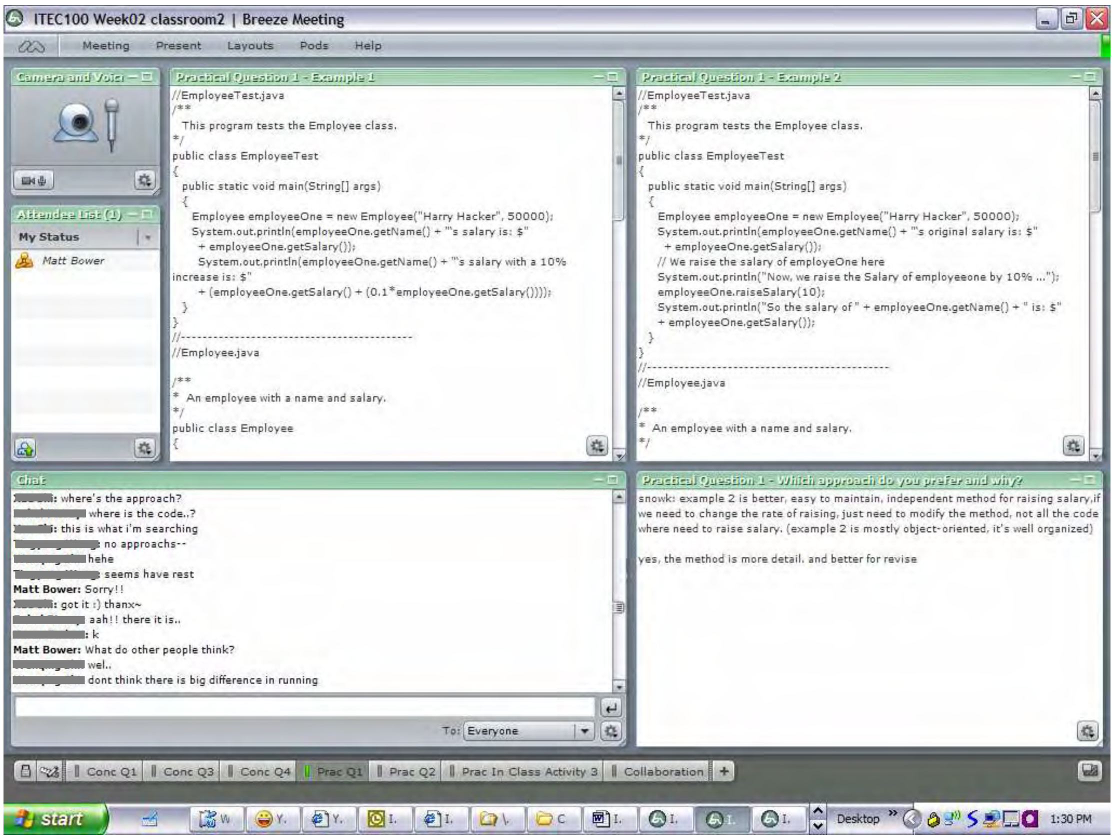

Figure 80 – Iteration 2 Topic 2 Using a purpose built layout with note-pods for an evaluative task

Students in each room spent over two minutes in silence as they read each approach. One student in each room (the most capable student) then provided their opinion, which was also the final solution presented by each group. There was some discussion, but not extensive debate as the most confident student in each room was able to justify their (correct) rationale.

OB5: For practical task 2 the two programs students were to compare were provided in a zip file for them to download and inspect on their computer. A pod allowing them to choose a whiteboard or screen-share was provided in the main area at the top of the layout (see Figure 81). Again students were not provided with directions on the ways that they should use the web-conferencing tools to collaborate during this exercise.

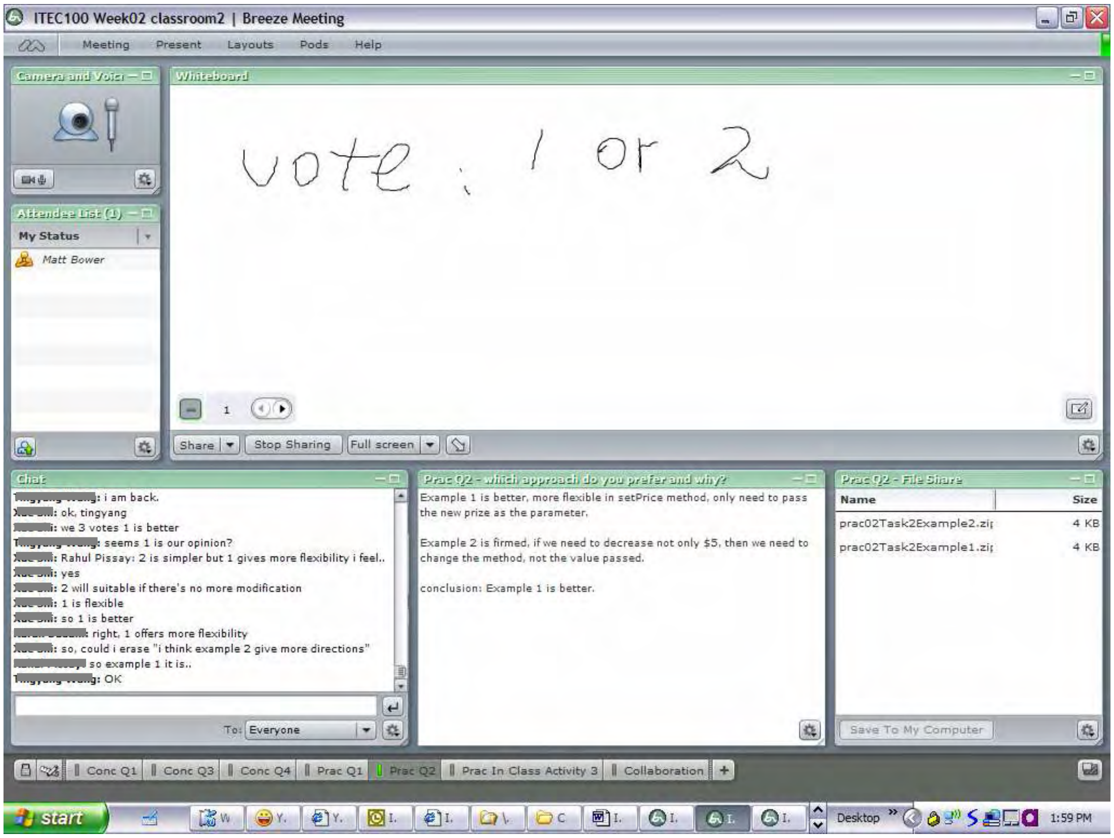

Figure 81 – Iteration 2 Topic 2 Using a purpose built layout with a whiteboard and file-share pods for an evaluative task

Each room recorded some discussion, with the solution once again proposed by the most capable member of the group (which subsequently became the group solution). For instance, in Group-work room 2:

> Example 1 is better, more flexible in setPrice method, only need to pass the new prize as the parameter.
> Example 2 is firmed, if we need to decrease not only $5, then we need to change the method, not the value passed.
> conclusion: Example 1 is better.

The large sharing space for the whiteboard or screen-share was hardly used (and used ineffectively) by the two groups. The fact that students downloaded the two separate files on their own machine meant there was no common artefact upon which they could work, resulting in split attention (Ayres & Sweller, 2005).

OB6: At the conclusion to the lesson students were asked whether they preferred the group-work approaches or the direct instruction approaches. Their feedback included the following comments:

> JB: im not sure if some people might have been lost in the shadows
> WS: prefer group disc
> XS: it's great in groups, that makes us more interactive
> JB: i thought it was ok though
> RP: group work is anyday better...
> GV: i find it a bit difficult to discuss the practicals in this breeze group work

TW: group is better, since more ideas
KD: Just the right mix of two
JB: coming back and discussing what we done tied it up nicely though
GV: straight discussion makes me feel better
RV: but its better physically... not virtually... tho..

Students expressed general support for group-work approaches, but identified that completing group-work through the web-conferencing environment proposed some difficulties.

Key Incidents:

- KI1: Moments after commencing this lesson the internal network crashed, a problem that was not rectified for 31 minutes. Remarkably all students were still online and the lesson continued (though with less time in total to complete).
- KI2: The program code in the group-work rooms for preliminary conceptual task 1 was different to that on the question sheet because during development of the room the teacher had not realized the note-pods were linked between layouts. This meant that when a note-pod on a separate layout was furnished with a different program it automatically changed the program on the previous layout. This was disconcerting for students, however they were still able to complete the activity using the second program.
- KI3: For practical task 1 in group-work room 2 the teacher had not switched to the appropriate layouts meaning the students had a delayed start on this task.
- KI4: For the review of program code tasks in practical tasks 1 and 2 the most capable student in each group presented their solution first (assumedly because they could comprehend the programs faster than their peers and were more confident in providing answers). The task only required conceptual level (evaluative) consideration without any need for them to engage in an authentic process together. The amount of collaboration is correspondingly small.
- KI5: The interface design for practical task 2 resulted in significant split attention (Ayres & Sweller, 2005), with students having to work between the IDE on their machine and the group-work browser window. As well, a time overhead was incurred for students to download, unzip and open the project files. Students had not been taught how to screen-share (which was a possibility afforded by the interface) and did not use this feature. In group 2 a whiteboard was used to request the group "vote 1 or 2", which was both an inefficient approach to using the technological affordances available (for instance, text-chat would have provided a faster communicative mode) and a sub-optimal approach to collaboration (focus on deriving answers rather than discussing concepts). As in previous examples, task prescription aimed at discussing approaches to using the technology to interact may have improved the amount and quality of collaboration that resulted.
- KI6: Conducting group-work discourse using text-chat allowed the teacher to monitor collaborations in each room and subsequently ascertain whether any remediation or commentary was required back in the main room to conclude a learning activity.
- KI7: Some students indicated that switching rooms several times to complete the group-work activities was disconcerting. This may have been more easily accepted later in semester once students were more familiar with the technology and group-work approaches.

- KI8: Students expressed qualified approval for group-work approaches, citing interactivity and exposure to more ideas as advantages.
- KI9: Only one person can type in a note-pod at any one time. To overcome this, students were encouraged to type contributions in a separate text area and copy-paste into the solution space.

Reflective Notes:

- RN1: Designing and developing the group-work room layouts for this lesson takes a considerable amount of time (3.5 hours for this week).
- RN2: The task design for preliminary question 3 could have been improved by refining the interface, explaining to students how they should use the interface that had been provided and explaining their collaborative objectives regarding the task. In small group tasks the voting tool can distract students from forming negotiated meanings (contributing to constructed understanding rather than 'majority rules' approaches to knowledge building) and in terms of collaborative learning the tool is a more appropriate strategy for harvesting perceptions of large groups. Use of webcams does not contribute to discourse and should therefore not occupy such space on the interface. Providing the question in the note-pod solution space would have reduced split attention (students having to refer to separate printed or digital notes).
- RN3: On the one hand the group-work activities allowed students to be more involved, take initiative, and perform discussions uninterrupted by the teacher. On the other hand the group-work required more time than teacher-centred or teacher-led approaches. An appropriate mix of each approach is required, and thus the decision about which activities should adopt group-work approaches is important. For instance, for declarative knowledge tasks, is may be more efficient and effective to use teacher-led question-response approaches rather than student coordinated group-work. Perhaps one or two group-work tasks per week involving negotiation and extension would be most appropriate.
- RN4: Audio may be a considerable advantage for conducting group-work in terms of facilitating rapid discussion and debate between students.
- RN5: Adding group-work markedly increased the amount of in-class technology management work required of the teacher (troubleshooting, switching layouts, monitoring progress between two rooms and so on).
- RN6: There were several lessons learned from this first attempt at incorporating group-work into the class, including the following:
    - Interfaces should only include the tools that were anticipated for use (Sweller, 2005b)
    - Designs should aim to reduce split attention (Ayres & Sweller, 2005)
    - Instruction should be provided regarding how students should collaborate using the technology
    - Tasks with which to apply group-work approaches should be carefully selected.

Iteration 2 Topic 3

Topic: Introduction to types and numbers.

Attendees: 9

Summary

OB1: After introductory comments the first three preliminary tutorial questions were covered using a standard question-response approach (teacher broadcasting solutions using a standard sharing interface and asking questions using audio while the students responded using text-chat). This allowed the approach to collaboration that was to occur for the next seven questions to be modelled by the teacher, thus reducing the amount of explanation that was required to introduce the next student-centred activity.

OB2: Students were then separated into two group-work rooms and asked to identify concepts from the preliminary tutorial questions 4 to 10 which their entire group did not understand. The 'Collaboration' layout used for the activity provided students with a sharing space (for a whiteboard or to share their screen), a communal note-pod, a text-chat area and a file-share pod (see Figure 82).

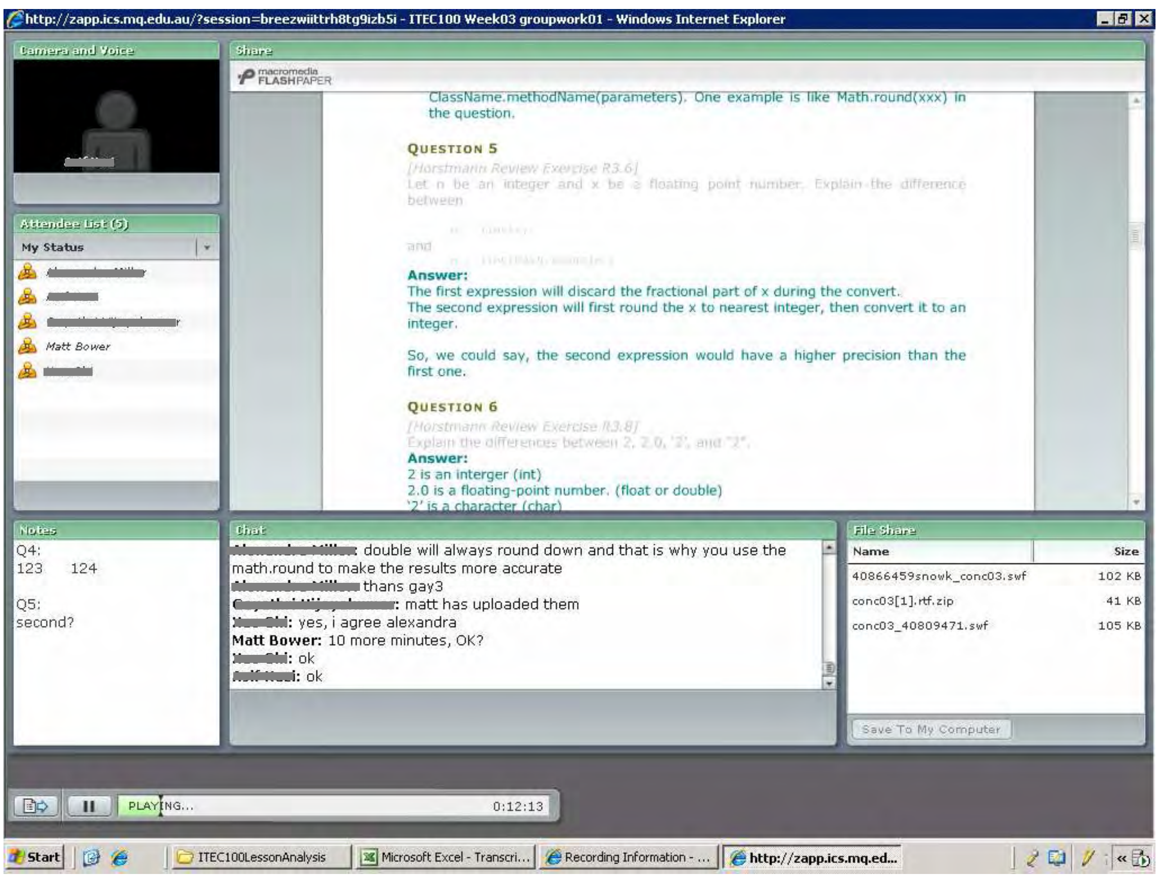

Figure 82 – Iteration 2 Topic 3 Student-centred collaborative task using a 'Collaboration' layout

The teacher also asked if any students from each group would be willing to share their solutions as a starting point. One student (the most able) from each room volunteered to do so. Students proceeded to discuss each question in turn in their group-work rooms, using an approach similar to that adopted in the first three questions for this lesson. At the beginning of the activity students in group one misinterpreted the task and proceeded to create collaborative solutions in the note-pod rather than identify concepts they did not understand. The most able student in each room automatically led the discussion. This activity resulted in 214 separate student text-chat contributions in group 1 (4 students) and 134 in group 2 (5 students) over a period of 28 minutes.

OB3: The teacher used a transmissive approach to cover preliminary tutorial questions 11 to 13 in order to save time and also so that pre-prepared diagrams could be utilized. Again this coincided with little student contribution.

OB4: For the practical work the teacher presented two student solutions and explained the various parts. The interface had been designed specifically to display the two classes that comprised the program in two separate note-pods along the right hand side of the window (see Figure 83). This allowed students to instantly relate the code between the files whereas performing a screen-share of the IDE resulted in split attention (Ayres & Sweller, 2005) between the two files. The teacher identified the second solution (requiring students to make a sphere class) as an excellent template for writing other object oriented programs.

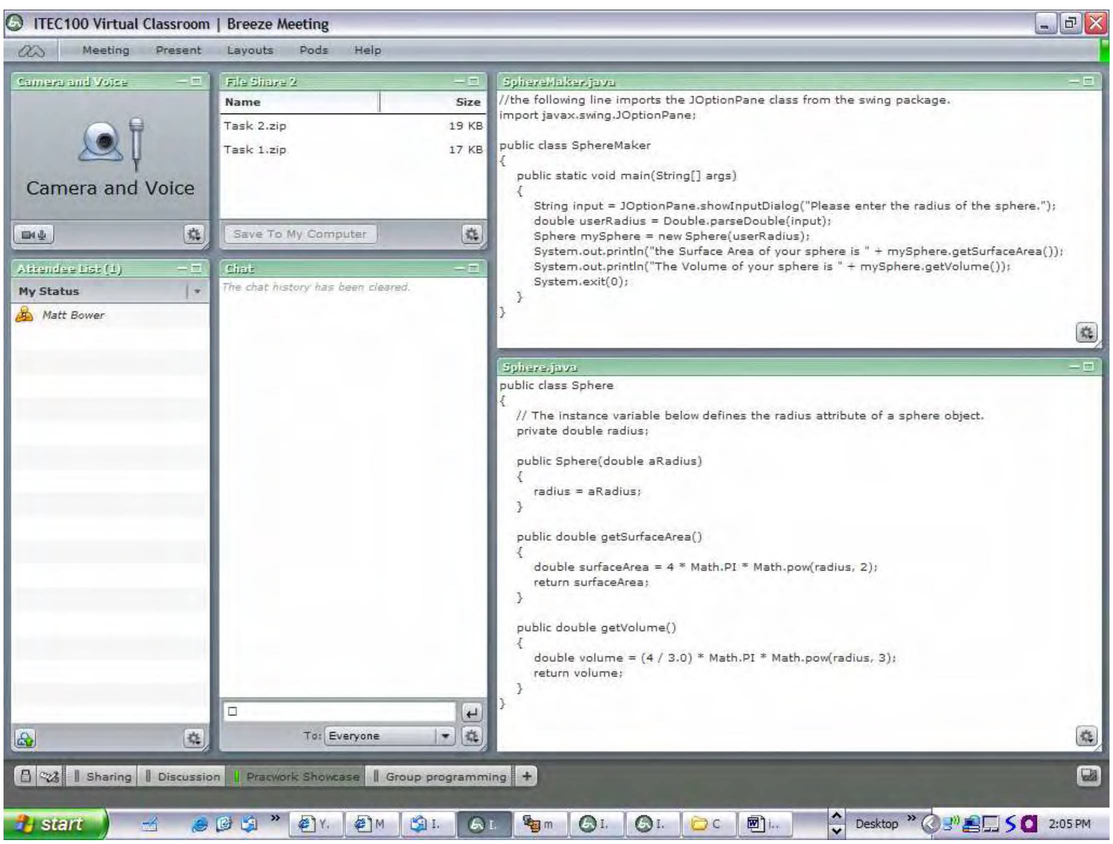

Figure 83 – Iteration 2 Topic 3 Presenting students with code for practical tasks

The template was then used to complete a programming task requiring the class to "write a program that prompts for two integers and then prints out the sum and the average". The teacher then switched to using screen-sharing to take instructions from the class about how to solve the problem. Students were comfortable with the collaborative pattern of the students "driving" the teacher and so very little explanation about how to collaborate was required on the part of the teacher. This then served as a model for the group-work programming task.

OB5: Students were then divided back into their group-work rooms (see Figure 84) and asked to complete the following task (with instructions provided as part of the interface design):

> *Write a class TinCan that creates cylindrical TinCan objects and has a method to return the volume. Write a class TinCanTest to test your class.*

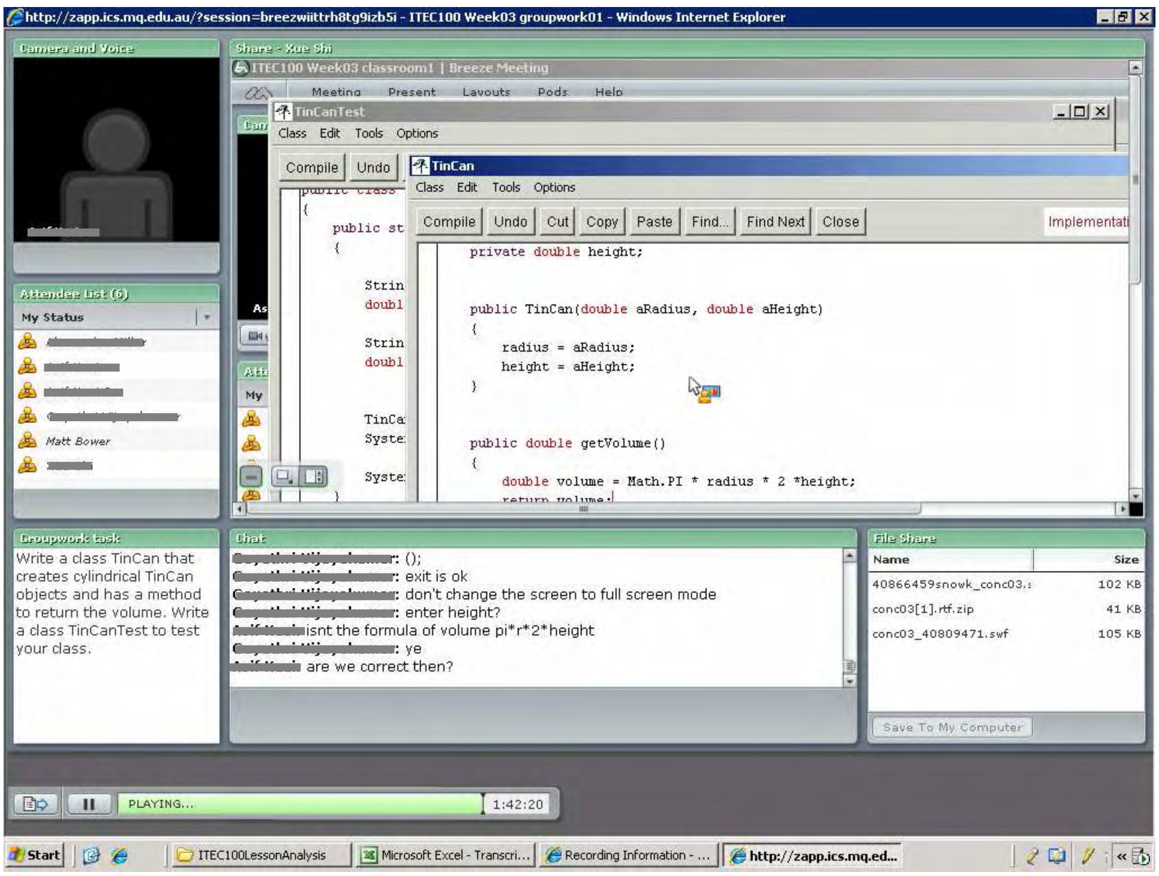

Figure 84 – Iteration 2 Topic 3 Students using screen-sharing to perform group programming

The two most capable students in the class were the facilitators to be guided by the other team members. However those two students had not been given prior instruction on how to facilitate such an activity, and as a consequence there were some skills that they lacked. For instance, they did not know that if they minimized the browser window popups containing text-chat comments would appear. As well, the teacher had not considered that without audio they would not be able to respond to their peers using text-chat without switching focus from the IDE that they were broadcasting. In group 1 this resulted in a disjointed broadcast with the student facilitator often maximizing and then minimizing the group-work room window to make a comment. In group two the facilitator adopted a work-around by typing text-chat comments in the code window of the IDE. Either way, if the teacher would have anticipated this then he could have insisted that students who were facilitators used audio. However, both groups were able to complete the task. The pace of collaboration was rapid and students appeared highly engaged.

OB6: Despite sub-optimal features of the collaborative designs, student feedback regarding the lesson was positive. The following sequence of discourse is in response to being asked about their impressions of the class:

> AK: team work makes programming fast!
> AM: team work is good. helps witht he learning
> JB: QA on the fly
> AM: good
> AK: cool
> MB: Team programming? Good?
> GV: yes it was good
> KD: helpful
> RP: team work makes learning easier...
> JB: good
> XS: robot wants to talk
> AK: :D
> WS: not bad.. but sometime one person can dominate the whole thing

Students appreciated that group-work supported learning programming process, but XS identified that the person performing the screen-share needed to be able to talk in order for collaborations to be more efficient.

Key Incidents:

- KI1: After only the second week of group-work students were already comfortable switching between rooms (and the teacher had developed skills in managing this online synchronous group-work, such as approaches to directing students to their rooms and calling them back to the main room). Note that having only one transition and layout for the group-work required less student reorientation than the multiple changes that occurred in the previous week.
- KI2: The approach by which students were to construct group solutions to question 4 through to 10 was not clearly preconceived or specified by the teacher, and as such the approach to collaboration was sub-optimal. For instance the solution space shown in group-work room 1 needed to be larger to match the importance of that note-pod in aiding the collective cognitive approaches of the group (Hollan, Hutchins, & Kirsh, 2000). As well, students struggled to collaborate about the questions relating to mathematical formulae because of the inappropriate match of the text-chat modality to fully express the mathematical "Symbol System" (Salomon, 1994).
- KI3: For the group-work tutorial questions it was intended that one room would use audio (based upon students who had setup their voice facilities prior to the lesson) other room text-chat. However the students in the audio room struggled to establish clear audio communication between all group members and eventually chose to revert to text-chat.

- KI4: In the group-work activities students with less complete mental models were more likely to ask questions of their peers than during whole-class teacher led activities.
- KI5: For the group-work tutorial task the activity prescription was not included in the interface, which resulted in group one forming collaborative answers as opposed to completing the intended task of finding questions that the group did not understand.
- KI6: The capacity to broadcast audio to one or many groups simultaneously (by switching the level at which audio was being broadcast – class, or individual group rooms) allowed the teacher to more accurately direct commentary.
- KI7: Preparation in terms of having files that students would require preloaded into the file-share was far more time efficient than uploading during the lesson.
- KI8: The approach of modelling the group programming collaborative pattern and then requesting students perform a similar task in groups allowed them to immediately commence on their group task. However further instructions were required to explain the micro-skills of performing screen-share and how to compensate for not having audio.
- KI9: Using screen-sharing for the group programming interface allowed students to volunteer short snippets of code through the text-chat, however at times they experienced difficulty explaining where those snippets should be placed. Also, typing code in the text-chat took time, by which stage the person performing the programming may have moved location. A note-pod would allow students to contribute code directly into the place it was required.

Reflective Notes:

- RN1: The importance of insisting upon student input and feedback at least every two or three minutes (especially when using transmissive approaches) is again noted. Text-chat affords the advantage of the teacher being able to instantly ascertain everyone's understanding simultaneously, if they are providing comments regarding the curriculum matter.
- RN2: It would be interesting to compare and contrast the screen-share approach to group programming adopted in this lesson to using a note-pod. The screen-share had the advantage of displaying all the programming processes that are required in terms of operating the IDE, which is particularly important to reinforce when students are first learning to program. However, once students are confident with these procedural aspects of programming, using a note-pod may afford easier contribution.
- RN3: If a task has been placed on a separate layout it is advantageous to have the task written on a notepad somewhere – especially in absence of teacher instruction. This supports students in commencing confidently and immediately on the task.
- RN4: Once again, by being able to see students' group-work chats the teacher is able to gain an appreciation of any student misconceptions and instantly adjust whole class instruction to respond to these weaknesses. This is a recurring theme.
- RN5: In order for students to more effectively collaborate in groups students need to develop more web-conferencing competencies. Students need to be taught how to create new share pods, how to submit their solutions in the proprietary FlashPaper format which the web-conferencing environment uses, and how to contrast multiple FlashPaper solutions using share pods. They need to be reminded to use the whiteboard to represent non-textual information (e.g. mathematical formulae or diagrams), and encouraged to spontaneously adjust the web-conferencing environment tools and layout to meet their collaborative requirements. Teacher

modelling of patterns of collaboration in earlier exercises appears to support the development of students' collaborative competencies.

- RN6: Only 2.5 hours was required to design and create the group-work activities as opposed to 4.5 the previous week. This decrease in time required was due to an improved familiarity with how to create the interfaces, as well as a clearer understanding of how to design the interfaces.

Iteration 2 Topic 4

Topic: Introduction to Applets and Graphics

Attendees: 8

Summary of Lesson

OB1: The first activity required students to create integrated solutions to the following three preliminary questions in pre-designed group-work rooms:

1. *Explain the difference between an Application and an Applet.*
2. *What are the advantages of applications over applets? What are the advantages of applets over applications?*
3. *What security features do applets have?*

The note-pods were seen as an efficient means of facilitating group response formation, and attempting three questions together in one activity was considered appropriate both in terms of the amount of learning and the amount of content that could reasonably occupy the interface. The interface design is shown in Figure 85. Groups were allocated 10 minutes to complete this task.

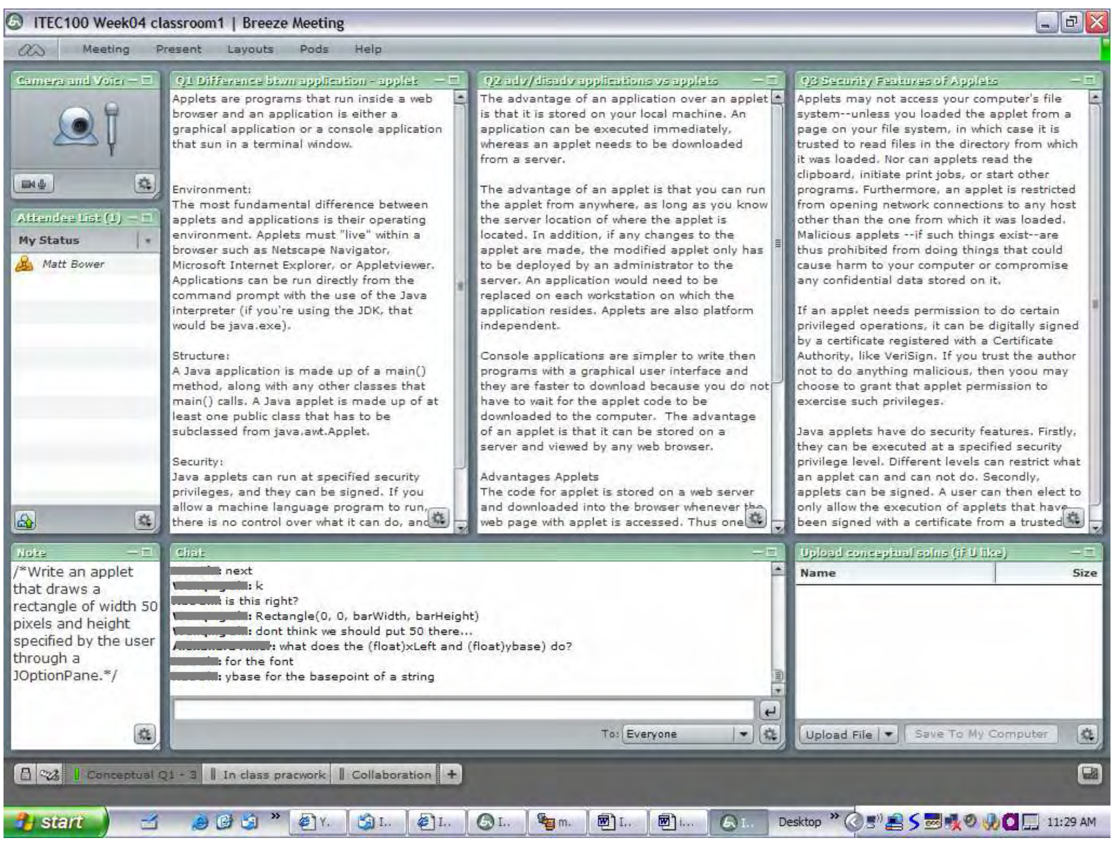

Figure 85 – Iteration 2 Topic 4 Purpose built interface for collaborative authoring of tutorial responses

One group room was designated as the audio room, and the other the text-chat room (although the identical interface was provided in each room). Students who had their audio configured and ready for use were allocated to the audio room and the other students were allocated to the text-chat room. The first part of the audio group discourse related to establishing reliable group collaborations. However once audio was established students in this room were able to contribute approximately double the amount of discourse than students in the text-chat only room. Students were able to construct reasonable solutions to these comprehension based tutorial questions.

OB2: Preliminary tutorial questions 4 to 6 were covered using a teacher-led question-response approach. Two network outages at the campus level led to everyone being evicted from the virtual classroom twice in a row for a period of approximately 15 minutes each. This resulted in the questions being covered by the teacher using text-chat rather than audio (to avoid network load) in combination with revealing the document solutions.

OB3: Approaches to combining students' applet pictures for the first practical task on the one canvas was demonstrated by the teacher. After this students were provided with ten minutes to complete the same approach themselves at home using other pictures from the zip file they downloaded. The teacher then presents a solution to the task, showcasing and discussing different student solutions. The solution to practical task 2 (that required students to write an applet that prompts for their name and places it in an ellipse) was also made available for student download and is discussed by the teacher. Students asked questions, for instance regarding the getWidth and getHeight methods inherited from the Applet class, and the teacher explained these concepts. These approaches to reviewing the practical work are similar to those adopted in Iteration 1, and result in similar levels and types of discourse.

OB4: The group-work programming took place using the note-pod, offering a means of comparison and contrast to the previous week's screen-share approach (see Figure 86). Students are provided with the following instructions in the note-pod:

> *"write an applet that draws a rectangle of width 50 and height specified by the user through a JOptionPane."*

In the audio room (group 1) only the most capable student uses the voice collaboration facility, with other students opting to use text-chat. This student is not supported by his peers and ends up completing the exercise by primarily writing the program himself. Students in the text-chat room (group 2) adopt a more collaborative approach, but do not make as much progress towards the solution (not finishing by the time the activity draws to a close). Due to time restrictions the teacher recommends that students complete the task at home.

OB5: Students are asked: "What did you think of using the notepad to group code instead of screen-sharing?" using the polling tool (with responses much better, better, the same, not as good, much worse). All responses except one indicated that using the note-pod was at least as good as the screen-share, if not better, however given there were only five students left in the class by the end of the lesson, this was not strong evidence in support of using a note-pod. The note-pod was observed to allow communal access to a shared authoring space, whereas the screen-sharing allowed procedural knowledge relating to the use of the IDE to be shared (as well as debugging to be performed without copy-pasting being required).

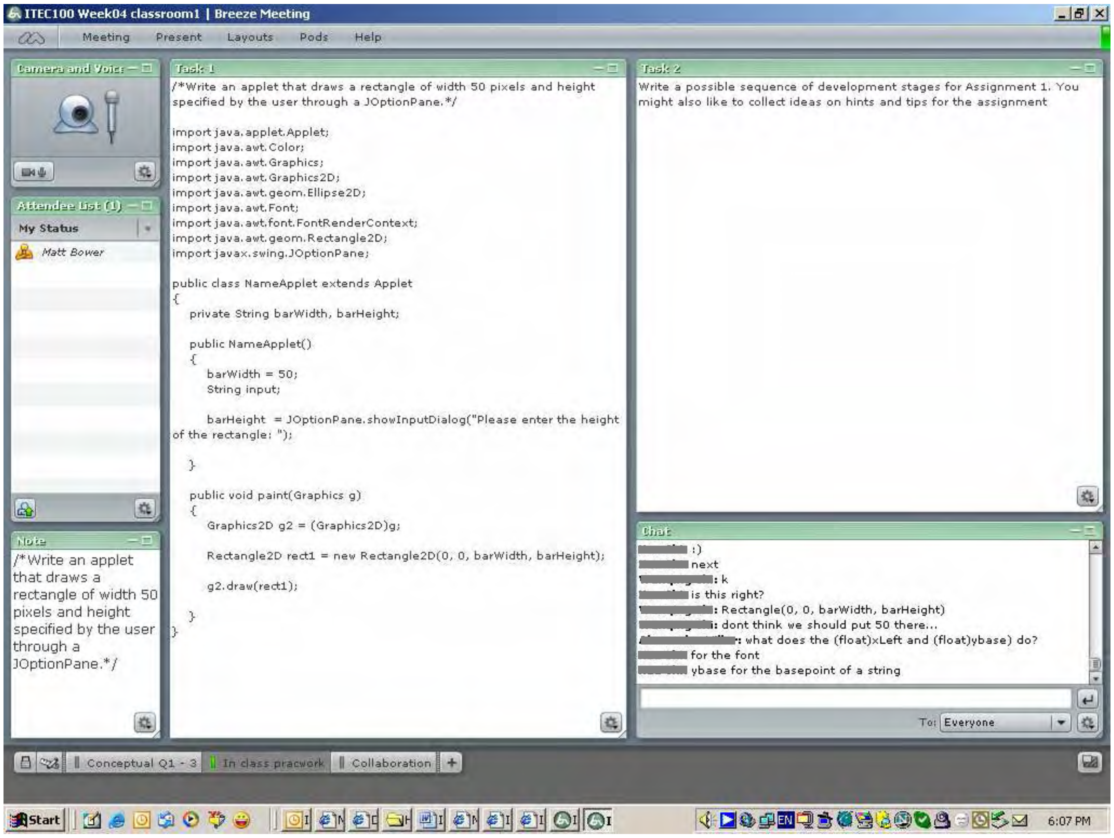

Figure 86 – Iteration 2 Topic 4 Interface for group programming using note-pod

Key Incidents:

- KI1: For the task requiring students to construct collaborative answers to preliminary tutorial questions 1 to 3 far greater interaction occurred in the audio room as opposed to the text-chat room. As well, audio comments could be contributed at the same time as textual interactions in the note-pods without causing interference (Modality Principle, Low & Sweller, 2005). In contrast, students in the group using text-chat to converse had their attention split between the note-pod solution space and the text-chat area. The two modes were not suited in combination for this group-work task.
- KI2: Although there was no clear student preference for using the note-pod over the screen-share for group programming (1 better, 3 same, 1 not as good), the capacity for all students to have equal access to the note-pod represents a significant advantage.
- KI3: In the group programming activity less confident students in the audio room choose to use text-chat to communicate, perhaps as a way to withdraw from collaborations. The availability of audio does not ensure more rapid collaboration will occur, but rather allows the possibility.

Reflective Notes:

- RN1: Asking students to combine their answers for preliminary tutorial questions one to three did not require any substantial reprocessing of the information. A ranking task or some sort of reformulation task would have required them to perform more critical evaluation. Clear task prescriptions requiring students to succinctly integrate points as opposed to everyone merely copying and pasting their answers to the same space may improve the quality of solutions constructed.

- RN2: The use of audio in one of the group-work rooms corresponded with different types of discourse. For instance, one student (AM) contributed far more social and supportive comments.
- RN3: Audio appears to be an excellent modality for rapid and 'tightly-coupled' (Neale, Carroll, & Rosson, 2004) exchange of information between users, for instance when negotiation is required. Text-chat appears useful for broadcasting ideas in a manner that doesn't interfere with others, such as with medium size classes.

Iteration 2 Topic 5

Topic: Introduction to conditional statements ('if' statements)

Attendees: 8

Summary

OB1: In order to provide students with more out-of-class interaction with their peers, students were asked to attempt the preliminary questions in groups before this week's lesson. The nine students were split into three groups, each of which was allocated its own virtual classroom that could be accessed at any time. The first twenty minutes of lesson time was spent by having students enter each other's rooms and provide feedback on the solutions posted by the other groups. Initially one student asked "how do we comment on the answers?" This was not specified by the teacher. However, once they had been told to simply use the text-chat in the individual rooms there were no further questions about how to use the technology to collaborate.

OB2: Providing feedback to the groups' preliminary conceptual solutions resulted in 65 student text-chat comments including 15 suggestions/corrections, seven positive feedback statements and four questions. As well, the activity provided the opportunity for social comments and gratitudes such as "I didn't know much about q5, thanks for that answer". The activity design allowed the level of student understanding to be effectively exposed, both through the solutions and through the feedback comments. Almost all of the discourse related to revealing and remedying weaknesses in students' mental models.

OB3: Students were asked to provide feedback regarding the pre-class group-work approach to attempting the tutorial questions. This was harvested using the voting tool and the text-chat pod. Five of the students thought the approach was better than working alone, with the two most capable students indicating that it was "not as good" as working alone (there was also one non-response). Students felt that they could "learn from each other", "discuss problems" and that they "got more ideas which [they] never thought of". However students also pointed out that attempting the preliminary exercises in groups before class took more time (up to four hours in this first trial) and required more coordination to find a common time for them to confluence. Students indicated that a significant amount of time was required to experiment with and learn the functionality of the web-conferencing environment.

OB4: In order to clarify student understanding of "if-then-else" sequences of tutorial question 3 students were asked to plot the output of the conditional statements on number lines. This interface was designed to facilitate exposure and sharing of students' mental models regarding how the four if-then-else sequences operated. Providing an exemplar representation (in accordance with Or-Bach & Lavy, 2004) allowed them to immediately provide a corresponding representation for their own sequence of code. Their representations immediately exposed their conceptions regarding the operation of if-then-else statements. Using the collaborative space allowed students to review each other's representations and provide critique and suggestions. Requiring students to interpret the code within each if-then-else sequence and represent it in an unfamiliar way was also a learning exercise in itself (see Figure 87).

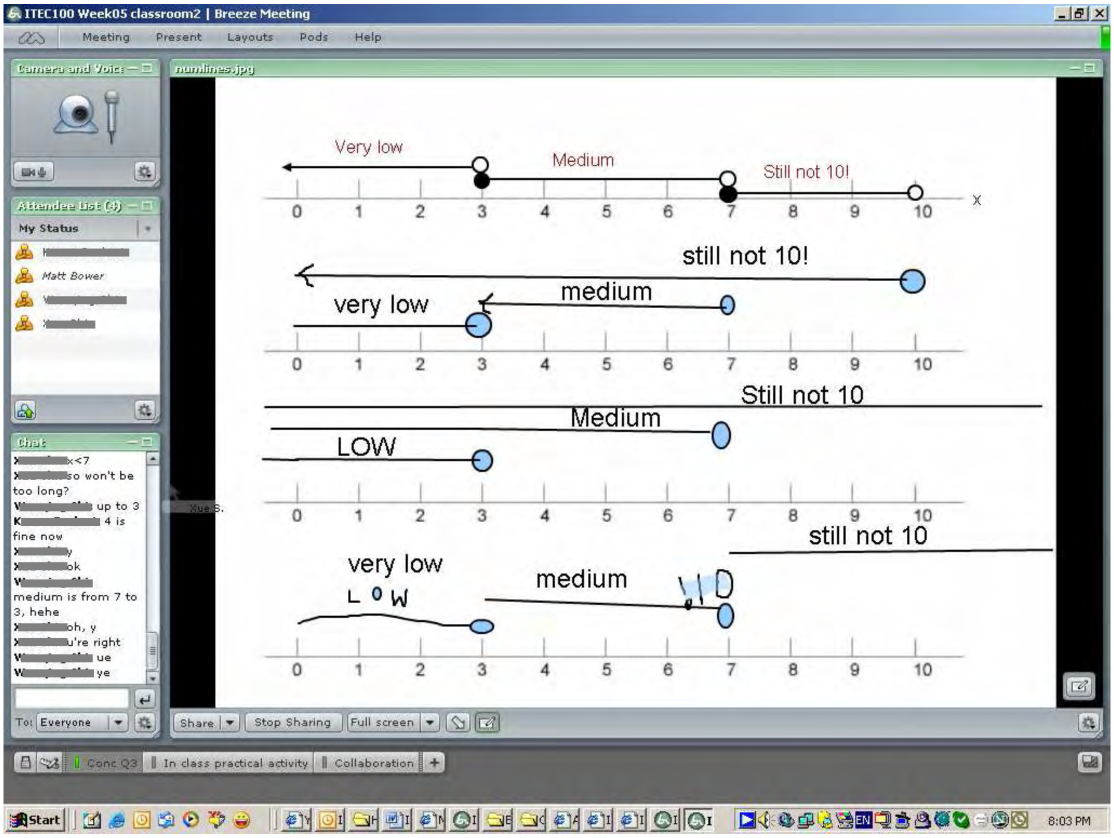

Figure 87 – Iteration 2 Topic 5 Using a whiteboard to facilitate student representation of conceptual information

OB5: When asked by the teacher "what did you think of doing Q3 on the number line" students responses indicated that it "makes it more clear", was "really interesting and helpful", and "fun". Students indicated that they "needed time to figure out the tools" and that it would be quicker once they were able to "get used to them". The teacher saw this activity as a way for students to become familiar with the whiteboard tools, so that their competencies would be improved for future collaborative activities involving the whiteboard.

OB6: Preliminary practical exercises for this week included a tax calculator, and two applets to test whether a point lied within a shape. A primarily instructive approach to covering these activities was adopted. The teacher presented students' solutions using note-pods (see Figure 88) which allowed students to make amendments if required (although this capacity was not utilized). As well, students were provided with a single zip file containing all practical files being presented so that they could download and run the programs for themselves. The teacher also used screen-sharing to demonstrate how the if-statements in one of the applets could be adjusted to be more efficiently coded. This teacher dominated approach resulted in a corresponding reduction in the amount of student collaborations, but commensurate with those observed when similar approaches were applied in Iteration 1.

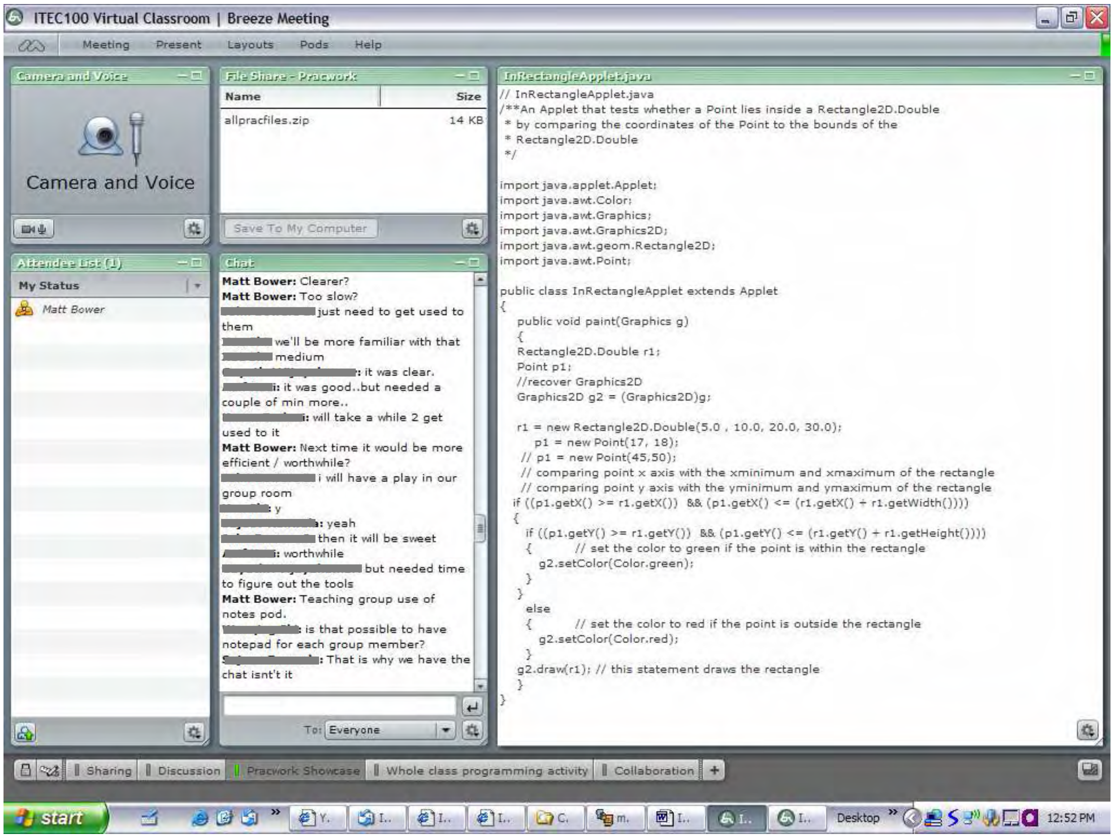

Figure 88 – Iteration 2 Topic 5 Teacher presented program using a notes-pod with file-share

OB7: The final episode for the lesson involved students completing the following programming task in groups:

> */*Last week you wrote the Applet below that draws a rectangle of width 50 pixels and height specified by the user through a JOptionPane.*/*
> */*This week you are to adjust it so that it draws two rectangles side by side but with bottom edge at 600. Each height should be specified by the user through separate JOptionPanes. (If you have time then also calculate the maximum height entered and prints it to the screen.)*/*

A note-pod was used in each group-work room rather than screen-sharing, as it allowed more equal access of group members to edit the code (see Figure 89). The teacher provided directions about how to collaborate, including specifying the strategy of appending initials when making an alteration to the code. From an Activity Theory (Engeström, 1987) perspective this was a teacher provided 'rule' for use by the community. Students followed this convention to some degree, which overcame the "distributed process loss" (Neale, Carroll, & Rosson, 2004) that could occur if the identities of contributors were unknown. Students were surprisingly slow at completing this task and the most capable student in each room dominated collaborations. By this stage of the lesson time was pressing, so once one group had achieved most of the specification the teacher called students back to the main room to observe the solution to the task. Students did not get a chance to complete task 2 in the right hand note-pod which required them to write a list of steps appropriate for developing the solution to assignment 1.

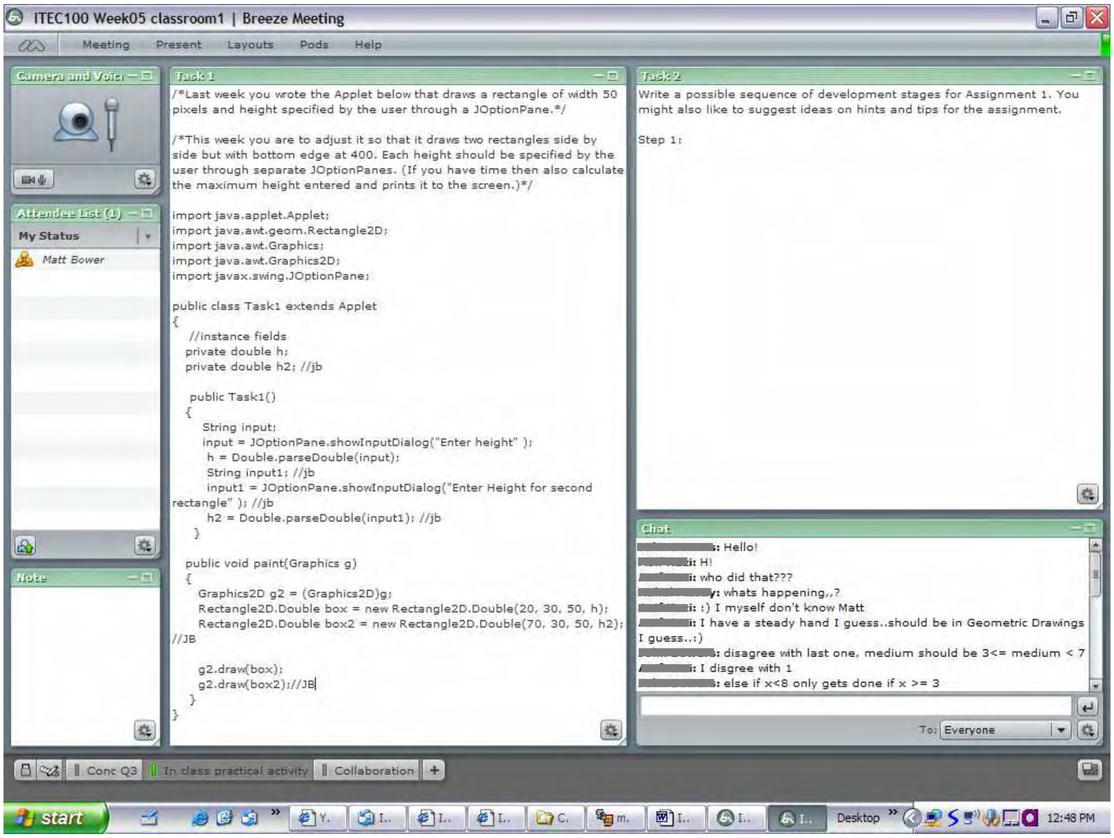

Figure 89 – Iteration 2 Topic 5 Student-centred group programming design

Key Incidents:

- KI1: The fact that students were providing feedback to the other groups' preliminary tutorial questions using text-chat meant that the teacher was able to intercept conceptual weaknesses from more than one group at once, whereas in face-to-face lessons only one dialogue can be monitored at a time.
- KI2: Not including the question in or along-side the solution space for the "if-statement number-line" activity caused split attention (Ayres & Sweller, 2005). As well, using text-chat to conduct discourse split student input between two tactile means (whereas audio would have avoided this split). Having to do so meant 'dual processing capabilities' (Low & Sweller, 2005) could not be used to monitor interactions (as both the text-chat and whiteboard input media required visual observation).
- KI3: Having a single zip file for the lesson containing all files that the students would need rather than separate zip files for each activity (which was the approach adopted in previous lessons) was a more efficient way to distribute resources.
- KI4: The fact that for the group programming task the entire program fitted in the visible section of the note-pod was useful in so far as allowing alterations to be more easily perceived. This meant that less effort was required for the teacher (and students) to track changes and provide feedback.
- KI5: In terms of technological competencies, a student still asked if the font size could be enlarged (not remembering to use the scroll button which had been described several times in previous lessons).

Iteration 2 Topic 6

Topic: Introduction to iteration ('loops')

Attendees: 6

Summary

OB1: As in the previous week, students attempted the preliminary tutorial questions in groups before class. The first part of the lesson required students to provide feedback about the other group's solutions in the text-chat pods for those rooms. This led to 16 feedback comments in room 1 and 63 feedback comments in room two (where more mistakes had been made). The discourse in each room was focused upon students remedying the mental models of others.

OB2: When asked by the teacher, students had no questions about the preliminary conceptual exercises. The teacher chose to cover question 2 regarding the erroneous 'factorial' problem using the debugger (to model the debugger's use). An instructive style was used (see Figure 90).

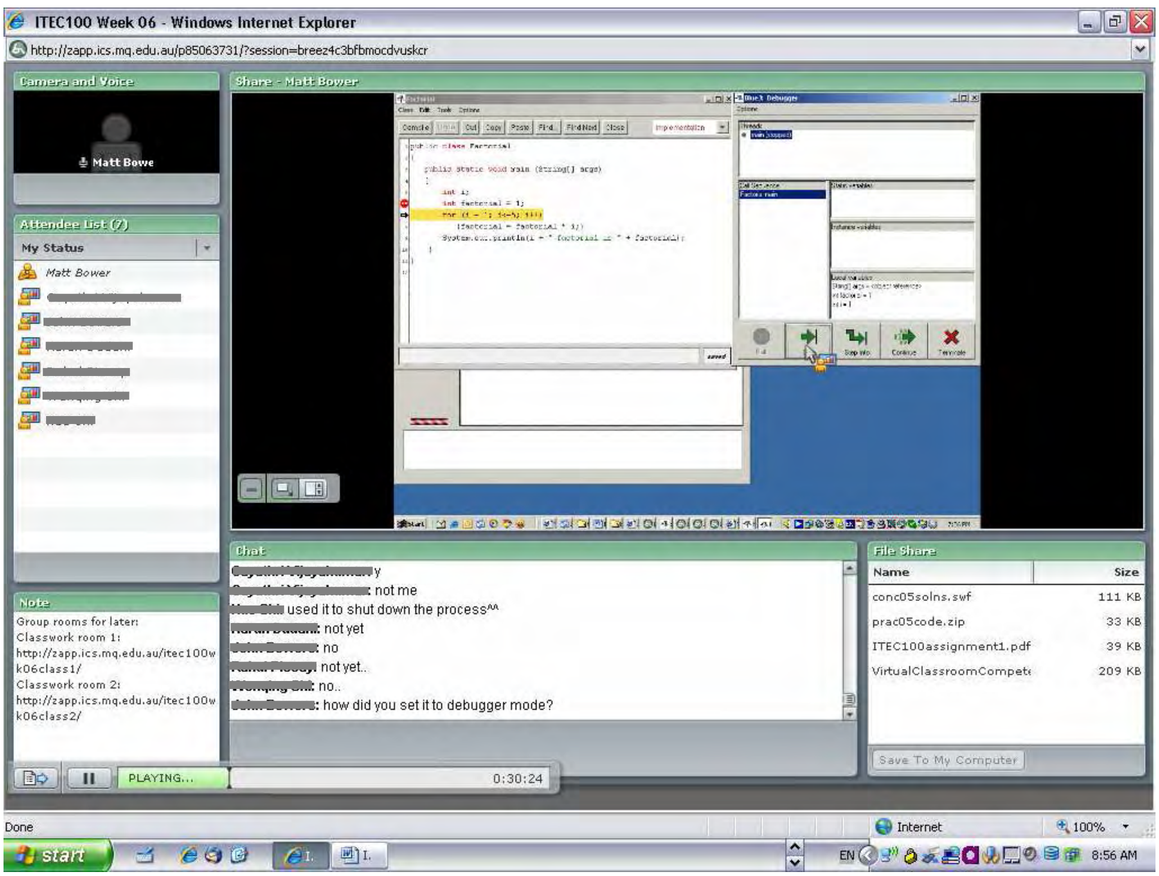

Figure 90 – Iteration 2 Topic 6 Transmissive approach to covering a programming problem

The teacher also covered question 6 by asking students to paste their shadowing examples to the whiteboard (providing them with more experience using the whiteboard and allowing them to compare and contrast their answers). When asked, students agreed that this was a useful approach to collecting and comparing their answers and that they could use it while completing the preliminary tutorial questions in their groups.

OB3: The next section of the lesson used the polling tool and a note-pod to determine students' impressions of completing the preliminary tutorial questions in groups (see Figure 91).

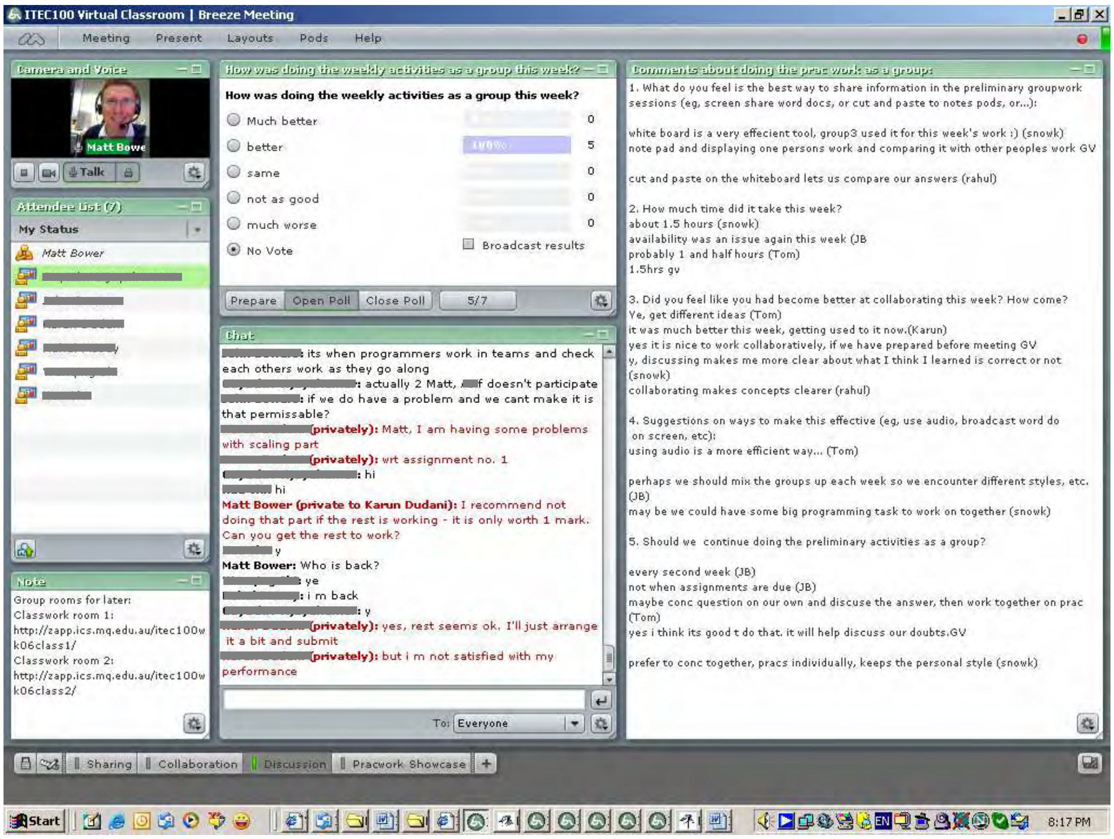

Figure 91 – Iteration 2 Topic 6 Soliciting student feedback regarding group-work approach to completing preliminary tutorial questions

In summary, students unanimously agreed that completing the preliminary tutorial questions in groups was "better" this week. Students indicated that they spent approximately 1.5 hours completing the preliminary activities in their groups (which is commensurate with the time that they might spend alone). Students felt that they had become better at collaborating this week, both in terms of using the technology and developing efficient patterns of collaboration.

OB4: Students were allocated to one of two groups and then asked to complete two in-class activities relating to 'loops'.

> *Task 1*
> *How often do the following loops execute? Assume that i is an integer variable that is not changed in the loop body.*
>
> *a) for (i = 1; i <= 10; i++) . . .*

b) for (i = 0; i < 10; i++) . . .
c) for (i = 10; i > 0; i--) . . .
d) for (i = -10; i <= 10; i++) . . .
e) for (i = 10; i >= 0; i++) . . .
f) for (i = -10; i <= 10; i = i + 2) . . .
g) for (i = -10; i <= 10; i = i + 3) . . .

Task 2
What does the following code print?

```
for (int i = 1; i <= 10; i++)
{
    for (int j = 1; j <= 10; j++)
        System.out.print(i*j % 10 + " ");
    System.out.println();
}
```

The interface had been designed to facilitate collaboration regarding the task. The question document had been placed in the major pod along the top-right of the room. A separate answer space was provided for each of the two questions in which students could collaboratively negotiate solutions. A medium sized chat-pod was provided at the bottom-left of the room for students to hold conversations (see Figure 92).

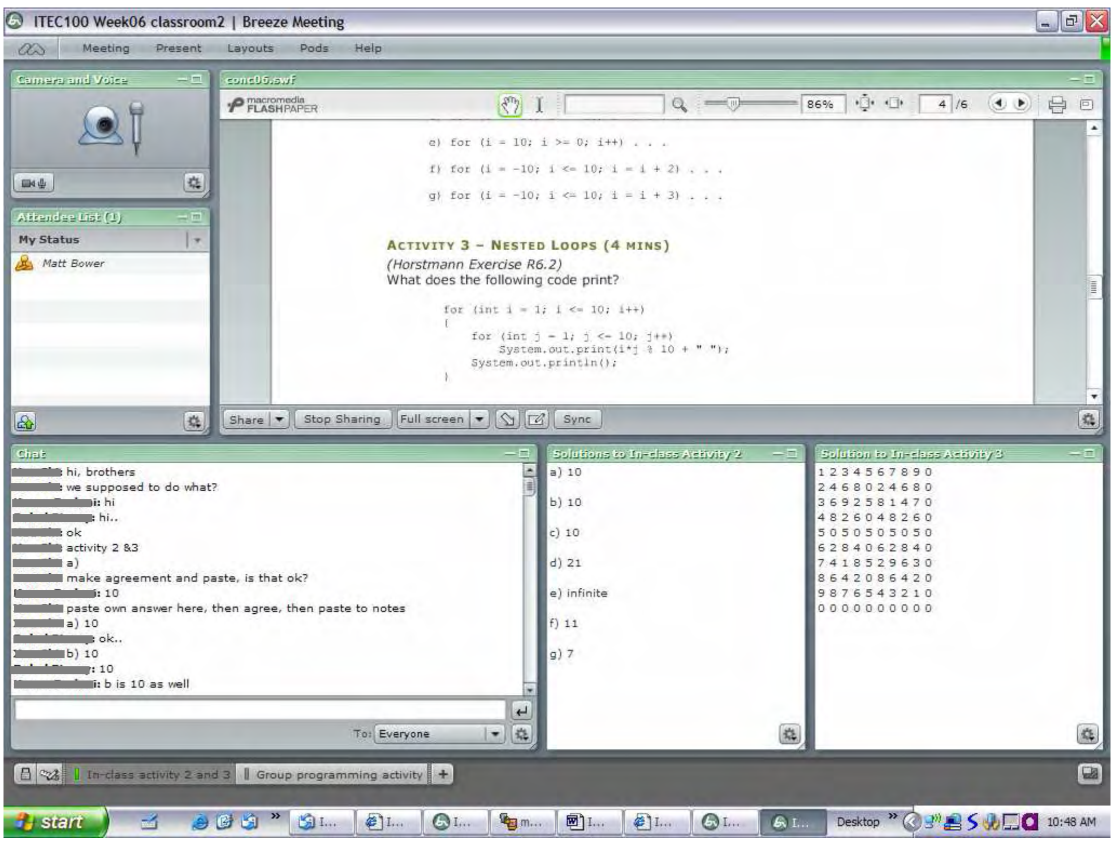

Figure 92 – Iteration 2 Topic 6 Purpose built interface for predictive tasks

Group 1 appeared to complete task 1 with ease, requiring little collaboration to complete it. For task 2 Group 1 did not conduct any collaboration, apparently because one student ran the program on their computer and copied the output directly into the solution pod. As a result there was less evidence of mental model formation in Group 1; their discourse related directly to proposing and agreeing upon solutions rather than explaining the underlying logic of the loops. In Group 2 students conducted much greater amounts of discourse (70 text-chat contributions as opposed to 25 in group 1). This was partly due to more discussion regarding how the loops function, but also due to more ancillary conversation. In Group 1 only one comment out of 25 was not directly discussing the subject matter at hand, whereas in Group 2 there were 30 out of 70 text-chat contributions that were discussing matters other than content such as how to collaborate, whether to move onto the next question, and social discourse. Note that for the second task Group 2 students still used the strategy of running the program in order to determine the output.

OB5: The task prescription for the next activity involved students completing the following task in their group-work rooms:

> *Write an Applet that draws a chess board. You may use any of the virtual classroom tools or approaches (screen-sharing, note-pod, whiteboard) to help you collaborate. To select new tools create them from the pods menu. Go!*

There were several similarities between the two groups' attempts at this task. Both groups of students chose to use a note-pod. The interface design illustrating the student's use of a note-pod to perform the collaborative programming is shown in Figure 93.

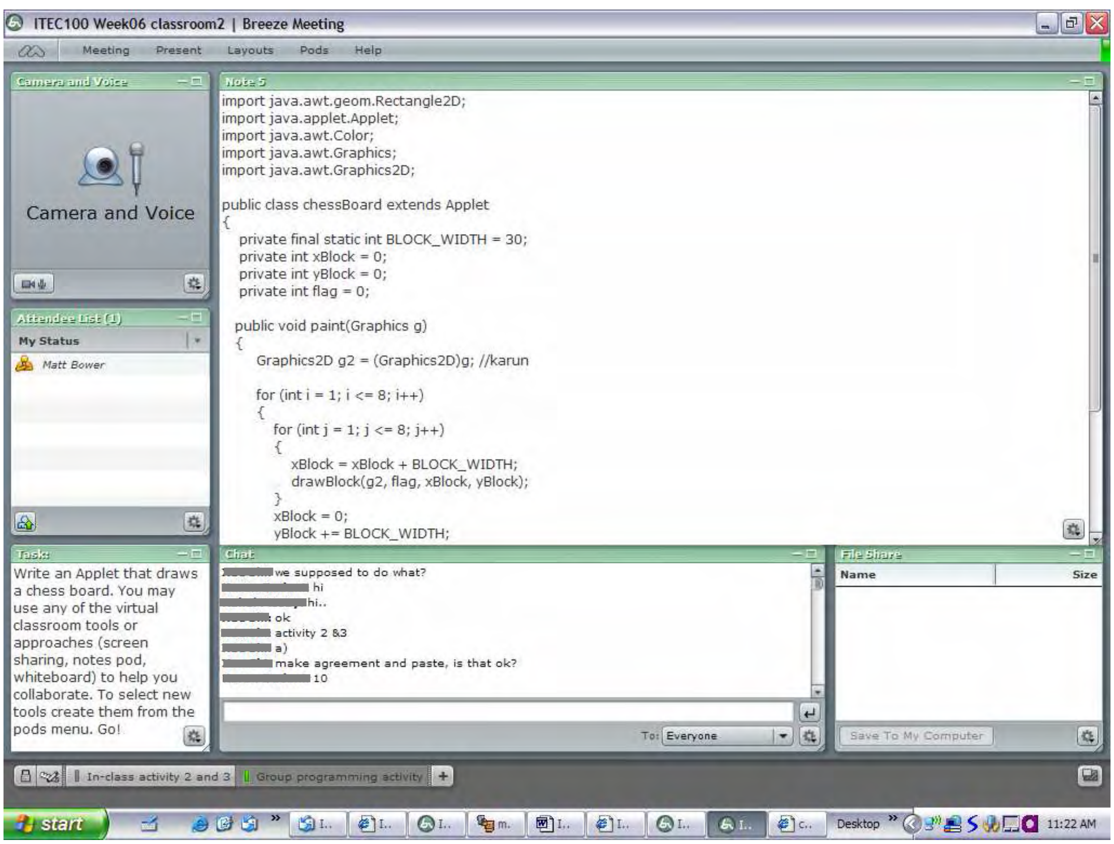

Figure 93 – Iteration 2 Topic 6 Students choosing to use a whiteboard to facilitate group programming

Both groups of students were able to fundamentally solve this problem within 25 minutes, although both groups experienced difficulty with filling the rectangles in alternate colours. Considerable task focused text-chat discourse was recorded in both rooms (66 contributions for Group 1 and 75 contributions for group 2). Both rooms had one student dominate the discussion, particularly in Group 2.

OB6: The final part of the lesson involved the teacher sharing student solutions to the RandomCircleApplet and FibonnaciApplet preliminary practical exercises. This was done by using note-pods and a zip file containing all programs. As well, the teacher used a transmissive approach to presenting and discussing each solution using screen-sharing of the IDE. The programs were covered quickly due to time constraints. This resulted in little student discourse.

Key Incidents:

- KI1: A great deal of mental model forming discourse was noted in the review of the groups' preliminary conceptual solutions. The fact that group two's solutions contained mistakes appeared to manifest more discourse than the other two rooms where solutions were for the most part correct.
- KI2: Both when asked directly and when responding to the survey question "How was collaborating in groups before class this week?" students indicated collaborating in groups was better than the previous week. It took less time this week because they were more familiar with the tools and had established collaborative patterns. For instance, students responded that comparing answers by copying / pasting to the whiteboard was an effective technique.
- KI3: Conducting student discourse using text-chat allowed the teacher to monitor student progress in both group-work rooms more easily, however was potentially not the most effective medium to be used while contributing to the note-pod (split attention and not utilizing dual processing capacities afforded by audio and note-pod).

Reflective Notes:

- RN1: Having students review other group's preliminary conceptual solutions reduces the intimidation that students may feel sharing and commenting on individual solutions.
- RN2: For the two in-class activities relating to loop outputs, explaining the purpose of the activity (to exchange and develop concepts of how loops operate) may have led to more constructive discourse in both groups.
- RN3: The collaborative learning redesigns are undoubtedly leading to greater quantities of discourse and more discourse relating to sharing and forming mental models.