A Case Study in Teaching Programming using a Hybrid Instructional Model

Jeanine Meyer[1]
and
Catherine Dwyer[2]

School of Computer Science & Information Systems
Pace University
1 Pace Plaza
New York, NY 10038
1-212-346-1569 FAX: 1-914-241-4489

Abstract

The debate surrounding distance learning versus the traditional classroom has often been presented as an either/or situation. Actual practice finds that many teachers use asynchronous tools to support a traditional course structure. The authors have tested an instructional model that does the reverse: a 'mostly' distance-learning course that uses required and optional face-to-face sessions to support learning. An additional challenge for this course was to teach modern programming concepts in a compressed time period. This paper describes the history of the course, starting from a decision to focus on programming games. It continues with a discussion of what factors influenced the design and results of the course, and concludes with reflections on the course's success.

Keywords: asynchronous support tools, hybrid on-line course structures, programming games, Visual Basic

1. BACKGROUND

Distance and asynchronous teaching is the subject of much experimentation and some study (New Media 1999). This paper is a case study of a course designed to blend normal, face-to-face contact with significant support provided through the Web. The face-to-face sessions take place in a high-function computer classroom, a 'closed lab' (Knox 1996). This hybrid model is not featured predominantly in the research or popular press but it may be the model of choice for many situations. The structure and the course content must be carefully designed. The content and pedagogical approach chosen here was to make students design and program their own versions of common games with very little lecturing, general discussion or reading on language features in the abstract.

2. FACTORS INFLUENCING DESIGN

The course was designed to fulfill several different needs of students. The Master of Science in Information Systems includes a requirement for students to have taken a programming course, preferably a 'modern' language incorporating event-driven programming, object orientation and graphical user interface design. Such courses must serve a population of students that include many who have avoided programming or have had what they term bad experiences. Practically all the students work full-time and as a consequence can only take one or two courses each semester. This means that their progress towards a degree is slow. The authors had developed materials for teaching programming in Visual Basic with a focus on games and wanted a chance to pilot the materials. The department chair and the associate dean proposed offering an intensive, 3-week class during January intercession. The IS students seemed an appropriate target since the intrinsic appeal of the games could overcome their resistance to

[1] jmeyer@pace.edu
[2] cdwyer@pace.edu

programming. Though IS professionals may not be involved typically in the creation of games, it is important that they appreciate how event-driven programming or, more broadly, the logic behind how human/computer interfaces are constructed. The general idea of a project-oriented course was also viewed as positive for this population and this setting. The January time period would be an extra opportunity to take a class. The course would include required and optional sessions in the evening and Saturday. On-line support would be provided so students could work independently on their own schedule. These factors would help address the challenge of serving a set of students with different backgrounds and skills.

The needs of graduate (adult) students working full time have been cited as a primary motivation for the delivery of education using distance learning. The authors were concerned, however, with the high drop out rate associated with distance learning along with the fact that programming is a difficult skill to teach (and learn) (Courtney 1999). The idea built into this course design was to have a small number of required sessions used for student presentations, and a greater number of optional sessions that could reinforce concepts introduced through the text or on-line material. The idea was that most students could manage the workload and minimal required sessions while a mechanism was in place for those that needed extra attention.

The Web based instructional system was the product CourseInfo (Blackboard 2000). The particular choice had been made previously by the School. The students would need to have a means of communicating with the teachers while they worked independently and the on-line discussion forums and the digital drop box along with ordinary e-mail facilitated this support. Sharing the teaching load was important given the intense nature of the course. The CourseInfo tool provided a shared space for the two teachers to design the course, receive student work and communicate with students.

3. HYBRID MODEL OF INSTRUCTION

The course was titled *Programming games using Visual Basic*. The authors designed the course using a subset of materials developed for teaching programming concepts by constructing games. Students enrolled received a Xerox copy of this material and a user account for CourseInfo.

CourseInfo was used to provide students with working versions of the games, a class schedule, description of assignments, and discussion forums on different parts of the course. Material that was reinforced during optional sessions was summarized and posted on discussion forums. When students sent questions via e-mail, the authors reposted the questions and answers on forums and required that future questions be posted on the forums so that the whole class could benefit from the question and answer. Students additionally used CourseInfo to take two anonymous surveys, submit assignments, and receive grades. The Announcement feature of CourseInfo was used to keep students informed on various issues and was updated regularly.

A survey was administered at the start of class to determine the background of students and a post-course survey was given on attitudes. Both surveys were done using the survey facility of CourseInfo, which provides anonymity but does reveal who has taken the survey.

The on-line grade book was used for posting grades on student work. An on-line quiz was given the third and last required class. Students could check their grades and see if more was expected on any of the projects.

Students were registered for CourseInfo prior to the first session so that the first session could be used for instruction in the tool and taking a survey on experience in programming.

The course included required and optional sessions over the three-week period. The following calendar appears in the CourseInfo Course Information section. (Note: the tool supplies a calendar but its format was cumbersome and so this additional presentation of the calendar was made.)

Table 1. Schedule.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
3	4 **REQUIRED** **6 to 9pm** **Introduction** **Random, RPS & memory demo**	5	6 Due 9am Rock-paper-scissors exercises *OPTIONAL* *6 to 9pm* *Help*	7	*8* DUE 9am Craps projects *OPTIONAL* *2 to 6pm* *Help.*	9

10 Due 9am Hangman & mixed-up dolls exercises	11 **REQUIRED** **6 to 9pm** **Presentations Hangman & mixed-up dolls**	12	*13* *OPTIONAL* *6 to 9pm* *Help.* *(Cannonball math/physics*)	14	15 *(OPTIONAL)* *2 to 6pm* *Help*	*16* Due 9am Cannonball projects
17 Martin Luther King Holiday	18 *OPTIONAL* *6 to 9pm* *General Help*	*19* Post topics that may be on quiz.	20 **REQUIRED** **6 to 9pm** **Quiz** **Presentations Final projects.**	21	22 Due 9am Feedback.	23 Due 5pm Final projects

The required and optional sessions were taught in a computer classroom with the facility for the teacher's display or any student's display to be projected on all computers. In addition, the teacher's display can be projected on a screen. This means that teachers can use a combination of

- demonstration while seizing control of the students' displays,
- demonstration and students following along on their own computers,
- individual students demonstrating what they have done.

The computers are sunk into the desks making it easy for everyone to see each other during discussion. All these facilities proved beneficial for instruction in programming.

The optional sessions were by appointment. That is, after the first one, which was set at Thursday of the first week, students were asked to send e-mail if any of them wanted an optional session.

The workload was 2 short-answer assignments, 4 programming projects, and a final quiz (taken on-line, in class). The text included explanations of the games and the Visual Basic features required by each game. The projects were

1. dice game of craps
2. mixed-up paper dolls (combining pictures) or the word game of hangman
3. cannonball: animation involving ballistics and hitting a target
4. enhancing cannonball, tic tac toe or the computer game minesweeper

The first required session was used to introduce the Visual Basic software and the tool CourseInfo. Subsequent sessions were used for student presentations and a final quiz. Optional sessions were taken up with detailed discussion of algorithms, step by step development of mini-programs that taught a kernel of a game, and one on one tutoring on assigned projects.

4. STUDENT INTERACTIONS

Before giving a narrative of what transpired in the course, it is important to note that the CourseInfo tool provides information that may not be recorded for most courses. For example, all discussion postings are recorded and can be sorted by author, date, thread or some combination. The CourseInfo survey facility means that we could ask students their background and have the results tabulated immediately.

Recruitment was done through a mailing to the graduate students who needed to fulfill the programming course requirement. The text for the course was the authors' draft material printed and spiral-bound in two volumes. Students paid the cost of copying.

Pre-registration in CourseInfo was tricky. We had one 'dummy' student ID that was used to give students not yet registered access to the system for learning on the first day.

The course took place January 2000 with 18 students registered. One student could not attend the first session because of a requirement to work to handle any Y2K problems. Our use of CourseInfo helped address this situation. Note: three or four students dropped out or chose not to register after the first session in which the teachers went through a small project with students following along. Having a required face-to-face first session meant that students could make an informed decision on whether or not they wanted to take a course with such a constrained schedule. This first session was also a time to associate names, faces and, to some extent, personalities. After the first session, all students who remained were able to complete the course. These appear to us to be vital for asynchronous courses, especially those with unusual formats and/or content.

The initial survey asked students to describe their experience with Visual Basic:

Table 2. Visual Basic Survey.

Response	**Percentage**
No experience	67%
Completed a beginning VB course	25%
Advanced	0%
Work experience	8%

Three quarters (75%) of the students indicated no experience programming Visual Basic in the last six months. The responses to the request to list "any other programming languages you may be familiar with" invoked the following:

Table 3. Programming Languages Responses.

6 responses of either "None" or no response
2 similar responses: "I was introduced to the basics of Java" and "Java"
1 response: "Cobol, C, C++: have not done any programming since completion of these undergrad courses"
1 response: "Assembler, C, C++"
1 response: "Pascal, C++"
1 response: "Visual Foxpro, C, peripro, clipper, sql, lisp, assembly"
1 response: "Cobol, C, Foxpro"
1 response: "SQL, Cobol"
1 response: "C, C++, Pascal, HTML, Java"

The optional sessions as we indicated were by request. Four out of the five sessions took place. The second Saturday optional session was not requested. Attendance at the others was: 6, 2, 7, and 9 out of a class of 18. For the three sessions with 6 or more students, we spent part of the time lecturing and working through parts of the projects. The rest of the time (and the whole session with just two students), we worked one-on-one with students.

The distinct capabilities of the computer classroom were used in the required and optional sessions. Both of the authors spend time in less ideal classrooms and appreciate these capabilities. The pedagogical technique of having students demonstrate programming features serves to confirm that students are doing the work and also decenters the classroom, allowing expertise to come from students and not just teachers. It also provides an authentic audience for the students and other aspects of 'authenticity' (Shaffer 1999).

Since the School of Computer Science and Information Systems selected the CourseInfo tool and several other classes had used it, the authors were optimistic that everything would work as advertised. To our surprise, there were problems in the use of the digital drop box for the sets of files generated by Visual Basic. The files were corrupted. The solution was to use a facility called WinZip (WinZip 2000) to bundle and compress the files. However, it was never clear what happened. We suspect it was some combined effect of CourseInfo and the UNIX operating system on the server. What CourseInfo did provide was multiple mechanisms (Announcements with the facility to include a link to a page to download WinZip, Discussion Forums, and broadcast e-mail) to inform students quickly on the problem and how to work around it.

The Discussion Forums were set up for two general topics and for each of the required projects. The table below indicates the Forum topic, total number of postings (including postings by each of the two teachers), number of students posting messages and number of student postings. It should be noted that the authors initially answered questions by e-mail but later advised students to post questions on the Forum so that everyone could read the question and our response.

CourseInfo provides information on students' use of the different features of the system. Total student accesses of the system were content areas (announcements, calendar document, and description of assignments) 39.1%, discussion forums 20.3%, and student areas (chief use: checking grades) 39.7%. If time were measured, the distribution would be more evenly

Table 4. Discussion postings.

Forum topic	**total number of postings**	**Num. of students posting**	**postings from students**
General Discussion	28	3	4
VB Web sites	4	1	1
The game of craps	9	2	2
Hangman and mixed-up paper dolls	12	4	5
Cannonball	22	4	10
Final project	33	8	15

divided, since composing a posting takes time but counts as only one access.

The students were given an on-line quiz at the last required session. Since all other work contributing to the grade could be done unsupervised, we felt it was important that something be done in class under monitored conditions. For the grading, the quiz was the equivalent of one of the projects. The quiz contained what we viewed as straightforward questions on features of Visual Basic plus questions of the form: "describe how you used feature X in one of the Visual Basic projects." The grades are shown below. The lowest scoring student complained the most before, during and after the quiz, saying, "In the real world, no one cares if you know this and you can always look things up." The authors have decided that in the next offering of the course, we will make use of the CourseInfo facility to have a quiz that displays a correct answer immediately to provide students a practice test to take at home prior to the real quiz. We did create a course document with acceptable answers to all the questions and broadcast its availability via e-mail and announcement at the same time that we told students that their grades were posted.

Problems did occur with the in-class testing. The computers of two students froze. The teachers could not reset the system so that each of these two students could re-start the test on other machines. This may not have been the best solution in any case. What we did was take each student outside and give the quiz orally. Our marks were probably overly generous because we felt sorry for these students.

Another problem with the quiz and the final survey is that after any assessment is made available and taken, if the authors choose to switch it back to unavailable, all student work is erased. The quiz was given to one student during the optional day before the last day because he could not attend on the last day. The teachers made the quiz available, made a hardcopy of this one student's quiz, and then made it unavailable for two days until class time. A similar situation occurred for the final survey. It had to be made unavailable to make the change that the survey could be taken again. To prepare for this, we again made copies of the initial results. These problems could be characterized as a combination of lack of knowledge by the teachers and lack of flexibility in the tool.

A quirky problem arose concerning making the quiz scores available to students. Any individual grade could be made available or not available to the student. This included the project grades, which we marked as "good", "fair", "okay", and "redo", and the quiz. We decided we wanted to make the quiz not available while we were working on the final grades. However, there is a feature called quiz/exam average that computes and shows the average of all numerical grades. This cannot be turned off and so in a situation with just one quiz, the one quiz score was revealed. Needless to say, a student who received a low grade on the quiz discovered this before we did.

The actual quiz results, our assessment of the student programming and the survey results are described in the next section.

5. RESULTS

Students' performance on the programming tasks was quite satisfactory. Specifically, all students completed the first three projects, with some taking more time and needing more assistance than others did. These are challenging assignments for non-programmers. A few students went beyond the minimum requirements for these projects and improved the graphics of the user interface. Most of the students did something on the final project. The easiest option here was to make minor enhancements of cannonball such as keeping track of successful hits and a few were content to do so. However, some made major enhancements of cannonball (see comments below) and several attempted tic tac toe with some success or made some progress with minesweeper. A couple of students said that they would keep working on these projects.

The cannonball project involves shooting a circle shaped object from a thick line representing a cannon towards a target represented by a rectangle. The speed and angle of the cannon can be changed. The project simulates the effects of ballistics. Concerning games in general and this project in particular, people have suggested that our approach may appeal to males and not females. A recent study of Internet use by women, however, found that women actually spend more time playing games on-line than men (Pew 2000). It was our experience that all the students liked the idea of making games. This is consistent with the idea that girls do like building things as opposed to merely playing games (Sanders 2000). The class did not involve students playing games. However, it should be noted that the enhancements of cannonball by the males were more violent than those made by the females. For example, one very able female student created a woodsy background and had the cannon fire at balloons tossed up and down by a clown. In contrast, one male student constructed a game with missiles firing back at a shooter with all hits vivid explosions. Another student displayed the words "You're Dead" when the projectile hit its target.

The quiz, as indicated above, was designed to confirm knowledge of the basics in Visual Basic and also that the students had familiarity with the projects. The quiz results were as follows:

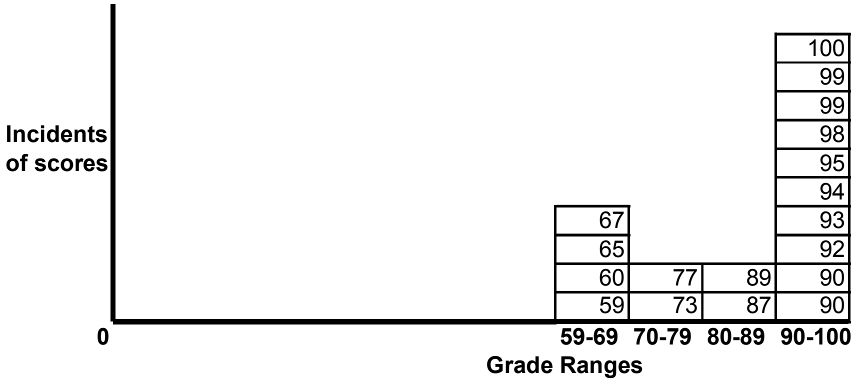

Figure 1. Quiz Results.

The marks were consistent with our assessment of individual students. We are pleased with these very positive results.

The survey results, as well as informal conversations with the students, indicated that the class was a success and, moreover, the individual features were successful. The survey questions cover the individual features and also the class as a whole.

To the question “would you recommend this course to other students?”, 95% of the students responded true. In the survey and in direct communication, several students said they wanted a follow-up course.

Categorizing the question on what was good about the course, seven students cited the asynchronous features (for example, “It fit my working schedule. We had two instructors, so we had help at any time we need it. Also communicating via e-mail with instructors was very helpful for homework problems. The web forum was an excellent learning tool.” “The on-line tool was very cool.”) Seven students cited the games' approach with four mentioning something else (for example, “a lot of hands-on”).

The question on what should be changed included requests for more choices of games, group projects, better editing of the textbook, and more time. One student wrote, “I feel this course would be better taught with more time in the classroom. The on-line feature of this course should remain intact. Uploading homework projects to the site is a very good idea.”

The statement: “I believe the class would work in a totally asynchronous format (no meeting as a whole group)” was labeled true by 38% of the responses and false by 62%. The statement: “The 3 required face to face sessions were essential.” invoked 78% true and 22% false. Though this could be interpreted as being evidence of inconsistency on the part of some respondents, our interpretation is that a minority of 38% believes that the course could be totally asynchronous with some of that number defining ‘totally’ as including required meetings. Our design with required and optional sessions served the students well. More than half attended at least two of the optional sessions.

The responses to the request to comment on programming were generally positive. Here is a sampling:

“I enjoyed the class tremendously. Because of this, my interest in programming has grown. I hope a follow-up course is taught next winter session.”

“I feel this is an area in which, I would like to explore when I graduate. Also I like visual basic, I am planning to get certification in this subject. Thanks for having this course.”

“The idea that programming is boring was done away with. I have little experience in programming but the class had a lot of enthusiasm and life to it. I will definitely continue to learn VB on my own. I wish there was another class to continue where we left off because I believe that VB can be a very important language for a systems analyst to learn.”

6. REFLECTION

This content and approach of this particular course was well suited to students doing independent work, namely programming the games that were discussed in the text. The minimum requirement for projects was specified but there were opportunities and challenges for the more ambitious students to be creative. This feature is especially important in a class for students with a variety of experiences. The asynchronous support provided a safety net for students that was essential given the spectrum of experiences and the intense time schedule. Since only a small number of students made active use of the Forums, we conclude that many students who did not make use of the asynchronous support still appreciated its existence. Students praised the organization and the appearance of organization. The students liked the method of delivering their work and even put up with glitches in the system. The technology level in the face-to-face part of the course, that is, the computer classroom, appears important to us and next time we will ask about it specifically in the closing survey. We believe the course features (stimulating content, high function computer classroom, constrained time, and asynchronous support) could be used on other project-based courses, especially if the work has some inherent appeal. The course did serve to give Information Systems students a basic level of understanding and experience with event-driven programming and user interface design.

7. REFERENCES

"Blackboard CourseInfo", 2000, http://company.blackboard.com/CourseInfo/index.html, (February 24, 2000).

Courtney, Mary and Allen Stix, 1999, "Survey Results and Observations on an Asynchronous Course," **Journal of Computing in Small Colleges**, Proceedings of the Fourth Annual CCSC Northeastern Conference, Volume 14, Number 4, May 1999.

Knox, Deborah and Ursula Wolz (joint chairs), 1996, "Use of laboratories in Computer Science education: guidelines for good practice", Report of the Working Group for Computing Laboratories, Integrating Technology into Computer Science Education, ACM SIGCSE, **SIGCSE Bulletin**, Volume 28, Special Issue, 1996.

"New Media in Higher Education and Learning", 1999, An International Information, Communication and Society (iCS) Conference organized by The Annenberg School for Communication, University of Southern California, USA and The Community Informatics Research and Applications Unit, University of Teesside, UK, October 27-30 1999, http://www-rcf.usc.edu/~ics/HigherEd.html (February 17, 2000)

Pew Internet and American Life, 2000, "Tracking On-line Life: How Women Use the Internet to Cultivate Relationships with Family and Friend", http://www.pewinternet.org/reports/toc.asp?Report=11 (May 11,2000)

Sanders, Jo, 2000, Interview: http://www.pbs.org/digitaldivide/gender-voices.html#jo (February 18, 2000) and private communication.

Shaffer, David Williamson and Mitchel Resnick, 1999, ""Thick" Authenticity: New Media and Authentic Learning", **Journal of Interactive Learning Research**, Volume 10, Number 2 1999.

"WinZip Home Page," 2000, http://www.winzip.com/, (February 24, 2000).