Integrating Interactive Computer-Based Learning Experiences Into Established Curricula: A Case Study

Anne Morgan Spalter
Department of Computer Science
Brown University, Box 1910
Providence, RI 02906
ams @cs.brown.edu

Rosemary Michelle Simpson
Department of Computer Science
Brown University, Box 1910
Providence, RI 02906
rms @cs.brown.edu

Abstract

Educators who wish to integrate interactive computer-based learning experiences into established courses must contend not only with the difficulty of creating quality digital content but with the often equally difficult challenge of reconfiguring their courses to use such materials. We describe our experiences with the Exploratories Project at Brown University [8] and the use of exploratories in an introductory computer graphics programming course [4]. We offer examples of both success and failure, with the goal of helping other educators avoid both painful mistakes and lost time spent coping with unforeseen logistical and pedagogical concerns. Among the lessons we learned: planning can't begin too early for the integration of such materials into an established curriculum, and all possible methods of integration should be considered before committing to any specific approach.

1 Introduction

Educators are intrigued by the potential for computer-based learning tools in their classrooms and put a substantial amount of time and energy into creating or locating impressive tools. However, there is often a "field of dreams" belief—a conviction that "if we build it, they will use it," i.e., if the material is compelling students will jump at the chance to use it. Unfortunately, our research and experience indicate otherwise. Tool integration into ones' curriculum, especially into long-established courses, can be just as challenging as making materials in the first place. In this paper, we discuss our experiences integrating a set of interactive computer-based learning experiences into an introductory computer graphics programming course at Brown University.

The course, Brown University's Computer Science 123 [ref], has approximately 70 students each year, mostly sophomores and juniors. Prerequisites include one of

Brown's full year CS introductory sequences, and a C++/Software Engineering course is highly recommended. The course meets twice a week for one-and-a-half hour lectures. Optional help sessions, run by the undergraduate teaching assistants, are offered for each assignment. While assignments are programming assignments - there are no exams or tests - recently the course has used homework assignments to review student's algorithm designs.

For over a decade, we have experimented with different approaches to bringing specific interactive materials into the course without disrupting its flow. Early work was all done with custom 3D development environments [8] and use was restricted to our computer graphics research lab. In 1995, we started the Exploratories project, which primarily uses Java applets. An exploratory is a combination of a exploratorium and a laboratory, a 2D or 3D microworld designed to help teach a specific concept or set of related concepts. While materials creation is essential, the long-term goal of the project is to create a Design Strategy Handbook based on the mining of our own and others' experiences. The Handbook includes guides, templates, patterns (as in Gamma et al's object-oriented "Design Patterns" [9]), and examples intended to guide teachers in the selection, creation, and deployment of interactive learning materials.

The material presented in this paper represents one set of experiences and will be incorporated into the Handbook.

2 Related Work

The attempt to integrate technology into the classroom is not a recent development. While we don't think of blackboards and pencils as technology, they were viewed as such when they were invented. During the last century, each major new technological invention—radio, movies, television, and more recently the computer—has had advocates predicting a revolution in education. While some individual projects, such as the PUMP project [17], have been quite successful, overall the results have not lived up to the promises [5], and there are claims that the attempt is misdirected in any case [10].

Although we may be seeing the birth of a truly new form of education, as the economic demands for just-in-time, distance, and lifelong learning combine with widespread Internet availability, the classroom remains the primary venue for education and is likely to remain important into the foreseeable future.

Approaches to integrating computers in the classroom have varied from complete replacement of parts of the curriculum (Papert [16], Apple Classrooms of Tomorrow (ACOT) [18], Classroom 2000 [1]), to use in virtual labs (Oregon [20], Moshell [12], Balsa [3], Brown [2, 8]), to providing specific topic support in the style of older technologies such as slide strips and movies (e.g., the Library of Congress project American Memory [14], Tom Snyder [6]).

For example, in the JavaLab project at the University of Oregon [20], computer-based lab sessions were introduced into a large introductory physics course. Although the computer lets these students learn more about experimentation (there were no labs before because of the size of the course), the use of a series of Java-based labs required a rethinking of the syllabus, the way that topics were presented, and the addition of lab times. The use of the Java labs had the positive effect, however, of "facilitating a more concept-based approach to teaching, as opposed to the traditional (and dysfunctional) topic based approach" [20].

In two courses at Brown computer labs were introduced with quite different results. In a biology course, simple computer labs seemed to fill a gap but had dull interfaces which made them unpopular with the students [13]. By contrast, in a multivariable calculus course [2, 13], high-end 3D graphics provided compelling interactive experiences, but the high-end graphics and interaction created difficult room scheduling problems and dramatically increased time commitments for the teacher, students, and support staff.

3 Using Interactive Learning Experiences in an Introductory Graphics Programming Course

To deal with these issues, and others, the Exploratories project has been creating 2D and 3D interactive learning experiences and exploring strategies for integrating these materials into traditional classroom settings as well as less traditional Web-based venues.

Each approach described below is discussed first in terms of our goals and expectations, then how we tried to achieve them, and finally the results and student response.

3.1 In-Class Demos

Our first approach was an obvious one, in-class demos. Since one of the important goals of an exploratory is to harness the power of interactive graphics to explain an idea, most function well as demonstrations. We selected about a dozen for use at appropriate places in the lectures.

Like any demo, use of an exploratory takes time away from the lecture, so we also had to slightly compress points or eliminate material. Our lecture room has computer projection, but otherwise we would have had to arrange for AV support and perhaps book a separate room. Given the unpredictable nature of Web-based Java execution, we had to test all our Java-based demos on the actual equipment, and even with the actual logins that would be used ahead of time. Finally, we rehearsed our undergraduate TA demoers.

We found in-class demos to be extremely successful. For example, demos of our applets about signal processing enabled us to explain convolution much more clearly than in previous years. The convolution applet in Figure 1 shows an original function, drawn by the user, a filter, also drawn by the user, the product of the filter as it slides over the function, and the changing area under the product—the convolution of the filter and the original function. In the past, the instructor had relied on colored pens and the layering and sliding of overheads to explain these relationships. With proper planning, we had few technical problems and students appreciated the change of teaching approach. It also served to awaken interest in the materials and increased the chance of students using our applets on their own time.

Student feedback, gathered in mid-term and final questionnaires, was overwhelmingly positive. The time commitments and logistical preparation thus seemed well worthwhile. The chief problem, aside from those created by lack of adequate preparation, is that software can always crash and hardware can always malfunction, A demo that blows up not only requires some fallback explanation (perhaps the old class notes) but can also have an unpleasant psychological effect on the class—a sense that things are not under control. This applies to lab sessions as well. For example, in one of our experimental lab sessions, a student's exploratory ceased to function when the whole class tried to use it at once. The student ended up leaving the lab in tears and her fellow students were understandably upset as well. When computer demos and lab sessions are commonplace, this problem's impact should lessen considerably, but for students unused to them, such technical difficulties can be very disconcerting.

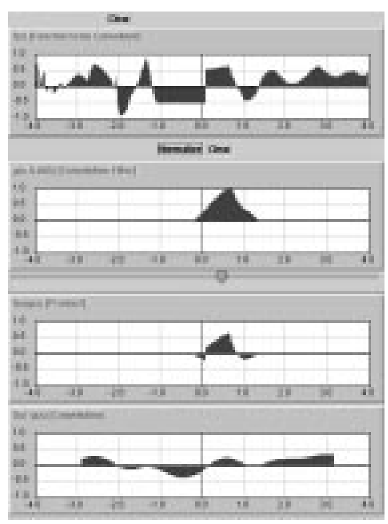

Figure 1: An interactive applet to teach convolution.

3.2 Optional Extra Curricular

We assumed that after seeing compelling demos in class, students would interact with them on their own time. Thus, our next step was make them available to the class and suggest different exploratories be used with different lecture topics. We added links to exploratories from the syllabus and then conducted informal polls and observations to see if they would be used. Most of these exploratories were also demoed in class.

We found that such optional exploratory use was largely ineffective. Students had too many other claims on their time and even those who expressed interest often spent little or no time with the applets. In addition, most of our applets lacked supportive text and students weren't sure what benefit they would receive unless they put in substantial time in first.

We learned from this experience that we needed to better motivate the exploratory use by showing how the exploratories would help is them in assignments. This was accomplished by having TAs use them in help sessions. It also reconfirmed the need for exploratories to be embedded in a hypertextual environment that explains their use, relevant background concepts, and relationship to other exploratories. In addition, when the use is optional, ease of use becomes even more important. These two points had been explored in a paper entitled "Granularity in the Design of Interactive Illustrations" [11] and led us to create simpler exploratories, with very simple interfaces, embedded in explanatory text.

3.3 In-Class Use, A User Study

In-class demos were working, but we believed that interaction was fundamental to the use of our material and that perhaps all students should be made to use them during the lecture. Since the optional extra curricular use had not been very successful, we did a user study [15] to see whether brief use of an exploratory in class was worthwhile.

We selected every other student as they entered room to either use the applet or to be in a control group, viewing a number of screen shots taken while an experienced user interacted with the applet (for example, Figure 2). Afterwards, all the participants were tested with randomly selected subsets of a collection of questions. Unfortunately, there was no statistical difference between the performance of our study group and the control. We concluded that five to 10 minutes was not enough time to spend with our style of learning experiences.

It had been difficult enough to get 15 minutes taken out of the lecture for the study and it was obvious that taking more time would not work. We thus abandoned the idea of annexing class time for exploratory use. Also, our lecture room has workstations, but otherwise in-class use requires two different classrooms and students may have to switch between them during the class session. We are also fortunate to have a large number of teaching assistants, but in other cases a course structure may not be able to support many students needing assistance at the same time.

Figure 2: Example of a screen grab from the in-class user study

3.4 Laboratories

We decided to solve the time-constraint problem by having additional lab sessions for exploratory use.

We planned a lab during an upcoming evening and advertised in it the class. Again, we had to test software, train TAs and book appropriate rooms. The first lab was on color theory and was not in support of a specific assignment.

Students who came to the lab gave positive responses on a feedback form and most who came attended for the entire session. We had initial difficulty running more labs, however, first because of scheduling problems and secondly because an hour-plus lab session required substantial materials development around a given topic. When additional labs were conducted, there were mixed results. In the end, attendance was not high enough to merit the extra effort and time demanded by their continuation.

3.4 Homeworks

The lab format inspired another approach, however, which we are trying for the first time this year. We decided to have students use the exploratories in homework assignments in which they had guided experiments to perform. We could also tie these homeworks in with the rest of the course by asking questions relevant to the concurrent programming assignments. Another benefit was that, once designed, the homeworks could be used year after year.

The final question in each homework asked for feedback on the experience of using the exploratories and answering the lab-style questions. In general students enjoyed using the exploratories and felt that experience contributed to their understanding of the concept being taught. There were some problems getting the exploratories, which in this case were Java applets, to run smoothly for everyone. Some labs were more popular than others, and several students felt the questions should be more difficult. There were also repeated suggestions for more accompanying text.

Based on this feedback, we are reassessing the lab questions, creating more levels of difficulty and fine tuning the wording. We will then add these questions to our Web site so that exploratories users outside of the course can have access to them.

3.5 Conclusion

For us, in-class demos and homeworks have provided the most rewarding and workable methods for integrating interactive computer-based learning experiences into our introductory graphics programming course. The demos are universally well received and the homeworks not only compel students to take advantage of what our applets have to offer, but also seem to create a critical mass of student-driven applet use. The teaching assistants have recently come into the computer lab in the evening to find students not only using exploratories for their assignments, but also perusing ones that teach related topics or that add depth to the class presentations.

For all of the approaches, advance planning was essential in order to take care of logistics such as room reservations, AV setup, software testing, TA training, lecture timing, as well as to develop new content for use in labs or homeworks.

4 Future Work

We are working to create a comprehensive listing of categories of integration techniques, such as in-class demo and extra lab session, for our Handbook. These, and the issues and solutions that we have gleaned, will be presented in the pattern format [19] with examples from a range of projects. Our next step is to create guidelines, templates, example lists, and other materials that facilitate the use of these different approaches.

5 Acknowledgements

We would like to acknowledge Charlie Currie, 1999 Head TA for CS123 for his critical reading and suggestions, and our sponsors: NSF, Adobe, Microsoft Research, Sun Microsystems, and TACO.

6 References

[1] Abowd, G. D. "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment" in IBM Systems Journal, Special issue on Pervasive Computing, Volume 38, Number 4, pp. 508-530, October 1999. http://www.cc.gatech.edu/fce.

[2] Banchoff, T. http://www.geom.umn.edu/~banchoff/Calculus/Calculus.html; http://www.math.brown.edu/~banchoff/swork/research.html

[3] Brown, M.H. "Perspectives on Algorithm Animation", Proceedings of the CHI '88 conference on Human Factors in Computing Systems , May 1988, 33-38.

[4] CS123 - Introduction to Computer Graphics. http://www.cs.brown.edu/courses/CS123.

[5] Cuban, L. Teachers and Machines: The Classroom Use of Technology Since 1920. Teachers College Press, Columbia University, 1986.

[6] Dockterman, D. A. Great Teaching in the One Computer Classroom. Tom Snyder Productions, Inc. 1997.

[7] Educational Software Seminar, Brown University Computer Science Department. http://www.cs.brown.edu/courses/cs092/.

[8] Exploratory project: home page: http://www.cs.brown.edu/exploratory/; 3D creature construction kit: http://www.cs.brown.edu/research/graphics/research/creature/

[9] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns, Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[10] Gelernter, D. J. "Technology can't provide a quick fix for societal problems", NY Times, Nov. 30, 1997.

[11] Gould, D.L., Simpson, R.M. and van Dam, A. Granularity in the Design of Interactive Illustrations. ACM SIGCSE 1999 Technical Symposium, New Orleans, LA http://www.cs.brown.edu/exploratory/papers/CSE99-gould.pdf

[12] Hughes, C. E. and Moshell, J. M. "Shared Virtual Worlds for Education: The ExploreNet Experiment." ACM Multimedia 5 (1997), 145-154.

[13] Lerner, M. The Current State of Technology and Education: How Computers Are Used in K-12 and Brown University Classrooms October 27, 1997. http://www.cs.brown.edu/exploratory/papers/mrl/int_ed.html

[14] Library of Congress American Memory Project: Port of Entry. http://lcweb2.loc.gov/ammem/ndlpedu/educator.html.

[15] Majzner, Marc. Study of effectiveness of scaling applet based upon current interactive illustration model. www.cs.brown.edu/people/mgm/uistudy.html

[16] Papert, S. The Children's Machine: Rethinking School in the Age of the Computer. BasicBooks—HarperCollins Publishers, Inc., 1993.

[17] PUMP project: http://act.psy.cmu.edu/ACT/awpt/pump-home.html, http://act.psy.cmu.edu/ACT/act-home.html.

[18] Sandholtz, J. H., Ringstaff, C, and Dwyer, D. C. Teaching with Technology: Creating Student-Centered Classrooms. Teachers College Press, Columbia University, 1997.

[19] Spalter, A.M. and Simpson, R. M. "Reusable Hypertext Structures for Distance and JIT Learning", submitted to ACM Hypertext '00.

[20] University of Oregon, Virtual Laboratory. http://jersey.uoregon.edu/vlab/.