Digital Gaming as a Vehicle for Learning

Ursula Wolz
(moderator)
Computer Science and Interactive Multimedia
The College of New Jersey
Ewing, NJ, USA
wolz@tcnj.edu

Tiffany Barnes
Dept. of Computer Science
University of North Carolina - Charlotte
Charlotte, NC USA
tbarnes2@uncc.edu

Ian Parberry
Dept. of Computer Science & Engineering
University of North Texas
Denton, TX, USA
ian@unt.edu

Michael Wick
Dept. of Computer Science
University of Wisconsin Eau Claire
Eau Claire, WI, USA
wickmr@uwec.edu

Categories and Subject Descriptors
K.3.2 [**Computing Mileux**]: Computers and Education, Computer and Information Science Education.

General Terms
Design, Experimentation.

Keywords
Digital games, graphics, computer science education

1. Summary

During the past two years there has been a resurgence of interest in how to use digital games (e.g. video games, computer games and simulations) to support instruction in a variety of fields[3,9]. The focus is on how to exploit the rich interactivity of 3-D, multiplayer virtual worlds. Computer science education has, for the most part, taken a different approach: rather than having our students play video games to learn concepts we ask them to build games to learn concepts [2,5,6,7,8]. In the process of building games, students become immersed in gaming. Yet neither the IEEE/ACM CC2001 [1] curricular recommendations, nor the ABET/CAC [4] criteria mention the notion of gaming. This panel addresses the still controversial question of whether gaming is a legitimate component of computing, and if so, where does it fit within the curriculum.

Regardless of where or how gaming falls within the curriculum, it is touted as an approach that will be attractive to a diverse audience, thus increasing potential enrollment into more traditional computer science courses. However, implementing a fully robust, modern, visually compelling, multi-player game from scratch as a semester-long project is problematic. The members of this panel will share a range of experiences in how to exploit a game format to meet particular pedagogic goals.

The holy grail of modern commercial game design remains the "First Person Shooter," (FPS) a game in which a character views a 3-D world from a first person, rather than map or text-based perspective, and with weapon (gun) in hand, moves through an interactive story to attain some goal. Typically there is a lot of shooting and consequent blood and guts. The genre, despite its violent roots, supports some of the most sophisticated techniques of computer graphics, animation and visualization. FPS open source game engines also provide compelling vehicles through which to teach good software design including design approaches for agent-based artificial intelligence and peer-to-peer networks. As a group we will each present our views on this controversy and suggest ways in which FPS can leave its violent roots in a manner similar to how the "kill text" button in early text editors became a more benign "cut" or "copy."

There appear to be four approaches to incorporating digital gaming into CS curriculum: (1) to support foundations courses, e.g. CS 1, (2) to provide specialized content at the upper level to prepare students for the gaming and animation industry, (3) to provide a curriculum encompassing thematic approach to CS in order to make CS and game development accessible to a more diverse population, (4) to provide trans-disciplinary experiences for CS students where they learn to interact with experts from other disciplines.

A unique aspect of this panel is that all of us have had experience of some sort with all of these approaches. Consequently, the names attached to the sections below reflect a somewhat arbitrary assignment by the moderator. Like any good game, each of us will assume a role and run with it, supporting our assigned character. The format of the session will consists of a brief overview, a short presentation of each approach, a set of challenges to the audience, and hopefully, a lively interactive discussion of the place of gaming in the curriculum.

2. Game Immersed CS 1 – Wick

The "game-immersed" approach is proposed by Wick and Barnes to make the introductory curriculum in CS more "fun." Wolz likes to comment to her CS 1 students that if she must teach Fahrenheit to Celsius conversion one more time her brain will explode. All of us have used game snippets to teach introductory concepts, but game immersion takes this idea much more seriously. For example, Wick and Barnes are currently working on a unit in which basic file processing is introduced by having students give their avatars the ability to extract messages hidden in images throughout a game world. The game world is built for them and they are required to program (in Java) the behavior of one method for their avatar. The resulting behavior is then used in carrying out a mission within the game. Note that this is a methodology that exploits game playing to teach computer

SIGCSE'05, February 23-27, 2005, St. Louis, Missouri, USA.

science while simultaneously requiring students to build a computer game.

3. Stand Alone Upper Level Course – Barnes

All of us have had experience developing and teaching an upper level elective in game design and architecture. A fundamental problem is gaining legitimacy for it as a bona fide topic of computer science. Barnes lost the coin toss with Wick to become the reporter on this approach. Curriculum standards are becoming established through informal course sharing (e.g. via the internet). Many game design courses include issues of genre, design, team development, and game engine (including 3-D and AI) implementation. A crucial component is formal presentation and class discussion on the concept of "fun." A game, unlike any other computer application must be fun. Yet all applications have equally intrinsic elements that much be captured algorithmically. We will make an argument that "fun" is a legitimate vehicle through which to develop problem solutions.

4. Curriculum Encompassing – Parberry

Given that there are masters programs in game design and architecture, and given that the game industry is one of the few that is still looking for skilled American programmers, how can we prepare ordinary CS undergraduate students for the industry? Parberry presents a framework through which the requirements can be met by a traditional undergraduate degree program with a few specialized upper-level electives. He will also describe "LARC" the Laboratory for Recreational Computing through which both formal classroom and more informal research and extracurricular activities take place.

5. Trans-disciplinary – Wolz

Computer scientists no longer build industrial strength video games by themselves. They also don't create animated films. Game development and especially design is as much an artistic enterprise as it is a technical one. Storytelling and the visual manifestation of it are arguably more important than the underlying game engine. Wolz will describe how a trans-disciplinary team of faculty can support a year-long collection of courses that allow students from a range of contributing disciplines to design and then collaboratively implement a video game. Parberry also raises the question of what components of the enterprise are legitimately computer science and which are not. The trans-disciplinary approach allows students to develop working knowledge of concepts across disciplines while simultaneously developing strong technical skills in their area of expertise.

6. Controversy – The Panel and Audience

To come full circle we will look at whether gaming is merely a good domain for CS applications, or is rightfully a field within CS. Each panelist will raise one question that will be presented to the audience in order to facilitate discussion and debate.

REFERENCES

[1] Computing Curricula 2001, Report of the Joint IEEE Computer Society/ACM Task Force, 2001.

[2] J. C. Adams. Chance-It: An object-oriented capstone project for CS-1. In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education, pages 10–14*. ACM Press, 1998.

[3] C. Aldrich. Learning by Doing: A comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences, John Wiley & Sons (2005)

[4] Criteria For Accrediting Computing Programs, Computing Accreditation Commission ,ABET, Inc., Baltimore, MD 21202 http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/05-06-CAC%20Criteria.pdf

[5] N. Faltin. Designing courseware on algorithms for active learning with virtual board games. In Proceedings of the 4th Annual Conference on Innovation and Technology in Computer Science Education, pages 135–138. ACM Press, 1999.

[6] T. J. Feldman and J. D. Zelenski. The quest for excellence in designing CS1/CS2 assignments. In Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education, pages 319–323. ACM Press, 1996.

[7] R. M. Jones. Design and implementation of computer games: A capstone course for undergraduate computer science education. In Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education, pages 260–264. ACM Press, 2000.

[8] R. Moser. A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it. In Proceedings of the 2nd Conference on Integrating Technology into Computer Science Education, pages 114–116. ACM Press, 1997.

[9] M. Prensky. Digital Game-Based Learning, McGraw-Hill, 2004)