Lessons in Teaching Game Design

Anne Sullivan and Gillian Smith
Center for Games and Playable Media
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
{anne, gsmith}@soe.ucsc.edu

ABSTRACT

In this paper, we describe a video game design and programming class offered as part of the California Summer School for Mathematics and Science (COSMOS) program at University of California, Santa Cruz. The authors have co-taught this class for four years, and have had an opportunity to iteratively explore a number of approaches in teaching the class. The course has evolved a great deal over the last four years; this paper presents a post-mortem of the program so far, describes our different approaches, and provides insight into the game design process for students.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *Computer science education*

General Terms

Design, Human Factors.

Keywords

Games, outreach, high school, computer science fluency, lesson plans, education, project-based course, game design and development.

1. INTRODUCTION

Each summer, the University of California at Santa Cruz (UCSC)—along with three other University of California campuses—hosts the California State Summer School for Mathematics and Science (COSMOS) program. COSMOS is a month-long hands-on program for high-school students, focused on exposing students to university-level topics and projects in the Science, Technology, Engineering, and Mathematics (STEM) fields. Each campus provides eight or nine *clusters* within the program, each concentrating on a different STEM field. For the last four years, UCSC has offered a cluster called “Video Games: The Design of Fun, from Concept to Code,” which gives the students background on designing and programming games, as well as giving them the opportunity to create their own video game. We have been involved with the cluster since its inception, as teaching assistants for the first year, and then as instructors for the later years.

The video game cluster is made up of 18-20 high school students. The class is made up of approximately equal numbers of boys and girls, ranging in age from 13-18. The students who choose our cluster have a stated interest in game design, although their backgrounds can be very different. While some students may have taken a computer science course in high school, most students who enter the program have no prior programming experiences, and there are some students who have rarely used a computer at all.

Students must present their final game projects to their teachers and classmates at their home institution, as a requirement for completing the COSMOS program. The majority of our efforts are therefore focused on the production of a complete game. We have experimented with a number of different teaching strategies for ensuring that students make the best game they possibly can while also teaching a foundation in game analysis and core programming concepts that can be applied outside of game creation. All of the games that students create are 2D games with original graphics.

As part of a STEM enrichment program, our cluster is heavily computer-science focused; one of our primary goals is to ensure that each student learns fundamental programming skills. However, we are equally focused on providing our students with a foundation in how to play, discuss, and analyze games critically, and teaching them about the game design process. We feel that these goals are symbiotic; games have previously been shown as a great motivational force for learning computer science [6] [10].

The focus on creating a project in a short time period, as well as the variety of experiences and backgrounds of our students, provides a unique set of challenges and opportunities for us as instructors. This paper presents the evolution of the game design portion of the course, including the different teaching strategies we have used and changes to the class format.

2. COURSE DESIGN

Over these first four years of our program we have taught two different design methodologies. The course also evolved from being mainly lecture-based to incorporating significantly more activities for the students.

2.1 Lecture vs. Activity-Based Courses

Classes are broken up into two three-hour blocks. While there is easily enough material to fill a month worth of three-hour long classes, we found that some students struggled with successfully incorporating lecture material into their projects from lecture alone. Therefore, we began to incorporate activities to apply the knowledge they were learning in the lectures to help connect these abstract concepts to their games and projects. After integrating more activities, students began independently incorporating the design concepts into discussions about their projects as well as other games.

To integrate the activities, each class was started with a game demonstration. The game would be selected to highlight the topic of the class for that day (such as *One Button Bob* [2] to discuss game mechanics or *Flower* [12] to illustrate a focus on aesthetics), and was typically chosen from less well-known video games or even board games. During the lecture, the games were referenced

FDG'11, June 29-July 1, Bordeaux, France.

and tied into the material, and at the end of the lecture, the students were given a design task which incorporated the concepts they had learned.

2.2 Design Formalism Choice

We have taught two schools of game design over the four years of the program. Initially, we focused on theoretical design concepts such as Salen & Zimmerman's theories of the magic circle and game rules [11], Juul's definition of a game [8], and Crawford's definition of interactivity [5]. In 2010, we introduced the Mechanics, Dynamics, Aesthetics (MDA) framework [7] as a practical framework to incorporate with the previous theories.

2.2.1 Theoretical Design Concepts

When we began the program, much of the game design material was borrowed from our college-level game design course. The material focused on a definition-based approach to game design. This method gives the students a strong foundation to be able to perform critical analysis on game design.

To enable critical analysis, we covered different schools of thought on definitions of a game and play [4] [1] [8] [3] [11], game rules [11], games of emergence and progression [8], challenge and conflict and the definition of interactivity [5], rewards and goals [11], and game narrative [9]. As the program progressed, we found that the students had some difficulty taking the abstract theories of game design and applying them to their games. This was in part due to the accelerated nature of the program; the students start to design their games by the second week of the program. We were left feeling that we needed a quicker way to introduce design theory that the students could easily relate to and incorporate into their projects.

2.2.2 Practical Design Concepts

In 2010, we chose to move to the MDA framework, which stands for Mechanics, Dynamics, and Aesthetics. Mechanics refers to the rules of the game, dynamics are the ways in which mechanics interact with each other and with the player, and aesthetics are the emotional response of the player to the game—or in what way the game is engaging.

The MDA approach describes how making changes to the mechanics, dynamics, or aesthetics of a game impact the other two aspects of the game. To demonstrate this, we used a slightly modified *Bartok* activity (a card game similar to Uno, but played with a regular deck of cards), based on *The MDA of Bartok* workshop [13] presented at Foundations of Digital Games in 2010. The activity involves having the students play *Bartok* a number of times while incorporating additional or changed rules. Each rule is added, changed, or subtracted one at a time, so the students are able to see how changing one mechanic can change the dynamic and aesthetic of the game.

This turned out to be a successful introduction to the concepts of game design; the students were able to integrate the theory into their own projects, and they were able to talk about their projects on a much deeper level. After introducing MDA, we then began integrating more theoretical design formalisms, tying them back to the framework.

3. PROJECT DESIGN

One of the primary goals of the COSMOS program is for students to participate in large-scale research or design projects that they can take back to their home institutions to present to their teachers and classmates. In the case of the video game cluster, this is a small, 2D game programmed by students in groups of two or three.

3.1 Introduction of a game theme

In the 2010 class, we decided to add a theme for the students to design around. This decision was based on the success of a game design theme in both the annual Global Game Jams[1] as well as the Experimental Gameplay Project[2]. The theme was used to add constraints to the design space which would challenge the students' creativity. We chose the theme of "twilight", with the added constraint that it could not have anything to do with the novels by the same name. While some students addressed the theme on a superficial level (e.g. having the game end when the sun set), others had the theme be a major part of their game. For instance, one group's chose to focus on a more abstract definition of twilight: "a state of ambiguity or obscurity." The game is a puzzle platformer where the player switches between two parallel worlds. Each world had different geometry and physics rules.

We felt that, in general, adding a theme component was a success, but in the future we would choose a different theme. Using a verb instead of a noun could encourage the students to incorporate the theme at more than a story level. Some students had trouble decoupling the theme from the novels, and it was too easy to incorporate the theme as an artistic or minor element. The games that integrated the theme at a mechanic level tended to be more innovative. It is possible that this is somewhat self-selecting (the students who think about the theme more deeply will think about their game designs more), but this can be tested in future iterations of the class.

3.2 Team Selection

For the first three years of the cluster, we allowed students to choose their own teams with the thought that the students would rather work with their friends. However, team selection happened early enough in the program that students hadn't really learned who they would work well with, or who they would get along with on a month-long scale. Many times, team selection was based more on where the students were sitting than any other factor. This would lead to issues with students working together with drastically different goals for the game, and in some extreme cases, some students were barely talking to each other by the end of the project.

In 2010, we decided to assign teams instead of allowing the students to choose their own teams. Each student sent us an email with three game pitches they would like to work on, and three of their classmates that they would like to work with. To help with team selection, we posted each student's game pitch anonymously to the class wiki. We then created teams based on skill levels, personalities, game pitch compatibility and the student's choices. This turned out to be a long process but was a large improvement in the success of the teams.

By balancing skill levels, we were able to minimize the number of students who felt like they didn't contribute to the game effort. By allowing the students to choose pitches they wanted to work on, it meant that they were able to work on a game they were interested in, even if they weren't with their friends. Overall this was very effective in creating stronger groups who worked well together; some of the teams are still working on their games almost a year later.

[1] http://globalgamejam.org/

[2] http://experimentalgameplay.com/

4. DISCUSSION

Over the last four years of COSMOS, we have made significant changes to our curriculum, all with the goal of enabling student creativity, easing creation of their games, and improving theoretical understanding of games. By iterating on our course's design, we have learned a great deal about what works and doesn't work in teaching game design to students, resulting in a list of concrete recommendations for similar game design programs to follow:

Promoting student creativity within constraints leads to greater student motivation and learning. Students are more motivated to learn when they have creative control over their work. For example, instead of being taught about programming concepts in the abstract, students learn by asking how to solve a particular problem in their project. We also introduced a game theme and required individual game pitches before assigning groups. Students are forced to think within the theme constraint, and then later forced to adapt their original idea to meet the constraints imposed by their team mates. This results in initial game design documents that are well thought-out, and discourages the reimplementation of an existing favorite game.

Activity-oriented classes help students connect abstract theories to concrete design. Students understand concepts more fully when they are able to apply them shortly after learning them. For example, lecturing about the differences between a game of emergence and a game of progression conveys the knowledge. Following with an exercise to find a game they like, identify whether it is a game of emergence or game of progression, and then change the game to make it fit the opposite role gives the students opportunities to understand the concept.

Adding instructor control over team dynamics leads to greater learning. While we thought students would do a better job of picking teams than we would, it turned out that the students do not have enough information to pick successful teams. Having the instructor pick teams based on student input is more work but ended up being far more successful.

5. CONCLUSION

This paper has presented the iterative development of a month-long intensive game design course held as part of the COSMOS summer program at UC Santa Cruz. As the program has matured, we believe that it has gotten much stronger, with more innovative student projects. We believe that the lessons we have presented can be used to help shape other game design courses and outreach programs.

6. ACKNOWLEDGMENTS

COSMOS is a program under the UC Office of the President, run by the Educational Partnership Center, and funded by both the state of California and by private sponsors. The authors would like to acknowledge the efforts of Jim Whitehead (UCSC) and Nathan Whitehead (NVIDIA) as former instructors for the Structure of Fun and Technologies of Fun courses, respectively. Finally, we would like to thank the cluster's Teacher Fellow, Takeshi Kaneko, for his insight into teaching methodology, and all of the UCSC COSMOS staff for their support.

7. REFERENCES

[1] Abt, Clark C. *Serious Games*. University Press of America, Lanham, MD, 1987.

[2] ARMOR GAMES. One Button Bob. (Flash Online Game), http://armorgames.com/play/5286/one-button-bob.

[3] Caillois, Roger. *Man, Play, and Games*. Simon and Schuster, Inc., 2001.

[4] Costikyan, Greg. I Have No Words & I Must Design. In *Proceedings of Computer Games and Digital Cultures Conference* (2002).

[5] Crawford, Chris. *Chris Crawford on Game Design*. New Riders Publishing, 2003.

[6] Guzdial, Mark and Soloway, Elliot. Teaching the Nintendo Generation to Program. *Communications of the ACM*, 45, 4 (April 2002), 17-21.

[7] Hunicke, Robin, LeBlanc, Marc, and Zubek, Robert. MDA: A Formal Approach to Game Design and Research. In *Proceedings of the AAAI Challenges in Game AI Workshop* (2004).

[8] Juul, Jesper. *Half Real: Video Games between Real Rules and Fictional Worlds*. MIT Press, 2005.

[9] Mateas, Michael and Stern, Andrew. Interaction and Narrative. In Zimmerman, Katie Salen and Eric, ed., *The Game Design Reader: A Rules of Play Anthology*. MIT Press, 2005.

[10] Overmars, Mark H. Teaching Computer Science through Game Design. *IEEE Computer*, 37, 4 (April 2004), 81-83.

[11] Salen, Katie and Zimmerman, Eric. *Rules of Play: Game Design Fundamentals*. Massachusetts Institute of Technology, 2004.

[12] THATGAMECOMPANY. Flower. (PlayStation 3 2009), Sony Computer Entertainment.