Game Design for Computer Science Majors

Philip J. Bernhard
School of Computing, Florida Tech, Melbourne, Florida, 32901

Abstract - We describe our experience developing and delivering a course in computer game design to students majoring in computer science. Constraints and objectives for the course are described, plus lessons learned from our experience, including things that worked and things that didn't. In general, we have found the course provides a very good platform that integrates a variety of topics from algorithms & data structures, graph theory, software engineering, statistics, probability, and psychology.

Keywords: Game design, Compute Science undergraduate education

1 Background

Several years ago we decided to offer a course on games in the school of computing at Florida Tech. The decision was motivated by several factors. Most noticeably, we had observed a decline in enrollments in recent years, and we thought that adding a course on games might help attract students. There was also a lot of interest and demand from current and incoming students; over half of our incoming freshman routinely expressed an interest in game design or development and we felt compelled to respond. There were, however, several constraints and issues that we had to deal with.

First, although many different types of courses on games are possible (design, physics, graphics, engines, etc.), lack of resources limited us to the addition of just a single (elective) course. This is in contrast to some departments that offer several courses on the topic, sometimes in the form of a specialization, concentration, minor, or in some cases a complete major [1]. This led to the second issue – what topics do we cover from the multitude of those available?

Finally, although there were several members of our faculty in related areas, e.g., graphics, none specialized in games, per se. This created the obvious concern - how could we possibly teach a course on games to a sizeable group of students who have been exposed to computer games from a very early age, and who play them on a frequent basis?

We quickly realized, however, that although there were plenty of games and concepts with which we were not familiar, we did have more experience than we thought. We hadn't played Halo, Warcraft, or Sims, for example, but we had played Tetris, Minesweeper, Pacman, plus numerous card and board games. In short, we weren't serious gamers, but we had plenty of experience with casual and non-computer games. One of the things we discovered is that most of the important concepts directly transferred from the experiences that we did have. There was certainly plenty of room to grow, but there was also a decent basis from which to start.

2 Course Content

In this section we describe the course content, how it was initially selected, and how it evolved. We also describe some of the principles that guided us during development and delivery of the course.

2.1 Initial Delivery of the Course

Given our lack of experience in the area, and our lack of a specific topical focus, we decided to implement the first iteration of the course in a manner similar to what we personally experienced as students in several humanities courses. Specifically, a book was selected for the course, along with other readings from various conferences and magazines. At the end of each class students were given a reading assignment for the next class, and were required to bring a 1-2 page summary of that reading along with questions, issues, points of contention, etc. Class time was spent discussing those issues, and no formal lecture was given. In summary, our initial delivery of the course had an informal, seminar-style format. Finally, for lack of a better title, the course was called *Game Design*, and offered as a special-topics seminar.

This particular approach had several advantages. First, as novices in the area ourselves, it gave us time to learn the material, basically alongside the students. Second, we have always found student enthusiasm in this class to be very high, and with the seminar format we were able to exploit this and rely on students to drive the discussion. Often times very simple questions such as *"Which are better, console games or PC games?"* prompt an in-class debate that required very little input or additional direction from the instructor. On the other hand, since there were no formal exams or quizzes, the seminar-style approach did make it difficult to assess student outcomes and assign final grades.

Over the period of several years, the course evolved into a form more typically used for science or engineering courses. This included a more precisely defined set of topics, a semester-long game design group project,

individual writing assignments, plus quizzes, and a full set of power-point slides.

2.2 Current Course Content

One of the first things we learned was there is a broad range of subtopics under the topic of games, encompassing enough material for several courses. For example, individual courses on game design, implementation, physics, artificial intelligence, modelling, graphics, graphic design, storytelling, dialog, scripting, engine development, and project management, among others, could be offered. Courses that focus on specific game development tools such as Unity, Blender or Unreal Engine are also possible. As noted previously, due to our restriction to a single course, we had to select a relatively narrow set of topics that would be appropriate for, and of interest to our computer science students.

Currently, our course covers the following areas, all of which are part of the 2008 IGDA Curriculum Framework [2]:

- Game Vocabulary – What is a game? What are the key components of a game? How are games different from puzzles, toys, and sports?
- The Video and Computer Game Business – Publishers, developers, the International Game Developers Association (IGDA), the Entertainment Software Rating Board (ESRB) and other trade groups.
- Game Design Concepts – Mechanics, reward systems, level design, immersion, balance, uncertainty.
- Artificial Intelligence – Traditional vs. game AI, path finding, state machines, dynamic game balancing.
- Procedural Generation – Algorithms such as midpoint displacement (and variations), fault formation, cellular automata, random number generation.
- Game Physics – Collision detection & resolution, space partitioning.

2.3 Teaching from a List of Rules

Whether it's online resources, books or other sources of information, one thing we discovered is that a lot of game design material takes the form of rules. This is best exemplified by the 400 Project [3], which had the goal of compiling a list of 400 game design rules. In this project each rule is relatively short – consisting of an ID# (1 through 400), an imperative statement (summary or title), an explanation in 250 words or less, a domain for the rule, and the name of the rule contributors. Figure 1 shows an example of one rule from the project.

As of the writing of this paper, the list consists of just over 110 rules, submitted by several contributors, most of whom are well known game designers. The project started in 2001, and over the past few years the list does not appear to have changed.

ID #: 106
Imperative Statement: *Have Fun in the First Minute*
Explanation: *In casual games it is critical to make sure the player is having fun right away. If the game is an expensive, boxed game then this rule is not critical (although still good to follow).*
Domain: *Casual Games*
Contributors: *Steve Meretzky*

Figure 1: A sample rule from the 400 Project.

Stating game design principles or concepts as rules is natural and very common. Although not quite as explicit as the 400 project, we have seen game design principles specified as rules in magazine articles, conference and journal publications, and books.

In general, we have found most such rules to be concise, insightful, and easy to explain. The one problem we had with this format, however, is that by itself it doesn't make for the best classroom presentation. Early in the development of our course we found our in-class slides to be dominated by list after list of such rules. This did not make for the best classroom presentation, or the best basis for discussion, and students very quickly lost interest.

More recently we have avoided letting this format dominate our materials, and we would recommend that other course developers do the same. One way to do this is to focus on a small number of high-priority game design rules, and supplement their presentation with supporting material, such as examples of YouTube videos of published games.

2.4 On-Line Resources

It should come as no surprise that the internet provides an abundant source of both instructional and non-instructional game design material. As noted in the previous section we have found YouTube in particular to be a tremendous source of pre-recorded video transcripts presenting many of the concepts we teach. These frequently come from games the students are familiar with, making it easy to keep them engaged. Additionally, most game engines and related tools have instructional videos available, which help students learn the tools on their own. Finally, on-line books, courses, the Wikipedia, and even blogs and user groups provide more material than we could ever use.

Of course, quality is always a concern with materials on the internet, and anyone using such material should consider their sources and take appropriate precautions. Just like any academic topic, the internet is full of both good and bad material on the subject of game design.

2.5 Vocabulary

One area where this last point is particularly apparent is in the vocabulary of gamers, designers and developers. Put bluntly, there is a lot of jargon associated with game design & development, and different authors, bloggers, and teachers will frequently use terms and phrases in conflicting and ambiguous ways. For example, the phrase "game mechanic" is one that we struggled with for some time. The large number of explanations on the internet created just about as many problems as it solved when we were trying to provide a reasonable definition to our students.

Of course, sometimes this is more a reflection of the subject matter than any particular author. For example, the term *game* itself is notoriously difficult to define precisely, as has been discussed by many authors. In such cases, instead of asking how a word is, or should be defined, we change the question to focus more directly on what we are really trying to teach. For example, instead of asking how the word *game* is, or should be defined, we ask *what are the essential elements that make up a typical game?*

3 Assignments and Projects

In this section we describe the course project, individual assignments, and mechanisms used to evaluate the students. This includes a semester-long game design team project, individual research/writing assignments, and on-line quizzes.

3.1 Project Description

From the very first delivery of the course we decided to have a semester-long, group project that involved design and implementation of a game prototype. At the beginning of the semester the class is divided into groups, typically consisting of approximately half a dozen students each. Students are allowed to form groups themselves, both in terms of membership and roles. However, in some cases teams are modified for students who can't find one themselves. Depending on the specific semester, and class size, this has resulted in anywhere from 4 to 9 teams.

Each team is required to give three presentations throughout the semester:

- Game Concept, Platform & Tool Selection – A brief summary of the game genre, target audience, major goals & objectives for the player, plus an indication of the hardware platform (PC, console, mobile device) and software tools to be used during development (game engine, modeling tool, software development tools).
- Game Design – A summary of the game design (not the software), the mechanics, gameplay, characters, terrain, audio and graphic design, reward systems.
- Game Demonstration – An in-class demonstration of the final prototype.

The first presentation is typically given in the first 3 weeks of the semester, the second is given around midterm, and the third is given the last week of class.

3.1.1 Selecting and Enforcing Prerequisites

An important lesson we learned after several iterations of the course was that by enforcing appropriate prerequisites, we were able to rely on the students' abilities to figure out development tools on their own, outside of class. This was particularly true for game & physics engines, such as Unity and Unreal Engine, plus modeling tools such as Blender.

In our curriculum computer science majors are all required to take two introductory programming courses, plus a course on algorithms & data structures before taking the game design class. This, combined with the fact that some students had already used such tools, plus the multitude of instructional YouTube videos and other resources on the internet, ensured they were able to identify, download, install, and quickly learn the required tools with virtually no classroom discussion.

This selection of prerequisites was anecdotally supported by the fact that in the few situations where we did approve a student taking the class without all of the prerequisites – most notably algorithms & data structures – those students struggled with the class project, significantly more than other students. Frequently students from other majors that don't have the prerequisites still believe they have some other experience with game design that is sufficient, but in all cases we have found that not to be the case.

Conceivably, if the project were modified to focus only on design with no prototype, or if the course as a whole were refocused on teaching how to use tools such as game engines, then the course might meet the needs of such students better. This latter change would, of course, be a substantial deviation from our current course content, which does not focus on tool details at all. We also believe it is not necessary for our current audience. Being able to leave this responsibility to the students has proven to have many advantages and allows us to focus on issues that we feel are more appropriate for the majority of our audience.

3.1.2 Tool & Platform Independence

With few constraints we allowed teams to select whatever hardware and software tools they wanted. As noted in the previous section, students were then required to acquire and learn the tools on their own. We did this for several reasons. In particular, many students are already biased against certain game development engines and platforms, and this flexibility allowed us to attract the broadest possible audience. It also helped free us from having to teach a specific tool in class, and allowed the class to adapt from semester-to-semester as different tools enter and

leave the market, without us having to update course materials in a significant way.

As noted in the previous section, we do not teach specific tools or how to use them. It was not our personal interest, nor possible given the combination of goals and constraints that we had for the course. That having been said, courses on specific tools could be offered, and many are in various schools, but we did not want to take this approach.

We also found students more than willing and capable of selecting appropriate tools. A sizable number of the students have already used game design or development tools, mostly on their own time. We have had no complaints from students feeling overburdened by having to select and learn the tools and in only a few cases have we had a team suffer a substantial failure due to poor tool selection or inability to learn one.

3.1.3 Use of On-Line Assets

As noted in Section 2.4, the internet provides access to a large quantity of online resources that can be used during game design and development. This is particularly true for what is commonly referred to as game content or assets such as clip-art, character sprites, music clips, partially/fully developed terrain, fully functional objects or characters, and photos. Often times these are freely available, while other times they are sold with licenses that allow them to be integrated into games. In one recent, somewhat interesting case, students asked if they could use a "dancing zombie" in their game, which they were able to purchase over the internet for under $25.

Initially we allowed students to integrate such assets into their games. We allowed this for several reasons. Most notably, a typical game will incorporate many such assets, and during a one-semester course we expected that students wouldn't have enough time to develop all the art and music themselves. Furthermore, a lot of what goes into development of a game has little to do with the (art) assets, but rather the game rules, mechanics, and physics. These later items are precisely where we believe the focus of the course should be for our students.

Of course, it should come as no surprise that students took substantial advantage of their ability to use online assets. We now require however, that they 1) give credit whenever they use such assets, and 2) include in their game prototype something other than just a loosely connected collection of downloaded assets. In other words, the majority of the effort that goes into their game is expected to be original development, and reused assets should be kept to a minimum.

More recently we have also concluded that students should document proof that they are not violating a copyright restriction or intellectual property law by using such assets. This became particularly apparent when in a recent project students incorporated a scene and music from the movie Star Wars. Consequently, in future delivery of the course we plan to require students prove that their use of such an asset is legal, even if only in the context of a college course.

3.1.4 Game Project Theme Restrictions

One of the first things students are required to do at the start of their project is to come up with a game concept. For example, students might decide to develop a first-person shooter that takes place in Europe during world-war II, a Mario-style side scrolling game, or a car racing game. Sometimes students decide to develop their games in 2-D, while others use 3-D. In the past we have not restricted students too much in this regard, except that they not develop a game that will create a public relations problem for the university, should it get outside exposure. For example, several semesters ago one team proposed developing a marijuana growth and harvesting simulation game. Although such a game has the potential for several interesting horticultural aspects (fertilizing, watering, light exposure, etc.), students were nonetheless encouraged to choose a less controversial plant.

We have seen examples of courses in other universities where students are restricted to 2-D games in one course, and 3-D games in another. While we do see the merits of such an approach, we have not used this particular restriction, but we have used others. For example, many of our students have developed games that have a Florida Tech university theme. One game in particular, called "Trolley of Terror," simulated the driving of a passenger trolley around campus. This was based on the actual, real trolley that traverses the Florida Tech campus, picks up passengers and delivers them to their destinations. Similarly, faculty, administrators, and even the students themselves have made appearances in their games as avatars, and in general students have found this very entertaining and motivating.

The use of university themes has become so common that we have considered restricting the students from doing this. Although it has definitely helped keep the students engaged and entertained, it has made the game design process easier for them than it probably is in the real world. In other words, it's relatively easy for them to develop a university themed game that gets laughs from the entire class, but more difficult to do so otherwise. For similar reasons we have considered discouraging students from developing games around current topics such as a recent election and its candidates - something that appears interesting and funny at the beginning of the semester, but has frequently gone stale by the end.

3.1.5 Final Demonstration Evaluation

We have found the final in-class demonstration to be very informative and, frankly, a lot of fun. After each team gives their demonstration, one of the unexpected things we do is ask two or more non-team members to play the game.

The reaction of the players, and the observing class, helps give us a feel for the playability of the game and the degree of success at creating a product that's fun. Of course, performing such demonstrations in class does require classroom media support, including a computer, projector, screen, and speakers or a sound system.

Although we don't track attendance for most classes, we do require all students to attend all presentations. Otherwise students tend to skip those classes when their team is not presenting. We have found student engagement and interaction is especially high during the presentations, provided the students are in attendance. This is particularly true of the final demonstration.

3.2 Individual Assignments

As noted previously, and by other authors, enthusiasm on the part of students for this topic is typically very high. This made it easy to prompt classroom discussion. Even students who haven't prepared for class or completed required reading are more than happy to express their opinion on just about any topic related to games.

One way we harness this energy is to give students individual, quick turn-around assignments that requires them to research/investigate a topic, write a short summary of their findings, and discuss their findings in class. Typically, these assignments are based on a game design concept previously discussed in class, but in general their focus and format varies greatly.

Sometimes we ask students to find an example of a game concept as it appears in a specific game. Other times we ask them to evaluate a specific game to see how it implements a concept. Regardless, in all such cases the objective is to prompt classroom discussion. Simply put, on the day when such an assignment is due, we ask a select few to tell us what they discovered, and typically this results in a lively discussion.

3.3 On-Line Quizzes

Evaluating and grading student performance has continued to be a challenge. During early iterations of the course this was due to the fact that we simply didn't have a good feel for the material ourselves, so it was difficult to know what the students were capable of, and how we should evaluate them on this type of material. Over time we have developed a better idea of what to expect on the projects, what kinds of game designs were realistic and what were not, how close students should get to a completed game by the end of the semester, etc.

More recently we have started using on-line quizzes to evaluate students' understanding of the topics discussed in class. The quizzes are generally objective-style questions, such as T/F or multiple-choice. All quizzes are open-book and open-notes. The students are even allowed to use google and access the course slides during the quiz. The only restriction is that they do the quizzes by themselves. In order to minimize the effect of free access to materials, we also put a limit on the time students have to take the quiz; approximately 30 seconds per question. Finally, students are only given one attempt at each quiz.

Students are informed of the time-limit in advance, and advised to study sufficiently so that their use of materials during the exam is minimized. Collectively, all of these constraints, plus our own growing understanding of the material, has resulted a more reasonable and acceptable grade distribution.

4 Conclusions

We first offered this course in response to student demand and declining enrollments. At the time, we had several faculty members who specialized in various aspects of computer graphics and vision, but no one who was an expert in game design or development. This created some concern at first – how could we possibly teach a course on games to a sizeable group of students who have been exposed to computer games from a very early age, and who play them on a frequent basis?

We quickly realized however that although there were plenty of games and concepts with which we were not familiar, we did have more experience than we thought. We hadn't played Halo, Warcraft, or Sims, for example, but we had played Tetris, Minesweeper, Super Mario, plus numerous card games and board games. In short, we weren't serious gamers, but we had plenty of experience with casual and non-computer based games, and most of the important concepts directly transferred. There was certainly plenty of room to grow, but there was also a decent basis for which to start. We would encourage anyone considering such a class to give it a try.

5 References

[1] Colman, R., Krembs, M., Labouseur, A., and Weir, J., *Game Design & Programming Concentration Within the Computer Science Curriculum*, SIGCSE'05, pp. 545-550, 2005.

[2] IGDA Curriculum Framework, The Study of Games and Game Development, International Game Developers Association, February 2008.

[3] Barwood, H., *The 400 Project*, www.finitearts.com/pages/400page.html.