LAMAR UNIVERSITY
A Member of The Texas State University System

ISSN: 1940-8978

# A Foundation for Growth: An Introductory Course in Computer Game Development

Timothy E. Roden, Rob LeGrand

No. 4, October 2012

Computer Science Department Technical Reports - Lamar University

Computer Science Department, Lamar University,
211 Red Bird Lane, Box 10056, Beaumont, TX 77710, U.S.A.
URL: http://cs.lamar.edu/tech_reports
Email: tech_reports@cs.lamar.edu

# A Foundation for Growth: An Introductory Course in Computer Game Development

Timothy E. Roden
Department of Computer Science
Lamar University
Beaumont, TX, USA
troden@lamar.edu

Rob LeGrand
Department of Mathematics & Computer Science
Angelo State University
San Angelo, TX, USA
rlegrand@angelo.edu

## ABSTRACT

A one-semester introductory undergraduate course in computer game development is described. The freshman-level course has been taught for four years. The course was designed and implemented in the context of a new curriculum in computer game development. Goals of the course were to retain undergraduate computer science (CS) majors, attract new students to CS, and expose students to the content creation pipeline used in more advanced courses. A significant aspect of the course is the end-of-semester project to create a machinima – a video created using computer game technology. Besides the initial goals for the course, other benefits were discovered including fostering a sense of community by helping students get to know their peers early in the program. Results, both quantitative and anecdotal, show the course has exceeded expectations in meeting the goals originally established. In addition, the course content has been adapted to the format of a three-day summer workshop to recruit high school students to the university. The course has been integral to the growth of the CS program, enabling the program to nearly triple in size over four years and achieve both statewide prominence and national recognition.

## Categories and Subject Descriptors

K.3.2 [**Computing Milieux**]: Computers and Information Science Education [**Computer Science Education**]

## General Terms

Design, Experimentation.

## Keywords

Computer science education, game programming, game development.

## 1. INTRODUCTION

In 2007, the Computer Science (CS) Department at Angelo State University (ASU) sought to implement a new program in computer game development. The motivation was to increase the number of CS majors and turn around what had been a steady decline in majors over the preceding ten years, a trend not unique to ASU. Computer game development was identified as the single most significant enhancement to the existing CS program that had the potential to increase the number of majors, based on faculty interest and positive reports from other universities [1, 2, 9]. It was decided to add the new curriculum as a set of elective CS courses. The new curriculum was modeled not unlike other similar programs already ongoing in academia in which game development courses are typically targeted at upper-level students – juniors and seniors, and game development is kept within the a traditional CS degree as opposed to creating a separate game development degree [2, 9, 10].

A dedicated game development laboratory was established with keycard access to only those students identified as enrolled in the game development sequence. A projector was installed so classes could be taught in the lab. Three specific game development courses were added, all of which required students to complete the first three semesters of introductory CS programming courses as a prerequisite. Two of the courses focus on PC game development while one course features development for a handheld platform.

Near the end of the initial curriculum development an introductory course was envisioned, almost as an afterthought. In retrospect, the introductory course was a vital component to the success of the new program. The course had no prerequisites and was open to all majors. The three credit-hour course was named "Introduction to Computer Development."

The course had three goals:

1. Retain majors, especially from freshman to sophomore years.
2. Train students on content creation tools used in the upper-level game development courses.
3. Recruit new CS majors.

### 1.1 Goal 1: Retention

It has been widely chronicled that retention is a problem in CS, especially among new students, with many schools choosing various methods to enhance retention [1, 3, 5, 7, 11, 12, 14]. Some schools have chosen to add game development in early courses specifically to address the retention problem, among other goals [1, 5, 7, 11, 14]. At ASU, faculty were not interested in adding game development to the early CS programming courses.

*SIGCSE 2013*, March 6–9, 2013, Denver, Colorado, USA.

Without an entry level game development course, the result would have been students spending a year and a half of study prior to taking anything game development related due to the prerequisite requirements of the game programming courses. It was feared that many students would drop out of the CS program before their hopes of becoming game developers were realized due to the lack of game development experience in early courses. Experience has shown that many students can easily become disillusioned with computer science in early semesters if courses don't meet their expectations of software development as a fun activity. For example, students may respond more positively to programming courses involving games and graphics as compared to the traditional business-oriented number-crunching assignments like interest-rate calculators [1, 5, 7, 11].

The primary goal of the introductory course was to encourage retention of CS students attending ASU specifically for the game development sequence. Entering freshmen who indicated a desire to study game development would be encouraged to take the introductory course during their first semester or as early as possible. In addition, since CS students taking the course would be concurrently enrolled in the first C++ programming course, as part of the normal sequence of courses for a CS degree, adding a programming component to the introductory game development course was not a priority. A programming component has been identified as an important feature of some similar courses at other universities, especially where students may not be concurrently enrolled in another programming course [6, 11, 13].

## 1.2 Goal 2: Content Creation Training

Early on in the development of the new game development curriculum it was understood a limitation of the new program would be the lack of a synergistic curriculum in the university's art department. The art program was very traditional and lacked courses suitable to train artists to become content creators for computer games. For example, there were no courses in 3D modeling. This was not altogether negative since the instructor for the capstone game programming courses wanted students to be minimally proficient in content creation and create some of their own art assets for their games. A few other similar programs have had the same goal [6, 14].

Given the limitations of the art department, students would have to create all of their own art assets for their game projects. Doing so would require knowledge of several content creation tools including 2D image editing, 3D modeling and animation, texturing, audio editing, and video editing. From experience teaching similar upper-level game development courses at another institution, the instructor planned to spend about a month introducing these tools and techniques in the first PC development course. The introductory course offered an opportunity to jump-start students' knowledge of the tools. The benefit to the upper-level course is less time could be spent on teaching students how to use the content creation tools with a corresponding increase in time spent actually teaching students how to write game code.

With this secondary goal in mind, several in-class hands-on lectures were scheduled throughout the semester to walk students through demonstrations of how to use the various tools. To accomplish this, the introductory course would have to meet in the game development laboratory during the semester. One negative aspect of this is it limited the size of each section of the course to the number of seats in the lab.

## 1.3 Goal 3: Recruitment

While the first two goals of the course were paramount, it was expected the course could function as an on-campus recruiting tool. As such, course enrollment was not limited to CS majors and in fact was advertised as widely as possible through regular posting of flyers on billboards around campus. At least one other university has indicated recruitment as a motivation for the establishment of an introductory game development course [13].

To facilitate more non-CS majors taking the course, the course was added to the core curriculum as an approved course to fulfill the one-semester computer literacy core curriculum requirement.

# 2. COURSE OVERVIEW

The course meets twice a week during long semesters. Course enrollment is limited to 15 students per section due to space limitations of the game development lab. Since the course is very discussion-oriented, the small section size has proven to be an advantage.

The bulk of the course is a survey of computer game development both from a player perspective and also from a developer point of view. The course content is chosen to provide an overview of how computer games are made from concept to finished product. Hands-on exercises allow students to gain first-hand experience with content creation. Written assignments enable students to try their hand at game design and become more familiar with a particular role in the game development process such as programming, art, design or management. Finally, the machinima project ties together several themes in the course and validates students are able to successfully utilize several of the content creation tools presented during the course.

**Table 1. Grading**

| | |
|---|---|
| Projects (5) | 30% (6% each) |
| Machinima project | 20% |
| Hands-on Labs (4-6) | 15% |
| Exams (2) | 30% (15% each) |
| Industry news presentation | 5% |

## 2.1 Grading

Table 1 shows how the class is graded. All work in the course is individually done. There are no team assignments. Two written exams are given and cover all 12 chapters in the textbook. The midterm exam covers the first six chapters while the final exam covers the last six. Approximately every two weeks a short project is assigned consisting of a 3-5 page written report which is both turned in to the instructor and also discussed during class. Hands-on in-class demonstrations of software are scheduled periodically throughout the semester. Grading for these exercises is pass/fail. Basically, students who show up on 'lab' days receive full credit.

Each student is also required to give one short informal presentation to the class about a topic related to the game industry. Students typically choose to report on a new game or hardware device recently released or an interesting item of news about a game developer or publisher. Students are told they can present their news item at any point during the semester at the beginning of class. The instructor has experimented with not including the industry news presentation during some semesters. The only downside to inclusion of this element of the course is it requires a small amount of bookkeeping on the part of the instructor. Experience has show this small effort is well worth it since a short student presentation on a game development topic at the start of class often provides an interesting starting point for discussion of the lecture topic for the day. Also, students typically combine their presentation with a particular viewpoint about the item being presented. In many cases, other students in the class have different viewpoints which can lead to more interesting interactions among students, both during the presentation and later during the lecture.

**Table 2. Course SLOs**

| SLO | Implementation | Assessment |
|---|---|---|
| 1 | Textbook/Lecture | Exams |
| 2 | Textbook/Lecture | Exams |
| 3 | Textbook/Lecture | Exams |
| 4 | Textbook/Lecture | Exams/Projects |
| 5 | Textbook/Lecture | Projects |
| 6 | Textbook/Lecture | Projects |
| 7 | Textbook/Lecture | Exams/Projects |
| 8 | Hands-on lab | In class |
| 9 | Hands-on lab | In class |
| 10 | Hands-on lab | Semester Project |
| 11 | Hands-on lab | Semester Project |

## 2.2 Student Learning Outcomes

The student learning outcomes for the course are listed below. Upon successful completion of the course, students should be able to:

1. Discuss the history of electronic game development.
2. Distinguish between different game platforms and genres.
3. Define elements related to game strategy, theory and gameplay.
4. Identify the distinct roles and responsibilities of members of a game development team.
5. Analyze and develop game concepts and proposals.
6. Apply story and character development to games.
7. Evaluate the game industry and market.
8. Create a textured 3D model using 3D modeling software
9. Manipulate textures for use in 3D modeling.
10. Edit sound effects using sound editing software.
11. Create a video (machinima) using video editing software.

Table 2 shows, in general, how each SLO was implemented and assessed.

**Table 3. Two-week Course Schedule**

| Day | Week | Activity |
|---|---|---|
| 1 | 1 | Lecture on weekly book chapter |
| 2 | 1 | Hands-on lab or video |
| 3 | 2 | Lecture on weekly book chapter |
| 4 | 2 | Discussion of Project assignment |

## 2.3 Course Schedule

With the exception of the week of midterm and final exams, the course schedule revolves around a repeating two-week cycle as shown in Table 3.

One chapter is assigned for reading each week. A weekly lecture covers the chapter using slides adapted from the instructor resources CD of the textbook. The lectures are designed to be interactive, featuring group discussion. For chapters in which the material is shorter, a video is often used to augment and reinforce the lecture topic. A hands-on in-class graded lab is assigned typically once every two weeks. Alternatively, a video may be shown on that day to further reinforce important chapter material or to introduce topics outside the textbook such as machinima production. Since project assignments are due approximately every two weeks, one class is devoted to discuss and share students' results.

**Table 4. Lecture Topics from the Textbook**

| Topic | Description |
|---|---|
| 1 | History of computer games |
| 2 | Player motivation & demographics |
| 3 | Elements of computer games |
| 4 | Storytelling |
| 5 | Game characters |
| 6 | Gameplay |
| 7 | Game levels & level design |
| 8 | Interface design |
| 9 | Game audio |
| 10 | Development team roles & responsibilities |
| 11 | Production & management |
| 12 | Marketing & maintenance |

## 2.4 Lectures

The chosen textbook provides a broad overview of game development methodology currently used in industry [8]. The 12 chapters from the textbook correspond to 12 weekly lectures. The objective of each lecture is to both present and discuss the topic. The small class size fits nicely into a discussion format. Table 4 shows the lecture topics.

To augment lecture material, videos are shown throughout the semester. Several related documentary videos shown in the course were purchased from educational vendors. A video covering the early history of computer games describes the birth of the industry from early coin-operated machines and home consoles to second generation consoles and the first widely used handheld gaming device – the Nintendo GameBoy. Another documentary describes the creation of modern video games from concept to pre-launch testing from the point of view of several different game development studios.

Other videos are used to showcase in more detail specific roles of members of a game develop studio such as programmer, artist and level designer.  Many triple-A games are often released as special editions packaged with DVDs chronicling the game's development.  The instructor has shown some of these during the course with very positive feedback from students.  In particular, these videos typically feature young people at work in various settings in game development studios.  An important objective of viewing these videos is to help students interested in entering the industry identify role models in addition to letting someone else besides the instructor talk about how they perform their job.

**Table 5. Project Topics**

| Topic | Description |
|---|---|
| 1 | Play and review a computer game |
| 2 | Game design based on original Intellectual Property |
| 3 | Game design based on existing Intellectual Property |
| 4 | Organizational analysis |
| 5 | Career analysis |
| 6 | Machinima creation |

## 3. PROJECTS

Five short projects are assigned throughout the semester, approximately one every two weeks.  The assignments are completed individually by students and submitted in 3-5 page hardcopy format.  For each project one class session is reserved as a discussion session.  Students are not asked to stand in front of the room and make a formal presentation.  Instead, the instructor asks each student to tell the class a few details of their project results.  These informal sessions often lead to very lively discussions of the project and related game development topics. The sixth project in the course is a more detailed project in which each student creates a machinima.  The projects are listed in Table 5.

Project assignments are taken from the instructor resources from the textbook.  They are therefore designed to reinforce the reading material.  In addition, the assignments are chosen to, as much as possible, encourage student creativity.  Letting students be creative builds confidence, gives them a sense of accomplishment and reinforces the idea that their ideas are worthwhile.  The projects, collectively, teach students the important concept that game development is an industry based on creativity. Introductory game development courses at other universities have captured a similar theme of creativity [6, 13].  A particular focus of the projects is on designing games, an idea used elsewhere [4].

The first project asks students to play and review a computer game.  This project is assigned the first day of class.  The discussion session for the project often serves as an ice-breaker. Students get to know each other and what type of games each other prefers.

The next two projects ask students to design games.  In the first project the design is required to be unique and not based on any existing intellectual property.  For the second design project, the class collectively decides on four existing properties.  The assignment is to derive a game design based on one of the four. While the first game design project encourages an individual approach, the second encourages students to discuss ideas since some students will be basing their design on the same property.

For the third and fourth projects students are asked to delve deeper into a specific aspect of the game industry.  The organizational analysis project asks students to research a company involved in some way in the industry and write a report about the company with an emphasis on recent news.  The career analysis project asks students to research a particular industry job and outline a strategy to get that job.  The strategy includes education, work experience, networking, portfolio creation and a resume.  For students already planning to enter the industry after graduation this serves as a reality check and can help to guide their college career.  The instructor has often been approached by students completing this assignment to verify if their methodology is sound and if they are on the right track to getting the industry job they want.

### 3.1 The Machinima Project

The term machinima was created as a combination of the words "machine" and "cinema".  A machinima is a short film.  The video portion of the film is taken by recording sequences in one or more computer games while audio is often overlaid using live voice acting with custom music and/or sound effects.

Through in-class hands-on exercises throughout the semester, students are trained to use the software necessary to author a machinima.  The software includes a screen capture program, 2D image editor, audio recording, audio editing and a video editor.  A primary purpose of the machinima project is to validate students have learned to use the software tools.  Since the same software is used in upper-level CS game development courses, this introduction helps jumpstart students' technical abilities early in the program.

The machinima project is due during the week before finals week and serves as the capstone creative experience in the course.  One class day is reserved to watch and critique the videos in a group setting.  Students are also encouraged to share their videos with the larger community by posting online for public viewing.  Many have done so which is another public relations plus for the department.  It also enables the instructor easily to showcase work of previous students during the course in order to help motivate them and establish the expectation level for the quality of the machinima.  The technical requirements of the videos are as follows:

1. Contains in-game video from one or more 3D computer games (must be recorded by the student, not simply downloaded from the Internet).
2. A title screen.
3. A credits screen.
4. Background music and/or sound effects.
5. Recorded human speech.
6. Transitions (wipe, fades, etc.)

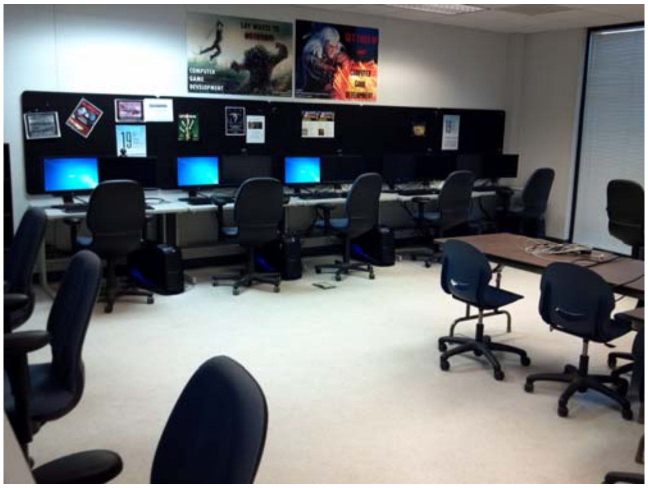

**Figure 1. The game development laboratory**

## 4. HANDS-ON LABS

The course meets in the dedicated game development lab, shown in Figure 1. Several classes during the semester feature hands-on demonstrations of several software packages. The content of these presentations has varied during the years the course has been taught. The core set of labs teach students how to use the software tools necessary to complete the machinima project. These include a session on how to capture video from a PC computer game using a licensed capture application installed on machines in the lab. The instructor also coaches on using freeware capture programs students can install on their home PCs. Another session covers audio recording and editing using Adobe Audition, also installed on lab computers. A third session covers video editing using Adobe Premiere. Students also need to be able to use a 2D image editing application in order to create title and credits screens for their machinima. The instructor initially scheduled one class to explain this but later realized it was not necessary. Students had this knowledge already or were able to pick it up without instruction. As a result, demonstration of 2D image editing was deleted from the course content.

Other hands-on sessions were designed to give students a taste of how industry workers do their jobs. Each iteration of the course has typically included a one or two-day demonstration using 3D modeling software to include basic modeling and texturing. One of the goals of this exercise is to get students familiar with the modeling package used in the upper-level game development courses. Another session typically included is an exercise using a program that automatically generates 3D human face models. The models can be procedurally generated from user parameters or optionally generated from a photo of a person's face. This exercise is often very fun for students who input their own photo using a web camera installed on each machine.

## 5. RESULTS

The introductory course has been taught continuously each long semester starting with the fall 2008 semester. It has also been taught several times during a five-week summer session.

The original three goals for the course have been realized with varying degrees of success. Freshman to sophomore retention of CS majors increased by 10% from academic year 2008-2009 to 2010-2011, according to the most recently available data on retention. Anecdotal evidence suggests a greater retention among CS students choosing the game development option. Almost all students in the latest upper-level PC programming course, taught in fall 2011, had taken the introductory class, compared to approximately half in fall 2010 and only a few in fall 2009. The increase in retention mirrors experiences elsewhere with entry-level courses in game development [5, 11].

The course goal of teaching students content creation tools was realized. The hope was students taking the first PC game development course would require very little training on the content creation tools because they had already learned some of this in the introductory course. However, since the introductory course was not a required prerequisite for the PC course, some students in the PC course were not as prepared. The end result was the instructor in the PC course spent as much time going over content creation tools as before in order to accommodate all students. While a time savings in course lectures was not realized, many students benefited by being better prepared, resulting in higher quality work. In retrospect the introductory course should have been required for students opting to take other game development courses. One reason it was not required is the limited number of seats in the course each semester meant the course would always be in short supply. In fact, this proved to be a continuing problem. Since the start of the game development focus, incoming CS freshman enrollment grew to nearly 100 majors by fall 2011, a figure nearly double what it had been prior to the introduction of the game development courses. Similar growth has been reported elsewhere [14]. At least half of incoming freshmen wanted to take the introductory course and the department typically admitted only 30 students per year in two sections of 15 each. In spring 2012 the number of sections of the introductory course was increased to two sections per semester as a result.

The course goal of recruiting new majors was realized. Each academic year, one or two students who had taken the introductory course changed their major to CS based on their experience in the course. However, it is certainly true that a much larger number of students regularly change their major from CS to something else as a result of taking introductory programming courses and discovering they do not like CS. Overall, the course did serve to recruit new CS majors but the benefit was small.

An unforeseen benefit of the course, based on anecdotal evidence, is the course has fostered a sense of community among CS majors. New students can often feel disconnected and alone. The small group interactions and interactive format of the course enabled students to get to know their peers early in their college career. Students from the course could often be seen together in the lab, open during non-class times, discussing assignments together or playing games in the lab. This aspect of the course contributes to fostering retention. This benefit of the course was particularly welcome in keeping with one of the original goals of the course as a retention tool.

At the end of each semester students are asked to complete a course evaluation to rate both the instructor and the quality of the course. Students can optionally write comments on the evaluation form. The course consistently receives very high evaluations and is one of the top-rated courses in the CS program. Students regularly comment the course is the best course they have ever taken. The feedback has been much like that reported for a similar course at another university [14].

In summer 2009 the CS department offered a three-day summer workshop on game development for high school juniors and seniors. The workshop was subsequently offered each summer thereafter. The goal of the workshop was to showcase the game development program as a recruiting tool. Much of the material for the workshop came from the introductory course including lecture slides, videos and hands-on exercises. The workshop was limited to 24 students. On average, two students from each workshop subsequently enrolled in ASU. While the recruitment benefit was less than expected, an unexpected result was the workshop received quite a bit of free publicity in local media which helped to advertise the new game development courses.

## 6. CONCLUSION

A one-semester introductory course in computer game development has proven to be an important foundation for building a new curriculum in game development as part of a traditional CS degree. The CS program at ASU grew from 71 to 213 majors over a four-year period since the introduction of the game development courses, including the introductory course. Two years after the courses were adopted ASU was named one of the Top 50 Game Design Programs in North America by Princeton Review in their 2010 survey of similar programs. By fall 2011 ASU CS enrollment had grown to 3% of total enrolled students, making it the CS program with the highest per capita enrollment of any Texas public university, according to a fall 2011 survey conducted by ASU.

Despite these successes, a significant problem remains to be solved. The small class size and limited number of sections did not scale well with the growth of the program. In the future the department will need to decide to either add more of these small sections or increase the course size. Increasing the course size may reduce the impact of the course somewhat since one of the benefits of the course has been to foster a sense of community through the close interactions of students in the course.

## 7. REFERENCES

[1] Becker, K. Teaching with Games: The Minesweeper and Asteroids Experience. *The Journal of Computing in Small Colleges*, 17, 2 (Dec. 2001), 23-33.

[2] Coleman, R., et. al. Game Design & Programming Concentration within the Computer Science Curriculum. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '05)* (St. Louis, Missouri, February 23-27, 2005) ACM Press, 2005, 545-550.

[3] Crenshaw, T. et. al. A Case Study of Retention Practices at the University of Illinois at Urbana-Champaign. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '08)* (Portland, Oregon, March 12-15, 2008). ACM Press, 2008, 412-416.

[4] Kessler, R., van Langeveld, M., and Altizer, R. Entertainment Arts and Engineering (Or How to Fast Track a New Interdisciplinary Program). In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)* (Chattanooga, Tennessee, March 4-7, 2009). ACM Press, 2009, 539-543.

[5] Leutenegger, S. and Edgington, J. A Games First Approach to Teaching Introductory Programming. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07)* (Covington, Kentucky, March 7-10, 2007). ACM Press, 2007, 115-118.

[6] Linhoff, J. and Settle, A. Teaching Game Programming Using XNA. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '08)* (Madrid, Spain, June 30-July 2, 2008). ACM Press, 2008, 250-254.

[7] Morrison, B., and Preston, J. Engagement: Gaming Throughout the Curriculum. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)* (Chattanooga, Tennessee, March 4-7, 2009). ACM Press, 2009, 342-346.

[8] Novak, J. *Game Development Essentials: An Introduction*. Third Edition. Delmar Cengage Learning. 2011.

[9] Parberry, I., Roden, T., and Kaxemzadeh, M. Experience with an Industry-Driven Capstone Course on Game Programming. In *Proceedings of the 2005 ACM Technical Symposium on Computer Science Education (SIGCSE '05)* (St. Louis, Missouri, February 23-27, 2005). ACM Press, 2005, 91-95.

[10] Roden, T., and Etheredge, J. Educating Game Programmers, In *Proceedings of the 2nd International Conference on Game Development in Computer Science Education (GDCSE '07)* (Orlando, Florida, February 22-25, 2007). Microsoft Press, 2007, 82-86.

[11] Sloan, R., and Troy, P., CS 0.5: A Better Approach to Introductory Computer Science for Majors. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '08)* (Portland, Oregon, March 12-15, 2008). ACM Press, 2008, 271-275.

[12] Talton, J., et. al. Scavenger Hunt: Computer Science Retention Through Orientation. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '06)* (Houston, Texas, March 1-5, 2006). ACM Press, 2006, 443-447.

[13] Whitehead, J. Introduction to Game Design in the Large Classroom. In *Proceedings of the 3rd International Conference on Game Development in Computer Science Education (GDCSE -08)* (Miami, Florida, February 28-March 3, 2008). ACM Press, 2008, 61-65.

[14] Zyda, M., Lacour, V., and Swain, C. Operating a Computer Science Game Degree Program. In *Proceedings of the 3rd International Conference on Game Development in Computer Science Education (GDCSE -08)* (Miami, Florida, February 28-March 3, 2008). ACM Press, 2008, 71-75.