

Growing a Computer Science Program with a Focus on 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 comprehensive undergraduate curriculum in computer game development is described. The program was created as a set of elective courses in the context of a traditional computer science (CS) degree. Primary goals of the program were to increase enrollment in CS and prepare students for careers in the entertainment software industry. In addition, the CS department sought to compete for students with larger state institutions. To do so effectively the department needed to offer a unique program. Results show the new program helped the CS department to nearly triple enrollment 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, 3, 13]. 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 a traditional CS degree as opposed to creating a separate game development degree [3, 13, 14].

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. An introductory course, open to all majors, was added. To add synergy to the program, the instructor teaching the Artificial Intelligence (AI) course adapted the course to include a focus on games. A dedicated laboratory was also established.

The new program had three goals:

1. Increase the number of students majoring in CS.
2. Prepare student for careers in game development.
3. Align with industry trends and government priorities.

1.1 Goal 1: Increase enrollment

At the beginning of the fall 2007 semester, enrollment in CS at ASU was dangerously low. There were not enough students for the faculty. Small class requests were common. Increasing enrollment was seen as mandatory to maintain the CS department. In addition, the department wanted to hire more faculty to diversify the capabilities of the department with an eventual goal of adding a graduate program within a few years. To increase enrollment significantly the department had to be competitive with larger state schools. With far fewer faculty and resources than other schools ASU needed a unique program. A survey of curricula at other state universities indicated only a few schools had any undergraduate programs in computer game development. The most notable program was at University of North Texas where a faculty member had been teaching game development since 1993 [13].

1.2 Goal 2: Prepare Students

Computer game development was an easy choice to add to the CS program at ASU. A faculty member in the department had experience developing games. Student interest was very high, as has been reported elsewhere [1, 7, 13]. Jobs in the industry pay well with average starting salaries for new graduates near \$60,000. Due to the highly competitive nature of securing a job

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'13, March 6–9, 2013, Denver, Colorado, USA.

Copyright © 2013 ACM 978-1-4503-1868-6/13/03...\$15.00.

in the industry after graduation the department knew any curriculum would need to prepare students adequately to be successful.

The new program was a good fit since the existing CS degree prepared students well for game development. For the first three semesters, CS majors take programming courses in C++. We believe a foundation in C++ prepares students for careers in game development better than another language such as Java. Other similar programs have taken the same approach in response to perceived industry demands. Also, like similar programs, we want students to graduate from the program with one or two significant game projects to showcase to potential employers [13, 16]. We also want students to leave with a basic understanding of how to develop games for a popular handheld platform since we believe handheld gaming will continue to grow in popularity and market share. In addition, a goal of the new program was to teach students not only how to program games but also how to create game art assets. Since the university art department did not offer classes in content creation for computer games, the department would take on this responsibility as part of the new program. Educating students about the industry was also a priority. Understanding the industry gives students a sense of perspective when working on particular platforms and helps them better identify a segment of the industry they want to target in their job search after graduation. The textbook chosen for the introductory course [12] does a good job surveying the industry.

Some universities have reported success with creating specialized game development degrees [8, 10, 16]. At ASU, we chose to keep the new program in the context of the traditional CS undergraduate degree for several reasons: First, implementing the new program as a set of classes simplified the approval process since only university-level approval was needed. Second, faculty believed a traditional CS degree was more beneficial to a graduate over the course of their career in case they wanted to work in a field of computing other than game development. Despite this, our experience has shown there is a significant population of prospective students who would rather earn a specialized degree.

1.3 Goal 3: Program Alignment

To make sure the new program was successful consideration was given to how the new program aligned with other priorities, including university, local, state, and national.

At the university level, a new president had been hired in summer 2007. One of the president's top priorities was growth. The CS department was able to leverage this to obtain university funding for a new lab and equipment and also to move the new program quickly through the university curriculum approval process. The department also worked with the university communication department to publicize and promote the new program as much as possible including web, newspaper and radio advertising. CS faculty also met regularly with university recruiters to discuss strategies for marketing the new program to prospective students.

At the local level, representatives from the Small Business Development Center, located on campus, were eager to establish new relationships with businesses potentially seeking to relocate to San Angelo. This proved to be very beneficial a few years later when publicity surrounding the program's mobile development

course attracted several companies to give the university grants to development mobile software.

At the state level, attracting the game development industry to the state was a priority of the state's Governor. The Governor had recently attended the annual Electronic Entertainment Expo to recruit businesses to the state. It was anticipated the program would apply for state funds to enhance department facilities and equipment. Also, the state has more game development companies than any other state besides California. This was another fact that helped in recruiting.

At the national level, computer science had been identified for several years as one of the best growth areas for jobs by the U.S. Bureau of Labor Statistics. Starting salaries for computer science graduates are typically highly competitive as compared to other disciplines. Both of these facts were highlighted in communication to prospective students. The program also expected to apply for grants at the national level.

2. PROGRAM OVERVIEW

The new program in game development consisted of four elective courses. This presented a problem since the existing CS degree contained only two electives. To manage this, the existing degree was restructured. The existing degree's large kernel of required courses was trimmed to a smaller kernel to incorporate five electives. This allowed students to take the four game development courses and the artificial intelligence course. Courses moved out of the required kernel were programming languages, architecture and software engineering, all of which were retained in the curriculum as electives.

At ASU the academic year is divided into two semesters. Students can optionally complete a third semester in the summer. Beginning students in the CS program take three semesters of C++ programming. Subsequently, many other required and electives courses become available. The core of the game development program is the two-course sequence in PC game development. Since students must take the first three semesters of C++ and the first course in the PC development sequence begins in the fall, students are typically juniors when they enroll in the PC courses. A course in handheld game development is offered in the spring so students can begin developing games in the second half of their sophomore year. An introductory course can be taken at any time in the program. Freshmen interested in game development are especially encouraged to enroll. Having an early course in game development keeps student's enthusiasm high in addition to other benefits as reported elsewhere [3, 8, 10, 17]. Students are also encouraged to take the artificial intelligence course given the topic's importance to game development; they typically take it in their junior or senior year. In addition, students are given the opportunity to enroll in independent research if they express an interest in pursuing a specific topic in game development not covered during the regular courses. These sections require faculty sponsorship and are only approved for outstanding students.

The department allows students to minor in any discipline. Game development students are encouraged to minor in a discipline related to their particular interest in game development such as art, mathematics or English (story-telling).



Figure 1. The game development laboratory.

3. LABORATORY

Before the first course was offered the department received internal funding for a dedicated game development laboratory, shown in Figure 1. The lab has keycard access for only those students enrolled in the game development sequence and new freshmen who indicate they intend to take the game development courses. A projector was installed for both teaching and presenting student projects. Equipment includes twelve high performance dual-monitor PC workstations loaded with appropriate game development software including compilers, image editing software, video editing software, 3D painting software and content libraries. Table 1 lists important installed software. All of the game development courses are taught in the lab if space permits. For larger sections of more than 15 students a lecture room is used. The lab is similar in scope to those found at other universities with dedicated game development labs [13, 14, 17].

Table 1. Software Installed in Game Laboratory

Software	Purpose
Microsoft Windows 7	Operating system
Microsoft Office	Word processing, etc.
Microsoft Visual Studio	PC development tool
Eclipse	Android development tool
Adobe Photoshop	2D image editor
Adobe Audition	Audio editor
Adobe Premiere	Video editor
Lightwave	3D modeling/animation
BodyPaint 3D	3D painting
FaceGen	3D face generator
SnagIt	Video capture program

The importance of the lab and its benefit to the new program cannot be overstated. Small classes can be taught in the lab using hands-on methods of instruction. The lab also serves as an open lab for student use during non-class times. Students use the lab for both working on assignments and meeting with other students. Playing games in the lab is encouraged and promotes student

interest and enthusiasm in the subject matter. Even if a general access lab could be equipped with the necessary hardware and software, we agree with others that hosting these activities in a general access lab would be too distracting to other students [13, 14].

In addition to the benefits described, the dedicated lab afforded a unique recruiting opportunity. When shown the lab, visiting prospective students often commented how surprised they were that a university would have the level of equipment contained in the lab and how pleased they were the university appeared to have a high level of commitment to the program. The lab was often featured in marketing materials and on television.

4. COURSES

The following sections describe each of the courses in the game development program in addition to the artificial intelligence course which is complimentary to the program.

4.1 Introductory Course

As the first course in the game development sequence, the introductory course was important to provide a foundation for the program. The course originally had three goals:

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

Since students needed to complete three courses in C++ programming prior to taking a game development course it was expected that attrition would factor heavily in reducing the number of majors. To keep students enthusiastic and interested in the program the introductory course was created. The content of the course includes lectures covering aspects of the industry starting with the history of computer games. There are no programming assignments. Instead, assignments are heavily oriented toward enabling students to express their creativity through game design and other multimedia-related projects. For the final semester project students create a machinima: a video created using computer game technology. Students record video of a 3D computer game and insert voice, sound effects and still images using a video editing program. The resulting movies are presented in a group setting.

Equally important to retaining majors, the course is designed to teach students how to use content creation tools used in later semesters. Through hands-on demonstrations in the lab, students learn basic 3D modeling, audio editing, video editing and a tool that creates 3D human faces that can be exported to the modeling program for further customization. Since the course was created as an elective and not required for the game programming courses some students did not take the course. As a result, not all students entering the game programming courses were familiar with the content creation tools. A remedy for this would be to require the introductory course as a prerequisite for the other courses. One reason this was not done is the course was open to all majors and due to the popularity of the course CS majors often had a hard time enrolling in the course due to the limited number of sections offered each semester.

Despite the popularity of the course among non-CS majors, the course did not serve to attract a large number of students to the CS program. However, an unexpected benefit of the course was realized. The course fostered a strong sense of community among CS majors and helped new freshman get to know their peers early in the program.



Figure 2. Student project from the PC development course demonstrating lip sync morph animation.

4.2 PC Development Courses

Due to the complexity of PC development and the breadth of topics that can be covered, a one-year, two-semester sequence was created. A similar approach has been taken at other universities teaching game development [3, 13] and recommended as appropriate [11].

The choice of programming tools is an important one. An application programming interface (API) for graphics must be chosen that students can digest and become proficient using in a short amount of time. There is currently no standard in academia for which API to use. Many have used a high-level tool such as Microsoft XNA with mixed success [2, 5, 6, 8, 10] while others have used a custom API [5, 14]. In our experience the best tool to use is what the instructor is most familiar with. We used a custom API based on the Microsoft DirectX 9 fixed-function pipeline. Since a custom API will likely not have extensive documentation the instructor bears the responsibility for carefully instructing students in its use whereas with an industry-standard API students can be left somewhat on their own. During semesters with small groups of students the custom API worked well and students used it with good success. During semesters with larger classes of more than 15 students the instructor was not able to spend as much time with students individually. As a result, the quality of work was lower and student satisfaction was not as high.

Topics covered during the first semester include:

1. 3D computer graphics – 3D concepts, mathematics, hardware and software, APIs, object representation, file formats, camera control.

2. Windows programming – OS architecture, SDKs, event-driven programming, multi-threading.
3. Game architecture – major software components of a 3D game engine.
4. Texturing – texture coordinates, texture filtering, texture animation, mipmapping.

Assignments guide the student from loading and displaying a simple model they create in a modeling program to creating a walkthrough of an outdoor environment. The final semester project requires students to use their knowledge to craft a game. Since the focus of the semester is on outdoor environments most students typically create a game set in a forest or other outdoor space.

During the second semester two primary topics are covered: indoor rendering and character animation. Portal rendering is presented as a method for creating and managing large indoor environments. The instructor provides source code for a project that procedurally generates indoor environment from a set of student-created prefabricated 3D models of sections of the environment such as rooms and corridors. Students can also manually create their own indoor environments although few students have chosen to implement their own portal rendering code due to complexity and time constraints. Since students in the PC development courses are required to author their own artwork they also have to balance the time spent coding and creating art assets.

A second emphasis during the second semester is character animation. Multiple techniques for character animations are explained including morphing and skinned bones. Significant project assignments highlight both methods. To demonstrate their skills with morphing, an assignment requires students to create a 3D head animated using morphing and lip synced with audio. An example of a student project demonstrating morph animation is shown in Figure 2. An example of a student project demonstrating skinned bone animation is shown in Figure 3.



Figure 3. Student project from the PC development course demonstrating bones animation.

4.3 Handheld Development Course

The handheld game development course was designed both to provide students the opportunity to develop games for a mobile platform and to afford students an opportunity to take a game development course during their sophomore year. Another premise for inclusion of the course was to accentuate the uniqueness of the ASU game development program since, at the time it was first offered, no other university in the state offered a similar course.

For the first two years the course was offered the development platform used was Nintendo Gameboy. This was quite popular with students, many of whom had owned a Gameboy at some point in their lives. An example of a student program created for the platform is shown in Figure 4. An advantage of the platform is simplicity—programs are written to run directly on the hardware without an operating system. The course was redesigned in 2011 to use Android. The best supported development language for Android is Java. In the first iteration of the course, in spring 2011, the instructor used Java exclusively. Reactions to the new format were mixed. Students who had not taken the department’s only elective Java course expressed some dissatisfaction with the course since they were required to both learn Android and learn to program using Java. A different textbook was chosen for the second iteration of the Android course with good results [15]. Students were still required to learn Java as needed but the textbook helped abstract away many of the operating system specific details, enabling students to concentrate on creating games.

Overall interest in the handheld course among students has been very positive. Many students not interested in game development wanted to take the course in order to learn more about Android programming.



Figure 4. Student project from the Handheld/AI courses.

4.4 Artificial Intelligence Course

The artificial intelligence (AI) course fits into a traditional CS curriculum and covers important AI topics. However, we take every opportunity to relate traditional AI approaches to game development. Almost every topic covered is put into a game-development context, at least briefly. Some techniques, such as A* search, depth-limited alpha-beta pruning and game-theoretical analysis, are widely used in commercial games of different types.

Other techniques are pointed out as less useful in game development. For example, most machine learning approaches are too time-intensive to use in games, which usually must react to the human player in real time. Simpler, even hard-coded, approaches often work just as well and can run much faster. Even though a traditional AI approach can be overkill and a simple, approximate solution is often better in game development, students in the AI course benefit from a big-picture understanding of how computers can be made to think.

The programming assignments in the course are organized as a series of “agent challenges.” A C++ framework is presented that sets up an environment in which agents can compete and/or cooperate. Each student writes an agent function that takes as input a situation and returns an action. The agent functions are run inside the framework to determine the outcome and the agents’ performances are scored and compared. Agent challenges range from simple game-theoretic games (rock-paper-scissors, prisoner’s dilemma) to games that require search (Connect Four, Chomp) to explorable environments like a Wumpus world. Agent functions are given time limitations, so realistic efficiency/accuracy tradeoffs must be considered. The students enjoy the competition and chances for cooperation. The agent challenges add excitement and a real-world-application flavor to balance the theoretical feel of some of the course material.

The course culminates in a final project. Students propose their own projects. Subject to instructor approval, game-related ideas are explicitly encouraged. In fact, many students have overlapped the AI project with the final project in a game-development course. For example, a student concurrently enrolled in both the handheld game development and AI courses produced the program shown in Figure 4 to navigate a maze using help from a custom AI search algorithm.

5. RELATED ACTIVITIES

After the establishment of the new program there were inquiries from prospective students who already had an undergraduate degree in computer science and wanted to attend ASU specifically to enroll in the game development courses. A certificate program was created as a result. A “Certificate of Computer Game Development” is awarded to any student completing the two PC development, handheld development and artificial intelligence courses. The certificate proved to be attractive to both post-graduate students and new undergraduates. It also provided another selling point for marketing purposes.

It was understood that marketing would be vital to the success of the new program and growth of the CS department. The department website was redesigned early to emphasize the game development program. Large posters advertising the program were printed and displayed on campus bulletin boards. Flyers were posted to advertise specific courses. One benefit of this was summer enrollment went up.

A three-day summer workshop was established to attract new students to the program. The workshop was advertised to prospective high school students. While the benefits of the summer workshop were smaller than expected in terms of the number of new students enrolling, the workshop generated an unexpected amount of publicity via newspaper and television.

6. RESULTS

The computer game development curriculum has been offered since the fall 2008 semester. The program has been instrumental in growing the CS program at ASU. The program grew from 71 to 213 majors over a four-year period since the introduction of the game development courses. A fall 2011 survey of 155 freshman and sophomore CS majors indicated 63 chose to attend ASU specifically to study computer game development and 73 planned to enter the game development industry after graduation. Freshman-to-sophomore retention of CS majors increased by 10% from academic year 2008-2009 to 2010-2011. 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. 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. The growth of the program mirrors experiences elsewhere where game development courses have increased enrollment [7, 9, 13, 17].

7. CONCLUSION

A new program in computer game development as part of a traditional degree in CS has been described. At the present time nearly half of CS majors at ASU have indicated they chose to attend ASU specifically to study computer game development. It remains a significant challenge to keep the game development curriculum up to date and produce graduates who are competitive in securing industry jobs.

8. 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] Bidarra, R., et. al. Bringing a Pioneer Games Project to the Next Level. 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, 11-15.
- [3] 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.
- [4] Distasio, J. and Way, T. Inclusive Computer Science Education Using a Ready-Made Computer Game Framework. In *Proceedings of the 12th SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '07)* (Dundee, Scotland, June 25-27. ACM Press, 2007, 116-120.
- [5] Duvall, S. Creating a Games Class: A Walkthrough. In *Proceedings of the 4th International Conference on Foundations of Digital Games (FDG -09)* (Orlando, Florida, April 26-30). ACM Press, 2009, 279-284.
- [6] Harris, J. Teaching Game Programming Using XNA: What Works and What Doesn't. *Journal of Computing Sciences in Colleges*. 27, 2 (Dec. 2011), 174-181.
- [7] Haungs, M. et. al. Improving First-Year Success and Retention Through Interest-Based CS0 Courses. In *Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education (SIGCSE '12)* (Raleigh, North Carolina, February 29-March 3, 2012). ACM Press, 2012, 589-594.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] McGill, M. Defining the Expectation Gap: A Comparison of Industry Needs and Existing Game Development Curriculum. In *Proceedings of the 4th International Conference on Foundations of Digital Games (FDG -09)* (Orlando, Florida, April 26-30). ACM Press, 2009, 129-136.
- [12] Novak, J. *Game Development Essentials: An Introduction*. Third Edition. Delmar Cengage Learning. 2011.
- [13] Parberry, I., Kaxemzadeh, M., and Roden, T. The Art and Science of Game Programming. In *Proceedings of the 2006 ACM Technical Symposium on Computer Science Education (SIGCSE '06)* (Houston, Texas, March 1-5, 2006). ACM Press, 2006, 510-514.
- [14] 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.
- [15] Zechner, M., and Green, R. *Beginning Android 4 Games Development*. Apress, 2011.
- [16] Zyda, M. Computer Science in the Conceptual Age. *Communications of the ACM*. 52, 12 (Dec. 2009), 66-72.
- [17] 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.