International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

# Operating A Computer Science Game Degree Program

GamePipe, Chris Swain

USC GamePipe Laboratory 941 W. 37th Pl, SAL 300 Los Angeles, CA 90089-0781 1-213-740-4494
vlacour@usc.edu

USC EA Game Innovation Lab School of Cinematic Arts Los Angeles, CA 90089 1-213-740-3317
cswain@cinema.usc.edu

## ABSTRACT

The USC Department of Computer Science is in its second year of operating its BS in Computer Science (Games) and MS in Computer Science (Game Development) degree programs. We have developed an interesting educational architecture inside of that degree program that allows the students to become strong game developers, strong computer scientists, strong programmers, strong systems developers, and facile with working in cross- disciplinary, collaborative groups. We believe that educating students in this fashion strengthens our department's ability to do cutting edge research in computer science as well as provide great graduates for the game industry. In this paper, we share our lessons learned, some detail on our courses and processes, as well as detail on our impact on recruitment and retention for the Computer Science undergraduate degree program.

.

**Keywords** Computer science game education

## INTRODUCTION

In a previous article about game education [1], we described two new degree programs we were in process of creating for the Computer Science Department at the University of Southern California – a BS in Computer Science (Games) and an MS in Computer Science (Game Development). There were multiple goals for these degree programs – one was to graduate students capable of being able to immediately enter and be productive in the game development field (game industry request), another was to stem the loss of students to the computer science field (Computer Science Chair request), and an additional goal was to create new research directions for the department (personal and Chair goal). We have been successful with these programs and



*GDCSE'08*, Feb. 28 – March 3, 2008, Celebrity Century Cruise Lines. Copyright 2008 ACM 978-1-60558-057-9/08/02...$5.00.

have some 245 student seats filled in our program in its fourth semester! We have created what we believe is a unique architecture for that degree program and have some interesting lessons learned. The degree program has enabled interesting new research directions. We built these degree programs in discussion with interested game developers as well as the Computer Science Department's regular faculty. Below we detail our degree programs so that we can discuss the architecture and lessons learned in close context.

## 1. BS IN COMPUTER SCIENCE (GAMES)

Figure 1 shows the breakdown of courses for the BS in Computer Science (Games) degree program. We created the BS degree with some 37 units of computer science and 42 units of game development courses. We created this degree not by watering down the regular BS in Computer Science but by modifying and strengthening that degree and replacing elective slots with the game development courses. It looks very much like a double major.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

## Changes to Computer Science Courses

### *1.1.1 Programming and Software Development*

When we began our discussions with game developers, we asked them their thoughts on how we should modify the computer science component of the degree. We were uniformly told that they wanted students that were strong programmers, good systems developers and cognizant of how to work in the game development process. We were told that our students needed to know C++ and be strong programmers in that language and that the game industry was not interested in hiring graduates who either learned Java as a first programming language or spent the majority of their programming time using Java rather than C++. We started by revising the first four courses in programming and software development: CSCI 101 – Fundamentals of Computer Programming, CSCI 102 – Data Structures, CSCI 105 – Object- Oriented Programming, and CSCI 201 – Principles of Software Development

We revised the 101 course to use C++ (from C) and made the laboratory assignments the development of a small game using DXFramework, a simple game development framework (that hides most of Microsoft's DirectX) written by the University of Michigan from Microsoft sponsorship [2]. We had great results from this, with students being highly motivated to extend and enhance the small semester-long game project (ping pong).

We left the CSCI 102 Data Structures course as it was as it was already in C++. We left CSCI-105 Object-Oriented Programming as it was, in Java, as we felt that knowledge of Java was important

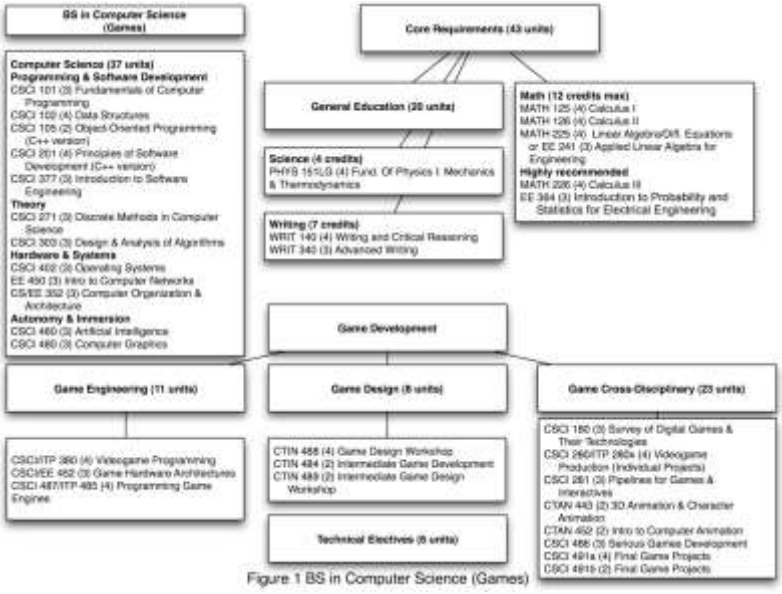

Figure 1 BS in Computer Science (Games)

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

for our students. We then created a special section of CSCI 201 Principles of Software Development that ran in C++ instead of Java. The project for the 201 course was to build a simple 2D networked game with the entire class participating as a group. The first offering of this course built a game entitled *Crosswinds*, a 4- player game based on a student-originated design [3].

### 1.2 Hardware & Systems

We replaced the four course sequence of EE courses on circuit and hardware design with two newly created courses – EE-352 Computer Organization and Architecture, and EE-452 Game Hardware Architectures. We wanted to make the 452 course a parallel programming course, where the students learned how to program the Playstation-3, the Xbox-360, and GPU hardware. To be able to get students to that point, we had to create a well- targeted 352 course that focused on a computer science understanding of computer organization and architecture (how to program it) rather than an electrical engineering understanding (how to design and make it). We were very lucky to find an adjunct with game console programming experience willing to design and create this two-course sequence. We were additionally lucky to have an EE Chair who understood and allowed us to create the courses our program required. The first offering of the 452 course is Spring 2008 but the laboratory exercises being developed now look exciting and quite promising.

## 2. GAME DEVELOPMENT COURSES

### 2.1 Game Engineering Courses

We created a two-course sequence of game engine programming, CSCI-380 Videogame Programming, and CSCI 487 Programming Game Engines. The 380 course shows the students how to build games using the Microsoft XNA Game Studio Express toolkit, with the 487 course showing the students how to utilize a game engine we have constructed out of the Ogre3D rendering engine [4]. The idea behind this two course sequence was to familiarize the students with building games from a high level in the first course, followed by a second course where they got their hands significantly dirty at the level at which they will probably work in industry. The only difficulty in this sequence is that the XNA environment requires the students to program in C# [5]. Computer science students familiar with programming in C++ and Java readily learn this new language with little difficulty.

### 2.2 Game Design Courses

When we began the Computer Science Games degree program, already running was an MFA in Interactive Media (IM) at the USC School of Cinematic Arts [6]. Inside of the IM program was a three-course sequence on game design – CTIN 488 Game Design Workshop, CTIN 484 Intermediate Game Development, and CTIN 489 Intermediate Game Design Workshop. This three- course sequence is an excellent sequence and teaches fundamentals of game design and experience design independent of technology via a process called "Playcentric Design" [7]. Before the CS Games program's existence, occasionally computer science students would take the CTIN sequence and help build software from the developed playable paper prototypes and design materials. We decided to place our computer science games students into this sequence, putting engineers into the same design course as designers. Our belief is that engineers that understand game design make better game engineers. The cross-disciplinary excursion is also good experience for their future careers in industry. We also find that many of our engineers that move through this sequence are happy at their ability to express themselves through game designs.

### 2.3 Game Cross-Disciplinary Courses

The courses listed as game cross-disciplinary in Figure 1 are courses where we expect students from multiple curricula, not just computer science. In fact, students from computer science, interactive media, animation, and fine arts co-exist in these courses by design. The Fine Arts students come from the 2D and 3D Game Art and Design Minor programs created to add art into the mix of on-campus game development degree programs. We worked closely with the Dean of Fine Arts to make that program happen.

**CSCI 180 Survey of Digital Games and Their Technologies** is a comprehensive survey of the history of videogame technology and game design with a focus on the language of game development through deconstruction of gameplay! The instructor for this course combines historical lecture, discussion, and student research presentation in combination with a lab that enables

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

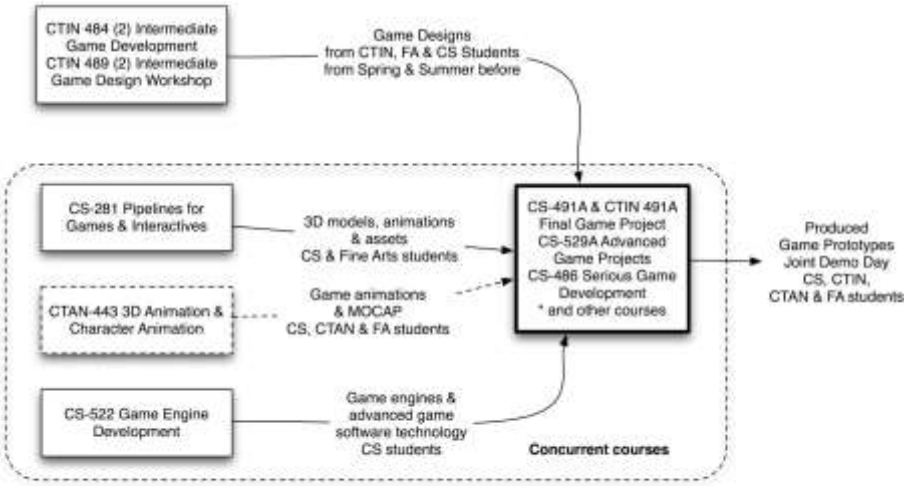


Figure 2 Cross-disciplinary coordination architecture for the Final Game Projects and Advanced Game Projects courses.

students to play old consoles and game emulations from early 70's to today. This process allows them to deconstruct the architecture of both the game's design and enabling technologies and understand how each genre and technology evolved together. This brings the students from their first semester and their relatively short personal experience with games up to a working knowledge of digital game history. Students take this in their first semester – they walk out saying things like "this is the best course ever" and are immediately hooked on staying in the degree program. Unlike many majors, we begin games courses in the first semester to solve the retention problem the computer science field has. We believe this works (see below discussion on recruitment and retention).

**CSCI 280 Videogame Production** is an introduction to all the pieces and parts of game development. Students in this course build individual games using GameMaker. Additionally, game industry speakers are brought in to further motivate the students. We find that visits by actual industry developers are hugely important in retention and often have the students saying things like "I hope to work for that company when I graduate". Let us know if you want to speak here!

**CSCI 281 Pipelines for Games and Interactives** teaches the students how to build 3D models, and animations, and how to do asset management for game development. A lead pipelines director for THQ Interactive teaches that course. Students in the 281 course build models, animations and assets for students in the 491 Final Game Projects course (Figure 2)!

**CTAN 443 3D Animation and Character Animation** teaches the students how to develop and animate game characters. Additionally the course provides animation and MOCAP services to students in the 491 Final Game Projects course (Figure 2).

**CTAN 452 Introduction to Computer Animation** teaches the fundamentals of animation using commercial tools.

**CSCI 486 Serious Games Development** teaches students how to build games for learning and/or training. The course goes beyond traditional design principles and creates features that enable learning of critical processes, procedures, and skills. All the projects are built for a real funding sponsor and provide the students the experience of working directly with a client weekly to achieve a goal. The course builds different 16-week prototypes

each semester in order to determine best practices for different pedagogical purposes. Thus far, we have created games in immunology, Russian political history, fish biology, disaster command, fire-fighting command, airplane assembly, and financial management. The best practices that are learned evolve the lexicon and structure of the course material. In essence, it is an ever-evolving course that is defining how to best build serious games.

## 3. CSCI/CTIN 491ab Final Game Projects & CSCI 529ab Advanced Game Projects

This course is the final course that brings together all of the skills learned by the students in a large game project course. The students get a sample of a real industry level experience where they must work in large multi-disciplinary teams over the course of 16 weeks to build a game. This is the most ambitious part of the degree program and is illustrated in Figure 2. Figure 2 shows how the students in the Final/Advanced Game Projects participate in a full development pipeline embedded in the degree program, receiving assets from the 281 and 443 courses and game engine technology from the 522 course. All of those courses run concurrently and have a mix of students from all

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

different disciplines in them. The 491/529 course is co-taught by CS and the School of Cinematic Arts, Interactive Media and has students from computer science, interactive media, animation and fine arts. Game designs are created and pitched by individual students during the Spring semester to an advisory board made up of faculty and industry mentors. Winning students are given the title Team Leader. These Team Leaders are then tasked to recruit team mates, and prototype their designs over the Summer. They then produce their designs in the 491/529 course that runs in the Fall with a full crew of fellow students. In Fall, students in the course are teamed up by preference and by skill to match the winning projects. The team leaders get invaluable experience directing and managing while still in school. The engineering students at this stage are working in open source engines in C++, DirectX and OpenGL. By this stage, they have been taught how to build games from the ground up, they know how to architect a proper framework, and know how to think critically about optimal architecture. When our students graduate, they are quite experienced in developing games in cross-disciplinary,

collaborative teams. The 491/529b course is used to further develop games built in the 491/529a course, getting those games ready for competition.

We do a joint Demo Day at the end of each semester and invite industry to view the developed games. A typical Demo Day has some 40 game industry visitors. Fall 2007 had representatives from Sony Online Entertainment, Sony Picture Imageworks, Sony Computer Entertainment of America, Electronic Arts Mobile, Electronic Arts Los Angeles, Activision, THQ Interactive, Digital Domain, Seven Studios, iSportGames, Lockheed Martin, Disney – Animation, Interactive and VR Studios, Emsense, Motorola Research Laboratories, Applied Minds, Northrup Grumman, Big Stage, Sandia National Laboratories, Konami, NaughtyDog, Google, Blizzard, and Creative Artists Agency.

### *4.1.1 Issues with Building This Architecture*

USC is architected such that each School (Engineering, Cinematic Arts, Fine Arts) is a revenue center that must have balanced books. The consequence is that Cinematic Arts students enroll in CTIN 491ab and Engineering students enroll in CSCI 491/529ab. Fine Arts students enroll in either course! The problem with revenue centric university financial architectures is that we have a course with 25 students in it, half of whom are from interactive media, and the other half from computer science and at the Assistant Dean level we find ourselves in conversations that indicate that the relevant Assistant Dean only wants to provide TAs for the students from their School. This is a common issue with cross-disciplinary degree programs, one that will be worked out over time with each change of the Assistant Dean of each School.

### *4.1.2 Experiences with 491/529ab*

We ran the first offering of this co-taught course Fall 2007. On the first day, the interactive media students sat all in a group at the back of the room, while the computer scientists and fine arts students sat separately at the main lab tables. This lasted through team selection. Once game teams were selected for the winning projects, everyone got down to work in teams (teams of 14 to 19 students, by the way!). Three games were greenlit in Spring 2007 and built in Fall 2007 and Spring 2008. Each was chosen based on the merit of the design and strength of the team leaders. And each was chosen based on original thinking about play mechanics.

Each game is ambitious and original from a game system design perspective. This is because we believe time in university is time to stretch for new ideas and that student teams can differentiate themselves from industry teams in this realm. The three games built were - *Ragnarokk*, a heavy metal themed adventure game in which the player uses a *Guitar Hero* guitar as input device. and battles his way through the demons of Norse hell using power cords and metal riffs as his weapons; *Errantry*, a medieval rap battle in which players gesture with Wii-motes to act out tall tales of chivalry, romance, piety, and heroism before a on-screen royal court. The court is amused or bored or annoyed based on the story-telling skills of the competing players; and the

*Misadventures of PB Winterbottom*, an adventure set in the milieu of silent film era scenarios. The game has unique time-based play mechanics that allow a player to record instances of the play character and then play back those instances in interaction with the game player's current character. The Winterbottom team has been invited to show their game to many publishers including Electronic Arts and Sony, and, additionally, is a finalist in the 2008 IGF Student Showcase award.

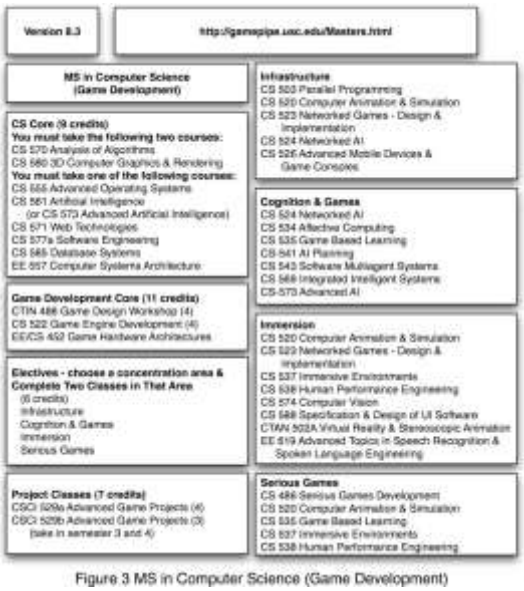

Figure 3 MS in Computer Science (Game Development)

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

## RECRUITMENT AND RETENTION

Part of the reason for creating the BS in Computer Science (Games) degree program was as an attempt to increase the number of undergraduates into the Computer Science BS program. In the Fall of 2006, there were only 223 applicants to the BS in Computer Science program, 57 were admitted, with 36 actually enrolling. In Fall 2007, there were 383 total applicants to the BS in Computer Science programs, 164 of whom applied to the BS in Computer Science (Games). 49 out of the 164 applicants to the BS in Computer Science (Games) were admitted and 24 new undergrads enrolled. In Fall 2007, the number of undergraduates into the BS in Computer Science program grew to 48, half of who were Games. So Games doubled the incoming undergraduate enrollment.

The freshmen year return rates for students who began in Computer Science versus Computer Science (Games) in Fall 2006 and who returned to the same major as sophomores in Fall 2007 are as follows: Computer Science: 81.2% freshmen year return rate, and Computer Science (Games): 77.8% freshmen year return rate. It is far too early to compare success in retaining students within Computer Science for the Games program at this time, since Fall 2006 was the first year students could apply directly to the program as freshmen. While freshmen year return rates can be key indicators of future retention and graduation rates, it will require additional data to reach any substantial conclusions about the programs ability to retain Computer Science students overall. These numbers are freshmen cohort numbers, which is how the university overall reports retention rates, tracking freshmen from entrance into a single program through graduation from that same program. It does not take into consideration students who moved from one major to the next within the Viterbi School of Engineering.

A comparison of GPA averages was made between straight CS
majors and Games majors. For 2006 – 2007, the Games majors GPA average for all students was 3.21, with the GPA for first year students 3.16. The regular Computer Science GPA average for all students was 3.14, and GPA average for first year students was 2.94. So students in the BS in Computer Science (Games) performed somewhat better, on average, than regular Computer Science students despite there being a higher percentage of applicants to the Games program not admitted.

## MS in COMPUTER SCIENCE (GAME DEVELOPMENT)

The game industry realizes that it is reaching the point of diminishing returns with respect to how it technologically currently builds games. There is more and more the realization that a greater investment needs to be made in technology and personnel. Recognizing that, we built an MS in Computer Science (Game Development) that is focused on building technology for the future of interactive games and then prototyping and testing that technology in our Final Game Projects, Advanced Games Projects sequence. The MS program has a graduate level Computer Science core, a Game Development core, and the notion of concentration areas of study. Concentration areas include infrastructure, cognition and games, immersion, and serious games. We have a well-developed set of new courses for these concentration areas and consider those areas as directions for research and PhD topics in Computer Science.

## NEW RESEARCH DIRECTIONS ENABLED BY THE GAMES DEGREE PROGRAM

It has taken us three years to get to our fourth semester of degree program operation and the question remains as to the ultimate dollar volume of research we will eventually produce. Most of us closely associated with the program now know that we have the full game development process embedded in the degree and see great potentials with that engine of production. We have the ability to experiment with serious games and their evaluation and have had many interesting funded efforts in that domain. We have been funded to look at entertainment games possible with next generation cell phones. We have begun to create a research program on human aware computing focused around what new input devices and games can we build with currently available low-cost sensors. We have developed a research agenda on emotion-cognizant games, something closely aligned with our focus on sensors. Our sensor and serious games focus has lead towards a potential effort on games for fitness and preventive health and we are in discussions with a large health insurance company for funding in that area. We are building an MMOG (massively multiplayer online game) infrastructure as a platform for a number of game experiments. We have interest in tying that MMOG infrastructure to real-time sensor suites to Formula-1 racecar telemetry. We additionally are experimenting with what we call model-embedded MMOGs, MMOGs that have the capability to have embedded agent-based models that track game play in the MMOG and generate reports and response based on observed game play. There is quite rich and interesting research to perform once one has a game development and technology pipeline built. We cannot imagine actually performing games research without having a pipeline of students in game development degree programs. We see a great games R&D future!

## ACKNOWLEDGMENTS

We wish to thank Gerard Medioni, the Computer Science Department Chair who strongly supported the creation of this new academic program. Gerard let us create sixteen new Computer Science and two EE courses over an 18 month period and let us focus on that rather than proposal writing. Dean Yannis Yortsos of the Viterbi School of Engineering provided us laboratory support in the newest building on campus and helped us create a great program for our students. Victor Lacour contributed

enormously to the creation of this degree program and has operated both the Serious game research course structure and Final Game Projects sequence from the start. Chris Swain of the School of Cinematic Arts Interactive Media Program provided guidance on how to utilize the game design workshop sequence in our degree program. Victor Lacour and Chris Swain have now bravely created and operated

the first joint Final Game Projects course sequence between the School of Cinematic Arts and the Viterbi School of Engineering. Victor has been great at helping stand up this program and throwing in his heart and soul to the students and the program. Steve Schrader made the paperwork happen for the degree program and he is an irreplaceable part of the Computer Science Department. Massoud Ghyam was a tremendous help in revising the computer programming and software development sequence of the undergraduate degree program. David Harr helped create no fewer than 5 of the syllabi for this program and developed the entire set of course notes for the EE-452 Game Hardware Architectures sequence (with Nathan Greenfield and Devin Rosen). Louise Yates, Associate Dean for Admissions and Students Affairs, has been outstanding in handling admissions so that we get excellent students and has been wonderful in providing statistical support to show how the BS in Computer Science (Games) degree program has helped to strengthen and grow the Computer Science Department. Dean Ruth Weisberg of the School of Fine Arts was wonderfully receptive to the idea of creating a Game Art and Design minor program to complement her schools' degrees. Ann Page of Fine Arts carried the ball in designing that Fine Arts minor program. We had a host of graduate and undergraduate students that made the USC GamePipe Laboratory happen even when there was no financial compensation or reward – Pamela Fox, Dhruv Thukral, Sumeet Jakatdar, Fred Zyda, Danny Parks, Donna Djordjevich, Erin Reynolds, Rob Cheng, Nathan Greenfield, Devin Rosen, and Lily Cheng. We consider them the founding students for this laboratory and academic program. We had wonderful contributions to our degree program from Scott Easley, and Jose Villeta of THQ Interactive; Steve Seabolt, Colleen McCreary, and Bing Gordon of Electronic Arts; John Bojorquez and Pat Griffith of Activision; Mike Lee of Emsense, and Anthony Borquez of Konami (formerly USC ITP). We had donations from Motorola Research Laboratories, Microsoft Research (John Nordlinger), Lockheed Martin, Sandia National Laboratories, THQ Interactive, Activision, EA Mobile, and Alon Carmel that keep our program operating.

## REFERENCES

Michael Zyda "Educating the Next Generation of Game Developers," IEEE Computer, June 2006, pp. 30-34.
DXFramework web site http://dxframework.org/
Crosswinds http://viterbi.usc.edu/news/news/2007/games- students-play.htm
Ogre3d web page http://www.ogre3d.org/
Microsoft XNA Developers web page http://msdn2.microsoft.com/en-us/xna/default.aspx
USC School of Cinematic Arts Interactive Media Program web site http://interactive.usc.edu/
Fullerton, Tracy, "Play-Centric Games Education," IEEE Computer, vol. 39, no. 6, pp. 36-42, Jun., 2006.
USC GamePipe Laboratory http://gamepipe.usc.edu