Design Issues for Undergraduate Game-Oriented Degrees

Michael Mateas
Dept. of Computer Science
Univ. of California, Santa Cruz
Santa Cruz, CA 95064
michaelm@cs.ucsc.edu

Jim Whitehead
Dept. of Computer Science
Univ. of California, Santa Cruz
Santa Cruz, CA 95064
ejw@cs.ucsc.edu

ABSTRACT
The paper describes the most significant design issues concerning the development of game-oriented undergraduate degree programs. These issues fall into two broad categories, those that concern the organization of the degree, including its framing and naming, as well as issues concerning the degree's content. Content issues include the amount of computer science content, use of digital media content, game design and game projects, ethics requirements, breadth requirements, and the impact game degree programs can have on the existing computer science curriculum.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education

Keywords
Curriculum design, computer game degree, game education

1. INTRODUCTION
Motivated by several factors, many universities are considering or implementing degree programs at the graduate and undergraduate level focused on computer games. One reason these programs are being instituted is the goal of increased enrollment, leveraging the widespread cultural interest in computer games into student involvement in game-oriented degree programs. Computer games are also intellectually exciting, sitting at the nexus of computer science, film, digital media, theater, art, literature, economics, and social science, thereby offering new opportunities for dramatic expression, nonlinear storytelling, social commentary, and interactive education. This intellectual interest in the expressive and interactive potential of computer games is also driving the creation of degree programs.

Like any complex design activity, the creation of a new degree program involves the simultaneous consideration of a wide range of issues. Some of these issues are universal across all kinds of degree program creation—gathering resources, building political support—and some are distinctive for game-oriented programs. The authors of this paper have been involved in the design of game-oriented undergraduate degree programs at the University of California, Santa Cruz (Bachelor of Science in Computer Science: Computer Game Design) and the Georgia Institute of Technology (Bachelor of Science in Computational Media). Within, we discuss many of the core design issues involved in the development of new undergraduate degree programs focused on computer games, drawing upon our personal experience, as well as other notable degree programs. While much of this discussion is useful for the creation of any game-oriented degree program, due to the authors' backgrounds the discussion will be more focused on technically focused and interdisciplinary degrees, with less discussion of art-focused game degrees.

The sections below begin by discussing issues concerning the framing and naming of undergraduate game-oriented degree programs (Section 2). Following is an exploration of a series of curricular design issues (Section 3) that affect the academic content of game-oriented degree programs.

2. CURRICULUM ORGANIZATION
The following two sections present issues concerning the overall emphasis of a gaming oriented degree program, and the naming of such programs.

2.1 Degree Emphasis
Perhaps the most important issue in designing a computer game degree program is determining its emphasis. Reflecting the broad and interdisciplinary nature of professional computer game creation, degree programs have emphases along a continuum from very art focused programs to very technically focused programs, with a number of broadly interdisciplinary programs exploring different niches in between. Undergraduate computer game degree programs in the United States tend to fall into one of three categories:

Art focused:

These programs emphasize the artistic and graphic design aspects of computer games, with only a small number of programming courses. Students graduating from these programs are well suited to join the art track of a computer game company. Example: BA in Game Art and Design, Art Institute Online.

Evenly interdisciplinary:

These programs have strong computer science foundations, but do not go into computer science topics with the same depth as technology focused programs. Instead, they offer a broader mix of courses on game design topics, and tend to emphasize game design. These degree programs can also provide students some degree of choice as to whether to focus on technology or arts. Example: Georgia Tech: BS in Computational Media; Worcester Polytechnic Institute: Interactive Media and Game Development.

Technology focused:

These programs are strong computer science degrees, with additional courses adding depth in computer game design. Students graduating from these programs can enter the technical track in computer game companies. As compared to the other kinds of programs, the technically focused programs provide greater depth in computer science topics. Examples: UC Santa Cruz BS Computer Science: Computer Game Design, Univ. of Southern California, BS Computer Science (Games); Univ. of Denver, BS Game Development and Animation.

The difference between the evenly interdisciplinary and technically focused programs can be subtle. One way of thinking about the difference is that the evenly interdisciplinary programs emphasize technology-inflected design, while the technically focused programs have curricula that emphasize design-inflected technology.

The choice of emphasis will often be strongly determined by the kind of organization developing the degree. An art-focused department or school, such as the Art Institute Online, will naturally develop an art-focused degree program. In a similar vein, computer science departments will tend towards technically focused degree programs, since it permits building upon their existing strength.

Interdisciplinary programs are more challenging to attempt, since they require more new courses to be developed. Most colleges and universities in the United States do not have dedicated game studies departments or research groups, and hence do not have a ready base of game design or game studies courses to draw upon when creating a new major. Prospective undergraduate students are typically very motivated by degree programs that permit them to engage in game design. As a result, evenly interdisciplinary programs tend to better match this desire.

One concern when developing programs is the career prospects for students after graduation. For students in art focused or technology focused programs, there is a straightforward story to tell, with art focused students going into the art track of game studios, and technology focused students going into the game development track. Additionally, art-focused students could perform a wide range of digital art work, and technology focused students are sufficiently well trained that they can take almost any kind of information technology job. With evenly interdisciplinary programs, the story is more complex. Many game development studios are unlikely to hire freshly graduated students to perform game design, since this is typically a senior role. However, a job as a level designer is certainly a reasonable expectation. Additionally, students in the interdisciplinary programs do receive a solid technical background, though it is not as deep as the technically focused degrees. As a result, they are likely qualified for some game development tasks. Since these degree programs are very new, they do not yet have enough graduates to fully understand their career trajectories.

While many game-oriented degree programs focus on vocational outcomes for their students, it is important for programs to ensure their students are adequately prepared to embark on graduate study. There are an increasing number of graduate programs focused on computer games. As well, students who complete a game-oriented degree in their undergraduate studies may very well choose a different type of degree for their graduate studies.

2.2 Naming Game-Oriented Programs

At present there is no consensus on what title is granted to students after completing a game-oriented degree program. An extensive listing of game degree program names can be found in Davidson [Davi05], which is current as of 2005. We see a few emerging trends in program names.

Game Development

Several programs have game development in their title. This title is directly descriptive, since students are taught how to develop computer games. The title gives the impression of being more vocational than a science or engineering degree, similar to the term "programming". Examples:

Game Development (FullSail)
Electronic Game and Interactive Development (Champlain College)

Computer Science plus

Technically focused programs that are emerging out of computer science departments are choosing to use the BS Computer Science title, and then tack on additional descriptive terms. These titles carry the gravitas of the existing BSCS, itself an upstart scant decades ago. These titles fuzz the issue of whether these are specialized computer science degrees, or completely novel degrees. Examples:

BS Computer Science (Games) (Univ. of Southern California)
BS Computer Science: Computer Game Design (UC Santa Cruz)

Media oriented

Titles using the term "media" recognize that computer games have created a new form of computational media. This ties computer games into ongoing work in digital media, and creates a broader intellectual space for the degree program, as it can expand beyond an initial focus on computer games into different forms of computer afforded media. The drawback is that the general population currently does not have a strong understanding of interactive/digital/computational media, and does not necessarily equate that with the creation of computer games. Examples:

Computational Media (Georgia Institute of Technology)
Interactive Media and Game Development (Worcester Polytechnic Institute)

Simulation oriented

Some degree titles focus on the virtual world simulation aspect of creating computer games. This is an interesting move, evoking aspects of Simon's *Sciences of the Artificial* [Simo68] or Gelernter's *Mirror Worlds* [Gele91], which emphasize the simulation science aspect of computer science. These degree titles also broaden the programs, opening up the possibility of covering more simulation-focused content in the future. Like degrees with media in their title, it is not clear that prospective undergraduate students equate simulation with computer games. Examples:

Real Time Interactive Simulation (DigiPen)
Digital Simulation and Gaming Engineering Technology (Shawnee State University)

3. CURRICULUM CONTENT

3.1 Computer Science Content

Technically focused programs have to strike a balance between how many traditional CS courses are required, which provide students with a firm CS foundation, vs. how many game specific courses to offer (both design and technology), which give students depth in games. This is an opportune moment for CS programs to create game specializations, as many computer science departments are already reconsidering their core curriculum. Significant downturns in CS enrollments have caused CS departments to question why current CS curricula fail to attract students [Denn05].

One way of characterizing the resulting curricular debates is in terms of a big-kernel vs. small-kernel understandings of what it means to be a computer scientist. Big-kernel approaches assume that, in order to be a computer scientist, all students must acquire a large shared body of knowledge consisting of "core" CS topics before taking more specialized courses. The large "core" generally consists of the union of the research specialties of the department faculty (everyone thinks their specialty is something everyone should know), skewed towards specialties that were already well established in the 1970s and 1980s. Thus, for example, systems, compiler and theory topics may comprise a large portion of big-kernel curricula, while software engineering, human-computer interaction and ubiquitous computing will not, precisely because the former were well-established subfields before the later. Obviously there are only so many topics that can be fit into a four year degree. In a union-of-research-interests big-kernel model, early subfields will naturally dominate more recent subfields for the simple reason that, in a curriculum that grows by accretion, the "core" is already full before the arrival of more recent subfields. Small-kernel approaches acknowledge that CS has grown into a large and unruly confederation of often only loosely related research areas. The core of topics every computer scientist should know is therefore small, with students branching into sub-areas of computer science more quickly, and consequently more choice available earlier in the curriculum. A number of highly-ranked programs are making the move towards small-kernel CS, the most publicized being the Georgia Institute of Technology's Threads program [Furs06].

Within this climate, the creation of a technically-focused game specialization can energize the departmental discussion and debate around curriculum reform. A game degree puts pressure on CS departments to offer more specialized classes, and to make those classes available earlier in the curriculum. The high-level of interest in such programs among entering freshman guarantees that game specializations will quickly have a large number of students, further putting pressure to reduce the number of "core" courses, as the number of students in the game specialization equals or exceeds the number of students in the traditional major. Curriculum changes necessary to accommodate a game degree have the happy side-effect of setting up departments to be able to quickly establish additional specializations in the future, making the department more nimble in its ability to address future technological and social changes.

While the argument above has counterpoised big-kernel CS vs. small-kernel plus specialization, big-kernel CS is actually a specialization itself. Since the sub-disciplines that were established early (and which dominate big-kernel curricula) tend to focus on computers and computing as ends in themselves, while more recent sub-disciplines tend to connect computing to specific social and cultural contexts, big-kernel CS can be characterized as the specialization focusing on computing for computers, while more recent sub-disciplines can be characterized as computing for people. Interestingly, this directly relates to the much decried lack of women in computer science programs. The seminal, multi-year Women in Computer Science study run at Carnegie Mellon University [Marg03] found that women attracted to computing, more so than men, tend to care about the context and connections of computing to other arenas; it is these connections that make computer science meaningful to them [Marg99]. Thus, traditional big-kernel CS programs, with their focus on "computing for computers," are almost perfectly designed to chase women from the program. Games, as the emerging and potentially dominant expressive medium of the 21st century, are poised to connect to every facet of cultural life, from politics to public policy, from education to entertainment. The creation of a game degree, and the resulting structural changes within the CS department that ease the creation of future specializations, create opportunities for students to connect computer science to other disciplines, and to broader social and cultural concerns, addressing the female retention issues that have plagued traditional computing curricula.

3.2 Digital Media Theory

Any game program has to decide how much of the design and theory side of the curriculum should be explicitly focused on games vs. on situating games within the broader framework of media studies, particularly looking at digital and interactive media. While at first blush it may seem appropriate for game degree programs to focus entirely on game design and theory, so as to maximize the students' depth of knowledge in games, we feel that this would be serious mistake. This is for a quite simple reason: what it means for something to be a "game" is not stable. An over-focus on the design elements, approaches, and rules of thumb that are used to create what is currently understood as a game will leave students unprepared to track the ongoing evolution of the medium as the very definition of game continues to change and morph.

As an example of the fluidity of the concept of "game", in [Ward05] Wardrip-Fruin examines a number of interactive media work that have been explicitly declared "not games" by their creators, and yet have strong game-like qualities that, in some cases, have resulted in the definition of "game" broadening to accept the new type of interactive experience. For example, the website for Electronic Arts declares *The Sims* to be "The #1 best selling game of all time", yet both academic commentary as well as the commercial game press have described *The Sims* as more of a software toy than a game. Creator Will Wright has also referred to his sim games (from *Sim City* on) as software toys. *The Sims* has no winning condition, no score, and no explicit game goals. Yet now it is fairly commonplace to refer to such open-ended "playable simulations" as games; the concept of "game" has expanded to include such experiences. As another example, alternative reality games (ARGs) explicitly smudge the boundaries of the magic circle, creating conspiracies for groups of players to uncover through a mixture of real-world and web-based sleuthing. ARGs purposefully obscure the distinction between fiction and fact, mixing players' everyday lives with the game world. Such games often explicitly declare "this is not a game" as a way to help players maintain a belief that the events in the game are really happening. But ARGs, as the name implies, are now considered a standard game genre; the ever-plastic concept of "game" has again expanded to encompass this new form.

In order to prepare students to not only participate, but hopefully play a leadership role in this continuing innovation in the nature of games, they must not only understand the current state of game design, but how games participate in the broader media ecology. A firm foundation in the history of interactive media prepares students to understand how new media forms come into being, and how communities of practice develop conventions that simultaneously realize and constrain the technical possibilities of the medium. By investigating media phenomena such as networked communities, mobile communications and geo-

positioning technologies, ambient media, and electronic literature, students become familiar with how different combinations of social and technical infrastructures function as expressive forms. A broad foundation in digital media, in addition to depth courses in game design, prepares students to create the games of today as well as invent the new game genres of tomorrow.

3.3 Game Design Content

Even technology-focused game design programs must have significant game-design content in the curriculum. The game industry as a whole is moving towards interdisciplinary work practices, as exemplified by the organization of the team working on *Spore*, Will Wright's "Sim Everything" game. As Wright has described in his Game Developers Conference talks on *Spore*, designers and programmers are tightly coupled in the design process, with designers knowing enough programming to write lightweight prototypes that demonstrate design concepts, and programmers knowing enough design to iterate tightly with the designer as they write robust production code. The UC Santa Cruz degree program aims to create design literate programmers who have the CS chops to design the architectures and write the engines necessary to realize the game, while being first-class participants in the design process.

The most basic element of game design is the notion of rule systems, formal systems that govern player interaction and the evolution of game state. Students learn how to think about rule systems, and learn standard rule patterns found in many games, such as the rock-paper-scissors pattern governing relationships between units in RTSs and spells and equipment in RPGs. Additionally, students learn how to use rapid prototyping, including paper and pencil prototyping, to understand emergent interactions between rules. Concepts from psychology, such as flow, cognitive understandings of problem solving, and affective response, prepare students to think about player response during game playing. Concepts from sociology provide students with tools to think about community design for MMOs. Students learn about the game design process, including the typical roles found on game teams, milestones, resource management, and lessons from design science. In addition to focusing on design for the interactive entertainment industry, students learn about serious games, including political and policy games, training games, and games for health. Finally, students learn about emerging game design topics, including alternative reality gaming, and mobile and casual gaming,

3.4 Project Based Learning (Game Projects)

One area on which there is rough consensus is the need for at least one substantial game development project in the curriculum. Typically this project is viewed as a final year capstone, in which students work as members of a team to create a large computer game. Like most project-based learning activities, the senior game project is intended to allow students to synthesize knowledge developed in the classroom by applying it to the construction of a computer game. Game projects also allow students to experience all aspects of a typical game development lifecycle, from initial game concept through coding and art asset creation, to testing and deployment. Additionally, the project provides students with a completed game they can add to their portfolio and demonstrate to prospective employers or graduate schools. As an example of one such project, in the UC Santa Cruz degree program, students take the Game Design Studio sequence (3 quarters) during their entire senior year. Many other programs have similar project sequences.

Game project classes can also be used to energize students early in the program. Incoming students are very eager to begin learning the primary focus of their chosen degree in their first year. However, this is the year when most degree programs focus on background courses, such as an introduction to programming, calculus, physics, and institution-specific general education requirements. This leads to a mismatch between student interest and excitement to engage in game design and development, and the coursework they are required to take. One way to address this is to have students engage in game project development early in their curriculum. The DigiPen BS in Real-Time Interactive Simulation (RTIS) degree program is an excellent example, with students taking a game project class in each semester, starting the second semester of their freshman year. Another approach taken by the UC Santa Cruz program is to have a freshman year game project course, called the Game Design Experience. This course combines lecture material introducing game design with a small team development project.

3.5 Ethics Content

There are several negative aspects of computer games that appear in mainstream media. Some games have graphic violence, and there are concerns that repeated exposure to this violence leads to more aggressive behavior. Other games, especially massively multiplayer online games, can be extremely engaging, resulting in extensive gameplay hours that can lead to loss of relationships or job loss. These concerns are a mixture of both real issues and general societal concern with any new and powerful medium that is adopted by youth. Addressing these concerns is an aspect of designing a game-oriented degree program, since they will likely be raised by faculty examining any new proposed curriculum, and may affect the degree of enthusiasm among faculty members towards the degree. In addition to this internal consensus-building concern, the parents of prospective students, who are often quizzically amused by the game-playing activities of their children, are happy to hear that ethical issues, rather than being ignored or side-stepped, are dealt with squarely within the curriculum. Finally, for the students themselves, an ethics course ensures that students are familiar with some of the societal debates swirling around games.

One way to address these concerns is to introduce an ethics requirement. An ethics course provides students with a framework for critically examining a wide range of ethical issues, including those that are associated with computer games. We do not anticipate that students taking an ethics course will, for example, immediately stop using violence as an interaction mechanism in their games. Depictions of violence may, in fact, be an entirely appropriate design choice for a particular game. The goal is instead to ensure that inclusion of violence, or creation of extremely compelling gameplay, will be done after considering the ethical implications, and with full knowledge of any tradeoffs that are being made. Furthermore, if a new game ends up creating unforeseen negative issues, students will have the ability to reason about the ethical implications.

3.6 Breadth Content

One tension in creating a game-oriented degree program is the need to provide sufficient depth in either the artistic, design, or game development areas so that graduating students are competitive in applying for game development jobs, while at the same time providing a broad education across a wide range of topics so that students can draw upon this rich background

knowledge when designing games. Job advertisements for computer game professionals focus on specific skills, such as specific programming languages, graphics libraries, AI techniques, etc, and say nothing about broad backgrounds. In contrast, game designer Chris Crawford lists a very broad set of books, movies, and periodicals in his, "Education of a Game Designer." (Chapter 9, in [Craw03]).

One solution to this tension is to require students to take some number of breadth courses that are not directly related to computer games. Colleges and universities with a wide range of degree programs tend to have a variety of courses covering history, literature, psychology, music, film, theater, and so on, which students can draw upon for this breadth. Most 4 year colleges and universities have some kind of formal breadth requirement for all students. At UC Santa Cruz, 80 credits out of a 180 credit degree program must satisfy campus General Education requirements, which are split across a wide range of subjects, and include a writing intensive course. The UC Santa Cruz game degree has a series of "Art and Social Foundation" requirements in which students select three from a list of five categories: Art, Film, Theater Arts, Music, and Economics. In each category, students select a single course from a list of approved courses for the major. In this way, students have part of their general education requirements targeted towards courses of particular relevance for game design, while still giving students substantial freedom in picking additional breadth courses.

More specialized institutions need to explicitly create courses to provide some of this depth. FullSail has required courses on Historical Archetypes and Mythology and Media and Society, and DigiPen has courses on Mythology for Game Designers and Society and Technology. One of the clear tradeoffs is that specialized institutions offer fewer breadth courses, thereby allowing them to offer more game design courses, while traditional colleges and universities can draw upon a broad array of breadth courses, but with fewer slots available for focused game design courses. The lack of required breadth courses also permits more compressed schedules, with FullSail's program taking just three years.

3.7 Impact of Computer Game Topics on Existing Curricula

The creation of a game degree program is not only an opportunity to add new game-specific courses to the curriculum, but also an opportunity to add gaming related content to existing CS courses. Games are a great student motivator and, due to the breadth of topics CS topics that play a role in games, can be incorporated into almost any class in the CS curriculum. As an example of organizing a CS course around games, one of us (Mateas) developed a version of the sophomore introduction to hardware systems course at Georgia Tech organized around the Game Boy Advance (GBA) portable game system.

The Media Device Architectures course covers the basics of binary representation, memory and processor architecture, memory-mapped I/O, interrupts, and low-level C programming (bit masking, bit shifting, etc.), but using the GBA as the reference architecture. Though a fairly contemporary game system, the GBA is organized like many classic consoles, with no OS, a memory-mapped architecture for I/O (requiring explicit memory manipulation to, say, draw on the screen), and interrupt-based input handling. Thus, students can gain real experience coding "close to the silicon", while having the motivation of programming a real game console. Further, unlike many introductory hardware systems courses which make use of an artificial simplified reference architecture for teaching assembly language programming, GBA programmers have a real motivation to embed assembly blocks in their C code, as assembly may be needed to optimize critical paths within the game code, or to write interrupt handlers that can require the use of opcodes that aren't generated by the C compiler. Finally, GBAs can be networked together with serial cables, giving students the opportunity to play with simple network protocols.

4. CONCLUSION

This paper has presented a series of design issues that confront any group seeking to develop a new game-oriented undergraduate degree program. Our goal in presenting these issues is to make these design considerations explicit, so that future designers can more easily create new, compelling degree programs by understanding the issues involved in the creation of existing ones.

References

[Craw03] Chris Crawford, *Chris Crawford on Game Design*, New Riders, 2003.

[Denn05] Peter Denning and Andrew McGettrick. Recentering Computer Science. *Comm. ACM*, 48 (11), Nov. 2005, 15-19.

[Davi05] Drew Davidson, "Games by Degrees: Playing with Programs," *On The Horizon. Special Issue. Second Generation E-Learning Part 2: Serious Media*. 13(2), 2005, pp. 70-74.

[Furs06] Merrick Furst and Richard DeMillo. Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment, The Georgia Institute of Technology, 2006, http://www.cc.gatech.edu/images/pdfs/threads_whitepaper.pdf.

[Gele91] David Gelernter, *Mirror Worlds*, Oxford University Press, 1991.

[Marg99] Jane Margolis, Allan Fisher and Faye Miller. Caring about connections: gender and computing. *IEEE Technology and Society Magazine*, 18 (4), 1999, 13-20.

[Marg03] Jane Margolis and Allan Fisher. *Unlocking the Clubhouse: Women in Computing*. MIT Press, 2003.

[Simo68] Herbert A. Simon, *The Sciences of the Artificial*, MIT Press, 1968.

[Ward05] Noah Wardrip-Fruin. Playable Media and Textual Instruments. *Dichtung Digital: Journal on Digital Aesthetics* (1). 2005. http://www.dichtung-digital.com/2005/1/Wardrip-Fruin