Game Design & Programming Concentration Within the Computer Science Curriculum

Ron Coleman, Mary Krembs, Alan Labouseur, Jim Weir
Marist College
School of Computer Science and Mathematics
Poughkeepsie, New York 12601-1387
1 845-575-3000
{Ron.Coleman, Mary.Krembs, Alan.Labouseur, Jim.Weir}@Marist.edu

ABSTRACT
This paper describes initiatives at Marist College to develop a Game Concentration in the undergraduate Computer Science curriculum. These initiatives contemplate recommendations for existing courses as well as adoption of new courses. We also consider activities of the Association of Computing Machinery (ACM) in this area and opportunities for students beyond the classroom.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computers and Information Science Education – *Curriculum.*
K.8.0 [**Personal Computing**]: General – *Games.*
D.1.m [**Software**]: Programming Techniques – *Miscellaneous*

General Terms: Algorithms, Design, and Experimentation.

Keywords: Curricular initiative, Game Programming

1. INTRODUCTION
Experience tells us that students are keenly interested in videogames. For many, game systems like the *GameBoy*, *PlayStation*, *Xbox*, etc. represent not only a student's initial experience of computers and computing, in all likelihood these students may well have played videogames before even using the Internet. Indeed, videogames are the initial draw to technology for a number of students in the first place.

Students are also drawn to cutting-edge and emerging technologies. They want to be "where the action is", as it were, and videogames are all about action and interaction. Advances in hardware and software have made game programming accessible to virtually anyone, including teenagers, with even modest programming skills. [6]

Yet videogames embody a considerable amount of computer science, aspects of which are not typically covered by computer science curricula. Consequently, as we explain further below, students have had to look outside traditional academic settings to formally study game development.

The paper focuses on recommendations for existing courses as well as adoption of new ones as we perceive them at Marist. Given the similarity of programs of study across different institutions, we believe these initiatives may potentially have applicability for many other institutions as well. On the other hand, this paper does not consider game development in broad artistic, cognitive, and socio-communicative terms. For instance, while there are some artistic considerations in our analysis, we are mainly concerned with issues related to computer science, physics, and mathematical coursework—the "hard core" of game development, as it were. Thus we have no specific recommendations for coordinating with or influencing the Art, English, or Psychology departments. While an initiative as such may be possible, perhaps even desirable, it is beyond our scope here.

2. MOTIVATION
A review of computing history suggests a curious and distinctive divergence between conventional and videogame applications.

Conventional applications generally emphasize managing and supporting organizational operations and objectives. These include business functions like accounting, sales, inventory tracking, student registrations, etc. These applications are supported by systems programming and employ best practices for optimization, searching and sorting, messaging, and the like, which are fundamental computer science issues usually taught in traditional academic settings.

Videogame applications, on the other hand, generally emphasize entertainment objectives.[†] Once relegated to hulking consoles in dimly lit juke bars and noisy bowling alleys, videogames have truly come into their own on home consoles, on cell phones and personal data assistants (PDAs), on handheld game machines like the *GameBoyAdvance*®, and – thanks to the Internet – online with multiple players scattered around the globe.

While videogames employ fundamental computer science, they also work in real-time and bypass the operating/windowing system, directly accessing hardware devices like video processor

SIGCSE'05, February 23-27, 2005, St. Louis, Missouri, USA.

[†] Many videogames emphasize educational and training objectives (e.g., flight simulators) that are also relevant for our purposes.

units, sound cards, analog, digital, and force feedback controllers—subjects not typically covered by mainstream computer science curricula. Games furthermore incorporate physics simulations, mathematical models, and game-specific principles that distinguish them from conventional applications. For instance, games are *non-linear*, meaning they provide the player with more than one way to be successful or victorious. Games are often written in non-traditional, game-specific languages and scripts supported by a *game engine*—the underlying software that automates game functions from loading and playing animations and sounds to handling collision detection and particle systems. Furthermore, videogames routinely employ artificial intelligence, which broadly speaking is relatively rare in non-entertainment applications.

Thus, we have the status quo: an applications bifurcation. It implies that conventional and videogame software designers rarely talk to one another or compare notes. [1] This situation similarly explains, in our experience anyway, why many game programmers are often self-taught; why they learn the art of game development in their spare time, outside of coursework, as a "labor of love"; why they have little or no formal computer science training, sometimes reinventing solutions to problems long since solved by computer scientists. By the same token, many computer scientists and IT professionals we know are frequently unaware of the challenges and diversity and uniqueness of games, that is, as serious applications. Legendary game designers and pioneers are virtually unknown outside game circles, although their games, e.g. Pong, Pac-Man, Quake, Halo, are widely admired and enjoy, in some cases, cult-like appeal. [4]

Game trade schools have emerged to fill this gap. However, game development is very encompassing and integrative. Indeed, industry experts are now recognizing that students need to be exposed to a broader range of subjects such as history, art, music, creative writing, etc. — the very strengths of liberal arts colleges. [1]

Game trade schools, furthermore, exist for another, perhaps an even more important reason: They satisfy a demand. In other words, game schools, by their existence, suggest that students not only want to be formally trained, but students (and their parents) also consider this training to be worthwhile.

3. OBJECTIVES

We approached the above challenges with the following three objectives (not necessarily in this order):

(1) Coordinate courses consistent with the International Game Developer's Association (IGDA) *Curriculum Framework* with focus on the study of Game Design and Game Programming topics. [5] We give more details below.

(2) Attract, retain, and prepare new Computer Science students for further game studies and/or to pursue careers in the game industry.

(3) Remain competitive as an Institution of higher learning.

We found these objectives could be facilitated within the context of an existing Computer Science curriculum with relatively minor changes. To accomplish this, we take advantage of a few opportunities presented by the policies and the currently recommended schedule.

- Full-time tuition at Marist buys students 16 credits per semester. Most semesters use only 15 of them. This is especially true of the fall semesters. By offering a one-credit GAME seminar in the fall of each year, the Gaming Concentration students get their own "special" course sequence where the faculty can present very game-specific topics, all at no additional expense to the students – more "bang for their buck".
- We are also adding the one-credit Computer Networking Lab in the spring of their junior year in support of massive multiplayer online games (or MMOGs).
- Since game programming requires knowledge of Physics, we specify "General Physics" in the program. This does double duty, also satisfying a core science requirement, so it doesn't reduce the number of electives available to the student.

4. NEW FEATAURES

The Computer Science department has already adopted a yearlong sequence of two courses, Game Design & Programming I and II, which form the core of the proposed concentration. Game Design studies interface design, world design, design of play rules and mechanics, and integration of audio and visual components. Game Programming considers not only "hard core" Computer Science – searching, sorting, path finding, finite state automata – it also incorporates real-time programming for animations, input control, sound playback, collision detection, and applications of physics simulations (e.g., forces, rigid body collisions, particle systems, explosions, etc.) and mathematical methods (vector transformations, random deviates, etc.).

The Concentration has additional features. For instance, we are proposing a Game Seminar as well as recommendations for related courses like Artificial Intelligence, Computer Graphics, and others. We've furthermore included a new math course on computational geometry.

4.1 The Gaming Algorithms Methods & Execution Seminar (GAMES)

This one-credit seminar provides a forum for teaching and exploring issues specific to game design and implementation. It sets apart game concentration students from the outset.

The GAME Seminar meets once a week throughout the fall semester of each year. Various topics are discussed. There are quizzes, tests, and projects. A letter grade is awarded.

Potential topics, in no particular order, may include:

- Game design
- Games and strategy theory
- Business of games in industry
- Game computer architectures
- History of games
- Language processing and adventure games
- Wireless games
- Internet games
- Humor in games
- Non-linearity in games

- AI in games
- Physics in games

Each topic may be covered in a single meeting or span several meetings. We can use this seminar to cover areas that don't require a full course, but are still important to the topic. (Finite State Automata comes to mind. A few classes should suffice.)

4.2 Topics in Applied Geometry

This upper-level Computer Science or Applied Math class covers topics primarily from Computational Geometry and Curve and Surface Design. Subjects to study include plane-sweep algorithms, convex hulls, Voronoi Diagrams, triangulations, parametric equations of curves and surfaces, spline curves and surfaces such as Bezier and NURBS. The course covers the mathematics, algorithms and data structures of these two areas.

These areas of study have applications in Gaming but also Graphics, Computer Aided Design, Molecular Biology, GIS, etc.

Pre-requisites are Calculus II, Linear Algebra, and Computer Science I or equivalent programming expertise. Assignments embody rigorous mathematics as well as programming projects.

5. INFLUENCE ON OTHER COURSES

Since we have neither the faculty time nor classroom space to entirely fill the gaming concentration with brand new courses (we already have three: Game Design and Programming I & II and Topics in Applied Geometry), we recommend small changes and adjustments to some existing courses and a few exchanges in the course requirements. See the appendix for more detail on this subject.

- We recommend that students in the Gaming Concentration substitute our required "Logic Design" course with "Computer Networking and Distributed Systems".
- The language study course uses the language of the Game Design and Programming I and II and gaming concentration capstone course. Currently this is C++.
- Students in the Gaming Concentration take Game Design and Programming I in place of Software Development, which is currently a requirement.
- Advanced Data Structures is required for the Concentration. Currently it is an elective.
- Artificial Intelligence is required for the Concentration. Currently it is an elective. AI would be taught with an eye towards gaming applications in addition to the traditional academic AI topics. As such, we don't need to have two AI courses. Some topics applicable to gaming and AI include chasing and evading, pattern movement, flocking, path finding and navigation, and targeting and threat assessment under uncertainty. [2, 9] Some of these are not typically covered in traditional academic AI study.

6. TRANSFER STUDENTS

As the gaming concentration develops, no doubt new students that didn't start at Marist as freshmen in the Concentration will want to enter the program. At the very least, we need to support transfer students at the Junior year level. (We are thinking, in particular, about transfers from community colleges in the Marist area.) We have considered this as a strong possibility and structured the Concentration accordingly.

Ideally, the transfer students will have already taken Computer Science I and II and Calculus I and II. If not, these courses will need to be taken before matriculation into the Gaming Concentration can be granted. Hopefully, the transfer students will also be ahead on their Core and Liberal Studies courses, so that Topics in Applied Geometry can be worked into their schedule early on.

Since no transfer student will have had the benefit of the GAME seminars we mentioned earlier, a special two credit "transfer version" of the GAME seminar is offered in the spring semester (so as not to add further burden to the Faculty responsible for the fall GAME seminars.)

7. OPPORTUNITIES

A concentration in games presents a number of opportunities for students to further their studies in games or perhaps pursue game careers.

- Every year Marist hosts two large, student-organized gaming events. These events don't have to be all play and no work. We are considering setting up a booth, developing a flyer, and whatnot, to reach out to prospective students and tell them about the Concentration and encourage them to participate in *game programming* competitions rather than just a game playing contests.
- We are considering setting up a forum for game vendor representatives (e.g., NVIDIA, Alienware, and others) and students to discuss game internships, careers, requirements, etc.
- We hope further to engage game vendors to donate hardware to support game design, programming, and student research. For instance, Marist already conducts non-game related joint studies with a major computer vendor. This vendor also makes processors for Nintendo's *GameCube*® and graphic processor units for NVIDIA, and will soon be making the next generation of processor chips for Microsoft's *Xbox*®. Also, *Direct3D*®, which our students use in the Game Concentration, is the software driver for the *Xbox*®. We believe there are opportunities here for the vendor to help prepare the next generation of game developers.
- Indeed, anticipating that students may have expectations for careers in gaming, we plan to work with Career Services to reach out to game vendors to support Marist's Career Fair.

8. THE ACM AND GAMES

The ACM has published no official position on game curricula, as far as our research indicates.

The ACM has, however, approved an agreement to cooperate with International Game Developers Association (IGDA) in 2002. As we mentioned earlier, the IGDA authored *The Curriculum*

Framework, [5] which is the basis for our Game Design and Programming courses.

We note further that key ACM publications have featured game-related papers. For instance, the February 2004 issue of the forward-looking *ACM Queue* was dedicated entirely to game development. [9] In the flagship journal, *Communications of the ACM*, a series of related papers in 2002 were published dedicated to game engines for scientific research. [8] In 2000, the *CACM* published another series of related papers related to physics-based game simulation. [7] Earlier still, Nolan Bushnell (1996), game design pioneer and founder of Atari, also published in the *CACM* a perceptive paper on entertainment computing.

We may conclude from these activities that there is growing interest in and support for game-related development and curricula within the larger computer science community. We believe the ACM will (eventually) need to formalize its position on computer science related game studies if demands from students for formal training grow as it would appear poised to do.

9. OPEN ISSUES

Adding course requirements to an already rigorous field of study is challenging. We don't want to overburden the students with so many required classes that they have no time to explore other intellectual interests through free electives. Marist students have an option use free elective credits for work internships, which is absolutely crucial, as that often enhances opportunities obtaining employment upon graduation.

Our recommendations accommodate free electives in both the major and general areas of study by maximizing any "double duty" opportunities (e.g., the Artificial Intelligence course) and minimizing the number of entirely new courses (e.g., the GAME Seminar is not a new course but replaces the computing studies seminar. The presence of general electives means that there is room for internships and some diversity in course load. However, from the Concentration perspective, the course load structure is still a little too strict. We would do much better shape if we could get two Art courses to count as Core / Liberal Studies courses. That would greatly help to preserve the remaining electives. This needs to be worked on further, and is not accounted for here.

Workload is another issue. After four years, all four levels of the GAME Seminar will be running simultaneously. There may also be a spring "GAME I and II for transfer students". This may present scheduling and/or teaching load challenges.

We furthermore have been asked several times about game design and programming at the graduate level. We have not fully taken these requests into account, although the opportunities would seem considerable since graduate students generally have deeper computer science backgrounds and programming experiences.

Then there is the issue of whether or not we need game-specific hardware. If so (and this is likely), who will control access to it? How do we support it?

10. EXAMPLE COURSE SEQUENCE

Table 1 presents an example course sequence along with notes specific to the Gaming Concentration.

11. ACKNOWLEDGMENTS

Thanks are due to Dr. Roger Norton, Dean of the School of Computer Science and Mathematics at Marist College for his encouragement and support of the initiative. We'd also like to thank the faculty in the School for their comments and feedback on this topic.

12. REFERENCES

[1] Adams, E., *Break into the Game Industry*, Osborne, 2003

[2] Bourg, D.M., Seemann, G., *AI For Game Developers*, O'Reilly, 2004

[3] Bushnell, N. "Relationships between fun and the computer business," *CACM*, vol. 39, no. 8, p. 31-37 (August 1996)

[4] Burnham, V. *Supercade*, MIT Press, 2003

[5] Education Committee, *The Curriculum Framework*, International Game Developer's Association, v2.3, igda.org (February 2003)

[6] Sethi, M., *Game Programming for Teens*, Premier (2003)

[7] Various authors, "Physically Based Computer Animation," CACM, vol. 42, no.2 (February 2000)

[8] Various authors, "Game Engines in Scientific Research," *CACM*, vol. 45, no. 1, (January 2002)

[9] Various authors, *ACM Queue* (2004). vol. 1, no. 10, (February 2004)

Table 1 – Example Course Sequence

	Number	*Name*	*Credits*	*Gaming Concentration Notes*
Freshman	CMSC 110	Computing Studies Seminar	1	
Fall	CMSC 120	Computer Science I	4	
	MATH 241	Calculus I	4	was Introduction to Statistics I
	PHIL 101	Introduction to Philosophy	3	
	ENG 116	College Writing I	3	
	CMSC ?	GAME Seminar I	1	
			16	

	Spring	CMSC 121	Computer Science II	3	
		MATH 250	Discrete Mathematics I	3	
		IS 130	Computer Studies Concepts	3	
		ENG 117	College Writing II	3	
		MATH 242	Calculus II	3	was Core / Liberal Studies
				15	
Sophomore		CMSC ?	Lower-level CS elective	3	
	Fall	MATH 210	Linear Algebra	3	was Calculus I
		Science	Lab Science I	4	should be PHYS 211 - General Physics
		Core/LS	Core / Liberal Studies	3	
		Core/LS	Core / Liberal Studies	3	
		CMSC ?	GAME Seminar II	1	
				17	
	Spring	CMSC 230	Assembly Language Programming	3	
		MATH ?	Topics in Applied Geometry	3	was Calculus II
		Science	Lab Science II	4	
		Core/LS	Core / Liberal Studies	3	
		Core/LS	Core / Liberal Studies	3	
				16	
Junior		CMSC 406	Computer Networks/Distributed Systems	3	was Logic Design
	Fall	CMSC 331	Theory of Programming Languages	3	
		CMSC 233	Language Study	3	may be OOP in C++
		CMSC 446	Computer Graphics	3	was Math or Science elective
		Core/LS	Core / Liberal Studies	3	
		CMSC ?	GAME Seminar III	1	
				16	
	Spring	CMSC 415	Computer Organization and Architecture	3	
		CMSC 414	Game Design and Programming I	3	was Software Development
		CMSC 335	Advanced Data Structures	3	*Required for Gaming Concentration*
		PHIL 300	Ethics	3	
		Core/LS	Core / Liberal Studies	3	
		IS 407	Data Communications Networking Lab	1	
				16	
Senior		CMSC 422	Operating Systems	3	

Fall	CMSC 404	Artificial Intelligence	3	was Upper-level elective
	CMSC 424	Game Design and Programming II	3	was Upper-level elective
	?	General elective / Internship	3	
	Core/LS	Core / Liberal Studies	3	
	CMSC ?	GAME Seminar IV	1	
			16	
Spring	CMSC ?	Upper-level elective	3	
	CMSC ?	Upper-level elective	3	maybe a Gaming Project, a capstone course
	?	General elective / Internship	3	
	?	General elective / Internship	3	
	Core/LS	Core / Liberal Studies	3	
			15	
			Total = 127 credits	