Teaching Soft-skills: Digital Game Development in a Multi-Discipline Environment

Paul J. Diefenbach

Drexel University, Philadelphia, PA 19104, USA

Abstract

While academic video game development has garnered widespread publicity due to the size of the industry and the potential impact on faltering Computer Science enrollment, many academic programs fail to address the expansive nature of modern game development. In addition, many game development companies are reluctant to hire recent graduates or co-op or intern students—not because of their technical skills, but rather due to their lack of real-world soft-skills. Most student courses fail to address the complexities, communication, and cooperative skill sets required in today's games, and the role that project management plays in development. This paper addresses the emerging approach that Drexel University is taking through a joint Digital Media/Computer Science partnership that serves as a foundation for university-wide application, and in particular examines the Game Development Workshop sequence of courses.

Categories and Subject Descriptors (according to ACM CSS): D.2.9 [Software Engineering]: Management
K.3.1 [Computers and Education]:Computer Uses in Education

1 Introduction

The emergence of digital games in academia is well documented in trade publications and the mass-media[Wad04][Sch05]. While academic video game development has garnered widespread publicity due to the size of the industry [Kol06], the growth of the serious game movement [Tho06], and the potential impact on faltering Computer Science enrollment [NSF06], many academic programs fail to address the expansive nature of modern game development. In addition, many game development companies are reluctant to hire recent graduates or co-op or intern students—not because of their technical skills, but rather due to their lack of real-world soft-skills. At Drexel, we are trying to address these issues by leveraging the strengths of the entire university.

1.1 Multi-Discipline Approach

Drexel University's original name in the 1800s was the "Drexel Institute of Art, Science, and Industry," which demonstrates the original tenets of the university to provide educational opportunities in the "practical arts and sciences." Today, the growth of technology has permitted Drexel to be a leader in the area of practical and applied education and training, and our gaming sequence is designed to subscribe to this applied, inter-discipline philosophy. Having one of the oldest and largest co-op programs and a true multi-college university hosting engineering, media and design arts, medical and law schools, the liberal arts, and a business school provides Drexel the breadth to mirror the commercial coordination of skills required for both entertainment and serious video games.

A quick survey of representative game development programs in the United States[Sak06] shows that these programs are usually an outgrowth of either Computer Science or an Art program. The gaming curriculum at these schools, especially where gaming is a major or offered as an advanced degree, is typically either heavy on programming (i.e. Full Sail), heavy on content creation (i.e. Art Institute of Pittsburgh), or relies on one comprehensive curriculum for all students (i.e. RPI).

At Drexel University, game development does not "live" in solely one department, and so mirrors the true nature of game development in commercial settings. Digital game development is offered in a coordinated, cross-listed series of courses in both the Computer Science (CS) and Digital Media (DIGM) majors, and production courses are open to other majors as well. Computer Science courses instruct on foundation software development skills, and offer software design courses for prototyping game concepts. Drexel's Digital Media major is one of the oldest such programs in the United States. The DIGM program instructs students on the foundation skills of design, art, programming, modeling, animation, audio, and video production, and the use of industry tools such as Maya and 3D Studio Max. The gaming courses and projects bring these two majors together, with the additional participation of students and faculty from other majors including Music, Music Industry, Screenwriting and Playwriting,

© The Eurographics Association 2008.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

Engineering, and Business and Law. The culmination of this curriculum is the Game Development Workshop sequence, which teaches soft-skills such as project management, teamwork, scheduling, milestones, and asset management in a multi-discipline setting. These skills offered by both programs are further developed in the graduate coursework in the multi-discipline game research lab: Drexel's RePlay Lab.

1.2 History

Prior to 2004, game-related courses were taught in both the Digital Media and Computer Science majors, but there was little or no coordination between these efforts. In 2004, Dr. Paul Diefenbach joined the staff of Digital Media to help establish its new Master of Science degree program and to revamp the game development curriculum. Bringing a CS background and years of experience as CTO in the game/simulation industry, Dr. Diefenbach along with Dr. Glen Muscio, Program Director, agreed that the game development program needed to better reflect the multi-disciplinary nature of modern games. Computer Science faculty member Dr. Frank Lee, specialist in gaming and HCI, and the CS department head Dr. Jeremy Johnson, had initiated an outreach to Digital Media in 2003. Dr. Lee and Dr. Diefenbach jointly began work on coordination of the two programs, and each was awarded a courtesy appointment in the other's department. Today's gaming curriculum leverages the strengths of each program and extends them with an education in both the technical skills and soft-skills required to function in multi-discipline team environments.

1.3 Soft Skills

In the 1990s, the US Secretary of Labor appointed a commission to determine the skills people need to succeed in the modern workforce. This new framework for workplace skills was based on three components: functional skills that describe what people actually do at work; enabling skills which are specific knowledge and procedures developed through the traditional teaching and learning activities of schools; and the scenario, a communication device to demonstrate the way in which work integrates these skills into a productive outcome.[KBGM90] The gaming curriculum assumes a foundation of enabling skills and focuses on teaching the functional soft-skills through the use of workplace scenarios, especially in the Game Development Workshop series.

The following sections of this paper address the current game development curriculum and detail the Game Development Workshop series of courses. The paper then examines the impact of this inter-discipline effort, and finally discusses the future expansion of these efforts.

2 Curriculum

The game development sequence currently spans both the Digital Media and Computer Science curriculum, and expansion including courses offered by other departments is planned. Game engines are now even being used in the introductory programming courses to teach concepts within a larger, more visual framework. The current sequence of courses, with their host programs noted, is seen in the following table and described below. Game releases and research from this sequence are available via the RePlay Lab main page at www.replay.drexel.edu.

Table 1. Game Development courses.

Game-based CS I & II* - experimental introductory CS sequence using game engines as platform for teaching programming and software design concepts
Gaming Overview – covers the history and theory of game development as well as analysis and critique games.
Computer Game Development – introduces, compares, and contrasts a variety of game and simulations engines, their asset pipelines, and scripting and programming environments. Students learn to evaluate and utilize different game engines and tools for specific tasks.
Scripting for Gaming – covers concepts found in game development such as time and timelines, event-driven programming, state transitions, etc.
3D Gaming Workshop I – initial game pitch, design, and prototyping (pre-production effort)
3D Gaming Workshop II – full production effort of design from GDWI
Educational Game Design – focuses on the unique aspects of designing games for educational purposes (2D)
Experimental Game Design – rapid prototyping of novel game mechanisms (2D)
Graduate Game I & II – advanced topics in game research, development, and commercialization. Recent topics include sponsored user-interface development, multi-point touch screen research, and the commercialization of undergraduate games (bulletproofing game, marketing, licensing, copyright and IP, gameplay testing, production, release, etc.)

3 Game Development Workshops

3.1 Concept

Drexel operates on a quarter system of 10-week terms with an 11th week for finals. This short schedule is very challenging, but also helps reflect the deadline-driven nature of game development companies. A two-course sequence, Game Development Workshop I and II, attempts to follow the traditional cycle of game production. The first course serves as a design and pre-production phase where teams of 4-6 students create the game concept and write a 1-page "sell" document, an executive summary document, and eventually a full game design document (GDD). They also create a working prototype of the game—all in 11 weeks. At the end of the term, students pitch their game at a presentation in front of faculty and students, and the audience votes on their favorites. The best and most viable games are selected for full production in Workshop II the following term, and teams are merged and expanded.

3.2 Incorporating Soft Skills

The Workshop courses offer a perfect platform for addressing the issues of soft skills such as leadership, interpersonal communication, punctuality, respect for deadlines, working well with others, and time management. For these courses, the individual foundation skills of the teams have already been learned, and game engines, their scripting systems, and their asset pipelines have been introduced in prior courses. The workshop courses teach students how to use these individual skills in a managed team environment. The following sections address the application of these techniques to various aspects of game development. The use of role models, mentors, weekly meetings, brainstorming sessions, and other techniques are used. Teams are taught to be wary of groupthink, leverage tribalism, and avoid mistaking “pigs” from “chickens.”

3.3 Teams

A variety of majors now take the Game Development Workshop (GDW) sequence. DIGM majors enter this course with a foundation of art, design, modeling, and animation skills, yet these skills have only been applied to individual or small team production animation projects. CS majors enter with a foundation in programming and software design, but most projects are small, individual assignments. Students with other majors such as Music, Screenwriting and Playwriting, Engineering, and even Business are also represented, yet these too have traditionally not been exposed to a heterogeneous team setting. During the first class, papers labeled with core skills such as modeling, animation, programming, music, art, and story are laid on the floor, and students are asked to stand near their interests/skills. Teams of 4-6 people are then created based on these distributions. In GDW-II, teams are expanded to approximately 16 students, with an approximate breakdown of 35% DIGM majors, 35% CS majors, 30% others. As the program, faculty, and facilities expand, other majors (i.e. philosophy, English, history, etc.) may be included.

3.4 Design

Introductory brainstorming exercises, such as combining student-supplied themes, characters, et cetera, are used in the first class to spark creativity in unexpected ways. Once teams are created, teammates must agree on a game concept and create a “one-pager” to describe and sell the concept to the instructor, who acts as an external Executive Producer. Originality, feasibility, and gameplay are primary considerations; titles include Project Bolt (Figure 1), Chroma Chaos (Figure 2.), and Moach Rotel (Figure 3), which introduce music-based gameplay, color-based gameplay, and a story-driven game from the perspective of a bug, respectively. After the concept is approved, a 5-page summary document submitted by week 2 fleshes out the concept and addresses the marketability of the game. Early revisions are crucial in this shortened cycle. The students then begin to work in tandem on the GDD and the prototype. Sample and prior GDDs help guide the students. It is repeatedly stressed that the GDD is a living document.

Figure 1. “Project Bolt”- music-based 3D platformer

programming in a time-based, event-driven, high-level scripting language; using finite state machines for character AI; and creating interfaces for level designers to control environmental parameters. Programmers provide sandbox environments for both the DIGM modelers/animators and the music/audio team in an effort to promote coordination and cooperation between teams.

3.6 Asset Management

Asset management is a critical issue for any commercial game company, yet many programs fail to address the soft-skills aspects of it. Perforce™ is now used throughout the Digital Media major beginning in freshman year, and other majors are introduced to this system in the gaming sequence. With a platform for

Figure 2. “Chroma Chaos”. Pre-production version.

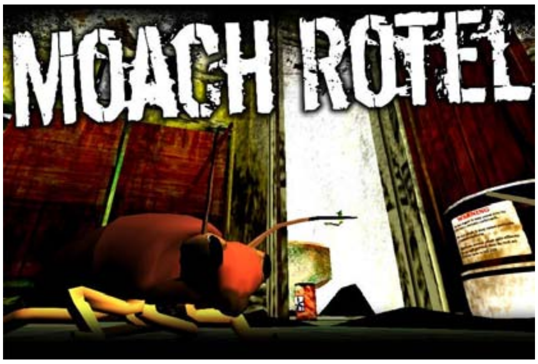

Figure 3. "Moach Rotel". Production publicity shot.

revision control, GDW-I can concentrate on the larger organizational and translational issues. Standard software management procedures have been introduced such as multiple asset trees for asset creation and for both engine-formatted build and release versions. Naming conventions are taught such as how to best incorporate level-of-detail and type information into an expressive name for quick identification and retrieval despite large numbers of assets and heterogeneous teams. Asset lists and guidelines are maintained for models, textures, and audio files. Application of these formal methods are in fact sometimes ahead of the haphazard implementation of the industry; a guest GM from one of the leading game production companies was surprised to learn that our students had been using for over a year a particular method for asset management and game builds that they had only adopted six months prior!

In addition, the disparate nature of the subteams means that communication of asset properties is critical. A common template is used by artists, modelers, animators, and technical directors to convey look, scale, motions, behaviors, timing, and other information for each central object or character. A student serves as Technical Director for each team, and manages formats, translation, and all asset pipeline issues. Documentation of all methodologies is archived in the GDD for team use and future reference by later classes.

3.7 Project Management

In contrast to small programming assignments or individual games, the 16 person teams of GDW-II present for many students their first large-scale project development. For this reason, project management and component soft-skills are taught and enforced for the smaller teams in GDW-I so that good practice is in place for GDW-II. Each team has an internal producer/manager. Teams are assigned a "budget" for the project, with payments for meeting both weekly milestones as well as longer-term goals such as an alpha or beta release or the production of the GDD. These payments, in points, are awarded each week by the external producer (the instructor) based on milestones, similar to a typical pay arrangement. The internal producer manages point distribution within the team, and in GDWII, this is often portioned out to the managers of each subteam such as modeling, programming, sound, etc. In addition, a team review of each manager/producer has been added to help foster leadership responsibility.

This payment system has solved several of the problems of the past. Students tended to either perform the bulk of the work at the end of the term or to concentrate too heavily in one area of development (i.e. hi-resolution modeling), to the detriment of the project as a whole. By losing unrecoverable points each week, they realize early that keeping to the schedule is crucial.

In addition to forming and managing the team structure, the course introduces basic project workflow, milestone projections, scheduling, inter-team communication, documentation conventions, and other soft-skills. The use of Gantt charts, forums, Basecamp^{SM}, telemeeting software, and other tools are taught and extensively used. Weekly team-wide reviews are used for all production assets including art and code. Development is centered around the scrum form of Agile development method [DS90], where weekly sprints are used and "pig" roles (committee parties) and "chicken" roles (interested parties) are brought together.

3.8 Assessment

Grading is based on a variety of metrics. Team members must keep logs of their individual work. In addition, each team member reviews up to four other members with whom they worked closely. As mentioned previously, milestones are an important element throughout each of the workshops, and the points each team member earns (i.e. their project payment) directly corresponds to that student's grade. The final component in grading is the evaluation of the end products: the GDD, the prototype, and the presentation of the game.

4 Results

While little empirical data on Drexel's sequence is available, the program has been very successful in a variety of ways. The mixing of CS and DIGM majors permits creation of complex, high-quality games that neither group could produce independently. DIGM students learn the transition from static to dynamic animations. They additionally learn the real-time performance considerations on models and textures. Computer Science students learn to code in a large, complex system that better reflects the nature of most software development jobs. The real-time nature of gaming requires concrete and visible application of the performance theory to which they have been exposed. Other concepts such as AI for behaviors and path planning; network communication in a client/server setting; and transition from pre-production, hard-coded "hacks" to full-production, extensible, and documented code are invaluable real-world skills. Most importantly, all students learn to work in teams where design elements may not be their own; where they must frequently communicate and work with other majors; where their individual production affects the entire team's work plan; and where scheduling, design, and documentation is as important as the code itself. Games produced from the workshop sequence are complex, creative, and comprehensive despite the condensed schedule.

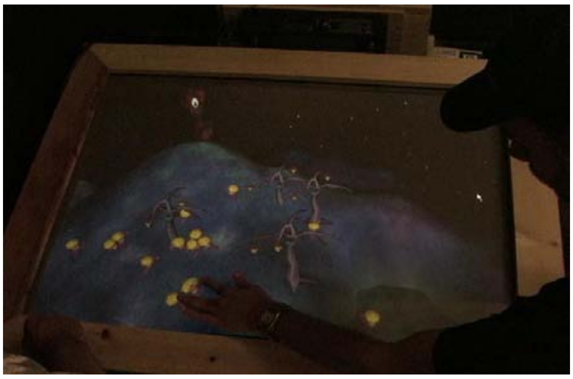

Figure 4. "Planet Diggum". Multipoint Touchscreen

The course curriculum has taken more than three years to be finalized and adopted by both programs, and has evolved from an initial offering of two courses. The computer game development sequence now counts as one of several "tracks" that CS majors must take. Scheduling between two programs (one a 4-year program and the other a 5-year program) and two schools continues to prove difficult, but manageable. The fruits of this labor are now being seen in the first joint Senior Projects between the two majors as well as graduate work including Planet Diggum [Die07], a multi-point touchscreen kiosk game.

Planet Diggum is an investigative gaming application being developed by Computer Science and Drexel Digital Media (DIGM) graduate students as part of their gaming curriculum and utilizes a multi-point display built by Computer Science (CS) and Electrical and Computer Engineering (ECE) undergraduate students. In addition, Computer Science students have provided customized gesture code and assisted with the client-server code. This project is coordinated by Drexel's RePlay Lab.

5 Future Curricular Changes

While the joint CS/DIGM sequence has been very successful, there are several revisions under development.

The addition of an intermediate game development course is planned to alleviate the ramp-up required in GDW-I. An experimental section of the introductory CS courses for Digital Media majors leverages visual-based projects (such as the Alice system), and an expansion of this pilot program is being developed which would extend the teaching of fundamental concepts through use of the Torque game engine platform. Not only would this enhance the introductory CS sequence, but it would also further reduce the GDW-I ramp-up. In addition, this would permit leveraging Torque in later CS courses to teach concepts such as Client-Server Networking.

The most sweeping changes would be in further coordination outside of the Digital Media and Computer Science majors. Discussions with the Music Industry program have considered not only preparatory courses aimed at game scoring and foley, but also leveraging existing courses that center on IP, business, and legal issues within the entertainment industry. Discussions with Drexel's Baiada Center for Entrepreneurship in Technology have focused on tailoring graduate-level courses on the commercialization of games to qualify as an Entrepreneurship Course for an annual business plan competition. Many other such cooperative efforts exist university-wide, and are being investigated on both undergraduate and graduate levels.

References

[Die07] DIEFENBACH, P. J. et. al. "Planet Diggum: Multi-point Touch and Gesture Control of Gaming Environments", *Proc. Thirteenth International Conference on Distributed Multimedia Systems (DMS 2007)*, San Francisco, USA. pp. 235-240.

[DS90] DEGRACE, P AND STAHL, L. *Wicked problems, righteous solutions*, Yourdon Press, 1990.

[KBGM90] KANE, M. BERRYMAN, S., GOSLIN, D., MELTZER, A. *Identifying and Describing The Skills Required by Work*, 1990. http://wdr.doleta.gov/SCANS/idsrw/idsrw.pdf

[Kol06] KOLODNY, L. *Global Video Game Market Set To Explode.* http://www.businessweek.com/innovate/content/jun2006/id20060623_163211.htm

[NSF06] National Science Foundation, Div. of Science Resources Statistics *Science and Engineering Indicators 2006* Arlington, VA (NSB 06-01) [February 2006]

[Sak06] SAKEY, M. *Fundamental Learning – GDC 2006 Curriculum Workshop.* April 2006. *http://www.igda.org/articles/msakey_gdc-curriculum.php*

[Sch05] SCHIESEL, S. Video Games Are Their Major, So Don't Call Them Slackers *New York Times* November 22, 2005, Tuesday Late Edition - Final, Section A, Page 1, Column 1

[Tho06] THOMPSON, C. Saving the World, One Video Game at a Time. *New York Times.* July 23, 2006, Tuesday Late Editio - Final, Section 2, Page 1, Column 1

[Wad04] WADHAMS, N. *Academics turn their attention to video games*. http://www.usatoday.com/tech/news/2004-02-13-ludology_x.htm