Emphasizing Soft Skills and Team Development In an Educational Digital Game Design Course

Quincy Brown
Drexel University
3141 Chestnut Street
Philadelphia, Pa 19104
215-895-2669
qb23@drexel.edu

Frank Lee
Drexel University
3141 Chestnut Street
Philadelphia, Pa 19104
215-895-4953
fjl@cs.drexel.edu

Suzanne Alejandre
The Math Forum @ Drexel
3210 Cherry Street
Philadelphia, Pa 19104
215-895-1080
suzanne@mathforum.org

ABSTRACT
Engineering education has evolved from providing students solely with technical skills to providing them with courses that provide students with the non-technical "soft skills". Among the soft skills desired by employers are student's ability to receive and respond to feedback, manage multiple requirements, and work in multidisciplinary teams. The use of Capstone courses in Computer Science education has been a means of providing students with the aforementioned soft skills and experiences prior to graduation. We define and implement a model for an educational video game design and development course that provides students with real-world game design experiences. Students learn the complexity of developing an educational game while functioning in multidisciplinary teams. Additionally, we provide students with an opportunity to visit K-12 schools to witness, first hand, the conditions in which their video games will be used. Finally, we present lessons learned and discuss methods for the successful development of multidisciplinary courses in Computer Science Education.

Categories and Subject Descriptors
K.3.2 Computer and Information Science Education

General Terms
Design

Keywords
Multidisciplinary, Games, Computer Science Education

1. INTRODUCTION
Capstone courses have been used in the education of students from disciplines such as, Computer Science, Engineering, and Agriculture [1-4]. Despite the variety in disciplines, the courses are designed to provide students with project-based experiences. While there is no precise definition of a capstone course, those using the title do share similar goals. Namely:

1) Integration of functional knowledge from previous coursework.
2) The application of acquired knowledge.
3) The opportunity to improve oral and written skills
4) The use of teamwork.

Additionally, Fairchild and Taylor [1] further describe the role of the course instruction as one that supports students' transition from dependent learners to independent learners.

This transition can provide students with the skills desired by employers to complement the technical knowledge acquired through traditional coursework. A survey of required skills for available game programmer jobs include:

1) People – ability to work with various groups of different skill sets, good people skills, collaborative attitude; ability to work effectively with other disciplines and external development partners [5-7]. While the advertisements use multiple adjectives to describe the desired attributes, the gaming industry desires individuals who have the ability to communicate effectively with team members who possess diverse skills and backgrounds. (e.g. User Interface programmer communicating with an animator)
2) Process – ability to follow processes and meet objectives; working from project requirements and helping to define software specifications [5-7]. Although companies may differ in the specific game design process employed, they desire game designers and developers who understand the game design process.
3) Planning – ability to understand project objectives and translate them into detailed tasks; fleshing out, expanding, and implementing game designs [5-7]. The game design industry seeks to hire individuals who are self-directed and are able to transform feedback and project goals into actionable items and carry through on those items.

Game design courses in CS curricula have served as the foundation for project-based and capstone courses [3, 8-10]. Within these courses Computer Science curricula developers have sought to prepare students for industry employment by providing training in these aforementioned skills. While we agree that the introduction of project based gaming courses has benefitted students, we believe that additional emphasis is needed on the collaboration of computer science students with peers from disparate disciplines.

In this paper we describe experiences from an Educational Digital Game Design course at Drexel University. Although games for education and entertainment share similarities, the design of educational games introduces a layer of complexity not present in the development of entertainment games. In addition to being entertaining, an educational game must also support the learning of users [11]. While it is easy to identify games that do achieve this balance and those that do not, it is quite difficult to design and develop such games [11]. We sought to instruct students on the process of designing and implementing games that entertain and facilitate learning. In addition to teaching about the complexities of educational game design we also sought to provide students with an experience working with team members who possess different skill sets.

To attract students from a variety of majors who possess specialized knowledge, the course was cross-listed in the Computer Science, Digital Media, and Education departments. The instructors sought to leverage the strengths and expertise of students to create multidisciplinary groups who collaboratively developed an educational video game.

The goals of the course were to:

- Provide students with an opportunity to develop math games for middle school students that can be immediately available for use.
- Enable students to receive and respond to feedback from multiple sources.
- Teach educational game design and development concepts to Drexel University students by providing them with a real world application for prior knowledge.
- Provide students with an opportunity to express their ideas using multiple media such as online-only discussion board exchanges, storyboards and class presentations.
- Provide students with an opportunity to work in teams of students from technical and non-technical disciplines.

In this paper, we describe the structure of the course, multidisciplinary teams, issues encountered with sharing work among teams, development process model, and course evaluations. Finally, we conclude with lessons learned and future plans. The paper describes two versions of the course, Fall 2006 and Fall 2007. While we discuss the course design in general there are differences in the 2007 version, which we describe when relevant.

2. Course Design

The course is offered in the Computer Science Department of Drexel University. To achieve the desired team balance the course is cross-listed in the Digital Media and Education departments. The university operates on a 10-week quarter system, which allows for an eight-week long development cycle.

In addition to providing a capstone experience the course was designed to provide a multi-faceted learning environment. Bransford, Brown, and Cocking [12] identify four perspectives for an effective learning environment; Learner-Centered, Knowledge-Centered, Assessment-Centered, and Community-Centered. We sought to align the course with these perspectives and address each in the course as shown in Table 1.

Table 1 Multi-Centered Course

Learning Environment	Demonstrated In Class
Learner-Centered Addressing beliefs, attitudes, skills brought into class	• Critique existing educational games • Discuss student rationale for making games.
Knowledge-Centered Activities and information required for students to learn necessary information	• Game development lectures • Individual and group projects • Middle school math subject lectures • Learning theory overview
Assessment-Centered Using assessments to provide feedback to understand what the learner understands	• Peer assessments of completed projects • User assessments of prototype by middle school students • Teacher assessments of final project
Community Centered Student's connection to their classroom, school, local, global community	• Classroom – Peer review • Local – Middle school testing • Global – Math Forum participation

Each perspective provides a distinct contribution to the learning environment. We embed lessons and activities into the course to support each perspective as follows:

Learner-centered: In sessions at the beginning of the quarter we discuss, as a class, student's experience playing digital games for fun and for educational purposes. As instructors we facilitate this discussion to support students in understanding their current beliefs about educational games. We seek to help students make connections between the games they have played in the past and their current goal of creating an educational game for younger students.

Knowledge-centered: Students are provided with several lectures in which we dissect educational games. We discuss the role of feedback, and hints in support student learning. The connection between how well a player performs, via scoring, and the amount that is learned is also discussed. We provide lectures on educational pedagogy and discuss the National Council of Teachers of Mathematics (NCTM) standards the games should address [13].

Assessment-centered: Throughout the game development cycle students receive both formative and summative assessments. The formative assessments are in the form of feedback from the course instructors, K-12 classroom teachers, and middle school students.

The goals of these assessments are to provide students with an opportunity to view their game from different perspectives. After each assessment the students are expected to revise their project and describe how they changed their game as a result of the assessment. The summative assessments are in the form of grades students receive from the instructors and their peers. The instructor's grade submitted artifacts for completeness and well as for their ability to revise their games based upon the feedback they receive. Class peers evaluate completed games during the last class session. Students are provided with a rubric to score six aspects of the completed game. This score is used as a component in their final course grade.

Community-centered: The integration of K-12 classroom teachers from across the nation as well as local middle school student testers enables students to realize the potential impact of their work. Upon graduation each student will be connected to a larger network comprised of individuals from different disciplines. The introduction of this connection in this course provides students with an opportunity to experience this connection prior to graduation and see how their disciplines are interconnected with other disciplines as well.

3. Multidisciplinary Team Structure & Roles

Collaboration between educators and game developers, as shown in Figure 1, has been recommended as an effective team composition for developing educational games [14, 15] . This structure enables the integration of games and pedagogy and suggests that there are multiple competences and skills required to develop educational games.

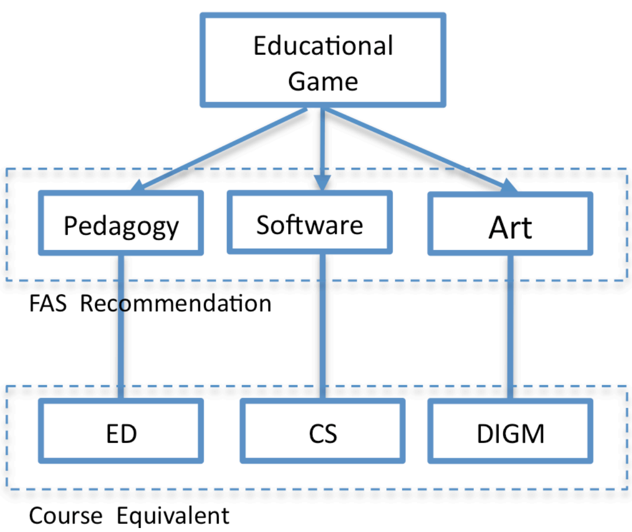


Figure 1 Design team structure

Students worked in teams composed peers from each discipline. Each team was composed of 2 Computer Science students, 1 Education student, and 1 Digital Media student. The expertise from each discipline plays an integral part in development of the educational game as shown in Figure 1. The role of the education students was to provide pedagogical knowledge to the team. They were tasked with communicating with K-12 classroom instructors and developing artifacts related to the educational content of the game. The Digital Media students were primarily responsible with creating assets for the educational video game. Their prior coursework in scripting for game development includes the concepts of state and time and has provided them with experience in creating assets for digital games. The Digital Media student's scripting coursework is completed in Flash and Actionscript and includes the concepts of time, states, and user interaction. The primary task of the Computer Science students was to develop the software to manipulate the assets created and package the game into an application that could be executed on computers in the schools. The code written by the Computer Science students integrated these assets using programming constructs such as variables and if-then statements drawing upon their prior coursework in computer programming. This structure was imposed to provide the Computer Science students with a team environment comparable to what is desired by the game development industry. Programmers in "real-world" environments are expected to develop code in collaboration with non-programming peers such as artists, educational researchers, and business people.

Previous developers of interdisciplinary project-based courses have encountered difficulties distributing coursework, balancing personalities, and disciplines while teaching student teams from different disciplines [16, 17] . In the first offering of this course similar issues were experienced by the course instructors. During course evaluations computer science and digital media students felt that they were contributing the majority of the work towards completing the project. Surprisingly, the education students also shared this sentiment. This perception was due in large part to all team members placing the majority of the value and success of the project into the game-like aspects of the product.

It was difficult for the instructors to explain to the students that the educational component was equally important. One education student emailed the following to the instructors:

" I have consistently tried to talk to my group about wanting to make it more educational but I feel as they don't care about that aspect. John has been very helpful but Alan and Robert seem afraid that the game will not be fun if we included any more learning." (names have been changed to protect student identities) [18]

Additionally, students expressed that the negative feedback received from the student testers and instructors was due to their lack of technical competence and low intellectual ability rather than any design flaws that they could correct. For example, this sentiment was evident in one summary students' submitted describing their experience with middle school student testers in which the group reported:

"We observed that the students' math abilities were not as great as we had expected. Even though we had tried to think in terms of an inner city middle school student, we had trouble. While half the students could easy count up the cost of items they had in the cart, the other half struggled…"

During the development phase the instructors named the attitude of the computer science students as "technical arrogance". As a result of these attitudes the education students were treated as unimportant and their tasks and inputs were devalued. This in turn, led to the education students not feeling empowered enough to request technical changes that may impact the educational content of the game. This unbalanced dynamic led to the creation of games, which were technically and visually acceptable, yet contained several mathematical and pedagogical errors.

To compensate for this imbalance in the second course offering the instructors provided additional writing tasks such as, describing the mathematics and how the game can be incorporated into a classroom, for the education students. These documents were written prior to the development of any code during the design phase shown in Figure 3. In addition to these artifacts, the education students were tasked with communicating online with a K-12 instructor. The education student was tasked with relaying information between the teacher and their development team.

To provide students with the skills to be able to work in teams with individuals from different disciplines, as desired by industry, and in an effort to minimize "technical arrogance", the instructors required each team to write a document detailing how they responded to the feedback received. We felt that it was our role to support the students in being more reflective of their role and to be less dismissive of the feedback from their users.

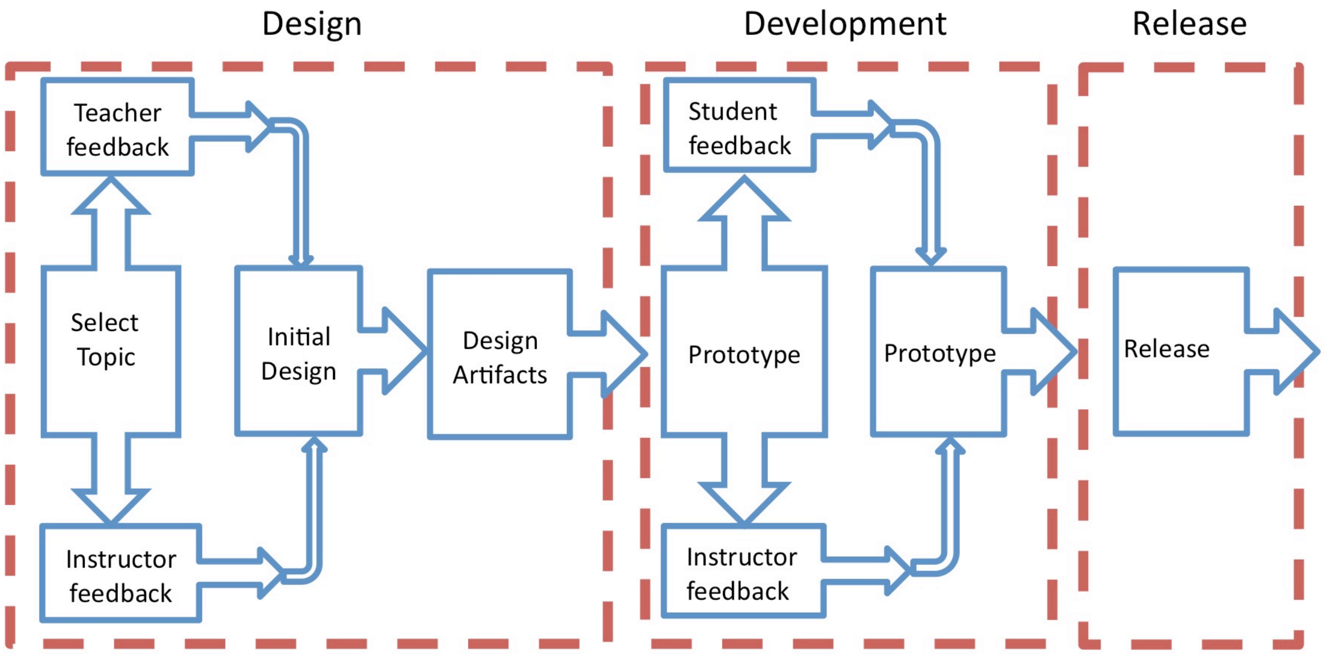


Figure 2 Game Development Process

4. Game Development Process Model

The development process was divided into three phases. The first, design provided teams with an opportunity to formulate their game ideas and learn about game design and pedagogy. The second, development, provided teams with several iterations of developing their game and receiving feedback on their game during development. This phase enabled the CS students to draw upon their prior programming knowledge. The third, release, required the completion of all documentation and game demonstration to the rest of the class. This section describes each phase in further detail.

4.1 Design Phase

In the design phase students were able to develop the game concept prior to writing any of the actual code. This phase was longer than any other to allow groups to receive feedback from the K-12 instructor and course instructors regarding their topics. To align with the goals of the capstone nature of this course and game industry desires this phase emphasizes multidisciplinary teamwork skills. Students are tasked with interacting with K-12 teachers, the course instructors, as well as each other to develop a game design.

The university students were given lectures on several learning theories such as Constructivism, and Behaviorism, and Learning Styles, and National Council of Teachers of Mathematics (NCTM) standards in topics such as, Geometry and Algebra for grades 6-8 [13] to assist them in selecting appropriate topics.

Although students in the course were game players, they lacked prior experience evaluating digital games from a developer's perspective. We provided them with opportunities to play and critique educational games during class sessions. We asked them to note the variety in game complexity and the relationship between success and learning. While developing their own games we wanted students to think about the feedback, scoring, and difficulty of the games they evaluated and how their beliefs of each would affect their design. In this phase, groups were responsible for creating storyboards and design documents detailing the plot of the game and what students were expected to learn.

4.2 Development Phase

In the development phase groups created two prototypes. Students were expected to use their prior knowledge of design, programming, and pedagogy in to implement their designs. To foster a connection to the broader community and give students an opportunity to observe their game played by users the class took a field trip. Prior to the trip all groups were required to implement a working prototype. As a class we visited a middle school classroom and allowed the middle school students to play and critique the developed games.

This experience was the first that many students had in delivering a product to a customer for testing. The students were able to rapidly assess various aspects of their games based on student initial responses. The instructors' felt it was important for the university students to receive feedback from multiple sources. In the first offering student "technical arrogance" resulted in their minimizing the feedback received from the middle school students. In the second offering students were more receptive to their feedback as they were less focused on the programming aspects of the project due to the simplistic Scratch environment and emphasis on the design prior to coding.

After receiving the feedback from the K-12 instructors and middle school students a second prototype was submitted. Along with this version, a document detailing how the feedback was used in the updated prototype was submitted.

4.3 Release Phase

The final 2 weeks of class were dedicating to refining and releasing the games. A pre-release was submitted at the beginning of this phase to enable instructors to assess the progress of each group. The last class session was our show and tell day. Each group presented their game to the class and was then allowed to play and critique each other's games. In lieu of a final exam, groups were given 3 additional days to submit the final version of the game. In addition to completing the games this phase was designed to provide students with an opportunity to make written and oral presentations to their peers.

4.4 Development Environments

In each iteration of the course, the instructors used different development environments. In the first version of the course the Flash [19] environment was selected. This was due to the familiarity of asset creation and Actionscript usage by the Digital Media students and the similarities of Actionscript to software languages familiar to the Computer Science students. While the instructors intended this choice to be welcomed by the students the CS students had more difficulty with Flash programming than anticipated. This difficulty coupled with the inability of the education students to participate in the programming aspects led the instructors to select a different environment for the second offering.

Scratch [20], a programming environment designed for middle school students, was selected for the second offering. We felt that this program enabled all members to participate in the programming aspects of the project, as none were familiar with the environment. We did not feel that this lack of familiarity would be an impediment to any student enrolled in the course because the software was designed for middle school students and utilizes a user-friendly drag and drop interface.

Surprisingly, the Computer Science students had a lot of difficulty developing in this simple environment. As none of the course goals were to instruct students in developing code for a particular language we felt that the deemphasizing the coding aspects enable the Computer Science students to focus on the non-programming skills of teamwork, planning, and oral and written presentation.

5. Course Evaluation

In each course students were asked to evaluate the instructors and overall course. In addition to the end of quarter evaluations students mid-quarter were required to complete a mid-quarter evaluation of the course.

From the Computer Science students, the course was rated 4.80 and 4.64 of 5, in the first and second year respectively, for the "Teamwork or group projects and assignments were an integral part of the course" category. For "The course had a multi-disciplinary perspective" category students rated 4.60 and 4.18, out of 5.0, in the two years of offerings.

The Math Forum is "the leading online resource for improving math learning, teaching, and communication since 1992" [21]. The online resource offers support for students, parents, and educators. They publish and catalog a variety of mathematics applications and software on their website. Three games, Ameba, figure 3a, Personal Shopper, figure 3b, and Football, produced from the first offering were published as part of their Math Tools collection.

An online teacher reviewer rated the Personal Shopper, shown in Figure 3b, an average of 4 out to 5 stars. One teacher review indicated that the game was effective in the "Ability to meet my goals" category. The user also noted "This game allows students to see real world applications of money." The Amoeba game, shown in Figure 3a, was rated an average of 4 out of 5 stars. A different reviewer would "like to recommend [it] to students I tutor and to teachers I mentor". The reviewer offered, "The activity has the potential to allow students to discover the true meaning of proportion." A suggestion was also given "it is easy to get too many by accident…perhaps an option of allowing the ameba to spit out some would meet the needs of a wider range of students". This type of constructive feedback from educators is in keeping the course goal of receiving feedback from multiple sources.

The Scratch games were placed in an online gallery on the Scratch website and are available for use as well. While they have not been rated they have been viewed multiple times through the classes gallery.

6. Effects of Feedback on Game Designs

The feedback from K-12 educators, middle school students, and course instructors was an integral component of the course. Throughout the course groups were required to describe how they incorporated the feedback they received. We present the examples of the changes made as a result of this feedback in this section.

6.1 K-12 Teacher Feedback

Although the class was given overview of learning theory and NCTM math standards it was difficult for the game design students to identify the skill level of their target audience. Comments from the middle school students and teachers provided guidance in achieving the correct level. For example, one group designed a game based on angles and believed they were teaching about angle reflection. However, after sharing the prototype with the K-12 teacher the students were asked,

"When you talk about enforcing the learning aspect, do you mean learning that the angle of incidence equals the angle of reflection? Are you hoping to help them discover that, or are you thinking you will be reinforcing that concept that they've already learned somewhere else?"

The group then reexamined their game to clarify what their learning goal was to be. As a result of this exchange the group modified their game scoring to give students points based on the angle the student selects in during game play. Additionally, the group had decided to create their game to reinforce a concept already learned.

The above exchange of ideas exemplifies the valuable nature of the feedback the groups received from their K-12 instructor. We, the instructors, designed this aspect of the course so that the teachers could press the students to think through their ideas and how their game would facilitate learning. We believe that this continuous exchange during the game design and development deepened the experience of working in diverse teams.

Another valuable lesson learned through this exchange was that in addition to the game play attention had to be paid to the instructions and ease of use. Often the teachers would have difficulty figuring out how to play the games. The following exchanges occurred between multiple groups and their teachers:

"I clicked on Project 3's Game URL posted by Suzanne, but was unable to get any instructions. I tried playing the game without instructions but was not sure that I was doing it correctly. I chose 3 players for my team, chose the opponents I wanted them each to hit, then saw the statistics when I indicated the game was over. But I didn't see any snowballs fly! Was I doing it right?"

Another teacher wrote: "I have tried the prototype a couple of time and am unclear exactly what it is supposed to do. I see time clocks moving but can't figure out whether I am supposed to be able to actually interact with the game at this point. This is a component of the development process that I am totally clueless about."

Teachers often assumed that they were not playing the game correctly. Initially, the some groups felt that the teachers needed to be told how to play and their games did not need to be changed. This was another example of the technical arrogance students exhibited towards other students. As instructors, we wanted the students to realize that the difficulty encountered by the teachers was feedback that they could use to make the game play more explicit and understandable. Visual cues, hint buttons, and instruction pages were added to several games to minimize the difficulty experienced by users.

6.2 Middle School Student Feedback

As a result of the middle school field trip a group design that had originally included a timer was modified. The group realized that the student testers were distracted by the ticking timer and therefore not attempting to correctly convert fractions as the game required. This experience led the group to write:

"The first issue that we plan to tackle will be the timer; we now know that is better for the student to understand the math than it is for them to do it quickly. We thought that the timer had enough time to complete the task, but we didn't consider the pressure it caused and its effect on a seventh grader."

Subsequently, the timer was removed from the final version of this group's game. The student had also commented on the black and white only colors of the game. While the group designers thought the color choice was sleek, the student testers believed it was boring. The final version of the game included color.

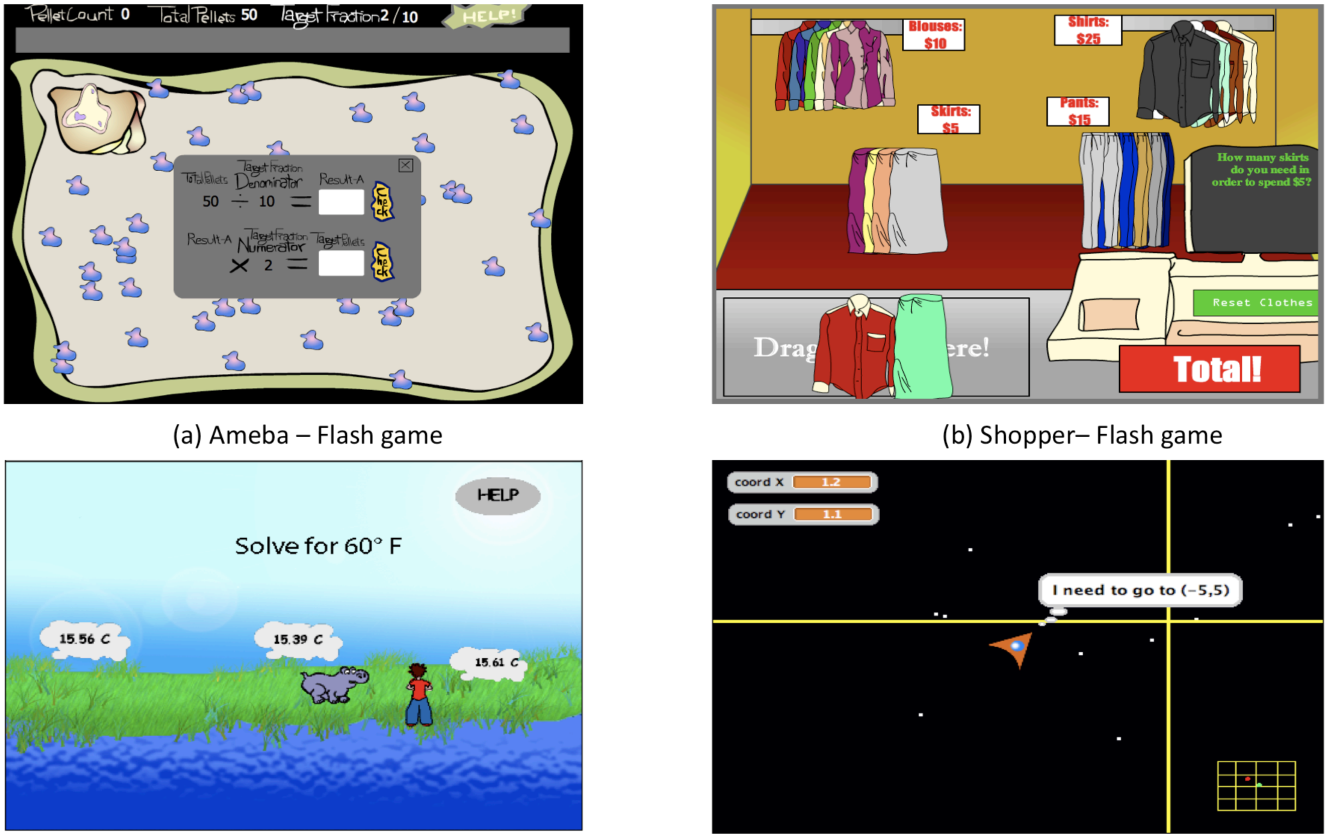
Another group realized that their game was too easy for the students to play. They reported:

"As they played, it became apparent that it was too easy to guess the answers. There were multiple selections for each answer; however the ranges of the answers let the actual answer too obvious if someone was guessing."

As a result of receiving this feedback they changed the incorrect answers so players would have to solve the problems to arrive at the correct answer. By comparison, as a result of student testing another group discovered that they had made their game too difficult and unfriendly for students to play. Subsequent changes such as, simplifying the drag and drop interface, were made to the final version.

Although students were presented with information on NCTM standards it was important for them to witness actual students play their games. Through this exchange, between university and middle school students, the game designers were able to understand how their game may facilitate learning in a classroom. Creating a game based upon general standards enabled students to make assumptions regarding facets of the games such as, level of difficulty and ease of use, however these facets become more concrete when utilized by student testers. It was our intention to provide students with direct feedback from users to facilitate the transition from abstract ideas to concrete game features. This type of feedback is similar to that experience in the game industry.

Students were also responsible for determining which parts of the feedback required changes to their designs. It was also important to instructors for students to experience the iterative nature of the game development process.



(a) Ameba – Flash game (b) Shopper– Flash game

(c) Zoo– Scratch game (d) Space – Scratch game

Figure 3 Games Created by Students

7. Discussion and Future Work

Using the context of educational game development we have implemented a capstone course to instruct multidisciplinary student teams to develop educational games. In addition to learning the game development process students are supported in building the non-technical skills desired by employers in the game industry such as multidisciplinary teamwork and project planning.

As new game development platforms become available we will consider integrating them into the course. While we do not desire to have Computer Science students focus solely on technical skill acquisition for programming games we do want to expose them to additional platforms when possible. This goal will also be balanced with creating a team environment in which all members' contributions are equally valued and required for team success.

As part of the feedback students received they were rated on six facets of their game; First Impression, Educational Content, What did students learn?, Game Play, Graphics, and Concept. As additional courses are offered we intend to refine this method of objectively evaluating the games developed.

We believe that the design and implementation of this multidisciplinary educational games design course can offer the following lessons learned to educators desiring to develop or improve their own courses:

1) Balance tasks from disparate disciplines: While there is a substantial amount of programming involved in the implementation of the game, placing emphasis on the design phase and other tasks prior to the code development can provide ample work for other majors. This distribution will demonstrate that programming is a portion of the entire task and not the only part that matters.
2) Simplify the programming environment: Selecting a programming environment that does not have a steep learning curve can enable all members, technical and non-technical, of the team to participate in programming tasks. Many of the education majors had never written software prior to enrolled in the course. The selection of Scratch, in the second year, provided them with new skills and enabled them to learn from their technical peers in a cooperative and supportive environment. Selecting a new platform also enabled the Computer Science students to focus less on the programming task and more on integrating pedagogical aspects such as feedback, connection between scoring and student's learned skills.
3) Emphasis on soft-skills: Providing students with opportunities to present their projects and communicate ideas, orally and written, to classmates, course instructors, K-12 teachers, and middle school students

reinforces the necessity of clear communication. The visit to the middle school provided students with a clear connection to their community-at-large and enabled them to view, first hand, how their game would be used.
4) Design Process: Enabling students to complete much of the design early in the development process reduced the amount of time coding. Stressing good design improves the clarity of game components and enables more sharing. Students reported difficulty in meeting outside of class to complete and share code and were provided with several class sessions to work as a team and receive instructor support.
5) Multiple Feedback Sources- Students are used to receiving feedback from instructors. We felt it was important for them to receive feedback from users (middle school students), K-12 teachers, and peers as well. This approach enabled students to receive a clearer picture of their work. Requiring them to explain how they responded to the feedback places a strong emphasis on value of the feedback they receive.

We believe our findings and experiences can be utilized by Computer Science educators as well as educators from other disciplines implementing multidisciplinary courses and courses in educational game design.

8. ACKNOWLEDGMENTS

The first author is supported by National Science Foundation grant #DGE-0538476 and would like to acknowledge National Science Foundation grant #HRD-0331559 for previous support.

9. REFERENCES

[1] G. F. Fairchild and T. G. Taylor, "Using Business Simulations and Issue Debates to Facilitate Synthesis in Agribusiness Capstone Courses," in *Western Agricultural Economics Association Annual Meeting* Vancouver, British Columbia, 2000.

[2] R. M. Jones, "Design and Implementation of Computer Games: A Capstone Course for Undergraduate Computer Science Education," in *SIGCSE*, Austin, Tx, 2000, pp. 260-264.

[3] I. Parberry, T. Roden, and M. B. Kaxemzadeh, "Experience with an Industry-Driven Capstone Course on Game Programming," in *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, St. Louis, Missouri, USA, 2005, pp. 91-95.

[4] S. Mosiam and C. Hiemcke, "Interdisciplinary Capstone Group Project: Designing Autonomous Race Vehicles," in *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, Austin, Tx, 2000, pp. 270-274.

[5] Insomniac Games, Incorporated, "Insomniac Games:: Careers." Retrieved December 23, 2008 from http://www.insomniacgames.com/careers/programming.php#gp.

[6] NCSoft, "PlayNC.com - Jobs in Austin, Tx.", Retrieved December 23, 2008 from http://www.plaync.com/us/jobs/jobs_austin.php

[7] Sony Computer Entertainment America, Incorporated, "Sr. Software Development Engineer." Retrieved December 23, 2008 from http://www.us.playstation.com/Corporate/About/Jobs#.

[8] Y. A. Rankin, T. Lechner, and B. Gooch, "Team-Based Pedagogy for CS102 Using Game Design," in *SIGGRAPH 2007* San Diego, California, 2007.

[9] T. Barnes, E. Powell, A. Chaffin, and H. Richter, "Game2Learn: Building CS1 Learning Games for Retention," in *Proceedings of 12th Annual SIGCSE Conference on innovation and Technology in Computer Science Education*, Dundee, Scotland, UK, 2007, pp. 121-125.

[10] A. T. Chamillard, "Introductory Game Creation: No Programming Required," in *37th SIGCSE Technical Symposium on Computer Science Education*, Houston, TX, USA, 2006, pp. 515-519.

[11] S. M. Fisch, "Making Educational Computer Games "Educational"," in *2005 Conference on interaction Design and Children*, Boulder, Colorado, USA, 2005, pp. 56-61.

[12] N. R. Council, *How People Learn; Brain, Mind, Experience, and School*: National Academy Press, 1999.

[13] National Council of Teachers of Mathematics, "Overview: Standards for Grades 6-8.", 2008.

[14] Federation of American Scientists, "Summit on Educational Games," Federation of American Scientists, Washington, DC 2006.

[15] J. S. Underwood, C. Hoadley, H. S. Lee, K. Hollebrands, C. DiGiano, and K. A. Renninger, "IDEA: Identifying Design Principles in Educational Applets," *Educational Technology Research and Development,* vol. 53, pp. 99-112, 2005.

[16] G. W. Zommerman and D. E. Eber, "When Worlds Collide! An Interdisciplinary Course in Virtual-Reality Art," in *SIGCSE*. vol. 33 Charlotte, NC, USA: ACM, 2001, pp. 75-79.

[17] J. McDonald and R. Wolfe, "Using Computer Graphics To Foster Interdisciplinary Collaboration In Capstone Courses," *Journal of Computing Sciences in Colleges,* vol. 24, pp. 83-90, 2008.

[18] Student, "Dear Suzanne," S. Alejandre, Ed. Philadelphia, 2006, personal communication via email

[19] "Adobe Flash CS4. Retrieved from http://www.adobe.com/products/flash/"

[20] Lifelong Kindergarten Group, "Scratch," Retrieved December 23, 2008 from www.scratch.mit.edu.

[21] The Math Forum, "The Math Forum @ Drexel." Philadelphia, Pa, 2008.