

A Hybrid Approach to Projects in Gaming Courses

Amber Settle
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5324

asettle@cti.depaul.edu

Joe Linhoff
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5861

jlinhoff@cti.depaul.edu

André Berthiaume
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5325

berthiaume@cti.depaul.edu

ABSTRACT

We describe an approach to projects used in game development courses that supports learning individual skills while also developing team skills. Early assignments focus on developing individual skills in coding and content creation, and when those skills are honed, students form teams to work on a larger and more complex game. Classes that use a hybrid approach, that is individual projects that build toward a large group project, allow students to solidly learn game development skills required of gaming graduates and yet stimulate creativity and challenge students to move beyond their comfort zone.

Categories and Subject Descriptors

K.3.2 [Computers and Information Science Education]: *game development, group projects.*

1. INTRODUCTION

Group projects are ubiquitous in computer science education and with good reason. Students need to be prepared to work on a team once they graduate, and educators have a responsibility to ensure that students develop the necessary skills to do so. Computer game development programs are no exception. Game development is nearly always a team effort, and gaming companies want to see evidence that students are team players, can develop games of a reasonable size and complexity, and can produce a game demo that showcases their skills [8]. It is difficult for students to achieve these goals in courses that do not make extensive use of group projects.

There are, however, a number of pitfalls inherent in technically-oriented team projects. Incompatible personalities can create tensions that prevent students from working together effectively [1]. Student teams often take an ad-hoc approach to a project, and the lack of structure can hinder progress [1], although the use of a team manager or leader has shown some promise in this area [2]. Determining the ideal number of students in each team can also prove challenging. Small groups allow better logistical control but larger groups create more possibilities for creativity and specialization [11]. Grading is another difficult issue. Judging how much each member of a team has contributed to the final product is crucial to ensure that the project is assessed fairly, and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The Third Annual Microsoft Academic Days Conference on Game Development in Computer Science Education, February/March, 2008, Celebrity Century.

Copyright 2008 ACM 978-1-60558-057-9/08/02...\$5.00.

there are a variety of sources of information that can be used when making this judgment, from minutes of team meetings to peer evaluations [10]. However, not all sources of information are equally valuable and finding a good balance is difficult.

Students have more positive experiences with group projects when instructors provide them with information and guidance about how to work together [3]. Instruction about interpersonal skills, the encouragement of positive interdependence, and reflection on the group process are all important. However, the conditions for group learning in higher education settings rarely meet the high goals advocated by cooperative learning scholars [5]. This is particularly true in courses where the main goal is to learn technically challenging material, as is the case for game development courses. A great deal of time in the course is spent conveying technical information, and there may be little time available for other topics. This situation is particularly noticeable at institutions using the quarter system.

Independent of group projects, creating effective game development courses is a challenge. Creativity is crucial for would-be game developers and yet they also need to learn structured programming processes that will allow them to develop games under tight deadlines [6, 8]. Student morale and creativity depends on an investment in the games being developed [8], but students often lack the experience to judge how much effort is required to implement their ideas. Ensuring that students have a quality project to demonstrate their abilities in a positive way is important for their employment opportunities [8].

For all of these reasons, creating game development courses that use group projects is at best a balancing act. Encouraging creativity, maintaining morale, improving team dynamics and project structure, enforcing deadlines, and judging the quality of the final result is a challenge for any instructor. In this paper we describe an approach to team projects in game development courses that simplifies this process. Our approach balances the need for structure and the development of individual coding and other technical skills with the need to inspire students to be creative and encourage enthusiasm. This hybrid approach has a number of advantages and addresses some of the challenges seen in game development courses and in managing group projects.

2. A HYBRID APPROACH

To successfully balance the many competing, and at times contradictory, demands in a game development course that uses group projects, we have developed a hybrid approach to projects in our gaming courses. In this approach, individual projects are used in the early part of the course. The main goal of these projects is to ensure that all students have a working knowledge of

the technical skills required for the platform used in the course. A secondary goal of the individual projects is to provide the instructor with knowledge of each student's strengths and weaknesses. The individual projects culminate in a comprehensive final group project.

Depending on the course, students are either required or encouraged to form teams for the final project, which is an expansion and extension of the final individual project. In one course, the final projects are chosen from the best of the comprehensive individual projects, adding a competitive atmosphere to the course and providing more structure for team roles during the final project. The teams ideally have members with all the skills necessary to produce a strong project, although this is not the case in both of the courses we discuss in this article.

This approach is used in several game development courses taught at our institution. The platform used in the course, whether teams are required or simply encouraged, the size of the teams for the final project, how teams are selected, and the goals of each course differ. In the remainder of the section we describe our game development program and provide details for each course that uses the hybrid project approach.

2.1 Computer Games Development Program

DePaul University's School of Computer Science, Telecommunications, and Information Systems (DePaul CTI) is one of the largest and most innovative information technology institutions in the country. Over 2,000 graduate students and nearly 1,000 undergraduates are enrolled in 14 Bachelors and 15 Masters degree programs, ranging from traditional programs like Computer Science to degrees focused on the digital arts such as Digital Cinema and Animation. Such a broad range of degree programs is highly unusual and has attracted large numbers of students during a period of overall decline in technology education enrollments [9]. Benefiting most from the influx of new students with an interest in the digital arts is the Computer Games Development program. DePaul was one of the first liberal arts universities in the country to offer a four-year degree in the area. In only 4 years the gaming program has grown to be the second largest at DePaul CTI with 150 majors.

All of the courses that use the hybrid approach to game development projects were created for the Computer Games Development program. GAM 244: Game Development I is a freshman-level course in game design and development that uses Game Maker. It is designed to ensure that first-year students have experience with game development before encountering their first C++ course, providing them with motivation to learn programming [4]. GAM 374: Action Games Programming is predominantly a programming class where the students learn to make 3D action games in C/C++ and OpenGL. The prerequisite for the course is Programming Languages II which is an intermediate course in C/C++. GAM 380: Console Game Development Environments also uses the hybrid project approach. Since GAM 380 is being offered for the first time during the Fall 2007 quarter, it is not included in this work but discussed in a separate article [7]. It should be noted that all undergraduate courses at DePaul University are taught on the quarter system, with 10 weeks of classes followed by a final exam week during which final projects are due.

In the remainder of this section we discuss the structure of GAM 244 and 374, the way in which the hybrid project approach was used in each class, and the benefits and drawbacks to the approach.

2.2 Course Structure

GAM 244 and 374 are very different game development courses with different goals and audiences, yet both utilize the hybrid approach to course assignments.

2.2.1 GAM 244

The purpose of GAM 244 is to expose the students to the entire game production cycle, from original idea to detailed specifications, to completed implementation. This is a challenge as this course does not have any prerequisites, is required by students in the Computer Games Development and Digital Cinema Programs, and is an option in a host of other programs including Computer Science, Interactive Media, and Information Technology. To simplify the process of teaching the complete game design process to a very diverse group of students, the course uses Game Maker as its tool and focuses purely on 2D games.

To ensure that everyone in the class experiences the design process as a whole and moves out of their respective comfort zones, a hybrid approach to assignments is used. The students have a total of five projects over the course of the quarter. The first four projects are all done individually and the final project is completed by groups of three or four students. The first project is a fairly simple exercise to get students acquainted with Game Maker. The next three projects all require students to implement fully functional games, with each project having progressively more complex requirements. Specifically, the first deliverable is a single-level game, the second a multi-level game, the third a functional demo, and the fourth a complete design document and implemented game. The deadlines for the first four projects are spread over the first eight weeks of the quarter. It should be noted that the grading of the first four projects does not take into account the complexity of the game, but rather how well the game idea was implemented. By the end of the eighth week of the quarter, all students are capable in principle of performing any task in the game design process.

The fifth and final project is quite different from the rest. Based on the design documents for the fourth project and the implementation quality of the resulting game, roughly a quarter of the fourth project deliverables are competitively chosen as the final projects for the course. Students whose projects are selected become the lead designer for their team. Two or three other students are also assigned to the team. A few days before the final project, all selected projects are available for the students to review. Students can then choose the game on which they would most like to work, with a maximum of four students per team. The only intervention by the instructor is to spread the skill sets more or less equally, for example, by making sure that there is no team consisting only of coders or a team without any coders.

It is important to note that the games selected for the final project are not necessarily the ones with the best implementation. Games are selected by the instructor based on how well the game ideas are presented in the game design document, how much the game

can be expanded in an interesting direction, and how well the implementation supported the design document. This in effect turns the fourth project into a game proposal, a good experience for the students. One might imagine that a competitive game proposal near the end of the quarter would be stressful for students. However, the diplomatic approach taken by the instructor of emphasizing the positives of being chosen while downplaying the negative aspects of not being chosen has made this a nearly uniformly positive experience for the students. While this is an interesting pedagogical technique in a game development course, we are unaware of any literature on the subject.

2.2.2 *GAM 374*

As mentioned previously, GAM 374 is a development course that focuses on the creation of 3D games in C/C++ and OpenGL. Other tools used in the course include source control, a wiki, and an educational game engine. Using a source control system eases communication between the instructor and students, provides a means for distribution of starter kits and other files, assignment submission, and provides students with a sandbox area. Communication between students is improved with the use of a wiki, a suggestion by Alex Seropian of Wideload Games, one of the DePaul CTI Gaming Advisory Board members. There is also a common instant messaging system for the course, used extensively for contact with the instructor.

The course is divided into eight milestones. The first seven milestones are the creation of a hello world program, a design document, the 3D display of moving objects, a project utilizing input and sounds, a 2D display with texture mapping, a project involving collisions, and the integration of all the previous projects into a game. The first seven milestones are done individually. These serve to guide the students through the fundamental elements of game programming. Milestones one through six are self contained and do not depend on prior milestones. The seventh milestone integrates the first six milestones into a game-like program. The students are then given the choice of forming 2-3 person teams for the final project.

2.3 Course Results

There are both benefits and drawbacks to the hybrid approach to projects taken in the DePaul CTI game development courses. In order to understand the experiences in each course, we first present data regarding student performance. We then discuss the results and outline the benefits and drawbacks of the approach.

2.3.1 *Student performance*

There have been a total of eight sections of GAM 244 and one section of GAM 374 offered prior to the Fall quarter 2007.

2.3.1.1 *GAM 244*

A total of 172 students formed 48 teams across 8 sections of GAM 244 between Spring 2005 and Spring 2007. To measure how the quality of individual training contributed to improving the quality of the work produced by teams, we used the following approach. Since Projects 1 through 4 served as the training phase, any student who did not achieve a passing score at that point was considered untrained and therefore removed from our data set. This left 139 students. All teams remained viable for our analysis.

Of the remaining students, we computed the difference between their score on the final project minus their score on Project 4. Project 4 and the final project are fairly similar in terms of scope and goals, and therefore one would expect that a team of four properly trained students would be successful once placed on a single team project.

Of the 139 students in our data set, 79 (57%) of them increased their score in the final project while 60 (43%) of them lowered their score. Students who did show an increase went up by an average of 18.3 point whereas those who dropped did so by an average of 13.0 points.

Often on a team project the hard work of the better students benefits the weaker students. By removing from the data set students who were failing the class by the end of Project 4, we reduced the impact of this effect on our analysis. Among the 139 students in the data set, the average score on project 4 was 75.8 while the average on the final project was 80.6. One may reasonably infer that while teams were composed of students with varying degrees of skills, all students had the ability to make significant contributions to the final project.

Anecdotal evidence based on the self-report completed by the students at the end of the final project show that, in general, the success of final project hinged on team dynamics. Cohesive teams with good communication and organization get the job done. But perhaps most interestingly, of all the final projects that ended badly, not one student has ever reported lack of preparation as the main cause of the problems.

It should be noted that the data is not as supportive of the anecdotal evidence as we would like. Particularly frustrating is the percentage of students who do not achieve competency in individual work and are dropped from the analysis in this section. As educators at an institution that focuses on first-generation college students, we are highly motivated to ensure that all students in our courses succeed. However, as noted earlier, GAM 244 is an introductory course that serves both computer game development majors and other DePaul CTI majors. Finding a way to provide the majors with the skills needed to succeed in their program while enabling non-majors to thrive is a distinct challenge. As the digital arts programs at DePaul mature, a better balance may be achieved by creating multiple introductory game development courses.

2.3.1.2 *GAM 374*

As indicated above, students had a choice in GAM 374 as to whether to join teams for the final milestone. Seven of the fifteen students in the course (47%) joined a team for the final project. The grades of the three team projects were close to the average grade of the strongest individual on that team. See Table 1 for the course data. Also, one of the three teams was over ambitious and produced only a partially working game.

Eight of the fifteen (53%) students continued to work alone on the final project. Motivations for continuing to work alone, rather than team up with other students, included confidence in programming ability, personality conflicts, and a strong desire to see their own game design come to life. Interestingly, even though the course favored programmers, artists did not always join programmers for the final project. The grades for the final project for individuals working alone generally show a drop from their averages over the other seven milestones. Again, see Table 1

for the complete data. Two of the eight individuals created only partially working games. The most advanced game and the most creative game was produced by individuals.

| Team number | M 1-7 average | M8 | Delta |
|-------------|---------------|----|--------|
| None | 50 | 87 | 37 |
| None | 92.14 | 87 | -5.14 |
| None | 91.14 | 90 | -1.14 |
| None | 80.86 | 60 | -20.86 |
| None | 99.71 | 90 | -9.71 |
| None | 83.29 | 90 | 6.71 |
| None | 81.43 | 90 | 8.57 |
| None | 95.43 | 95 | -0.43 |
| 1 | 74.14 | 85 | 10.86 |
| 1 | 68.93 | 85 | 16.07 |
| 1 | 86.86 | 85 | -1.86 |
| 2 | 89.14 | 87 | -2.14 |
| 2 | 86.14 | 87 | 0.86 |
| 3 | 74.43 | 95 | 20.57 |
| 3 | 92.71 | 95 | 2.29 |

Table 1: Student performance in GAM 374

2.3.2 Benefits

One of the clearest benefits of the hybrid approach to course assignments is that it provides the instructor with detailed knowledge about each student's strengths and weaknesses, allowing the instructor to form more effective teams. Students who excel at programming are ideally paired with creatively-talented students, producing a more balanced team. Of course, not every team will be ideally balanced as there may be an imbalanced skill set distribution in the class.

The competition of midterm projects in GAM 244 also motivates the students. It allows students with creative ideas or very strong programming skills to achieve recognition for their efforts. It also puts them in a leadership position for the group project, giving them ownership of the game. This can be both a reward and a punishment depending on their teammates. In either case it is certainly a good learning experience, as it allows the strong students to stretch themselves.

The creation of balanced teams and the structure imposed by a team leader improves the chances that quality projects will be produced. When possible, this can be encouraged by the formation of a small number of strong teams. This has been particularly utilized in GAM 244. An instructor who groups notably talented and/or motivated students on the same team can encourage those students to push themselves and produce a high-quality project, one that they can potentially use as a showcase of their talents when job hunting.

2.3.3 Drawbacks

Overall, the lack of time in a 10-week quarter is a significant problem. The hybrid approach to projects makes this worse, since students have a number of different assignments to complete often in a week or less. The time for the final project is also very short since much of the quarter is spent on individual work.

One of the most difficult things for students, or anyone trying to develop a game in a team environment, is finding ways to efficiently work on a project. It takes time to develop a good

working relationship with team members and that relationship is crucial for team success. In a course where the majority of time is spent working alone, students have less time getting to know each other and forming good working relationships. In GAM 244 this is particularly obvious when the team leader is weak, and in GAM 374 it manifests itself in poor stabilization of team dynamics.

Another drawback to this approach is the difficulty in finding an appropriate metric to measure its effectiveness. While we have good anecdotal evidence that this way of structuring course assignments is beneficial to students, we do not have compelling data to support those conclusions.

3. CONCLUSION AND FUTURE WORK

Anecdotal evidence suggests that a hybrid approach to assignments in a game development course has a number of benefits. Students gain stronger individual skills in each area covered by the course. Instructors gain detailed knowledge about student's strengths and weaknesses and can use this information to better assign students to teams and to evaluate student performance. When competitive individual assignments are used, students are more motivated to work hard, to stretch themselves, and to gain valuable experience as a team leader. Student creativity is also encouraged, and in the case of competitive individual assignments, rewarded with public recognition.

Unfortunately, it is difficult to assess the effectiveness of this approach and our data does not clearly support the anecdotal evidence. We believe there are two main reasons for this. First, the metric of individual grades versus team grades does not adequately address the contribution of each student toward the team work. If we are to measure how much hybrid projects improve each student's individual skills, we need more information. For example, we need to know what skills the students brought into the class and the precise contribution of each student to the final project. We could then measure if the improvement in individual skills is a result of the initial course projects, and if so, how that improvement manifests itself in the finished project. It would also be helpful to measure student experience with and attitudes toward group work since students' prior experiences on teams influence their interaction in group projects, the development of interdependence among team members, and issues such as conflict management and communication [5]. Second, individual skills are only a part of what each person contributes to a team. One of the most important things that a team member can do is to work efficiently, with as much of her work as possible ending up in the final product. To determine if the hybrid approach improves this aspect of team performance, we need to find a way to measure if the individual projects help students become more efficient.

There are several changes independent of hybrid projects that we anticipate making to the GAM 374 course. At the suggestion of Eugene Jarvis, another member of the DePaul CTI Gaming Advisory Board, the new version of the course will focus more clearly on the creation of complete game programs. To do this, two mini-game assignments will be added. The design document milestone will be changed to an in-class activity and the integration milestone will be eliminated in favor of one of the mini-games. For the final game project, the students have the option of extending one of the mini-games or working on their

own game design. These changes will increase the number of working games a student creates.

4. REFERENCES

- [1] Amzad, I. and Ortiz, A. J. 2006. Model Based Project Centered Team Learning. In Proceedings of the Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, (Bologna, Italy, June 2006).
- [2] Banerjee, S. 2007. Encouraging Equal Participation in Student Team Projects. In Proceedings of the Conference on Frontiers in Education: Computer Science and Computer Engineering, (Las Vegas, Nevada, June 2007).
- [3] Bosworth, K. and Hamilton, J. 1994. Collaborative learning: Underlying processes and effective techniques. *New Directions for Teaching and Learning*: 59.
- [4] Chamillard, A. 2006. Introductory Game Creation: No Programming Required. In Proceedings of the SIGCSE Technical Symposium on Computer Science Education, (Houston, Texas, March 2006).
- [5] Colbeck, C., Campbell, S., and Bjorklund, S. 2000. Grouping in the Dark: What College Students Learn from Group Projects. *Journal of Higher Education*: 71(1), 60-83.
- [6] Doughty, M. 2004. Computer Game Development Education at University. In Proceedings of the 5th International Conference on Computer Games: Artificial Intelligence, Design, and Education, (Reading, United Kingdom, November 2004).
- [7] Linhoff, J. and Settle, A. Teaching Game Programming Using XNA. Unpublished manuscript.
- [8] Parberry, I., Roden, T., and Kazemzadeh, M. 2005. Experience with an Industry-Driven Capstone Course on Game Programming. In Proceedings of the SIGCSE Technical Symposium on Computer Science Education, (St. Louis, Missouri, February 2005).
- [9] Perkovic, L. and Settle, A. 2007. Computing Branches Out: On Revitalizing Computing Education. In Proceedings of FECS 2007: The International Conference on Frontiers in Education: Computer Science and Computer Engineering, (Las Vegas, NV, June 2007).
- [10] Wilkins, D. and Lawhead, P. 2000. Evaluating Individuals in Team Projects. In Proceedings of the SIGCSE Technical Symposium on Computer Science Education, (Austin, Texas, 2000).
- [11] Wolz, U. and Pulimood, S. M. 2007. An Integrated Approach to Project Management through Classic CS III and Video Game Development. In Proceedings of the SIGCSE Technical Symposium on Computer Science Education, (Covington, Kentucky, March 2007).