Teaching interdisciplinary teamwork through hands-on game development

Prof. Scott A Kuhl, Michigan Technological University

Scott Kuhl is an Assistant Professor of Computer Science and an Adjunct Assistant Professor of Cognitive & Learning Sciences at Michigan Technological University. He received his Ph.D. in Computer Science from the University of Utah in 2009. He has been the faculty advisor for Husky Game Development Enterprise since Spring 2010. His research interests include immersive virtual environments, head-mounted displays, and spatial perception. A link to his web page can be found at http://www.cs.mtu.edu/.

Dr. Robert Pastel, Michigan Technological University

Robert Pastel is an Associate Professor of Computer Science and an Associate Assistant Professor of Cognitive & Learning Sciences at Michigan Technological University. He received his Ph.D. in Physics from the University of New Mexico. Robert help create Husky Game Development Enterprise in 2004 and was advisor to 2010. Robert's research interests include mobile device interactions and software development. A link to his web page can be found at http://www.cs.mtu.edu/.

Ryan George, Michigan Technological University

Ryan George is a 4th-year undergraduate student at Michigan Technological University pursuing degrees in both Computer Science and Computer Network/Systems Administration, with expected graduation in Spring 2015. He has been a member of the Husky Game Development Enterprise since Spring 2011, and was elected President in Fall 2012. He served as Co-Chair of the Enterprise Student Advisory Board from Fall 2011 to Spring 2013, during which he represented the Advisory Board on the University's Enterprise Governing Board.

Mr. Chad M Meyers, Michigan Technological University

Chad Meyers is a 3rd-year undergraduate student at Michigan Technological University pursuing a degree in Computer Science. He has been the Vice President of Communications for Husky Game Development since Fall 2013, and has been a participating member of the organization since Fall 2012.

Matthew L Freitag, Michigan Technological University

Matt Freitag is an undergraduate student in Computer Network and Systems Administration (CNSA) at Michigan Technological University. He is also the Vice President of Technology for the Husky Game Development (HGD) enterprise, which he has been a part of since April of 2012. His interests include cloud computing, home automation using wireless technologies, software development operations, and (of course) video games.

Jacob M. Lund, Michigan Technological University

Jacob Lund is a 4th-year undergraduate student at Michigan Technological University pursuing a degree in Computer Science with a minor in Mathematical Sciences. He has been the Vice president of Finance of Husky Game Development Enterprise since Fall 2012 and has been involved with the course every semester since Spring 2010.

Mr. Michael Paul Stefaniak

Teaching interdisciplinary teamwork through hands-on game development

Abstract

Husky Game Development (HGD) is an innovative project-based course at Michigan Technological University that gives a diverse set of undergraduate students the opportunity to develop video games in a business-like setting. The course allows students to exercise their technical and artistic skills on a multi-semester project while also developing their teamwork, problem-solving, leadership, and communication skills. Many of our former students now have careers in the video game industry or in a computer science related field. The course is modeled after a business where the instructor assumes the role of chairman or advisor and the students elect their own management and form teams to work on projects. Although most of the students are pursuing computer science related degrees, HGD also attracts students interested in 2D art, 3D art, sound design, and electrical engineering. The diversity of students in HGD gives students the unique opportunity to learn how to work effectively with people who have different abilities, backgrounds, and interests. This paper focuses on the policies and techniques that we have successfully used to attract, organize, motivate, and evaluate the students in the course. We also provide analysis of enrollment in Husky Game Development from Fall 2011 through Spring 2014 and describe how the faculty advisor and student management share and delegate responsibilities. We hope that HGD can serve as one possible model for instructors at other institutions who desire to implement a similar course.

Introduction

Traditional computer science undergraduate courses often fail to give students hands-on experience which will help them learn how to work with a team of peers to propose, design, and develop large applications. Traditional courses typically consist of detailed lectures and well-defined assignments that students complete independently. However, in many real-world projects, a diverse team composed of programmers, artists, human-computer interaction specialists, management work together to create a product. To prepare students for success in the real-world, they must be able to manage their time, delegate responsibilities, respond to unforeseen problems, identify solutions to problems, communicate, take advantage of the flexibility that they are given, and report problems to a supervisor. Husky Game Development (HGD) gives students the opportunity to work on large-scale video game projects in an environment that mimics the real-world. It also provides students with management experiences that are uncommon in traditional computer science curricula. HGD is a part of a broader initiative at Michigan Tech called the Enterprise Program which supports student-driven, multi-disciplinary, and hands-on courses.

Besides giving students the opportunity to learn about video game development, HGD has four primary goals. First, HGD aims to have students gain first-hand teamwork experience on diverse teams. HGD is diverse because it includes freshman through seniors, includes students from multiple majors, and attracts underrepresented students. Second, HGD seeks to use project-based learning to give students experience working on large multi-semester projects. Third, HGD should provide an environment where student managers can manage their peers and find ways to resolve unforeseen problems. Fourth, we hope that innovative methods used in HGD might attract more students interested in computer science, art, and entrepreneurism to Michigan Tech.

A significant body of research suggests that integrating gaming programming and project-based learning into computer science curricula can capture student interest in computer science. For a summary of institutions which have attempted to integrate these ideas into curricula, see.[1] For example, some[2] have found that teams of students working collaboratively on game projects can positively impact students' attitudes about computer science, programming, and game design and providing enough time for students to complete is important. Other studies show that game development can increase student interest in computer science[3] and software engineering.[4] Student interest in CS may also be increased by integrating mobile game development into breadth-first introductory computer science courses.[5]

The primary purpose of this paper is to describe HGD in sufficient detail so that it is possible for other colleges and universities to create similar courses. Because HGD is radically different than traditional courses in everything ranging from enrollment and assignments to grading and responsibilities, we attempt to provide detailed information and guidelines about the structure of HGD.

Enrollment and teams

Any student can enroll in Husky Game Development for as many semesters as they wish as long as they can contribute to the overall goal of developing games. We generally discourage first-semester freshman from enrolling until they become familiar with being an independent student at Michigan Tech. Depending on how long the student has been at Michigan Tech, HGD is worth either one or two credits. Although most of our students are pursuing computer science degrees, we also have 2D artists, 3D artists, sound design majors, software engineering majors, computer engineering majors, scientific and technical communication majors, and business majors. Individual departments on campus can decide if and how the credits might count toward their degree. Many departments allow some or all HGD credits to count as electives or as an alternative to a senior project. We expect students to put in approximately three to five hours of time per week into the course outside of the regularly scheduled meeting times.

All students in HGD are either assigned to a team or are in student management. As HGD has grown in size, we have found it useful to implement a hierarchy (Figure 1) to organize the enrolled students. Sometimes exceptions are made to this hierarchy when the faculty advisor

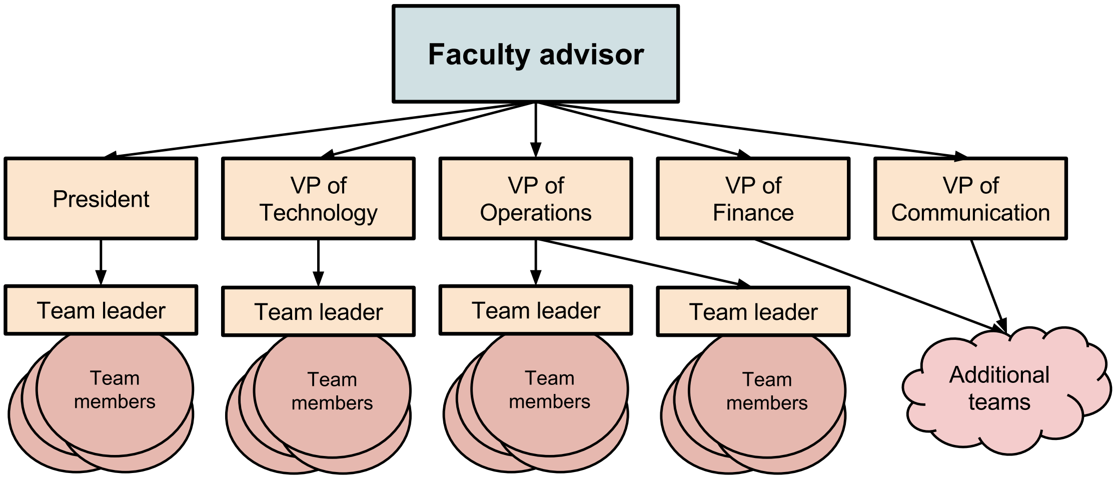

Figure 1: HGD uses a hierarchical division of responsibilities where five student managers are each responsible for one or more teams of students.

works closely with a team on an industry sponsored project, or when a student management member serves as both a team leader and a management member. For similar courses that have fewer students enrolled, it may be appropriate to have fewer management members. Most of the students in HGD are members of a team. Because HGD is open for anybody to enroll in, teams have a diverse group of students assigned to them ranging from freshman to seniors, artists, computer scientists, computer engineers, etc. Teams must identify a time that they can meet weekly and are encouraged to set up several "WorkJams" throughout the semester where the team spends several hours working together on their project. The team leader is responsible for organizing the team-meetings, making sure that everybody on the team has something to work on, and regularly checking to ensure that individual team-members are making progress on their tasks. We encourage team leaders to avoid being dictators, to make sure that younger team-members have the help that they need, to delegate responsibilities without micromanaging everybody, and to immediately report any concerns to the management member assigned to their team (or to the faculty advisor). One of the most common problems with inexperienced team leaders is an inability to recognize or report a problem. We try to counteract this by regularly reminding all team-members, team-leaders, and student management that problems are unlikely to be resolved if not reported up the hierarchy in Figure 1.

Student management

The five student managers illustrated in Figure 1 are not expected to perform game development. These managers, along with the faculty advisor, are collectively responsible for identifying and resolving problems within the teams, generating material for the weekly general meetings, grading, and reaching out to potential industry sponsors. In addition, individual management members have other responsibilities outlined below. When a management position is available, any student that applies is interviewed by the current student management. After the interviews and an advisory vote from all of the students in the course, student management selects one of the applicants to fill the position. The faculty advisor can override the decision made by student management if necessary. Managers keep their position until they leave HGD, voluntarily retire their position, or until someone successfully challenges them for their position. Management and the faculty advisor meet for an hour immediately before or after the general meeting. These meetings are open to anybody in HGD to attend except when management and the faculty advisor discuss grading.

- **President:** The president is responsible for setting the agenda of management meetings and working with others in management to ensure that others in student management are completing the tasks that they are assigned.
- **VP of Operations:** The VP of Ops is primarily responsible for assigning teams and organizing the content of the weekly general meetings.
- **VP of Communications:** This student manager is responsible for maintaining the HGD website (http://www.huskygames.com/), finding and maintaining industry connections, and recruiting potential students.
- **VP of Technology:** HGD has a small lab of computers that we maintain. The VP of Technology is responsible for maintaining these computers as well as the servers that host Redmine (project management and bug tracking software), Subversion (source code management software), and the HGD website.
- **VP of Finance:** HGD gets financial support via course fees paid by the students that enroll in the course and by industry sponsors. The VP of Finance is responsible for maintaining records of the accounts, coordinating the purchase of items for HGD, and interfacing with the CS department staff.

Meetings and course schedule

HGD holds a one hour weekly “general meeting” over the 14 weeks in the semester. Everybody enrolled in HGD must attend these meetings. We have scheduled our meetings in the evenings to decrease the likelihood of time conflicts with other courses. General meetings typically fall into one of two categories: (1) lectures and (2) progress updates from individual teams. The lectures can be on a variety of topics and are the responsibility of student management to arrange—although sometimes student management might delegate the responsibility of a particular lecture to another knowledgeable student. These lectures include topics such as:

- Communicating the course expectations to students.
- Guest speakers from industry on topics related to game design and development. We frequently use video-conferencing technology so guest can give lectures without visiting campus.
- Guest speakers from others on campus who overlap with game design. For example, faculty with experience in digital art, sound design, artificial intelligence, virtual reality,

networking, human-computer interaction all have useful knowledge that can be shared with students in HGD.

- Tips for finding jobs in the gaming industry or industry in general. Guest speakers from Career Services often provide useful advice for improving resumes and making a good impression during interviews.
- Suggestions of how to design compelling games
- Tutorials on using different game engines
- Providing tips to help teams effectively communicate their game idea and status during regular progress updates that the teams give to the students in the course.

Over the course of the semester, teams must give three progress updates at a general meeting. Since the enrollment in HGD has increased, two general meetings are required for all teams to give their presentations—resulting in six of the 14 general meetings being dedicated to progress updates. A typical progress report requires evidence that the team has made progress toward completing their game such as screenshots, images, videos, or live demonstrations of the game. At the last progress update of the semester, management typically requires a live demonstration of the game.

Although exceptions are sometimes made, we aim to start projects at the beginning of the school year and complete them by the end of the school year. We have found that extending a project beyond a single year is possible but often challenging since the team composition may have to change significantly when students graduate. The teams are also adjusted slightly between the fall and spring semesters as students are not required to enroll during any particular semester. At the beginning of the fall and spring semesters, every team-member is asked to complete an electronic survey indicating their major, experiences, and interests so that management can assign them to an appropriate team. Students must be quickly assigned to teams by the faculty advisor and management so that the team can promptly begin work. We also ask every student to submit a resume within the first few weeks of the semester so that we have them on-hand if an industry sponsor expresses interest in sponsoring a team of students in HGD.

At the beginning of the fall semester, game ideas must be identified. We have tried various techniques to generate and select game ideas for the teams to work on. For example, one year we had every student in the course write a game proposal and then management tried to filter out the best ideas and build teams around them. This process was time consuming and resulted in some students being put on games that they were not interested in. More recently, we have found that it works better to make teams meet for several hours over the first weekend of the semester and develop a one or two page game proposal document. Once student management and the faculty advisor review the game ideas and give feedback to the teams, the teams can begin work on their projects. Teams pick the platform, programming language and game engine that they prefer—but management encourages teams to try to stick to technologies that multiple people on the team are familiar with.

Financial support and industry sponsorships

HGD receives financial support primarily through course fees charged to the enrolled students and from industry sponsorships. Although the faculty advisor is the official financial manager of the accounts, student management is largely in charge of deciding how the funds are used to support HGD. These funds are typically used to purchase software and hardware to support game development, to purchase food for teams to use during long work-sessions, and to purchase one HGD t-shirt for every member who doesn't already have one. Excess funds can be used to host HGD-related events or as honorariums for guest speakers.

Industry or other organizations can sponsor a Husky Game Development team for a school year to work on a project for the sponsor. These partnerships allow students to gain experience working on a real-world project and gives sponsors the opportunity to create connections with talented students who will soon be entering the job market. We work with potential sponsors to ensure that the scope of the project is appropriate for a team of students to accomplish in a school year. Although HGD does not guarantee to industry sponsors that the project will be successfully completed, the students and faculty advisor make the best efforts they can to satisfy the sponsor. The students working on industry sponsored projects have weekly or bi-weekly conference calls with the sponsor to discuss progress, questions and concerns.

Grading

We have devised an effective grading mechanism that results in few complaints from students and gives student management the responsibility of assigning grades. In general, we use co-assessment[6] techniques where students use self- and peer-evaluation but the faculty advisor maintains the ability to, on rare occasions, override the grades assigned by management. At the beginning of the semester, the faculty advisor and student management outlines what "assignments" are going to be graded and their approximate weights. Current grading is weighted as follows:

- **10% of grade based on miscellaneous assignments.** These assignments include: completing the beginning-of-semester skills survey, submitting a resume, submitting a copy of their game at the end of the semester, writing a game proposal document (fall semester only) and developing some kind of promotional material to accompany the game. These grades are assigned individually to students.
- **30% of the grade is based on the three progress update presentations that the teams give throughout the semester.** During the progress update presentations, the faculty advisor and student managers complete a grading form and leave comments for the team. The grading rubric typically has three components: (1) Evidence of progress since prior progress update; (2) Clear description of what is completed and what they plan to complete next; and (3) Whether the presentation was well-organized and easy to understand. After the general meeting, one student manager is responsible for averaging the grades, assigning the grades to the team-members in the gradebook, and giving the comments from the forms to the team leader. All members of a team typically receive the same grade.

- **30% of the grade is based on documentation.** Every student is expected to use a source code management system (SVN) and to use a issue tracking system (Redmine). Students are expected to create tickets in Remine for anything that they work on and are also expected to electronically log the amount of time they spent working on any particular ticket. Team leaders are expected to ensure that all team-members have tickets assigned to them, that tickets are up-to-date and organized, and that tickets have appropriate deadlines associated with them.
- **30% of the grade is based on evaluation forms.** We have created several electronic forms that allow students to evaluate themselves and their peers and provide detailed feedback about their experience in HGD that often helps the faculty advisor and management to refine the course. These individual evaluation forms are one way that we hear summaries from multiple perspectives of what happened within a team over the course of the semester. The evaluation forms also ask students to grade themselves and their peers: Team-members grade themselves and their team-leader; team-leaders grade themselves and all of their team-members; and management grades themselves and each other. Although self- and peer-grading can save the instructor time, it may also improve student learning.[6–8] Management goes through all of the feedback and, along with their own knowledge of the teams, assigns recommended grades for all students in the course. If management and the evaluations agree that one person on a team contributed very little, they will likely be given a low grade for both their documentation and evaluation form grades.

Most grading occurs after a discussion between student management and the faculty advisor. We discuss each student's grade, agree on a grade, give feedback to the student being graded, and enter the grade into the gradebook. Although the students create the recommended grades, the faculty advisor observes the process, provides input, and reserves the right to later override management's decisions. In a couple of situations, student management assigns grades without the faculty advisor being present: (1) When we grade presentations, student management and the faculty advisor write their evaluations down on a form and a single management member averages the results; (2) We designate a single student management member who is responsible for assigning grades for miscellaneous assignments (such as giving students credit for completing their beginning-of-semester skills survey). The faculty advisor and student management can review all grades at any time. Any unethical behavior where management might manipulate other students grades would be handled through the Michigan Tech's academic integrity hearing process. However, we have never had any problems in the history of HGD.

Enrollment statistics

HGD was started in Fall 2004 and the statistics in this paper show the results from Fall 2011 through Spring 2014. Figure 2 shows that enrollment has steadily increased since Fall 2011 and has consisted primarily of students pursuing computer science related degrees. Since Spring 2012, at least 1 out of 10 students pursuing CS-related degrees at Michigan Tech are enrolled in HGD (Figure 3). The number and composition of non-CS students enrolled varies each semester. A large number of students pursuing audio-related degrees joined in Fall 2013. They were enthusiastic about the experience, and an even larger number of audio-related majors are enrolled

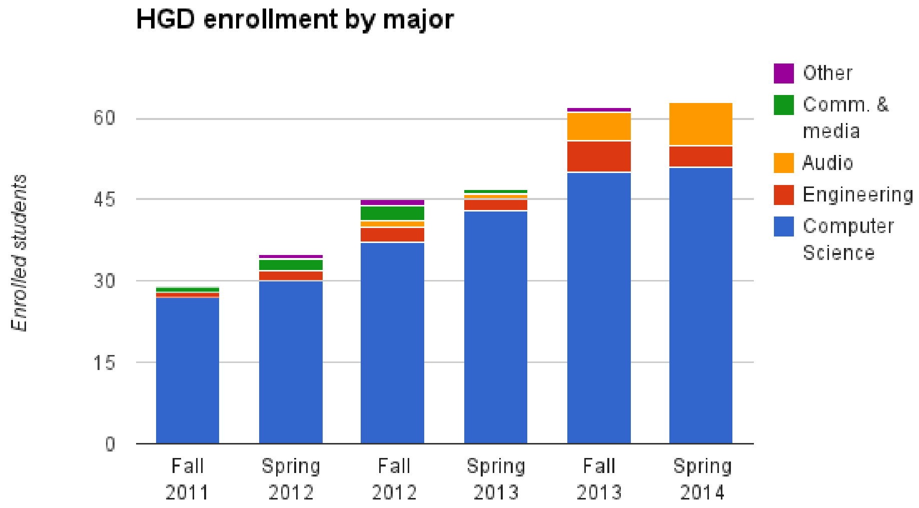

Figure 2: Total enrollment in Husky Game Development since Fall 2011. "Computer Science" includes all degrees offered by the CS department. "Engineering" includes electrical and computer engineers. "Audio" includes sound design, electronic media performance, and audio production majors. "Comm. & Tech" includes scientific and technical communication majors and media majors.

for Spring 2014. Some of the students enrolled contribute to HGD in ways that are not reflected by their major. For example, we regularly have CS students who want a primary focus on contributing 2D or 3D art and a secondary focus on programming.

Recent enrollment trends also indicate that HGD enrollments often have a higher percentage of women than the CS department (Figure 4). The higher percentage of women in HGD than in the CS department isn't necessarily due to the fact that HGD includes non-CS students. The gender ratios between CS-related majors and non-CS majors enrolled in HGD are typically similar. For example, in Fall 2013, 14% of the students pursuing CS-related degrees in HGD were women compared to a department-wide 6%.

When we sum all of the enrollment numbers from Fall 2012 through Spring 2014, we find that HGD is primarily comprised of sophomores, juniors and seniors (Figure 5). Although freshmen are allowed to enroll in HGD after they have been at Michigan Tech for a semester, students are most likely to join after they complete their freshman year.

Recent Projects

HGD has had numerous successful projects over the years since its inception. These projects cover a wide range of topics and purposes, from entertainment games like Arcane Brawlers which was released on XBox Live to training simulations including one developed to help ROTC cadets

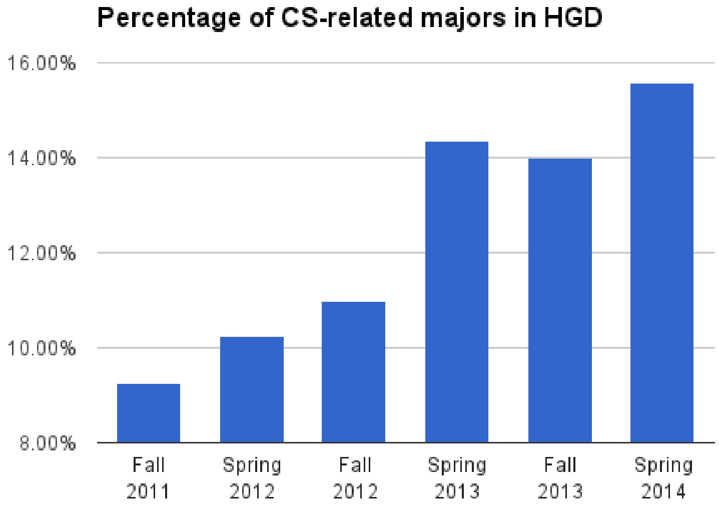

Figure 3: Percentage of CS-related majors at Michigan Tech that are enrolled in Husky Game Development.

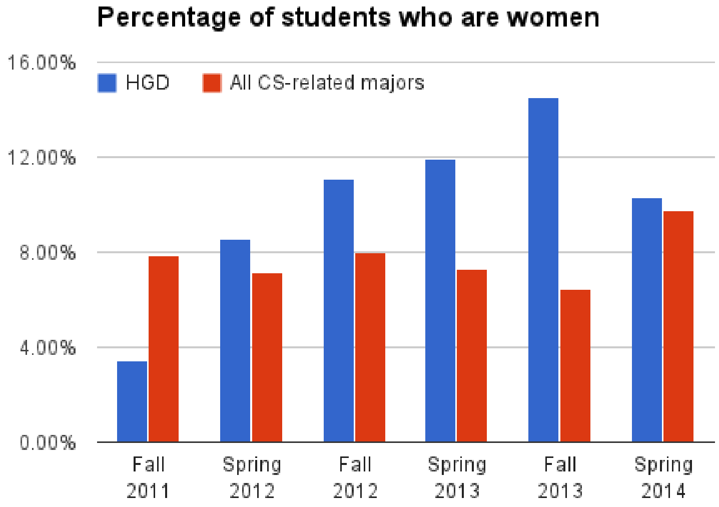

Figure 4: Percentage of women enrolled in HGD (out of all of the students in the course) compared to the percentage of women seeking CS-related degrees (out of all students seeking CS-related degrees).

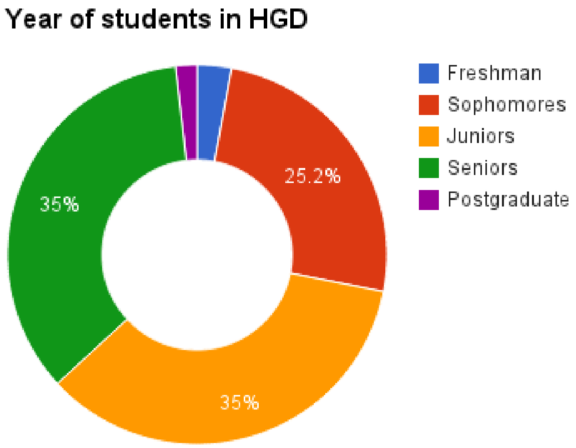

Figure 5: Enrollment in HGD by student's year at Michigan Tech (average from Fall 2011 through Spring 2014).

learn basic field tactics, movement, and mission protocols. In Fall 2013, HGD was actively developing ten different games that the students aim to complete by the end of Spring 2014. In this section, we highlight three higher-profile projects that we are working on.

RAM Truck Game: HGD is currently sponsored by Chrysler to develop a tablet-based game which features the towing capabilities and features of a popular truck. Initially, the game will be used by Michigan Tech's MindTrekkers in outreach events that target high school and middle school students at science festivals across the country. This project began in Fall 2013 and we are currently working to complete it sometime in 2014. The students and the faculty advisor have bi-weekly conference calls with the industry sponsor for the students to describe the progress that they have made and for the sponsor to provide feedback on the progress we've made.

BonzAI Brawl: HGD assigns one team to work on a platform that is used for an annual artificial intelligence programming competition. The competition is organized by several parties at Michigan Tech including Women in CS, HGD, faculty, the CS department, and graduate students. The competition in 2013 was the sixth annual event and attracted over 170 participants (including several faculty) from Michigan Tech and Northern Michigan University. The participants have eight hours to develop an artificial intelligence program that will play the game using the API developed by a team in HGD. At the end of the day, participants watch their programs compete against each other in a tournament.

Immersive Visualization Studio project: The faculty advisor for HGD, along with two other faculty at Michigan Tech, recently received an NSF MRI grant for infrastructure to support research and education. These funds supported the development of a large wall-sized screen consisting of twenty-four 46 inch high-definition televisions powered by a cluster of 8 computers. The space in front of the wall of screens is tracked by a twelve camera full-body tracking system.

Besides supporting research, HGD also uses the wall to give students hands-on experience with hardware that is typically only available for research at universities or for film or video game companies. This team is working on documenting how to use the full-body tracking system, integrating full-body motions recorded in the lab into HGD video games, and developing a game that people can play using both the display wall and the tracking system. The game that is currently under development involves displaying targets on the screen and then having the player move their hands and feet to align with the locations of those targets within a time limit. Besides giving students a unique learning experience, we hope to use this game as a fun demonstration for visitors who are touring the lab.

Other projects: The projects described above are highlights of the recent games developed by HGD, but are not representative of the average game the students develop. At the end of any particular school year, most of our games are playable but not completely polished. One of our goals is to increase the percentage of our games that are complete and polished. Many of these games are released on our website as a free to download game. Even if a game is not completed or widely played, the development of the game provides the students with significant learning opportunities. Besides learning how to work on an interdisciplinary team, they also learn about algorithms, networking, graphics, programming languages, and game engines. Since we allow students to choose the programming language and engine they develop in, they often pick technologies that they want to learn. For example, students have implemented games using programming languages such as Java, C, C++, C#, Flash, and JavaScript and have also used other technologies such as Unity 3D, Xbox, Xbox Kinect, OpenGL, DirectX and Android smartphones.

Guidelines used in HGD

We have developed a collection of guidelines for the faculty advisor and student management to follow that we believe has helped ensure the overall success of HGD. We have included these guidelines because we feel that they are an important component to making the course successful at Michigan Tech. However, we don't necessarily believe that all of these guidelines must be followed to create a successful similar course at another college or university. Furthermore, some of these guidelines may be impossible to follow when starting a program for the first time with a smaller number of students.

When a team (or student) fails to meet expectations, student management must ask themselves how they could have better supported that team (or student). We have found that it is easy for student management to assume that a team's problems are solely the fault of the team leader or one or more team-members. Therefore, the faculty advisor insists that management evaluate how they might have contributed to any significant problems that arise. For example, two common problems are: (1) a team has a serious communication breakdown or (2) a single team-member does not sufficiently contribute to the team. Management might contribute to these problems by failing to notice the problem in a timely manner or failing to regularly communicate with the team leaders and team-members of the teams that they are responsible for.

Student management and the faculty advisor must act quickly and decisively at the

beginning of a semester. Teams cannot begin work until member as assigned to the teams. Our goal is to have the teams organized within the first week of the semester. This requires that all students in the course have filled out a skills survey so that management can make intelligent team assignments. Since HGD has grown to 50 or 60 students in recent semesters, it is a challenging proposition to get every student to fill out the skills form and to get student management to organize the teams within days of the semester starting. Students who do not complete the skills form on time are randomly assigned to teams. However, if HGD has 50 students enrolled and we expect an average of four hours of work from each student, a one week delay in organizing teams is a loss of 200 person-hours.

Games should be simple. When we talk to students about developing game ideas, we repeatedly stress that their games should be simple. Mobile or tablet games are often a more appropriate project than a 3D first-person video game. One way to motivate students to pursue simple games is to focus on the idea of creating a game that they would be proud to show a potential employer. For example, a game that is technically complex but not complete, compelling, or fun to play may not be as impressive as a simple, complete, and fun game. There will be unforeseen development bugs that take a lot of time, time spent learning a new engine or programming language, problems with using source code management software, lost time due to trying to solve problems the wrong way, personal emergencies, and many other obstacles to completing a game. In addition, a polished and complete game needs artwork, sounds, an installer, testing on multiple computers, testing, and modifications based on feedback. Students will assume that they can complete a large project because they have two 14-week semesters to do so. However, each team-member is only committing an average of 4 hours a week and will therefore only contribute around 100 hours of work over a school year. This is equivalent to 2.5 40-hour work weeks per team-member. Therefore, we ask teams to imagine that we are asking them to work full-time for two weeks over the summer—and to propose a game idea that they believe they could actually complete in that time.

Games should be tested by others in HGD. At the end of each semester, we require all students to play at least two games developed by other teams and provide feedback about the games. This forces students to have a playable, but incomplete, game that their peers can play by the end of the fall semester. During the spring semester, teams primarily focus on adding features, adding levels or other content, and fixing bugs. We have found that many teams use the feedback that they receive to improve their games—despite the fact that we do not currently require teams to respond to the feedback they receive.

Develop a plan for releasing successful games. When a team is fully successful, they will have a game that could be released, marketed and sold (or given away) to the public. We have generally focused on trying to make free games successfully before focusing our attention on selling games. It is important to consider any licensing restrictions that might be applied to the game engine or other software that the students used to develop the game. For example, when a team uses a GPL licensed engine, they will be forced to release the game under a GPL license. If a game uses a proprietary game engine, it may be necessary to license the engine from the company that created the engine. Proprietary engines that are freely available for educational use may not allow for students to use the engine to make a game to sell without a different license. A plan for how to distribute the income generated by a game is important. At Michigan Tech, we have established a mechanism which allows students to receive some royalties for games that we sell—but have only had the opportunity to try this once with the release of a game called Arcane Brawlers on Xbox Live. Since students graduate, or move on to new games, we have found it nearly impossible to maintain released games. In the future, we hope to mitigate this problem by creating a plan that can effectively motivate everybody to fix bugs and release updates.

Ensure that the faculty advisor is adequately supported by the department and university. Although student management does do a lot of the work that is traditionally performed solely by the faculty advisor, being the faculty advisor for HGD is a major time commitment. In a traditional course, the instructor can make executive decisions about grading, the assignments, the course schedule, and the topics that should be covered during lectures. In HGD, many of these decisions get made by committee (with input from the faculty advisor) and the time for a committee decision is often longer. In addition, a lot of time is expended helping prospective students decide if HGD is appropriate for them, following up with management over email, managing sponsored projects, pursuing potential sponsors, and responding to problems that individual teams might be having. The time that the faculty advisor puts into courses such as HGD must be understood to count as service to the department or be counted the same way as teaching any other course.

The faculty advisor needs to strike a balance between dictating what students must do and suggesting what they should do. The faculty advisor should rarely overrule student management when they are in disagreement. If the faculty advisor always dictates to student management what to do and how to do it, the students will not learn how to manage. Instead, the faculty advisor should make suggestions, remind management of deadlines, indicate what they believe students are doing incorrectly, and explain the consequences of doing something incorrectly. For example, at the beginning of many semesters, the faculty advisor may have to remind management that a one week delay in establishing teams is the same as throwing away 200 people-hours of time on game development (50 students x 4 hours/week/student). The faculty advisor overrules student management only when he or she believes that the students are making a decision that will have severe consequences. The threat of management being overruled provides incentive for them to consider the faculty's advice and devise a plan to mitigate the faculty's concern.

Ensure that everybody is included and can contribute to their teams. The majority of students in HGD are pursuing CS-related degrees and the faculty advisor is in the computer science department. We aim to continuously increase the diversity of students in HGD. For example, we might assign an artist to a team of four or five programmers. At the beginning of the school year, the team might expend a significant amount of energy developing the core software necessary for the game. The artist (who perhaps has no programming experience) might feel left out during discussions and meetings. We regularly remind team leaders that it is to their benefit to include everybody in discussions whenever possible and assign them tasks as soon as possible so they have more time to accomplish their work. We have also found it useful to create pseudo-teams for groups of people with similar skills (artists, sound designers, etc) that meets occasionally to talk about their experiences and suggest changes to management that would help them be more productive. If new groups are not represented in management and do not communicate with management, we have found that it is unlikely that they will enroll in HGD for more than one or two semesters. We have also found it useful to have these pseudo-teams showcase their work during general meetings and talk about the things that teams can do to more effectively work with them.

Enrollment in a course is important. Although student clubs that focus on video game development are a positive outlet for some students, we do not allow students to participate in HGD unless they are enrolled. It can be destructive to an entire team when one team-member chooses to or fails to contribute to their team. In HGD, students that don't contribute will receive a lower grade. Although this doesn't eliminate problems, we believe that it does significantly reduce them.

Private workspaces can help productivity. HGD has its own small computer lab consisting of approximately eight computers that is only accessible to enrolled students. The lab is maintained by the VP of Technology, with help from others, and gives the opportunity for hands-on learning of how to administer and maintain a lab of computers. It also provides a private space for teams of students to have meetings without disturbing others, install software this is not provided on machines maintained by IT, and store any hardware that is owned for use with the course. Team leaders can reserve our lab for long meetings and work sessions via an electronic calendar.

Conclusion

We believe that HGD effectively teaches students how to work on multidisciplinary teams that closely approximates real-world software development than traditional computer science courses. Although HGD focuses on video game development, we believe that the experiences the students have are transferrable to other software development projects or to other kinds of interdisciplinary team projects. HGD has attracted more than 1 in 10 of all CS students at Michigan Tech since Fall 2012. Since Spring 2011, the course has regularly had a higher proportion of women enrolled than the overall CS department. Even when projects are not successful, HGD gives students valuable experience working on large projects and to exercise management and leadership skills through experiential learning. We have also provided useful guidelines and suggestions to help other colleges and universities replicate the success of HGD.

References

[1] Briana B. Morrison and Jon A. Preston. Engagement: Gaming throughout the curriculum. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 342–346, New York, NY, USA, 2009. ACM.

[2] Yolanda Rankin, Amy Gooch, and Bruce Gooch. The impact of game design on students' interest in cs. In *Proceedings of the 3rd International Conference on Game Development in Computer Science Education*, GDCSE '08, pages 31–35, New York, NY, USA, 2008. ACM.

[3] Timothy E. Roden and Rob LeGrand. Growing a computer science program with a focus on game development. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 555–560, New York, NY, USA, 2013. ACM.

[4] Kajal Claypool and Mark Claypool. Teaching software engineering through game design. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITiCSE '05, pages 123–127, New York, NY, USA, 2005. ACM.

[5] S. Kurkovsky. Can mobile game development foster student interest in computer science? In *Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's*, pages 92–100, 2009.

[6] F. Dochy, Mien Segers, and Dominique Sluijsmans. The use of self-, peer and co-assessment in higher education: A review. *Studies in Higher education*, 24(3):331–350, 1999.

[7] Donald Chinn. Peer assessment in the algorithms course. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITiCSE '05, pages 69–73, New York, NY, USA, 2005. ACM.

[8] Philip M Sadler and Eddie Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.