# A Survey of Final Project Courses in Game Programs: Considerations for Teaching Capstone

**Jose P. Zagal**
DePaul University
243 South Wabash Avenue
Chicago, IL 60604, USA
jzagal@cdm.depaul.edu

**John Sharp**
Georgia Institute of Technology
Digital Media/TSRB
85 Fifth Street NW
Atlanta, GA 30308, USA
john.sharp@lcc.gatech.edu

## ABSTRACT
Game design and development programs often include a final project or capstone course as a means of assessing the cumulative theory, processes and techniques learned by students through the program or department's curriculum. While these courses are prevalent in programs around the world, there has yet to be a study of how, why, and to what end these courses are designed and run. We review the literature on capstone courses, discuss the findings of a long-form survey administered in early 2011, and propose a set of framing questions for the design and implementation of capstone courses. Survey findings include common goals of capstone courses, make-up of faculty teaching these courses, the support obtained and desired for the courses, the technologies used to create capstone projects, the methods of project management used in the courses and the expectations of faculty teaching the courses. These results can serve as a baseline for faculty and administrators looking to develop or improve their game design and development curricula.

## Keywords
game education, capstone course, final project course, curriculum design

## INTRODUCTION
Game programs vary wildly in terms of their curricula, goals, and evaluation. The last ten years or so have seen a dramatic increase in the number and type of academic programs related to games (IGDA 2008). In this article we examine the different ways that academic game programs have developed and how they teach their courses. We focus primarily on the final project or capstone course. A capstone course is one in which students generally work in teams designing and developing a game project that ideally reflects the cumulative knowledge and experience they have gained throughout their course of study. We report on the results of a survey of instructors of capstone courses

**Proceedings of DiGRA 2011 Conference: Think Design Play**.

conducted during early 2011. Our results highlight the variety of ways that these kinds of courses are taught and also provide insights we can begin to explore for best practices that could be adopted more broadly. Our findings can also serve as a point of comparison for other programs.

## Capstone Courses and Game Programs

There are compelling reasons for a degree program to include a capstone course. The capstone course is a method of summative evaluation that not only assesses previous learning, but also allows an instructor to assess the student's overall collegiate learning experience (Moore 2006). Capstone courses also allow for a mix of evaluative styles that can assess the broad range of students' past experiences (Kemp and Smellie 1989, pg. 20). In addition to demonstrating they have developed and consolidated necessary knowledge, students are able to show they have the skills to apply it. Capstone also provides an opportunity to assess student's attitudes, values, and feelings regarding issues and topics related, or relevant to, the student's field of study (Moore 2006).

A capstone course is typically defined as the crowning course or experience coming at the end of an academic program. Its objective is often defined as helping students integrate a body of relatively fragmented knowledge into a unified whole (Durel 1993). It has been described as a rite of passage since it provides an experience through which "undergraduate students look back over their curriculum in an effort to make sense of that experience and look forward to a life by building on that experience" (Durel 1993). Other pedagogical alternatives commonly used (often together with a capstone course) include internships or student-work opportunities (Wright 2010), honors courses, and advanced seminars on special topics (Wagenaar 1993). Capstone classes are also supposed to allow students to apply what they have learned, hopefully in preparation for the workplace (Haas and Wotruba 1990). In some cases it is the needs of the industry that drive the design of capstone courses by seeking to address and overcome weaknesses in newly graduated students (e.g. Todd et al. 1993; Parberry et al. 2005). It is also common for these courses to be designed around and include the participation of an industry client, partner, or sponsor (Bruhn and Camp 2004). Capstone courses are used across a wide variety of disciplines and academic programs including marketing (Haas and Wotruba 1990), engineering (Todd, Sorensen et al. 1993), communications (Moore 2006), sociology (Durel 1993), computer science (Clear et al. 2001) and others.

Although capstone courses in game programs are quite common, most existing literature examines the use of game-related capstone courses in other programs such as computer science (e.g. Jones 2000; Kerbs 2007; Sumner et al. 2008). An exception to this is Linhoff and Settle's (2009) work proposing metrics for evaluating computer game development capstone projects in technically-focused courses. They describe the importance of ensuring that students produce a complete game while also rigorously evaluating the quality of computer code generated, the appropriate integration of various technologies, and the degree of success in realizing a game design (Linhoff and Settle 2009). More recently, Boudreaux and colleagues (2011) describe general characteristics of their final game course including the processes and tools used together with general notions on how they assessed student work.

There are also insights to be gained from examining literature on game courses with attributes common to capstone classes (e.g. project-based, collaborative, inter-disciplinary). Estey et al (2010) describe the importance of peer review for encouraging engagement, sense of community, motivation, and overall improvement of the game

projects developed. Fernández-Vara and Tan (2008) describe how it is possible to instill basic professional practices in their students, notably project management, through the use of the scrum methodology for software development (see also Schild et al. 2010). Tan (2010) argues for the importance of learning and practicing iterative game design as a best practice in game development. He warns that the tendency to focus attention on a finished product can imperil student's adequately learning and practicing the habits necessary for successful iterative game design (Tan 2010).

## METHODS AND DATA ANALYSIS

In early 2011 we invited games educators from all backgrounds and institutions worldwide to participate in an online survey regarding their experience teaching capstone courses in game design and development. Participants were contacted via mailing lists frequented by gaming education professionals or direct e-mail using publicly available information. Participants were self-selected and our sample should not be considered representative (in the statistical sense) of the larger population of those who have taught, or are currently involved with, game capstone courses. A total of 37 responses were collected (n=37).

Our goal in this study was to get a sense of how game capstone courses are taught, who they are taught by, and the issues faced when teaching this kind of class. Our long form survey included 34 questions in a variety of formats including multi-part, multiple-choice, and open ended. A couple of questions were inspired by McGill and Settle's (2011) work on faculty support in computer science departments. Our survey questions addressed the following themes: characterizing the institution and its academic calendar, understanding the spectrum of projects developed in the course, understanding how students are managed and organized, how students are evaluated and assessed, what kinds of institutional support are common, understanding student expectations, and finally, examining the strengths, weaknesses, and challenges of pedagogical approaches as reported by our survey participants. We also included an optional question soliciting participation in a follow-up interview study. With the exception of the optional question, survey responses were anonymous.

In our analysis of responses to open-ended questions, we used open coding to bring themes to the surface from inside the data (Neuman 2000). Here we assigned codes or labels to each answer. As we analyzed each response, new codes were developed and existing ones changed. We continued this process until no new codes emerged and then began to identify consistencies between codes. These consistencies, codes with similar meanings or referring to a common idea, helped us develop some of the themes we report on in this paper.

## FINDINGS

Rather than present the results of all the questions we asked, we choose to group them according to areas or themes each potentially covering several questions. When reporting our results, we identify which kind of question the responses are to and, unless otherwise noted, percentage numbers are based on the total number of responses for a particular question.

### Capstone's Most Important Feature

We asked our respondents to describe the single most important feature or characteristic of their capstone class. We were interested in better understanding the reason for having a course of this type in their program as well as the context in which the course is offered.

We obtained a variety of responses to this question. One of our respondents noted how he[1] calls the course, "the 'train wreck' module, as we make clear that students are expected and allowed to fail horribly, but to see that as a learning opportunity." This response highlights experiential aspects of the course that are hard to account for in a standard curricular model in which failure is not an acceptable outcome. For some capstone faculty, failing to finish a project can be acceptable, so long as students learn valuable lessons from that. In other words, failing the class is not necessarily the same as failing to succeed in a project, and passing the class is not the same as succeeding in a project. Another respondent described capstone's main goal as "producing a high quality, professional looking/playing game." Here, the focus is on a tangible artifact that can give students "something for their portfolio or [help them] get contacts with companies." Reading between the lines, it is clear that this sort of course is seen as an enabler for students' job prospects. A focus on employability was also noted by others who design their courses in such a way as to provide a "workplace simulation, where we attempt to construct a workplace that is more professional than the norm." Here the goal is for the course to provide students with a taste of what they might expect in their professional lives, with some instructors running the class in such a way that it "is effectively a 'simulated work day'." Finally, some instructors see capstone as providing a different kind of opportunity for their students: "the students can make any game they like." This freedom can extend beyond the creative, including other aspects such as choice of technology, management style, and more. For some, the capstone course is special because "it's student run. I am present during the first few weeks, but my role diminishes as the project goes on."

The variety of open-ended responses to the question of what a capstone class' single most important feature was surprising. However, our analysis helped identify four common features or themes. Capstone's most important feature is that it provides students with the opportunity to:

1. practice and learn "soft skills" such as communication, collaboration, and team work,
2. create a portfolio-quality project,
3. have an "authentic" industry-like work experience, and
4. exercise some independence

The features above are listed in order of importance as determined by how often they appeared. The first characteristic was reported more often (eight responses), with the fourth appearing the least (four responses). Of course, these features or goals are not mutually exclusive; the "ideal" capstone course would perhaps address them all.

The above list could also be seen in terms of instructor priorities: if things start going off the rails in a capstone course, what should the instructor focus on above all else? From a curricular development perspective, the four features listed above provide a clear guide of the priorities that can be chosen. Choosing one over the others would clearly have a significant impact on the course, how students would be evaluated, and ultimately the measure of its success. Making that decision is by no means an easy choice. Sticking to it can be even harder.

## General Characteristics

We were curious to learn how capstone courses are set up from an organizational point of view. Who are they taught by? How large are the classes? Most of our respondents (89%)

indicated that capstone is taught by fulltime faculty. Interestingly, some programs have faculty team-teaching. Although single faculty capstone courses seems to be the norm (57%), a significant number of courses are taught by two (26%), three (14%), and even four faculty members (3%). Due to the multi-disciplinary nature of game design and development, we find co-teaching is a practice worth noting. Some programs feature a fulltime faculty member teaching together with an adjunct from industry. This pairing might give the best of both worlds. Disciplinary-based co-teaching is probably also common: imagine a faculty team of programmer, game designer and artist, each able to provide detailed feedback and support to students in their respective fields.

The size of a capstone class can play a role in its success. The average number of students in a capstone class, per faculty member, is 15. This number seems ideal for both balancing teams as well as allowing faculty to provide individual hands-on attention as needed. Enrollment caps can be tricky to manage and depend on the particulars of each institution. However, we hope most institutions would avoid our reported worst-case scenario with 70+ students for a single instructor to manage. Generally speaking, we found that class sizes tend to be reasonable and manageable. Most capstone classes have an enrollment cap between 10-15 students (46%) followed by less common larger sizes: 16-20 students (21%), 21-30 students (17%) and more than 30 students (17%). Institutional circumstances and policies undoubtedly play a role in determining what limits to set. However, we were positively surprised to see these low numbers.

Capstone courses are often designed to allow students a longer and more involved experience than what a regular course might permit. This is often accomplished by designing the course to run over multiple academic terms. By asking how many terms are part of capstone together with the duration (in weeks) of those terms we were able to determine the average duration of the capstone class as 21.6 weeks. The shortest was 11 weeks, the longest 48 weeks, and the standard deviation was 8.61 weeks. The high standard deviation suggests a lot of variation across programs. Student involvement in capstone can also extend beyond regular terms. For instance, in one program “groups are formed at the end of the Junior year[2], and the majority of brainstorming, concept, and stage 1 preproduction work is done over the summer. So the capstone project can last almost an entire year.” Students would probably not receive academic credit for preparatory work done over the summer. Other programs require that capstone projects be submitted for approval before the course begins with more projects proposed than are ultimately accepted. Again, a successful capstone project proposal would require significant work for no academic credit.

We also saw variation in how often the class meets. Most capstone courses meet once (40%) or twice (40%) a week. The rest meet on three (8%), four (3%), and five occasions (6%). Surprisingly, there was one reported instance of a class that met seven times a week. We’re not sure if this is because they meet more than once in a single day (say, mornings and afternoons) or if there is another explanation. Similarly, we saw variation in the duration of class sessions. Some programs favor short meeting times (e.g. 90-120 minutes) while others offer intensive full-day sessions (e.g. 360-480 minutes). Some programs adjust class time as the course develops. For instance, capstone might start with an intensive first week during which students meet for seven hours each day and then only meet twice a week for an hour for the remainder.

### Institutional Support

We were interested in getting a sense of what kinds of support are commonly available to instructors of capstone courses as well as what support is deemed most valuable or critical to the success of the class. We asked our participants to identify (from a list provided) all those areas in which they received or had access to support and/or resources. The most commonly available resources are:

- Software purchases or upgrades (60%)
- Development kits (56% )
- Game library (56%)
- Guest speakers (48%)
- Sound recording/mixing studio (48%)
- Industry consultants (40%)
- Funds to attend professional meetings / conferences (40%)
- Motion stage/lab (36%)
- Funds to host demo session / final presentations / public event (36%)
- Facilities dedicated exclusively to capstone (32%)
- Hardware purchases or upgrades (32%)
- Teaching assistant/grader (32%)

Using the same list, we then asked participants to pick three items they felt would most greatly impact capstone if institutional support increased. The most popular options were:

- Funds to attend professional meetings / conferences (53%)
- Hardware purchases or upgrades (37%)
- Development kits (33%)
- Facilities dedicated exclusively to capstone (27%)
- Industry consultants (27%)
- Software purchases or upgrades (27%)
- Funds to send students to professional meetings / conferences (23%)
- Funds to run focus groups / playtest sessions (20%)

A comparison of both lists for similarities and omissions provides insight in areas for faculty to focus their fund-raising (or administration lobbying) efforts. This analysis also provides support for choices that may have already been made. For instance, we note that travel support for faculty, hardware and software purchases and upgrades, development kits, exclusive facilities for capstone, and industry consultants all feature prominently as commonly available and important to success. On the other hand, game libraries, sound recording/mixing studios, and motion stage/labs are commonly available, but aren't deemed areas in which additional support would be that beneficial. So, institutions considering investing in "high-ticket" support (e.g. a motion stage/lab) might want to consider investing in some of the other areas. There might be an opportunity for improvement in two areas seen as great contributors to the success of capstone if additional support was provided: funds to send students to professional meetings / conferences, and funds to run focus groups / playtest sessions. Providing support for students to travel (and present) at professional meetings and conferences can provide a huge motivational impact, especially if students are aware that such a possibility exists before any work begins. Similarly, formalizing focus groups and playtest sessions can greatly enhance the quality of the final projects.

## External Collaborators and Collaborations

Capstone course often plays the role of bridge or connector between the university and industry (e.g. Linhoff and Settle 2009). This is often seen in terms of collaborations with people or groups outside of the course. We asked which forms of collaboration (from a list that included an open-ended option) had been implemented with outside institutions or schools for capstone class. By far the most common form of collaboration is to invite outsiders to provide feedback on the projects under development by providing feedback and critique at key milestones (47%), serving in the role of client/executive producer (27%), or having external people (or companies) play the role of potential game publishers (27%).

A large number of courses also use collaborations as a way to provide access to expertise that isn't available in the class (38%). For example, a game program in a computer science department might collaborate with a nearby art school (or the art department in the same institution) or vice versa. In some cases, as noted by one of our respondents, "outsiders also provide specialist equipment such as motion capture facilities." In another example, "outside resources are used as needed, but seldom from other institutions. Programming is the core area that we are light on, and [we] have utilized programmers from [anonymized institution], Juniors within our program, and even really good programmers from 2 year institutions when necessary."

We were surprised by how few courses include outsiders in playtesting and QA roles (27%). This seems like a missed opportunity that should be considered. We were also surprised by the high number of responses that indicated that no collaboration occurred with outside institutions or organizations (35%). Even for institutions lacking local game developers in their vicinity, there are other ways to establish collaborations (see Ficocelli 2006 for ideas and examples). For example, Gabe Newell, co-founder of Valve software, participated in a teleconference with school students (Goldman 2011). We encourage faculty to pursue these kinds of opportunities although we acknowledge that collaborations can be challenging to maintain and manage effectively.

Our survey neglected to account for class policies regarding non-formal collaborations that students might engage in. One of our respondents noted how they "also allow students to utilize any outside resources that they can find and want to include in their projects - or students majoring at our college in areas such as music, animation, and graphic arts." She also pointed out that "we highly encourage collaboration between majors in the Capstone projects." By neglecting to account for possible informal collaboration, such as students working with friends or participating in external communities (e.g. indie developers who might provide feedback), it is possible that our results are biased against collaboration. In other words, the reported 35% capstone courses that did not engage in collaboration might be an overestimate.

## Student Teams and Autonomy

Allowing students to practice and learn "soft skills" such as communication, collaboration, and team work is perceived as one of the most important goals of capstone. This focus should reflect on how students are managed during the course. We asked our respondents to indicate all the team-sizes they had observed in their capstone class over the last two years. The most common responses were for small teams with 2-3 members (55.6%), medium-sized teams with 4-6 students (55.6%), and large teams of 7 or more students (41.7%). Roughly one-third (27.8%) of our respondents indicated having students work on projects by themselves. Given the reported importance of team work,

we were surprised by how high this percentage was. Since we did not ask what the distribution of team sizes was in a given capstone class, we're not sure if this number is reflective of exceptional cases (e.g. students fired from a team, see below). Further inquiry is necessary to better understand this value.

We were also interested in finding out how team composition is managed over time. In particular, how common team membership changes are. Generally speaking, team membership is static (52.7% indicate no changes in team membership) with changes sometimes allowed under exceptional circumstances (44%). Contrary to industry practice, where members of a team may change depending on a project's status or needs (e.g. a texture artist transfers to another team to help them ramp up their production), changes in team membership during capstone are usually due to personal and inter-personal issues rather than individual skills or abilities. One respondent described how one team "split due to [an] irreconcilable personality conflict." Flexibility in managing team membership is generally at the discretion of the instructor, other times it is the students who decide. One of our respondents noted that in his course "team members may be 'fired' from their teams, at which point they may be picked up by other teams or fail the class." Another instructor described how she "gave the team the option to 'fire' any of their own (they opted not to), or 'hire' new people into the second half of the class if they felt they needed more hands. There was only one team (it was a small class, ~8 people) so there was no option to transfer from one team to another, but I would have allowed that as well." From an instructor's perspective, allowing students to "fire" a teammate can prove challenging to manage: what happens to students that have been fired? One respondent describes how she had "2 students on a team of 5 who weren't communicating well, nor producing to their team's expectations. There was mutual disappointment and so the two members joined with a third student who hadn't been able to join a group yet, and they formed a new group that has been doing very well." Another notes the risk involved, while a "'fired' team member may petition a different team, but if not 'hired', must complete capstone any way he can."

Further work is needed to better understand how to best manage student team formation. Should students pick their teammates? It has been suggested for example that personality preferences as indicated by the Myers-Briggs Type Indicator may be useful for maximizing team cooperation in capstone courses (Magness and Roslewicz 2009).

### Tools and Technologies

The role that tools and technologies should play in a given curriculum has always been a point of contention. Should students become experts at using the same tools used in industry even if those skills may become obsolete? What programming languages should they learn (if any)? Also, who chooses what tools to use?

Only 11.1% of our respondents indicated that students were allowed to choose the technological platform for their projects. This seems to contradict the importance of allowing capstone students to exercise independence. However, in most cases (61.1%) students are allowed to make this decision so long as they obtain the instructor's permission. Presumably, the students must convince the faculty that they have done the appropriate research on the platform they want to use together with possessing the skills to develop on it. 27.8% of our respondents indicated that the choice of platform was made exclusively by the instructors.

Rather than ask what was considered 'best,' we chose to survey for tools actually being used in capstone class. For reasons of length, our list of options available to choose from was not comprehensive (e.g. more programming languages could have been chosen and we did not consider embedded scripting languages such as Maya's MEL or Unreal's Kismet). However, respondents were invited to add more options via an open text "other" field. None of these responses registered significantly (i.e. more than two responses). Respondents could also select multiple options.

- Programming Language / Development Platform
  - C / C++ (58.3%)
  - XNA / C# (55.6%)
  - Flash / Actionscript (61.1%)
  - Processing (8.3%)
- Game Engine / Authoring Tool
  - Unity (55.6%)
  - Gamemaker (27.8%)
  - Torque (27.8%)
  - Source (11.1%)
  - Unreal Ed (25%)
  - Unreal Developer Kit (UDK) (33.3%)
  - Havok (13.9%)
- Art / Modeling
  - Maya (55.6%)
  - Autodesk 3ds Max (3D Studio MAX) (38.9%)
- Sound / Audio
  - Fmod (30.6%)
  - Pro Tools (27.8%)
- Version Control
  - Subversion (38.9%)
  - Perforce (5.6%)

Based on some of the open text responses, we know that even for a single capstone project, multiple technologies may be used. For instance, Game Maker may be used for rapid prototyping purposes with the final game developed in C/C++. We were surprised by the popularity of Unity (55.6%) given its relative youth compared to other platforms. We were also concerned by the apparent low adoption of software versioning and revision control systems. Assuming that a wide variety of other version control systems are not being used, it seems strange that they're not being used more widely given the existence of freely available options (e.g. Git and Subversion).

### Project Management Skills

Given the importance of capstone in providing the opportunity to practice and learn "soft skills" as well as have an "authentic" industry-like work experience, we wondered where and in what way(s) project management was taught in the game programs we surveyed.

First, we asked who chose the project management method used: the instructors (40%), the students (20%), or students with instructor's permission (40%). Then, we asked which project management methods were used during capstone. Here, respondents could choose more than one option (e.g. different student teams may choose different methods). Rapid prototyping (56.7%) and SCRUM/Agile development (53.3%) were both popular together with milestone (33.3%) and design document (53.3%) driven approaches.

Finally, we asked how project management techniques were taught in the respondent's game program prior to (and/or including) capstone. Again, participants could choose more than one option:

- 41% - Smaller-scale group projects
- 30% - Part of capstone course
- 22% - Within capstone course
- 19% - Stand-alone project course
- 19% - Production methodologies course
- 4% - Not at all

While the fact that project management isn't explicitly addressed in a few programs might be surprising, our attention was drawn to the (relatively) high number of courses that address project management skills in capstone (30% as part of, and 22% within). We feel that programs should address project management techniques early in the curriculum with frequent exposure from that point on. Leaving this part of the curriculum to capstone is almost always too late. Further work is needed however to determine how often and in what ways project management is covered in gaming curricula.

### Managing Student Expectations

Research on games education suggests that one of the greatest challenges faced by capstone instructors is in managing and handling differing expectations for the course between instructors and students. Egert and colleagues (2007) note how game courses often deal with faculty expectations that do not align with students' actual skills and abilities. Zagal and Bruckman (2009) have noted similar issues, with students assuming that their experience as "gamers" or "fans" easily translates into game design knowledge and expertise. Aware of these issues, we asked an open-ended question for examining what faculty felt students going into capstone class expected to gain from the experience. We analyzed the responses comparing them with those from a question asking instructors what they felt was the single most important learning goal of their capstone class.

For the instructors, the most important learning goals were team collaboration and soft skills followed closely by developing an understanding of "the big picture" (seeing how everything comes together). For students (as reported by faculty), the most important aspects of capstone were overwhelmingly related to the creation of a portfolio piece and a polished game. One respondent describes student's aspirations: "they have big dreams: they expect to make a game that gets picked up for commercial distribution and makes them famous". Another notes how "most want to create a commercial quality game", and a third respondent adds that student want "a portfolio piece that they can hopefully publish in some way."

The brutal truth is that these goals are rarely met. Our respondents noted that, in the last two years, 40 capstone projects are either under contract for commercial release (6 projects), development for commercial release (26 projects), commercially available (4 projects), or have a revised/upgraded version available commercially (4 projects). These 40 projects represent little over 12% of all projects (284 projects were reported as not going beyond the course assignment). If we only consider capstone projects becoming commercial projects, the percentage drops to 2.5%. So, student's dreams of creating a commercial quality game from capstone class are a long-shot, to say the least.

We draw attention to this issue not because we think capstone courses are failing their students. We don't think that all or most capstone projects should become commercially available in some fashion. Rather, we think there is a wide gap between students' expectation of producing a game that is commercially viable in terms of quality and what they will be able to effectively accomplish. Explicitly framing the goals of the course as conducive to, for example, an entry in a game design competition (e.g. student IGF or Imagine Cup, see Parrish et al. 2010), might help in this matter. However, further work is needed to better understand this gap as well as identify best-practices for managing student expectations.

## DISCUSSION AND CONCLUSIONS

In reviewing our findings, a framing perspective became clear: in the design and execution of capstone courses, understanding how the course fits into the multiple layers of institution, culture, curriculum and industry is critical to success. This may seem obvious, but these questions are not always taken into account in the development of capstone courses.

Starting with the curriculum, our research makes clear to us that a big challenge is to design a capstone course that fits the skills developed during the courses preceding the capstone. If a curriculum emphasizes modeling and art pipeline skills for 3D engines, but covers little in the way of programming, scripting, concepting and design, then it is unrealistic to design a capstone course that expects a balanced development team that has the skills necessary to conceptualize, design and produce a full game using a 3D engine. A favored approach for getting around this is "outsourcing" to other departments, other institutions and friends, but these are all fraught with challenges as they are difficult to institutionalize within the structure of a course.

For programs that include a capstone course, though it may seem extreme, the curriculum leading up to capstone should be viewed as "capstone preparation." It is unrealistic to expect students to have developed deep skills in any areas on their own outside the goals and outcomes of the curriculum. A case in point is project management techniques. Due to the material covered in earlier classes, it is possible that many game design and development students never encounter a course focusing on project management methodologies such as the popular SCRUM/Agile method. It is unrealistic to expect students to learn and master this methodology on-the-fly during capstone course, and yet it happens more often than one might think.

The culture of a program is often just as important in the education of its students as the curriculum. If the culture in the halls and homework labs is not one that supports student success, self-direction and collaboration, it is unlikely that it will magically appear during the capstone course. What goes on in between the lines (and grades) is of the utmost importance to have students in the right frame of mind and with the right expectations as they prepare for and enter the capstone course. In the same way that the capstone should be designed around the curriculum, it should also be designed around the culture of the program or department. If the culture of a department is for students to come in for classes and then leave campus, then it is unlikely that a capstone course that requires student teams to meet outside class time will succeed.

The character of the student body should also be taken into account. This relates to the culture, but it is really a separate concern. What sort of students are attracted to and moving through the department or program's curriculum? If your school attracts students

with inclinations to join the level design staff at a major developer, and your capstone courses emphasize solo or two-person projects, then you are likely falling short of the expectations of your students. On the other hand, it is also important to ensure that student expectations are managed so that they are realistic and attainable.

Wrapped around all of this is the institution and its needs. From the institution's perspective, the capstone course is the make or break point of the degree. These courses are the cumulative outcome of a given degree by design, and so are often closely scrutinized. If the institution focuses on career preparation, but the capstone faculty take a broader view, then there are likely to be problems. Similarly, if the institution emphasizes research and scholarship, then career preparation may not be given credence. Students with an expectation of job placement as the primary goal of their college degree may find themselves at odds with faculty expectations. Faculty are more likely to take the perspective that a failed project is just as valuable, if not more, than a successful one. Students, on the other hand, are likely to be looking for the cornerstone of their portfolios as they prepare to enter the job market.

Finally, there is the relationship between industry and the capstone. Do game programs owe the industry anything? Must we function as feeders for the next wave of game developers? Are we indebted to prepare students for careers as game developers? Do we owe it to our students to help them get that one polished game project that will help them land their dream job? For capstone courses at research-focused institutions or art schools, career preparation may not be the emphasis. For vocational schools, is the only goal preparation for entry-level positions? Of the four features we identified from our respondents, it is perhaps the last one, "exercise some independence", that should serve as the counter to many of the above questions. Shouldn't capstone class be the opportunity for students to leave the beaten path? To think outside the box and perhaps, just maybe, devise a game that challenges us to think about the medium in new and exciting ways?

While this research is preliminary and has thus far focused on faculty, it is clear that capstone courses can be both symptomatic of underlying ailments of a program, and the most heavily-weighted of courses in terms of faculty expectations. It is our hope that this study provides a starting point, a baseline of sorts, for faculty and administrators looking to design and improve game design and development curricula.

## FUTURE WORK

As mentioned earlier, our survey solicited participation in an interview study that pretends to examine in greater depth how capstone courses are taught. This follow-up study will help us identify best-practices as well as gain in-depth information we could not obtain from the original survey. Participants for this phase of our research include survey respondents who indicated they were willing to participate in our second phase. We also think it can be productive to take a deeper look at, and compare, these kinds of courses across different socio-cultural settings. Although our survey results are anonymous, we have the impression that our data mostly reflects game courses in the US and Europe. A better understanding of capstone-style courses in game programs in Asia could be particularly enlightening.

## ENDNOTES

1 Since our respondents are anonymous, we have no way of knowing their gender. Although we use gendered pronouns interchangeably in our text, we make no claims or assumptions as to a respondent's gender in a particular quote.

2 In the US, a Junior year student refers to a student in their penultimate (usually third) year.

## BIBLIOGRAPHY


Boudreaux, H., J. Etheredge and A. Kumar (2011). "Evolving Interdisciplinary Collaborative Groups in a Game Development Course." Journal of Game Design and Development Education, vol. 1, no. 1.

Bruhn, R. E. and J. Camp (2004). "Capstone Course Creates Useful Business Products and Corporate-Ready Students." inroads: The ACM SIGSCE Bulletin, vol. 36, no. 2, pp. 87-92.

Clear, T., M. Goldweber, P. Leidig, F. H. Young and K. Scott (2001). "Resources for Instructors of Capstone Courses in Computing." ACM SIGSCE Bulletin, vol. 33, no. 4, pp. 93-113.

Durel, R. J. (1993). "The Capstone Course: A Rite of Passage." Teaching Sociology, vol. 21, no. 3, pp. 223-225.

Egert, C., S. Jacobs and A. Phelps (2007). "Bridging the Gap: Balancing Faculty Expectations and Student Realities in Computer Gaming Courses". FuturePlay 2007, (Toronto, Canada, November 15-17), ACM.

Estey, A., J. Long, B. Gooch and A. A. Gooch (2010). "Investigating Studio-Based Learning in a Course on Game Design". Foundations of Digital Games 2010 (FDG2010), (Monterey, CA, USA), ACM.

Fernández-Vara, C. and P. Tan (2008). "The Game Studies Practicum: Applying Situated Learning to Teach Professional Practices". FuturePlay 2008, (Toronto, Ontario, Canada), ACM.

Ficocelli, L. (2006). "Industry – Academia: How Can We Collaborate?" Loading...: The Journal of the Canadian Game Studies Association, vol. 1, no. 1.

Goldman, T. (2011, April 22, 2011). "Gabe Newell Divulges Valve's Secrets to High School Students." The Escapist Magazine. Available at http://www.escapistmagazine.com/news/view/107878-Gabe-Newell-Divulges-Valves-Secrets-to-High-School-Students (accessed April 5, 2011).

Haas, R. W. and T. R. Wotruba (1990). "The Project Approach to Teaching the Capstone Marketing Course." Journal of Marketing Education, vol. 12, no. 37, pp. 37-48.

IGDA. (2008). "Curriculum Framework - version 3.2 beta." Available at http://www.igda.org/wiki/images/e/ee/Igda2008cf.pdf (accessed September 19, 2008).

Jones, R. M. (2000). "Design and Implementation of Computer Games: A Capstone Course for Undergraduate Computer Science Education". SIGCSE 2000, (Austin, TX, USA), ACM.

Kemp, J. E. and D. C. Smellie (1989). Planning, producing, and using instructional media (6th Edition). New York, Harper Collins, 1989.

Kerbs, R. (2007). "Student Teamwork: A Capstone Course in Game Programming". Frontiers In Education Conference - Global Engineering: Knowledge Without

Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual, (Miwaukee, WI, USA, October 10-13), IEEE.
Linhoff, J. and A. Settle (2009). "Motivating and evaluating game development capstone projects". Proceedings of the 4th International Conference on Foundations of Digital Games, (Orlando, Florida), ACM.
Magness, J. B. and E. A. Roslewicz (2009). "An Innovative Myers-Briggs Type Indicator Educational Experience Used in an Information Technology Capstone Course". SIGITE '09, (Fairfax, VA, USA), ACM.
McGill, M. M. and A. Settle (2011). "Computing Faculty Tenure and Promotion Requirements at U.S. and Canadian Post-Secondary Institutions". To appear in SIGITE 2011: The Special Interest Group for Information Technology Education Conference., (West Point, New York, October 2011).
Moore, R. C. (2006). The Capstone Course. Assessing Media Education: A Resource Handbook for Educators and Administrators. W. G. Christ. Mahwah NJ, Lawrence Erlbaum Associates, pp. 439-462.
Neuman, W. L. (2000). Social Research Methods (4th Edition). Boston, MA, Allyn & Bacon, 2000.
Parberry, I., T. Roden and M. B. Kazemzadeh (2005). "Experience with an Industry-Driven Capstone Course on Game Programming". SIGCSE 2005, (St. Louis, Missouri, USA), ACM.
Parrish, J. J., J. Bailey and B. Jensen (2010). "Using the Imagine Cup SDI as the Foundation for Computer Science Capstone Projects". SIGCSE'10, (Milwaukee, WI, USA, March 10-13), ACM.
Schild, J., R. Walter and M. Masuch (2010). "ABC-Sprints: Adapting Scrum to Academic Game Development Courses". Foundations of Digital Games 2010 (FDG 2010), (Monterey, CA, USA), ACM.
Sumner, R. W., N. Thuerey and M. Gross (2008). "The ETH Game Programming Laboratory: A Capstone for Computer Science and Visual Computing". 3rd International Conference on Game Development in Computer Science Education (GDSCE 08), (New York, NY), ACM.
Tan, P. (2010). "Iterative Game Design in Education." International Journal of Arts and Technology, vol. 3, no. 1, pp. 118-123.
Todd, R. H., C. D. Sorensen and S. P. Magleby (1993). "Designing a Senior Capstone Course to Satisfy Industrial Customers." Journal of Engineering Education, vol. 82, no. 2, pp. 92-100.
Wagenaar, T. C. (1993). "The Capstone Course." Teaching Sociology, vol. 21, no. 3, pp. 209-214.
Wright, K. (2010). "Capstone programming courses considered harmful." Communications of the ACM, vol. 53, no. 4, pp. 124-127.
Zagal, J. P. and A. Bruckman (2009). "Novices, Gamers, and Scholars: Exploring the Challenges of Teaching About Games." Games Studies: The International Journal for Computer Games Research, vol. 8, no. 2.