

MIT Open Access Articles

The game studies practicum: applying situated learning to teach professional practices

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Clara Fernandez-Vara and Philip Tan. 2008. The game studies practicum: applying situated learning to teach professional practices. In Proceedings of the 2008 Conference on Future Play: Research, Play, Share (Future Play '08). ACM, New York, NY, USA, 25-32.

As Published: <http://dx.doi.org/10.1145/1496984.1496990>

Publisher: Association for Computing Machinery (ACM)

Persistent URL: <http://hdl.handle.net/1721.1/100268>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



The Game Studies Practicum: Applying Situated Learning to Teach Professional Practices

Clara Fernández-Vara
Singapore-MIT GAMBIT Game Lab
Massachusetts Institute of Technology
telmah@mit.edu

Philip Tan
Singapore-MIT GAMBIT Game Lab
Massachusetts Institute of Technology
philip@mit.edu

ABSTRACT

The inclusion of a practicum is one of the main challenges in the game studies curriculum, especially when it comes to teaching professional practices to students. This paper presents how professional management methodologies (Scrum, in this case) can be related to models of Situated Learning, as we demonstrate through our case study, the Singapore-MIT GAMBIT Game Lab. Being aware of the connections and the pedagogical potential of professional practices can improve both how we teach and how our students learn how game development works. In our case study we also propose ways in which the practicum can be related to research in videogames.

Categories and Subject Descriptors

K.3.2 Computer and Information Science Education

General Terms

Education, Curriculum

Keywords

Education, game studies, videogames, learning, teaching, practicum, Scrum, Situated Learning, Cognitive Apprenticeship, Legitimate Peripheral Participation.

1. INTRODUCTION

There are many issues to be dealt with in game studies, starting with the identity of the field. What disciplines does game studies relate to? More importantly, what makes it different from those disciplines? To complicate things, game studies is no longer a specialized topic of research, it is now becoming part of the curricula of higher education institutions, even when the nature of the field is unclear. Of all the issues arising from the introduction of a new discipline into universities, we will focus on the incorporation of practice into the game studies curriculum, and how a practicum can serve both the pedagogical aims of a program and can inform and support research.

The goal of this paper is to suggest ways to improve the game studies practicum based on pre-existing models of learning and education. We will resort to two models within the Situated Learning paradigm: Cognitive Apprenticeship and Legitimate Peripheral Participation (LPP). These suggestions are based on our case study, the Summer Program of the Singapore-MIT GAMBIT Game Lab, which we will describe and analyze, and

then propose improvements according to these models.

2. FRAMEWORK

Before delving further into our case study, there are a series of general concepts that must be taken into consideration. These concepts have to do with the complexity of the growing field of game studies, based on the necessity to define its identity within multiple disciplines and the difficult conciliation of theory and practice, among other factors.

2.1 The Map of Videogame Studies

The landscape of videogame-related education is complex and craggy. An inherently interdisciplinary field, the study of videogames can be tackled from the practical/production side – teaching how to make videogames – or from the viewpoint of theory and criticism – teaching and developing theoretical concepts and vocabulary, to analyze games and their history and provide tools to study players and their behavior –. Both sides are not mutually exclusive, they must inform and support each other even when the emphasis falls on either theory or practice. A quick look at the recently released IGDA Curriculum Framework also evidences the variety of disciplines that make up the field of studying games, including Management, Psychology and Cognitive Science, Computer Science, Visual and Audio Design, and Media Studies [1]. This is partly due to the modernity of the field, where there are very few departments in the world devoted exclusively to videogames, and due to its necessary interdisciplinarity, which allows all these different areas to contribute to the advancement of the new field. Interdisciplinarity also requires schools incorporating videogames into their curricula to decide where in the map they want to situate their program—more practice- or theory-oriented, or somewhere in between; focused on art, or software development, or developing applications for education, to name a few options. Ideally, every program should incorporate more than one approach, even if it focuses on just a couple of places in the map, and should establish pointers to other related disciplines.

Of all the challenges of creating new curricular tracks and educational programs, one of the most pressing is incorporating practice in the curriculum, especially when the educational aim is to groom students who want to make games. This is the main issue that we tackle at the Singapore-MIT GAMBIT Game Lab, a research laboratory where we have established an international collaboration between Singapore educational institutions and the Massachusetts Institute of Technology. In this paper, we want to present our Summer Program as a case study of how we teach students professional practices while still developing games at the service of research. After the first run of our Summer Program, we have resorted to Situated Learning theories [8, 10] in order to

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.

FuturePlay 2008, November 3-5, 2008, Toronto, Ontario, Canada.
Copyright 2008 ACM 978-1-60558-218-4...\$5.00.

understand how learning takes place and to improve our teaching. Situated Learning theories have made us realize that the structure of our Summer Program bears strong relationships with the model of Cognitive Apprenticeship [2], with other aspects relating more directly to Legitimate Peripheral Participation [8]. These are the theories of learning that have informed the redesign of our Summer Program.

This paper analyzes and proposes a model related to one single aspect of videogame studies, which is teaching game development in an academic setting. In the process, we also suggest some ways in which game development can connect with other areas of game studies, such as research, analysis and game studies courses, all of which are also part of the activities and concerns of the GAMBIT Game Lab. We are aware that there are other schools that follow a similar approach, applying professional practices and methods to the development of student projects. Our contribution with this paper intends to highlight and describe the learning processes that take place, according to specific models of learning, and propose how to improve learning using professional practices based on those models.

2.2 Interdisciplinarity and Teaching

The issues arising from the interdisciplinary nature and the necessary relationship of theory and practice are not unique to videogames. Other budding fields that thrive on the crossover and pollination between disciplines have similar problems of identity. For example, some of the issues that affect performance studies can be directly transferred to game studies [11]. The first issue is the variety of goals of the students in the discipline—many institutions of higher education have to cater to the students who want to make games, as well as the ones who are still undecided whether they would want to pursue a videogame-related career, or just want to take a course because they are interested in the topic. Which department should be the host is another shared issue with between performance and game studies, since a program or laboratory's identity is very much influenced by the school it is under, even within the same institution. The choice of a host department can pose a problem when it comes to establishing relationships with the industry¹—Humanities departments, for example, are a lot less prone to form ties with non-academic partners than Computer Science. All these are factors that must be taken into account when designing our curricular activities.

The Singapore-MIT GAMBIT Game Lab is a research laboratory, rather than an academic program. It is a research initiative within the Comparative Media Studies (CMS) program at MIT, where interdisciplinarity is one of the program's defining traits. CMS also follows the principle of "applied humanism", where research supports practical application of its social and cultural expertise in contexts outside of academia. GAMBIT incorporates these principles into a five-year project to sponsor innovative research

¹ By *industry* we are not only referring to big videogame companies, but also independent developers, smaller companies and educational institutions that may produce games as part of their didactic materials. For a more extensive discussion on the relationship of industry and academia, see [5].

on video games, both in Singapore institutions and MIT, to develop new and innovative games, and to prepare Singaporean students from universities and polytechnic schools for the games industry. Our principles require students and researchers to study and write about games and players, as well as to develop playable games to support and communicate research goals.

Game studies is just one of the disciplines covered by the CMS curriculum, which also lists courses in film, literature, television, digital media, and photography, to name but a few. Currently, there are four videogame-related courses in the class offerings of the department, on videogame theory, game design, games industry and games and education. Several of these are taught by GAMBIT staff, a team made up of specialists in the field, either as professionals from the industry or as academic investigators. Games are also incorporated into many of the unit's transmedia and interdisciplinary offerings, integrating games into the full picture of the changing media landscape. The students that come to study and work at GAMBIT are not necessarily CMS majors, but come from all over the institution, from Mechanical Engineering to Brain and Cognitive Science. In the case of the Summer Program, the Singapore students that arrive in Cambridge come from a variety of backgrounds, from polytechnic schools to art and management colleges. The heterogeneity of the backgrounds has two sides; on the one hand, we cannot provide a four-year curriculum focused on videogames. On the other, we benefit from the richness and diversity of backgrounds of our students during the production of videogames, given their specialization in their respective fields (such as Computer Science, Audio Engineering, Visual Design or Management), thus creating a truly interdisciplinary environment.

2.3 The Difficulties of Teaching Professional Practices

Incorporating professional practices into the curriculum is a significant challenge, especially in institutions without a specific game studies major. Development is time-consuming; even in the case of more hands-on disciplines where a practicum may be required, such as Computer Science or Art and Design, there is little room to learn how to manage workflow and scope projects. Students may learn their skills in school, but they usually lack organizational and social know-how to bring a project to completion. One of the most desirable skills for future game makers is the ability to work well in teams, as well as knowing how to divide the work and communicate with each other. Even when teamwork may be a natural part of the practicum, it usually takes place within one single department, so students do not have the opportunity to work with students of other disciplines. The difficulties of communication across disciplines are also an obstacle in the professional world, as noted by Crawford [3]. Giving the opportunity for students to learn how to work with others with a different educational background will give them an advantage when they move on to the industry.

Therefore, it is essential that students who intend to work in videogames learn the necessary skills, and build a good foundation of basic professional practices. Colleges and universities usually shelter themselves in the ivory tower, not giving particular emphasis to the management of projects but rather to the results. This is not a flaw in itself; in fact, academia offers the possibility to fail within a certain extent—the purpose is not to produce complete, polished (or even fun) games, but to give

the students the opportunity to learn, even if it means they make mistakes. In this setup, professional practices usually fall through the cracks, given the already packed syllabi that instructors have to cover.

Our proposed case study was designed as one-semester program focused on teaching professional practices to students. The GAMBIT Summer Program takes up nine weeks during summer term, where student teams work on their games full time, following a work structure akin to that of professional game development.

3. CASE STUDY: THE GAMBIT SUMMER PROGRAM

The GAMBIT Summer Program had its first run in 2007, and has started its second one as we write. The program transformed our lab space into a development house for eight weeks, where six teams each developed a game prototype. An additional team produced sound assets (i.e. music and sound effects) for all the others. Most of our students came from Singapore, working along with MIT undergraduates and graduates. The purpose of the Summer Program is to develop prototypes at the service of research purposes, based on research questions posed by related faculty both from Singapore institutions and MIT. Thus the program served as an opportunity to teach students the process of game development, while their games were also research prototypes.

3.1 Using Scrum As A Teaching Methodology

One of the basic professional practices that we try to instill in our students is project management. In 2007, we chose a specific type of Agile Development called Scrum, an existing practice in software development [12]. The Scrum methodology is rather flexible, so we could easily adapt it to a range of different game projects. Scrum emphasizes the decentralization of the software production process, iteration, continuous re-evaluation of the project's scope, and focusing authority on a Product Owner.

3.1.1 Decentralization Of The Software Development Process

In the Scrum methodology, scheduling and scoping do not come from a central managerial entity; each team manages its own part of the development process. The team makes decisions as a group, keeping the ownership of the product rather than serving the decisions of an external manager. The methodology flattens the development hierarchy, in favor of individual teams that work in incremental feature sprints. Each team has a producer (Scrummaster), who coaches the rest of the team in the Scrum process and makes sure that it is applied efficiently.

The Summer Program mentors remained outside of the teams, as top-level coordinators of resources available to all the teams, or took the role of clients that each team would need to deliver their product for. Every team had the final goal of delivering a finished, polished game at the end of the Summer Program. The Scrummaster of each team was a student familiar with the methodology; (s)he was also the main contact of the team with the client (Product Owner in Scrum terms). Typically, the rest of the team would be two programmers, two artists, a game designer and a test lead. This team structure corresponds roughly to the structure of professional teams, and introduces students from different disciplines to each other, so that they learn what other

team members may require from them, and what they may need from others, encouraging the much needed communication between disciplines.

3.1.2 Iteration

The nature of software development requires it to be constantly iterative, so that features are implemented in code, tested, fixed for bugs, and re-tested to confirm if those bugs had certainly been resolved. Scrum is a methodology that revolves around iteration, stressing on having a functional piece of software at the end of each sprint. Multiple iterations are usually rather difficult to accomplish in an academic course, since usually students only have enough time to complete one version of the program, hoping to fix bugs as they encounter them.

In our Summer Program, every team had a lead tester in charge of Quality Assurance (QA), who initially worked as support for the designer, and then took charge of stability testing and of focus testing. Being able to incorporate iteration is a luxury resulting from a full-time schedule, where students can work exclusively on one game—we have found it is more difficult to implement testing as part of the production pipeline with students who only work part-time in development.

3.1.3 Continuous re-evaluation of the project's scope

The whole production period was divided in two-week intervals called *sprints* (again, according to the Scrum jargon). At the beginning of each sprint, every team created a sprint backlog, a selected list of features that they reckoned could be implemented during the following two weeks. At the end the sprint, each team had to deliver a playable version of the game with all the anticipated features for that sprint implemented. The aim was always to have a playable version at the end of each sprint, even if all the product's intended features were not in place. During the sprint, the students had to meet at the beginning of every day. The daily Scrum meetings lasted about 15 minutes, and in them each team member would state what they had done the day before, what they were going to do that day, and whether there was anything getting in the way of their work.

At the end of each sprint, every team would present its current version of their game to their client, the Product Owner. In that presentation they would explain what had been implemented and how; if there had been any features that were not implemented, what problems they had come across, and how they were planning on overcoming them. After those presentations, the students would open up their studios to everyone else, so that all the members of the lab and their friends could come and play their games. Seeing external players having fun with the games they were working on, even if they were incomplete, was also a morale boost for the students.

3.1.4 The Product Owner

Every team had one or two clients, the Product Owners, who were researchers investigating the questions that their game needed to address. Having a Product Owner replicates a professional environment; it is also useful for catering the research interests of the scholars working with GAMBIT, so that the prototypes developed also served as research tools. For example, the game *Audiodysey* has become the core of its Product Owner EitanGlinert's thesis [6]; the puzzles that appeared in *The Illogical Journey of Orez* have become part of a larger game developed by

The Education Arcade, another research project in CMS. The degree of involvement of Product Owners varied from team to team—some worked with their teams daily, some only met with their teams at the end of each sprint. Moreover, they were also able to detect and anticipate potential pitfalls and difficulties that the students were heading towards. However, Product Owners did not make major design decisions, leaving those to the students in each team.

3.2 The Problems Of Using Scrum In Rapid Development

Even though there are important pedagogical advantages in using Scrum in an academic setting, it also posed significant problems during development. Our Summer Program requires a rapid development cycle (not to be confused with Agile Development), since the students have to deliver a prototype in a short span of time. Scrum is a method intended for major software development projects that undergo constant revisions and reworks, such as online applications or operating systems, not for software that must be completed within eight weeks.

Scrum is a flexible methodology, but it does not provide many guidelines to deal with finalization of a product within a tight deadline. Our prototypes would need revisions after the summer, but those revisions would not be carried out by the same team. The hard deadline for the students to deliver their game meant that the students had to go into crunch mode for the last week or two. The mentors worked hard to inculcate students with good scoping and management practices, luring them from working overtime, for instance by scheduling leisure activities that would take them out of their studios. However, as the end of the program loomed, students still spent more and more time in the lab in order to finish the games with the level of polish that was required by GAMBIT. Thus most of the Scrummasters threw Scrum out of the window in the last two weeks, in order to complete their games on time.

3.3 The Importance Of Testing And Polish

The GAMBIT mentors (staff and Product Owners) emphasized the importance of polish of the final product during the production process. We encourage students to give their projects a limited scope—rather than working on over-ambitious projects that cannot be completed, the students must deliver smaller prototypes, where all the implemented features are fully functional and tested. The iterative development of Scrum facilitates this kind of “completeness” that we aim for in our prototypes. The look and feel of the game must include quality graphics, sound effects and music. The games must have an installer and instructions, so that users will be able to download the games from our website. This is where the role of the lead tester, in charge of the Quality Assurance (QA) of the game, becomes vital to making a difference in the final result.

3.3.1 Focus Testing

The polish of the game does not only refer to the look and feel of the game, but also to the proper testing with players outside of the production team. Lead testers did not only have to test every new build themselves, they were also in charge of supervising focus testing, evaluating issues such as whether new players understood what had to be done in the game, whether they found the

difficulty appropriate, and whether they were engaged and enjoyed the game. The students had to have other people play their game from the end of the first sprint, so that their audience was very much present throughout the development cycle.

There were several ways in which we brought in external testers. The MIT students invited their friends to try their games, and we also recruited students from a pre-orientation program. We also organized two Open House events during the summer, which had a double function: the events served as another opportunity for the students to present their work; they were also an easy way to get many visitors to play our games and became an important part of our focus testing. On the one hand, focus testing was a sobering experience for the students, since issues and problems that they had not thought of became glaringly obvious when their games were played by new players. On the other hand, the Open Houses were also a morale boost in the last two weeks of the Summer Program, when they had to work the hardest. The students saw that their games were generally enjoyable and fun, and they could also see the goal of completing a polished prototype close at hand.

4. SITUATED LEARNING AND GAME DEVELOPMENT

The initial design of the GAMBIT Summer Program was not based on any learning models; rather, it was an attempt to introduce professional practices into an academic setting. In the process of revising our program, we resorted to theories of Situated Learning, which emphasize the importance of the context in learning processes. Our concern about professional methods seems to find a good fit in these theories, where communities of practice shape and inspire the models where learning will take place. We have found out that Situated Learning is relevant to what we do, and that a good part of our program already suits some of its principles. Thus, it seems only natural to base the intended improvements to our program on these learning theories.

We will be focusing on two theories of Situated Learning: Legitimate Peripheral Participation (LPP), which presented a theoretical model based on actual apprenticeship practices [8], and Cognitive Apprenticeship, which was inspired by LPP but focused on how different types of knowledge may be acquired [2]. There are other approaches that relate to Situate Learning, such as Activity Theory [4], which focuses on the study of actual work groups and their dynamics. However, we have found the apprenticeship models more relevant, since they account for the process of learning professional practices, how knowledge is acquired, and what role the mentors play in the process.

4.1 Social Learning And The Importance Of Language In Cognitive Processes

Situated Learning has its foundations on Vygotsky’s theories of learning and child development [13]. Vygotsky argued that children learned from their cultural setting, by internalizing the knowledge and practices of their social environment. Learning results from problem-solving activities that the child does not have the knowledge to deal with on her own yet. The child is assisted by someone who is more experienced in how to solve that problem and takes the lead at first. As the child becomes able to solve the problem, the responsibility is gradually transferred to the her, thus the support (“scaffolding”) of the instructor fades.

Vygotsky also emphasized the importance of language in cognitive development as a fundamental tool for understanding, by internalizing the knowledge and processes that are presented to the child externally. This internalization of concepts helps children regulate their own behavior and activities. The next step is to externalize that inner speech, so that the child can explain what she has learned to others. The externalization of knowledge also creates a cycle by which verbal expression helps clarify and understand that internal knowledge more thoroughly.

4.2 Situated Learning: Legitimate Peripheral Participation

Inspired by Vygotsky's foundational theories, and focusing on the social components of learning, Situated Learning proposes a model inspired by traditional apprenticeship situations [8, 10]. According to this model, learning takes place within a social context, where learners follow the models set by experienced practitioners (the "old-timers") in the field, as well as from their peers. In the specific model of Legitimate Peripheral Participation [8], learning takes place by becoming part of a community of practice, as evidenced by apprenticeship models, such as that of tailors, midwives, or meat cutters. The LPP model describes how the learner performs a small, non-specialized task at first, to help and contribute within the community of practice. By participating with a small but relevant job, the learner observes how the old-timers work, becomes familiar with the jargon of the practice, and the organization of tasks. The learner steadily takes on tasks of more complexity and responsibility, growing from novice to expert in the process.

4.3 Situated Learning: Cognitive Apprenticeship

A related model of learning is that of Cognitive Apprenticeship [2], which emphasizes on the importance of language as a tool and on the processes of knowledge acquisition proposed by Vygotsky. According to the model of Cognitive Apprenticeship, instructors must design activities that require expert-like strategies, providing the necessary scaffolding to the completion of that activity. During the learning process, students are encouraged to explain what they are doing and why, critiquing their own work as well as that of their peers, thus externalizing their thinking processes. The processes of internalization and externalization through verbal expression help them understand and acquire the higher-order skills that they can apply to a variety of situations. The first phase is that of *modeling*—the instructor tackles a problem and explains what his/her reasoning is while trying to solve it. After that, it is the students' responsibility to solve a similar problem, verbalizing their thinking processes as they work, with the guidance and support of the instructor—this is the *coaching* phase. As the activity advances, the scaffolding fades steadily as students prove that they can continue on their own—this last phase is that of *fading*. The key of the Cognitive Apprenticeship model is dealing with real-world issues, rather than "textbook" problems—the outcome must have professional relevance, i.e. it is applicable/usable outside of the learning environment. This applicability can encourage and motivate students throughout the activity, especially if the result is an artifact that can be used by others, shown outside of the academic environment, or become part of their portfolio.

5. UNDERSTANDING THE GAMBIT SUMMER PROGRAM IN TERMS OF SITUATED LEARNING

As we applied the models of learning described above to our summer program, we realized that our approach was rather consistent with the model of Cognitive Apprenticeship. The Scrum methodology seemed to be very much compatible with its pedagogical tenets, thanks to the decentralized structure that leaves most of the authority to the teams. The daily meetings where team members have to explain what they are doing and what problems they have, also parallel the processes of self-regulation and verbalization. We realized that the way in which we had incorporated testing and Quality Assurance into our structure was interestingly accounted for by LPP, particularly because it resembled the role of testing and QA in the professional world.

5.1 Modeling: The Problem Of Mentorship

One of the basic features of an apprenticeship environment is the presence of an expert mentor, which serves the students as a model for the work they have to carry out. In our case, both GAMBIT staff and a good deal of the Product Owners had professional experience in working in videogames. For instance, Chor Guan Teo, Executive Director of the Singapore section of GAMBIT, had worked in Electronic Arts and Lucasfilm Animation, while Scot Osterweil worked at The Learning Company before joining The Education Arcade, to name but a few.

However, modeling could not take place in the Summer Program as it were an apprenticeship situation. The mentors were not making the games directly, nor were they observed by the students while they work. Rather, they helped the students become familiar with the vocabulary and methods of the practice. They provided the guidelines and helped with the overall organization of the lab, such as setting up lab-wide deadlines and meetings across teams that the students had to reach. This provided a framework to work within, but was not modeling as such. This is a drawback in our program that is difficult to address, since there seems to be no room for the mentors to make a game and have the students as apprentices.

5.2 Coaching And Fading

While the mentors had very little room for modeling, most of their effort focused on coaching the students, working along with them all the way through the development process. The students were relatively independent, with their Product Owners and GAMBIT staff checking on them regularly. It was the job of the mentors to anticipate what problems the students may be running into, and steer them into the right direction. The most difficult part of the job was deciding when to assist the students, since making mistakes can be the best way to get a point across. On the one hand, mentors wanted to see a complete game delivered after eight weeks, but they knew that the weight of the responsibility eventually fell on the students.

5.2.1 Self-regulation

As dictated by the Scrum methodology, groups were relatively autonomous. Mentors supervised and provided scaffolding as needed, although at times it was a tough call to evaluate how much help the students required. In our experience, the behavior of students seems rather consistent with the description of coaching and fading [2]. As the project advanced, the students would incrementally regulate their work themselves, rather than having the instructor manage the teams, and come up with their own way of dealing with problems. For instance, it was the initiative of the Scrummasters themselves to not follow Scrum in the last week or so, and come up with their own management structure, in order to finish the games in time and in compliance with the standards required by GAMBIT.

5.2.2 Distributed Knowledge

Cognitive Apprenticeship also describes how skills and processes are distributed amongst the different members of the group, so that the problem-solving is communal. This usually means that students may take turns in playing different roles, becoming familiar with the entire process. This is not completely possible in our lab—students usually take up the role that they already have the skills for, particularly in the case of programmers and artists. However, they need help from others to complete a videogame and depend on their knowledge, and in the process also learn about other aspects of development by working with people from other disciplines.

5.2.3 Verbalization Of Internal Processes

The revisionist nature of Scrum matches very well the aspects of Cognitive Apprenticeship that involve verbalization, where students are required to evaluate their work by writing it down or by explaining to their peers what they are doing. There were constant opportunities to explain and justify what they did, as well as to critique their own work and that of their peers: from the daily Scrum meetings, where students had to expound what they were doing, to their interactions with each other, which usually implied that they were talking to someone from a different discipline (and probably from a different country). Meetings with the Product Owners and Open Houses also expanded the variety of audiences, whose changing assumptions and knowledge they had to address in order to communicate effectively.

5.3 Intrinsic Motivation

Another aspect that is emphasized by theories of Situated Learning in general is the relevance of the work outside the academic setting. Cognitive Apprenticeship focuses on moving away from “textbook” problems, and to tackle issues that are relevant and applicable to the real world. Collins et al. [2] also emphasize the value of the work that is driven by intrinsic motivation, i.e. it is personally meaningful and its goal is interesting and coherent to the learner. This is different from providing external rewards to the learner in the form of extrinsic motivation, as in getting a good grade or a salary, for example.

Making videogames seems to be easier to become personally relevant than other jobs, so spurring that intrinsic motivation may seem slightly easier. However, we have observed that at times the students had certain expectations about what the development

process is like, which are shattered when they start working on an actual game. Some of these preconceived ideas range from thinking that the game has a sole “author” (usually either a programmer or the designer), or that if a game is not fun from the get-go they have already failed. When it dawned on the students that making videogames is a tough job, it was up to the mentors to remind them of further aspects that can make their work personally relevant, such as showing it to others or its relevance within a research project.

5.3.1 Showcase

During development, the students had to show their work to others periodically—to their peers, their mentors, or any visitor who came to the GAMBIT offices. By having them demonstrate their work, we emphasized its relevance outside the lab, reminding the students that there were actual players for their game. The showcase also stimulated intrinsic motivation, not only because of the thrill of seeing others enjoying your game, but also because eventually it would become part of their professional portfolio.

5.3.2 Research

Another way to make the work of the students relevant outside the learning environment is that all the games are intended to address a specific research question. The games did not only have to be fun, they were also at the service of the research question posed by the Product Owner. For instance, two of the games last summer used the Wii remote connected to a PC in order to explore different questions of interface design, *Wiip* and *AudiOdyssey*. *Wiip* was an attempt at trying to cater for expressivity in physical interfaces (simulating a whip, in this case), a question that Alex Mitchell, its Product Owner, wanted to explore. *AudiOdyssey* was the basis for the master’s thesis of its Product Owner, EitanGlinert, which focused on accessibility in games. In this case, the challenge was to design a game that could be played both by the visually impaired and people with normal vision, since the games that are designed for one demographic normally cannot be played by the other.²

Another research purpose was using the games as tools. This was the case of *TenXion*, an online multiplayer shooter, which was designed as a data mining tool for Artificial Intelligence research. The game logs in gameplay data that reflects the choices of the player, so that different strategies and gameplay styles can be recorded. This data will help provide insight into player decision-making processes, which is part of the research for the PhD thesis of the Product Owner, Jeff Orkin.

5.4 Legitimate Peripheral Participation (LPP) And The Role Of The Tester

The figure of the tester is instrumental for introducing students to game development; by working in Quality Assurance, students learn by following patterns very close to those accounted for by Legitimate Peripheral Participation (LPP) [8]. In the games industry, the position of the tester has been traditionally thought

² For a more detailed account of this specific project, the origins of the research question, how it was tackled, and the analysis of the final game, see [7].

of as an entry-level position for future game designers. Since there are still not many schools that teach videogame design, the industry itself inadvertently established an apprenticeship model for its newcomers that follows the processes described by LPP. This model is not likely to disappear, since it also helps newcomers learn the dynamics of a specific work environment, whether the company is large or small. The job of testing a game is a complex process, which demands methodical and exhaustive analysis in combination with creativity. However, it may not require extensive knowledge of coding, for instance, unless the novice does not intend to stay in Quality Assurance and wants to become later a programmer, or artist, or sound engineer, for example. Testers are also in a unique position to learn about the whole process, because they have access to everyone involved in production, from the programmers and artists to the managers, who have the responsibility of ensuring that the final product meets the quality required by the client. Conversely, testers who work off-site or in an outsourcing company are not part of an apprenticeship environment, and are therefore not likely to become familiar with other parts of production. The peripheral nature of the tester role seems to be ideal for students and scholars who wish to know about more the development process, even if they do not intend to make games.³

In the case of our Summer Program, the role of tester served as a position for students who had no previous experience in making games or in software development. The position also allowed students who were interested in studying games, rather than making them, to become involved in the lab activities. As coordinators of focus testing, lead testers learned how to deal with players as experimental subjects—how to select subjects, how to make them feel at ease in the testing environment, how to observe the subject without obstructing their gameplay, how to interview them after they played. This part of focus testing actually overlapped with the background of two of our lead testers, who came from Brain and Cognitive Science, giving them an opportunity to put their prior knowledge into practice.

Testers were also considered the “second designer”, the advocate of the player in the process, so their role was not as peripheral as it might be in a professional setting. They were in the studio working with all the other members, and their input was key for testing technical issues, and as part of the game design process. Even if the position of tester may seem peripheral with regards to the specific skillset required, it is actually vital in the whole game development process. In our case, it has served as an introduction to game production, encouraging students to become more central to the practice as Scrummasters of later projects. As mentors, testing also allowed us to evaluate the attitude and the aptitude of the students, so that we better understand how to match their responsibilities in the lab to what we already knew they could do best. For our students, it has also opened new careers options—some of our testers are now have taken jobs and internships in the videogames industry, even if they had not seriously considered making games as a career before coming to GAMBIT.

6. SUGGESTED IMPROVEMENTS BASED ON SITUATED LEARNING

For the 2008 GAMBIT Summer Program, we have planned a series of changes to improve the learning experience of our students. Many of these changes are inspired by becoming aware of the learning processes that take place during the whole program, and are based on our own observations as well as on the realization that we can incorporate Situated Learning models into the program.

6.1 Emphasis On The Role Of The Mentors

One of the main changes that we have introduced in GAMBIT is the emphasis on the role of the Mentors. The lab has hired a full development team (Lead Producer / QA Manager, an Art Director and a Technical Director and Lead Designer), all fresh from industry jobs. Having specialists on each discipline will hopefully help introducing the modeling phase in our Summer Program, since every specialist mentors the students within that discipline (Scrummasters, QA leads, artists, programmers, designers and audio design).

In our first summer, all Scrummasters and Product Owners had a meeting every two weeks, where they exchanged experiences, described what was going well or not so well in their respective projects, and gave each other feedback. Testers also ended up working together a couple of times a week towards the end of development, evaluating each other's games and proposing new testing methods for all games. These discipline-specific exchanges turned out to be quite beneficial, since students learned from their mentors as well as their peers, and it showed in their productivity and the improved quality of their work. In the 2007 Summer Program, we were providing general guidelines for all students, and only Scrummasters and testers had specific pseudo-modeling on their respective disciplines. Given the success of discipline-specific meetings, we are fostering discussions in our current session by having programmers meet together with our Technical Director, our artists with our Art Director, and our designers with our Lead Designer, in addition to Scrummasters and testers (QA Leads). This change is also inspired by the importance of mentors in modeling the practices of the students, as proposed by Cognitive Apprenticeship, although we have not found the best way to incorporate modeling in our teaching yet. The emphasis on this type of mentorship complements what students learn about working with people from other disciplines; in fact, the issues that Scrummasters or Testers had with the other disciplines involved were a recurring topic in the discipline-specific meetings.

By giving students more opportunities to meet with their peers, explain what they do and comment on the work of others, we are also emphasizing the importance of communication skills during the development process. Thus we give the students further opportunities for self-examination, criticism and communication with their peers and mentors; opportunities which are integrated in the production process rather than as an extra activity on top of all the development work.

³ See [9], where Niedenthal discusses the diversity of testing practices in a videogame company.

6.2 Communication: Use Of CSCW Tools

Communication is encouraged in face-to-face meetings, as well as through the use tools for Computer-Supported Collaborative Work (CSCW), from email to wikis and bug-tracking systems. In the summer 2007, these tools were available for students to use in their projects, but teams had the responsibility of figuring out how to best serve their purposes. They could use index cards or spreadsheets if that worked for them better. It was good for the students to try different management methods, as long as they worked for each individual team. However, the heterogeneity of approaches for keeping track of tasks was a problem when it came to communicate requests and issues to the Audio Team, which provided assets for all the other teams. The Audio Team would get lists of assets via email, the bug tracking system, and even on handwritten notes, requiring extra work from them to track sound asset needs. In our current Summer Program, we are giving them more specific guidelines about how to use the CSCW software, not only to encourage professional practices, but also to streamline the communication across teams. A new bug-tracking system, FogBugz, is being used for sound asset requests, as well as for reporting bugs and hosting each team's wiki. This facilitates comparisons and discussions between teams when they share their experiences in the discipline-specific meetings.

7. CONCLUSION

With our case study, our purpose has been to analyze one aspect of teaching a specific area of game studies, which is game development, and to suggest ways to improve and relate it to other areas of the field. As we have seen, the Scrum methodology matches very well the tenets of Cognitive Apprenticeship, while Legitimate Peripheral Participation has helped us understand how testers learn in a work environment, and how translate that to a research environment. Thanks to Situated Learning, we have realized of the importance of mentorship and communication in learning professional practices, and have come up with guidelines to improve our teaching.

Given how productive the application of Situated Learning theories has been, we are also considering setting up a new experiment in our Summer Program next year. The aim would be to include a section of the program where modeling could take place (as proposed by Cognitive Apprenticeship). After modifying our structure, we would also take data (e.g. video recordings, interviews with the students), in order to better evaluate the success of our experiment. This is still at a very early stage, since it may require an important overhaul of part of our Summer Program and requires very careful planning.

Another avenue worth exploring, inspired by the work expounded here, would be resorting to other theories of Situated Learning, such as Activity Theory [4], in order to study how videogame development teams work. Activity Theory focuses on the working processes, and the tools produced in the work environment. Those tools are usually conceptual models and processes, which could be not only be taught to students, but also analyzed and modified in an academic environment as an experiment, in order to find ways to improve those tools.

Professional methodologies can be beneficial when transferred into an academic environment, particularly in the case of methodologies that deal with teamwork strategies. By applying

professional practices to a school project, we are also translating the social dynamics of the workplace, giving the students the opportunity to learn how to work professionally before they finish their studies. In a way, our summer is a simulation of a professional environment, and as such, we offer our students more room for experimenting with new methods that we would outside of academia.

8. REFERENCES

- [1] 2008. IGDA Curriculum Framework - version 3.2 beta. http://www.igda.org/wiki/Game_Education_SIG/Curriculum
- [2] Collins, A., Brown, J. S., & Newman, S. E. 1990. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick, Ed. *Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics*. Lawrence Erlbaum, Hillsdale, NJ.
- [3] Crawford, C. 2005. *Chris Crawford On Interactive Storytelling*. New Riders, Berkeley, Ca.
- [4] Engeström, Y. 2008. *From Teams To Knots: Activity-Theoretical Studies of Collaboration and Learning at Work*. Cambridge University Press, New York.
- [5] Ficocelli, L. 2007. Industry - Academia: How can we collaborate. *Loading...* 1, 1.
- [6] Glinert, E. *The Human Controller: Usability and Accessibility in Video Game Interfaces*. Massachusetts Institute of Technology, Cambridge, MA, 2008. <http://web.mit.edu/glinert/www/thesis/index.html>
- [7] Grigsby, N., Glinert, N., Tan, P., Fernández-Vara, C., & Jenkins, H. In Press. Between Theory and Practice: The GAMBIT Experience. In Perron, B. & Wolf, M. J. P. *The Videogame Theory Reader 2*. Routledge.
- [8] Lave, J. and Wenger, E. 1991. *Situated learning : legitimate peripheral participation*. Cambridge University Press, Cambridge [England]; New York.
- [9] Niedenthal, S. 2007. Real-Time Sweetspot: The Multiple Meanings of Game Company Playtests. In *Proceedings of Third International Conference of the Digital Games Research Association (DiGRA): Situated Play*, Akira Baba, Ed.
- [10] Seely Brown, J., Collins, A. & Duguid, P. 1989. Situated Cognition and the Culture of Learning. *Educational Researcher*, 18-1.
- [11] Schechner, R. 2004. Performance Studies: The Broad Spectrum Approach. In H. Bial (ed.) *The Performance Studies Reader*. Routledge, London / New York,
- [12] Schwaber, K. 2004. *Agile Project Management with Scrum*. Microsoft Press.
- [13] Wertsch, J. V. 1985. *Vygotsky and the Social Formation of the Mind*. Harvard University Press, Cambridge, MA