Teaching 'hardcore science' to arts and design students: Reflections on the development of a basic programming course

Anders Nordby & Sidsel Karlsen[1]

Hedmark University College

Abstract: This article reports on the longitudinal development of a computer-programming course designed to meet the needs of students who enrolled in a specific higher education game development programme during the period of 2006 to 2010. The students came from three different arts and design-related strands of the programme, and had that in common that very few had taken advanced science classes as part of their upper-secondary education. This again meant that they were rather poorly equipped for learning computer programming, which they needed to master in order to tap into the full potential of the interactive and creative processes which their computers allowed for. Consequently, the programming course was designed in a way that allowed the students to practically engage in creating a computer game alongside being taught the actual programming skills, as well as the mathematics and physics needed in order to efficiently utilise those skills. A working hypothesis for the project was that if the responsible teachers were able to run the course in a way that cohered with the principles of problem-based learning, this would create an environment which would enhance the students' motivation to learn basic programming as well as the operative and innovation skills needed for fulfilling the course requirements. In addition, ideas developed within the field of situated learning constituted theoretical points of departure for developing the course. The article describes the practical and theoretical points of departure for developing the programming course and reflects on the experiences made from running it. Summing up, the authors conclude that the *why* and *how* of teaching needs to be in line with students' worlds in order for educational experiences to be considered as meaningful.

Keywords: arts and design students; computer programming; gamification; problem-based learning; situated learning.

Introduction: The gaming generation

During the past 20 years, videogames have become increasingly popular, in fact to such an extent that, in 2010, as many as 65 percent of all American households admitted to its members playing such

games (Education Database Online, 2010). Online games and communities are widely played and extensively used by young people, and to a rapidly expanding degree. This same generation of people enters our educational systems with the experiences and knowledge they have gained through playing. In this article, we take this phenomenon as a point of departure for investigating how students' gaming backgrounds can be taken into account when designing higher education. More specifically, we look into the longitudinal development of a specific course on computer programming designed to meet the needs of the arts and design students who enrolled in the Game Development Programme at Hedmark University College during the period 2006 to 2010.

So, if anything, what characterises the young gamers? Interviewed about his book on the net generation (see Tapscott, 2008), Tapscott elaborates on what the younger internet users demand from life, education and future employment, and claims that the concept of *fun* is crucial to their participation and motivation: "They want to have fun. In fact, 58 percent of them say that having fun with a product or service is just as important as what that thing actually does. If you employ any of these people, realize that they also want to have fun at work. They want to collaborate and have relationships. They want innovation and creativity. They want speed. They want to customize everything … this group wants to do things their own way" (Davis, 2008). When the members of the gaming generation enrol in higher education they may have similar needs and wishes without those necessarily being met, among other things because educations and courses are designed in quite traditional ways.

Gaming and the aspect of learning

Needless to say, the facts and claims exhibited above have not gone unnoticed by researchers and learning designers. For example, Gee (2004, 2007, 2010b), Prensky (2011) and Shaffer (2007) argue that well-developed games may be used as learning environments. Also, they point out that gamers normally learn in inductive ways, by doing, and not deductively by reading the written manual before engaging in the game. Gee (2010b) describes how he himself tried to do the latter, but found himself sadly unequipped for the actual playing. Then, he tried to do it the other way around – playing the game for a while and then reading the manual – with the consequence that the manual appeared as both lucid and instructive. In other ways, he engaged in the game as the average underage gamer does: he started to play the game and then moved on to the written instructions whenever that was necessary for solving particular challenges that occurred along the road. Such an approach might be labelled *learning by doing* (Dewey, 1916) or *learning just in time* (Gee, 2007).

Video or internet-based games are typically multimodal. The player's visual environments consist of movable graphics – often presented as splendidly and meticulously painted interior or exterior views – as well as an animated character that represents the player and which can often be transformed, to some extent, in accordance with the player's personal preferences. In addition, the environment has audible features, such as a more general soundscape and particular sounds and musics that trigger and underpin the player's actions and experiences. Designed well, these features enable the player to immerse herself completely in the game and seemingly forget the fact that she is engaging in tasks that require large amounts of focus and actual learning. As Vangsnes (2009) reminds us, this immersion often also involves role-play, which may enhance the gamer's experiences of learning in additional ways.

The player's learning may be understood as cross-disciplinary as well as multimodal. Since games often cover many topics, the player will learn them all at once and in an integrated fashion. Moreover,

games offer continuous and immediate assessment, which contributes to the learning, and with the evaluation procedures built in as part of the playing. In a keynote speech at the Meaningful Play 2010 conference, Gee (2010a) puts it this way: "All a video game is, is a set of problems that you must solve in order to win. The game itself does assessment all the time. No test is necessary when the game is completed to check whether the gamer has learned, it is proven throughout the game that he has."

Pearce (2011) points to some further key elements concerning games and learning. First, every good game has a clear goal and provides the tools necessary for reaching that goal. Since an easy game is considered boring, obstacles and challenges – and their related areas of learning – are obligatory components if a game should be perceived as 'fun to play'. Second, online games provide fruitful environments for collaboration, peer learning and competition. Third, as Kapp (2012) has shown, because most games provide different levels customized for players with various degrees of competence, the games may also offer personalized learning experiences and environments in which the player has the freedom to take chances and to fail without this having severe consequences for playing the game as such.

Recognising that games most often encompass features that require learning in one form or the other, some educationalists have sought to make use of these characteristics and develop educational games (Egenfeldt-Nielsen, Smith & Tosca, 2013) out of what traditionally have been thought of as 'boring topics'. This approach, sometimes named *gamification* [1], is defined as "using game-based mechanics, aesthetics and game thinking to engage people, motivate action, promote learning and solve problems" (The Learning Circuits Blog, 2012), and may be introduced in traditionally non-gaming environments, such as schools and universities, in order to create dynamic and competitive learning environments for students. Since most students nowadays come with game-related experiences of 'doing and learning before reading', which in many ways contradict the traditional approach to knowledge acquisition found within schools, aiming to transfer some of the learning-enhancing features of games into the educational sphere seems like a fruitful approach in order to strengthen students' motivation and build on their already acquired learning strategies.

With this knowledge about the gaming generation and its members' potential learning experiences in mind, the aforementioned computer programming course at Hedmark University College was developed so that it would better fit the students' needs and correspond more efficiently with their previously acquired skills and knowledge. Most of the students who enrolled in the course belonged to the gaming generation and had, as such, many years' experience of playing in video- or online-based multimodal, collaborative and fun learning environments. However, when it came to *creating* such or similar environments and artworks, and thereby tapping into the full potential for artistic and interactive processes that their computer-based tools allowed for, the students were sorely lacking in relevant knowledge and skills related to computer programming, as well as in knowledge of advanced mathematics and physics needed to efficiently perform such programming. Moreover, they did not even consider it interesting or important to learn any of these 'hardcore' topics. Describing how the computer-programming course developers accommodated to the situation at hand, and thereby laying out the theoretical framework, rationale and workings of this specific course, the following questions came to guide our explorations: How can the 'fun' experienced in gameplaying, modding, gamification and online communities be brought into so-called 'boring' topics in higher education? How can gameplaying and game development be used in order to teach 'hardcore' subjects like programming (and its related mathematics and physics) to art and design students in a university

course on computer programming? How can problem solving, collaboration, learning by doing, learning just in time and innovation be implemented as part of such a course?

Gamification in education – a brief overview

Using games as a part of formal schooling is nothing new; in fact games have been employed in education generally as well as in the teaching of natural sciences specifically for quite a while, and gamification has even been put to work in courses on basic programming in other universities. A well-known and still ongoing project in this respect is IBM's *Robocode* (Nelson, 2011; see also Long, 2007) in which the students create 'intelligent' tanks which have only one purpose, namely to beat other tanks and win the game. This particular game is designed to teach students the basics of natural science phenomena and presents them with challenges related to, for example, physics, mathematics and artificial intelligence. Each time the students change the programming code or try out a new theory in order to overcome the obstacles created by other tanks, they receive immediate visual feedback from observing the alterations in their own tank's movement and capabilities. Another relevant example is a Finnish game (Rajaravivarma, 2005) created with the purpose of enhancing students' learning of special programming skills through the programming of simple numbers and wordplays. Furthermore, a Danish game-based environment exists (see Andersen, Bennedsen, Brandorff, Caspersen & Mosegaard, 2003), which is formed to teach multimedia students with an artistic background basic computer programming skills. With respect to the informal features of web-based learning, Partti and Karlsen (2010) show how these are played out in an online music community, with a special emphasis on peer learning and reviewing, sharing of information and learning through experimentation. They also discuss how such internet-related learning strategies can be implemented as part of compulsory music education (see also Partti & Westerlund, 2013). Despite the subject areas being vastly different, we find these contributions to be highly relevant and adaptable to a course on computer programming, encouraging the students to learn as much as possible from each other instead of providing a traditional teacher-led learning environment.

Theoretical points of departure for developing the course

On the basis of the above, a working hypothesis for the project of developing the course in computer programming at Hedmark University College was that if the teachers responsible for the course were to be able to run it in a way that combined student tasks of actual game development with teaching and facilitating founded on the principles of problem-based learning (PBL), this would create a working environment which would enhance the students' motivation to learn basic programming as well as their operative and innovation skills in this particular area. Furthermore, their learning was expected to proceed more efficiently in such an environment than was the case when the course was run using more traditional teaching methods, such as lectures. In addition to the game development tasks and the PBL teaching methodology, another prerequisite of the course was that it should incorporate more informal online activities such as creating web pages and blogs and participating in online communities and forums into the formal sphere of the university context. The aim of this approach was to maintain a sense of 'fun' as well as linking students' outside-of-school experiences to their educational reality. Theoretically, the course design was informed and inspired by the already mentioned PBL-approach as well as by ideas found within sociocultural theories of situated learning (Lave & Wenger, 1991; Wenger, 1998). In the following we will aim to explicate these theoretical frameworks and their contributions to the course design.

Problem-based learning

The concept of problem-based learning (PBL) originates from medical school, and was first implemented in the late 1960s by Howard Barrows and his colleagues (Barrows, 1980). Originally, it was a model for facilitating student-centred learning in small groups, ideally six to 10 persons, led by a tutor whose task it was to guide the students, more so than to teach them, and the focus of the group was to solve real-world problems related to the professional world which the students aimed to enter. Since its beginning in the 1960s, PBL has established itself as an independent pedagogical concept or tool in higher education. Leaning on Pettersen (2005; see also Dochy, Segers, van den Bossche & Gijbels, 2003) the present-day understanding and principles of PBL could be described as follows:

1. The problems build on practice-related descriptions, case reports and case studies

2. Students – both individually and in groups – receive support, assistance and monitoring from a tutor

3. Students will engage in learning activities and develop ways of working and learning as well as strategies for studying along with practical problem solving and logical reasoning

4. Students have autonomy of their learning and study activities and emphasis is put on self-regulation and student autonomy

5. The teaching, the curriculum and the actual courses are organized in ways that allow and maintain cross- and interdisciplinary approaches

6. The study programme that the students follow facilitates their early contact with future and possible clients/users, and authentic tasks and challenges are given which qualify for the students' future professional lives

In addition to following the above principles, a teacher planning a course adhering to the PBL model will normally facilitate for the students to follow these seven steps (Maastricht University, 2013):

1. Clarify difficult terms

2. Define the problem

3. Brainstorm

4. Take stock of and analyze the possible solutions provided in step three

5. Formulate learning objectives

6. Self-study

7. A post-discussion

Step one to five points to the initial phase where the tutored groups of students define the learning goals of the given case through brainstorming, and identify which theory it is relevant to study for solving the case. Step six represents the implementation phase where the actual work with the case is done, while step seven designates the stage of summing up the results and discussing one's experiences from the work accomplished.

Situated learning

As mentioned above, the field of *situated learning* and the idea that learning is integral to participation in communities of practice (Lave & Wenger, 1991; Wenger, 1998) also informed the design of the programming course. Underlining that their theory is meant as an "analytical viewpoint on learning, a way to understand learning" (Lave & Wenger, 1991, p. 40) and not as a set of prescriptions for how to facilitate learning, not to speak of designing education, the authors still present ideas worthwhile pondering for teachers and university faculties who wish to design learning environments while taking

the participatory and interactional aspects of knowledge creation into account. According to Lave and Wenger, learning in a community of practice happens by legitimate peripheral participation, a position that is accessible to each and every one of the community's members simply by them "being located in the social world" (p. 36) which the community constitutes and affords, and partaking in its ongoing activities. However, just grouping people together does not automatically form a community of practice. One of its core characteristics is the "mutual engagement of [its] participants" (Wenger, 1998, p.73) centred around a joint enterprise and exhibiting a shared repertoire of, for example, actions, artefacts, stories and tools. Listing in all 14 indicators that a community of practice has been formed, Wenger emphasises, among other things, "sustained mutual relationships – harmonious and conflictual … shared ways of engaging in doing things together … knowing what others know, what they can do, and how they can contribute to an enterprise … mutually defining identities … specific tools, representations, and other artifacts [and] local lore, shared stories, inside jokes, knowing laughter" (p. 125). Since learning is considered to happen in and through shared and concrete undertakings, it can be accessed (and even facilitated) by considering the relations between the newcomers and the old-timers of a particular community as well as between its range of activities, identities and artefacts. The key to participation for a newcomer is to gain access to the community of practice in question "and all that membership entails" (p. 100), and while there exists no such thing as a centre or core of a learning community, prolonged membership and vivid participation may lead to the more mature stage of "*full participation*" (p. 37, italics in original). In the particular context of the programming course, the tutors employed (see below) and those of the students who had previous experience of creating computer games would typically act as more full members than the rest of the students – at least to begin with – and those with less experience would start the work from the newcomer's position.

For educators who wish that their students should grow and reach the full member-level of proficiency in relation to, for example, participation in the community of game designers and developers, it is crucial to take into account the intimate connections between practice, community, identity and meaning and acknowledge learning, not first and foremost as acquisition of knowledge, but as doing, belonging, becoming, and as experience (Wenger, 1998, p. 5). If knowledge is seen as "a matter of competence with respect to valued enterprises" (p. 4) – in this case the enterprise of creating computer games – information about what kinds of knowledge and skills are valued *within* this specific enterprise or profession is crucial for meaningful participation and future success. Moreover, considering Wenger's notion of *trajectories*, which addresses how participants move between, participate in and learn "both within and across communities of practice" (p. 154) as well as how the identity-work of "constant becoming" (p. 154) is key to forming such paths of learning, designing a course which allows the students to bring in their outside-of-university activities and identities, such as those of bloggers, gamers and participants of online communities, will most likely add to the meaningfulness and the feeling of belonging.

The design: Rationale and run-through of the basic computer-programming course

The course in basic computer programming at the Game Development Programme at Hedmark University College was implemented as a one-semester course, starting 2006. The first year served as a trial year, in which all the practical aspects of the course were tested and evaluated; for example which software [2] and game types to use, what kind of cases to present the students with, and how to mix the PBL-aspects of the course with lectures and lab work in efficient ways.

The students attending the course over the period covered in this article (2006 to 2010) mainly came from three different strands of the overarching programme in game development, namely from Virtual Arts and Design, Animation as well as Visual Simulation. They were all in their first year of studies. In addition, third- or fourth-year teacher education students for whom the course was optional and offered as one of several available specializations, would occasionally enrol. Typically, around 70 students attended the course each year, their age ranging from 18 to 25 years, mainly. As such, the students were typical examples of the gaming generation described in the beginning of this article.

The main reason for teaching computer programming to art and design students was, as mentioned above, to enable them to tap into the full potential for interactive creative processes which their computers allowed for, and which were seen as fruitful for the kind of artistic work that they engaged in. A big challenge, however, was that the students did not consider it interesting or beneficial to learn programming or any natural science topics for that matter. Very few of the students had taken advanced science classes as part of their upper-secondary education, which meant that the mathematics and physics needed in order to create a functioning game had to be learned alongside the other elements of the course. Although the main goal was to teach the students computer programming, we also wanted the students to experience why an understanding of certain 'hard science' topics was needed in digital arts education for them to be able to create the advanced computer-based artworks that they aimed for.

Before the implementation of the game-creating approach started in 2006, programming, mathematics and physics had been taught in more traditional ways. However, the general examination statistics of the university college showed that approximately 50 percent of the students failed, and consequent informal talks with the students indicated quite a lot of frustration with the teaching methods employed. Hence, new ideas for the design of this particular part of the students' education were needed. Gamification was much approved as an educational motivator around the time when the basic computer-programming course was first designed, and a decision was made to use this as one possible road of teaching the necessary skills. As mentioned above, during the trial year many different game types were tested until the course developers landed on a platform game which seemed to contain all the elements and challenges needed to justify the teaching of what the staff knew would be the skills and knowledge about basic programming, mathematics and physics that would be required in the students' future professional life.

Apart from some basic elements of a platform game, such as a main character that could be controlled from the keyboard and who should be able to walk, run and jump, the students were given very few specific guidelines or restrictions on how to design the game. The course developers implemented the PBL approach in a similar way to what had previously been done in Finland (see Kinnunen & Malmi, 2005; Nuutila, Törmä & Malmi, 2005). The students were divided into groups in which they were to work in collaboration throughout the course. Mostly, but not always, the groups were formed according to suggestions made by the students themselves about who they would like to work with. Within the frames of these groups each individual student was responsible for creating and completing his or her own game. In line with the PBL thinking, each group had their own tutor, which would typically be higher-grade students employed for this job specifically and with the necessary experience. Although the course was built around specific cases and tasks to be solved, lectures and guided lab work were not entirely abandoned as a way to facilitate the students' learning. Starting the course, short lectures were given on how to approach the PBL working method as well as the cases, and in order to prevent the students from becoming stuck while working on their projects, guided lab

work was implemented in-between the case periods. Following the PBL model, the game-development project was divided into smaller tasks or steps that were presented as problem-based cases. While the actual number of cases would vary somewhat from year to year, a typical order of tasks would be:

1. Design the game and make a storyboard

2. Design the characteristics of the main character of your game and then make a drawing of this character

3. Divide the character into layers, animate its walking cycles and jumping abilities

4. Connect the triggering of the walking cycles to the keyboard arrow keys [3]

5. Draw the ground of the game and make the main character follow this ground

6. Make a second character (an enemy) and let it follow the main character

7. Implement jump for the main character and design platforms to jump on

8. Make a starting screen with the necessary menus as a well as screens to be shown when winning or losing the game [4]

The cases were deliberately designed as open-ended so that eager students could develop them beyond the basic requirements without proceeding to the next case and thereby moving too fast through the planned steps of the course.

During the last part of the course, the student groups used Wiki pages to present their individual projects. In addition, each student wrote a blog after the completion of every step or case about how the game they had chosen to create progressed. The blog tasks were obligatory and they were also part of the final report that all the students submitted at the end of the course. Chat tools and an internet-based forum were used throughout the project to facilitate collaboration among the group-members in-between physical meetings. This way, the students had the chance to develop both place-based and virtual communities of practice (Barab, Kling & Gray, 2004), something that was believed to enhance their overall experiences of learning.

Most of the Wiki pages and blogs were written in English so that the students could have feedback from the international community of gamers and game makers. The students were also encouraged and enabled to seek for online game communities in which they could participate, study and play games similar to the ones they aimed to create themselves as well as present their own projects, including their technical and artistic solutions, and discuss them with the other community members. Finally, the students were told to play their own games as they developed and in this way explore the 'play' aspects embedded in them as well as reflect on the emergence of the games and how they could be made more immersive.

An absolutely necessary tool throughout the course was each student's own PC and its accompanying software. To enable all the students both to create the games as well as play them later on without too many technical difficulties, they were offered a beneficial leasing deal that gave them access to powerful computers and upgraded software.

Experiences from running the course: Practical solutions and theoretical afterthoughts

In this section we aim to describe some experiences gained from running the computer-programming course as well as relate them to our theoretical points of departure as laid out above. Our discussion is based on the fact that one of us – Anders Nordby – was responsible for developing, implementing and

running the actual course over the period 2006 to 2010. Furthermore, although this is not an empirical article, but rather a theorized description of a longitudinal course-development process, what follows is also partly based on the experiences of reading through a large amount of feedback material collected from the students and the tutors who participated in the course over the given time frame [5].

First of all, it quickly became evident when following and evaluating the course that 'making a game' in fact came to constitute a common goal for most of the participating students, despite the fact that they arrived with vastly different backgrounds and were enrolled in several and quite different study programmes. As such, the entire class, and perhaps even more so the student groups, made up vibrant communities of practice (Lave & Wenger, 1991; Wenger, 1998) which engaged in this task as their primary joint enterprise. Hence, the gamification approach was largely successful in that it provided the students with a collaborative social environment in which they could participate according to their level of knowledge and skills – in other words, more-or-less 'peripheral' or 'full' (Lave & Wenger, 1991, p. 37) – and where the newcomers to the field of game-making could learn from those considered more experienced, whether that be other students or the tutors. It is of course hard to tell the exact reasons for the student groups actually developing into communities of practice (as can be seen above, this can by no means be expected to happen automatically just by grouping students together). However, one fairly qualified guess would be that this development was afforded by the fact that the students – as a general rule – were able to choose their own collaborators. Thereby they could also select persons with whom they shared other features crucial for the formation of a community, for example they could know approximately "what they [could] do, and how they [could] contribute" (p. 125) and they would probably also have "shared stories" (p. 125) already prior to embarking on the making of the games. Another potentially community-developing factor could be that the students in fact did share quite a lot of "specific tools, representations, and other artifacts" (p. 125) given to them during the run of the course, and in order to complete the given task they had to figure out how to utilise these resources *together*, thereby strengthening the mutual commitment and engagement of the groups. The fact that the task was steered towards actually *making a game* and not towards learning *how to make a game* (ref. Folkestad, 2006) [6] also seemed to enhance the students' motivation towards solving the task as well as give them insights not just into the particular and minuscule operations and pieces of knowledge needed for completing it; they also came to see how these 'fragments' belonged to the bigger picture. Furthermore, they developed an overview of how the specific topics taught all fed into creating a larger academic, artistic and practical context to which they needed to relate in order to complete their game in a successful way.

The social environment and communities of practice created through the course also successfully provided a 'learning by doing' (Dewey, 1916) context, largely through the utilization of the PBL approach, which forced the students to learn in an integrated way since the theory they had to seek out was always connected to practical tasks and operations embedded in the problem-based cases. As such, their theoretical learning always happened 'just in time' (Gee, 2007) for them to solve actual problems, and not in a row of lectures preceding the application of the factual knowledge. As examples of practical, game-related tasks given that effectively led to students developing knowledge in hardcore physics and mathematics, we here choose to emphasize two.

First, when the students had finished their game characters and started the process of animating them, one of the features they needed to add was giving their characters the ability to jump. For this to happen, they first needed to apply a force that allowed an upward movement. However, without making any modifications to that force, the character would simply disappear out of the top of the screen. In other words, this was the right time to learn about gravity, since applying a gravitating force

would move the character back to the ground. Still, not just any kind of gravitation would do, since the aim was to make the character jump in a credible and convincing way with body-movements similar to those a jumping human or animal would exhibit in real life. Faced with this situation, very few students found it odd to read and talk about physics and gravity, and they usually realized quite quickly that they needed a solid theoretical foundation for understanding the forces and effects of gravity in order to be able to implement a well-working solution to the task at hand.

Second, when reaching the stage of the game development in which they should implement their characters' walking cycles and moves around the screen, the students needed to expand their knowledge about x- and y-axes, coordinates, angles, and conversion between degrees and radians. Eventually, they also had to learn trigonometry to be able to divide the movements into components. Usually, this is a kind of theory that students struggle to understand if it is taught separately and on an abstract level; however faced with practical problems to solve in order to make their games work, the rather complex physics and mathematics once again became comprehensible because it was to be applied to something very concrete and something which it mattered greatly to the students to be able to complete. In other words, the 'fun' (Tapscott, 2008) and 'motivation' aspects were evidently present all along the course.

As mentioned above, through the gamification and PBL-based approach, the students were provided with one or more communities of practice in which they could develop their knowledge and skills. Not just the different modes of participation or the fact that the focus was on actually making a game contributed to this learning environment. So too did the composition of the working groups, the access to 'qualified helpers' as well as a wide-ranging (and explicitly encouraged throughout the course) contact with other, relevant communities of practice composed of people with similar interests, knowledge and skills. The latter opportunity enabled the students to develop *boundary trajectories* and thereby "linking communities of practice" (p. 154) as well as experiences and knowledge acquired in different learning contexts.

In order to accommodate groups who could work together creatively, the teachers sometimes interfered with the group composition so as to make them as heterogeneous as possible (Darsø, 2000; Gee, 2007). After a trial period in which different group sizes were tried out, the groups stabilized on eight to 10 students, which seemed to ease many of the organizational aspects. At the same time, this group size correlates with what the theories suggest both when it comes to maximising of creativity (Darsø, 2001) and PBL learning (Pettersen, 2005). The heterogeneity approach encompassed variables such as study programme, interests, gender and age. In addition to the existing knowledge and skills within each student group, the groups were, as already mentioned, provided with a tutor, who represented a very significant 'old-timer' (Lave & Wenger, 1991) with respect to the actual game task. The tutors were carefully selected among the best students from previous years, and all of them put much effort and energy into their work.

Related to the already-mentioned concept of trajectories, Wenger (1998) also emphasizes the necessity for communities of practice to be engaged with similar communities (see above) in order to expand their practice-related knowledge, and coins the notion of *brokers* to denote individuals who "make new connections across communities of practice, enable coordination, and – if they are good brokers – open new possibilities for meaning" (p. 109). Likewise, he reminds us that people may have (or aspire to) community "*multimembership*" (p. 149) as well as a need to negotiate "local ways of belonging [and doing] to broader constellations and of manifesting broader styles and discourses" (p. 149). In the course development described in this article – and especially during the last two years of the period attended to – these aspects were facilitated and nurtured through the students being

encouraged to create web pages, blogs and promote their ideas and games in online communities and on web-based forums in which gaming and game development constituted the core point of interest for the members. Those of the students who belonged to such constellations beforehand, then had an opportunity to enact their multimembership and act as brokers between the university college-based game-creating communities and the web-based ditto. Furthermore, these outwardly directed "boundary trajectories" (p. 154) provided the students with valuable input and information from outsiders who were connected to international discourses on gaming. The input could range from practical and concrete help with getting the technical details right to broader discussions about what constituted a good, playable and interesting game from a gamer's perspective. The web pages and blogs also gave the students an arena for constructing and displaying – in a very visible way – their (often) newfound identities as game-developers [7]. Hence, these tools integrated the practice-, trajectory- and identity-aspects of situated learning in a very fruitful way, and might have contributed to the students experiencing their learning both as "belonging," "doing" and "becoming" (see Wenger, 1998, p. 5).

Concluding remarks

In the beginning of this article, we asked, among other things, how gameplaying and game development could be used in order to teach 'hardcore' subjects like programming, mathematics and physics to arts and design students in a university course on computer programming. Furthermore, we wondered how problem solving, collaboration, learning by doing, learning just in time and innovation could be implemented as part of such a course. Hopefully, the previous sections have provided the reader with some tentative answers and theorized practical solutions to these questions. Going back to Wenger's (1998) notion of trajectories, not all students enrolled in the programming course remained in the game-developer sphere. Rather, some of them entered "outbound trajectories" (p. 155) and chose to implement the knowledge and skills gained through the course in other parts of the world of virtual art and design. Other students, however, remained on the inbound trajectories afforded by the course and even chose to enter "insider trajectories" (p. 154), further participating in "new events, [meeting] new demands, [contributing] new inventions" (p. 154) and continued to renegotiate their identities as game-developers. Some of these students have since embarked on a course as professionals within the gaming industry (see for example Krillbite, n.d.; Moondrop, n.d.; Roofmoose, n.d.).

One of the main challenges of running the programming course is of course to be able to keep up with the rapid development of the field and to continue to provide new gaming and net generations (Tapscott, 2008) of students with fun and motivational educational experiences that correspond with *their* virtual and real-world realities. As such, this article is meant as a contribution to keeping the field afloat, showing that it is not just *what* you teach that matters. The *why* and, not least, the *how* of teaching also needs to be in line with students' worlds in order for them to consider their educational experiences as meaningful.

Endnotes

[1] While we have chosen to relate to a definition of gamification that specifically incorporates the element of learning, we are well aware that differing and even competing definitions of the term exist. For example, in Deterding, Dixon, Khaled and Nacke's (2011) investigations into the historical origins of the term, gamification is taken to refer to the use of game design elements in non-game contexts more generally.

[2] An important premise for choosing the software in which the game should be developed was that the students should be able to have a visual representation of all programming actions at any given time. Consequently, Adobe® Flash® was chosen since it allows the developer to carry out the object-oriented programming, the graphic design, the animation and the implementation of the game soundscape using only one tool, and at the same it provides an environment where the results of all actions can be viewed immediately. This was considered crucial for the work in the PBL groups (see below), where each session started with the students showing each other how their particular game had developed since their last meeting and commenting on each other's work. Furthermore the immediate visual and graphic feedback of the programming was seen as a way to connect the more theoretically oriented programming work to the course participants' identities as students of art and design.

[3] The actual programming tasks started here; this fourth case would typically last double the length of the three preceding ones.

[4] Additional cases nine and 10 were sometimes used, typically involving more advanced programming options, like the implementation of several game levels, more characters and more advanced character functionality et cetera.

[5] As part of a bigger action research project, a vast amount of quantitative and qualitative data has been collected which concerns the students' experiences of participation and learning within the frames of this particular course. Among other things, the data includes students' and tutors' reports after completing each case; students' reports after finishing the course; blogs created by the students during the run of the course; and a questionnaire distributed among all the students who participated in the course during the period of 2006-2010. While the findings from this study are not the concern of this article, the task of reading through and analyzing the student data forms an additional interpretative backdrop for our reflections in this section. Since the research was carried out within the frames of the researchers' own institution and among students which one of the researchers also met in the capacities of teacher and supervisor, the project was conducted with a special awareness of the ethical concerns that come with such situations (Brydon-Miller, 2008).

[6] In an article on formal and informal learning situations, practices and ways of learning related to music, Folkestad (2006) makes a distinction between *learning how to play music* versus *playing music*, claiming that the latter constitutes an important part of an informal learning situation or practice, and is often also imbued with a lot more ownership and agency on the learner's side than the former, more formally oriented approach. We believe this to be true not only for learning situations involving music.

[7] Similarly, Partti and Karlsen (2010) show how online music communities may provide an arena for people to construct and maintain their musical identities as for example musicians and composers.

On the contributors

Anders Nordby is educated in electronics, computer science and pedagogy, and he currently works as an assistant professor at Hedmark University College at IgA, a section in the Department of Fine Arts and Computer Science which focuses on interactivity, gamification and arts. His work is mainly focused towards games and learning, system thinking and sustainability, and recent writings centre on how gamification might be utilised in order to teach systems science and sustainability. Nordby teaches typical game design topics, such as general game design, 3D design, animation, sound production and programming, as well as pedagogy.

Sidsel Karlsen is educated as a music educator and classical singer and she currently works as professor of music education at Hedmark University College in Norway as well as docent at the University of the Arts Helsinki (Sibelius Academy) in Finland. She has published widely in Scandinavian and international research journals and has contributed to a number of international anthologies, such as *Sociology and music education* and *Collaborative learning in higher music education*. Her research interests span, among other things, the field of multicultural music education, the relations between music, identity, learning and new media, as well as the social and cultural significance of music festivals.

References

Andersen, P. B., Bennedsen, J., Brandorff, S., Caspersen, M. E., Mosegaard, J. (2003). Teaching programming to liberal arts students – a narrative media approach. *ACM SIGCSE Bulletin – Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*, *35*(3), 109-113.

Barab, S. A., Kling, R., & Gray, J. H. (2004). *Designing for virtual communities in the service of learning.* Cambridge: Cambridge University Press.

Barrows, H. S., & Tamblyn, R. (1980). *Problem-based learning: An approach to medical education.* New York: Springer.

Brydon-Miller, M. (2008). Ethics and action research: Deepening our commitment to principles of social justice and redefining systems of democratic practice. In Reason, P., & Bradbury, H. (Eds.), *The SAGE handbook of action research* (pp. 199-210). London: Sage Publications.

Darsø, L. (2001). *Innovation in the making.* Frederiksberg: Samfundslitteratur.

Davis, E. (2008). *Don Tapscott discusses the "net" generation.* Retrieved from http://blog.shop.org/2008/09/16/don-tapscott-discusses-the-net-generation/

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". *MindTrek '11. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 9-15).

Dewey, J. (1916). *Democracy and education: An introduction to the philosophy of education.* New York: Macmillan.

Dochy, F., Segers, M., van den Bossche, P., & Gijbels, D. (2003). Effects of problem based learning: A meta-analysis. *Learning and Instruction*, *13*(5), 533-568.

Education Database Online. (2010). *American households playing games*. Retrieved from http://www.onlineeducation.net/videogame

Egenfeldt-Nielsen, S., Smith, J. H., Tosca, S. P. (2013). *Understanding video games. The essential introduction* (2nd ed.). New York: Routledge.

Folkestad, G. (2006). Formal and informal situations or practices *vs* formal and informal ways of learning. *British Journal of Music Education*, *23*(2), 135-145.

Gee, J. P. (2004). *Situated language and learning: A critique of traditional schooling*. London: Routledge.

Gee, J. P. (2007). *Good video games and good learning: Collected essays on video games, learning and literacy.* New York: Peter Lang International Academic Publishers.

Gee, J. P. (2010a) Design, learning, and experience [Video file]. Retrieved from http://vimeo.com/16254768

Gee, J. P. (2010b). *New digital media and learning as an emerging area and "worked examples" as one way forward*. Cambridge, MA: MIT Press.

Kapp, K. M. (2012). *The gamification of learning and instruction: Game-based methods and strategies for training and education*. San Francisco: John Wiley & Sons.

Kinnunen, P., & Malmi, L. (2005). Problems in problem-based learning – experiences, analysis and lessons learned on an introductory programming course. *Informatics in Education*, *4*(2), 193-214.

Krillbite. (n.d.). *Current projects.* Retrieved from http://www.krillbite.com/index.php

Lave, J., & Wenger, E. (1991). *Situated learning. Legitimate peripheral participation.* Cambridge: Cambridge University Press.

Long, J. (2007). Just for fun: Using programming games in software programming training and education – a field study of IBM Robocode Community. *Journal of Information Technology Education*, *6*, 279-290.

Maastricht University. (2013). *Problem based learning preparatory website.* Retrieved from http://www.umpblprep.nl/

Moondrop. (n.d.). *Main page.* Retrieved from http://moondrop.no/

Nelson, M. (2001). *Robocode.* [Java], IBM.

Nuutila, E., Törmä, S., & Malmi, L. (2005). PBL and computer programming – the seven steps method with adaptations. *Computer Science Education*, *15*(2), 123-142.

Partti, H., & Karlsen, S. (2010). Reconceptualising musical learning: new media, identity and community in music education. *Music Education Research, 12*(4), 369-382.

Partti, H., & Westerlund, H. (2013). Envisioning collaborative composing in music education: Learning and negotiation of meaning in operabyyou.com. *British Journal of Music Education*, *30*(2), 207-222.

Pearce, K. (2011). *The 'Gamification' of education.* Retrieved from http://blogs.canoe.ca/canoetech/signs-of-the-times/72001/

Pettersen, R. C. (2005). *Kvalitetslæring i høgere utdanning: Innføring i problem- og praksisbasert didaktikk.* Oslo: Universitetsforlaget.

Prensky, M. (2011). Computer games and learning: Digital game-based learning. In Raessens, J., & Goldstein, J. (Eds.), *Handbook of computer game studies* (pp. 97-122). Cambridge, MA: MIT Press.

Rajaravivarma, R. (2005). A game-based approach for teaching the introductory programming course. *ACM SIGCSE Bulletin*, *37*(4), 98-102.

Roofmoose. (n.d.). *Factsheet.* Retrieved from http://roofmoose.com/press/

Shaffer, D. W. (2007). *How computer games help children learn.* New York: Palgrave.

Tapscott, D. (2008). *Grown up digital: How the net generation is changing your world.* New York: McGraw-Hill.

The Learning Circuits Blog. (2012). *What is gamification? And why it matters to L&D professionals.* Retrieved from http://learningcircuits.blogspot.no/2012/01/what-is-gamification-and-why-it-matters.html

Vangsnes, V. (2009). A dramaturgic perspective: seeing digital role-plays as drama and theatre. What are the epistemological and pedagogical consequences? *Nordic Journal of Digital Literacy*, *1*, 20-34.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity.* Cambridge: Cambridge University Press.