HOW TO DESIGN A GAMIFIED EDUCATION? THEORY AND DESIGN OF A GAMIFIED CURRICULUM FOR IgA, INN

Anders Nordby, Bård Gunnerud-Åhlén, Marit Christine Berg Strandvik
Inland Norway University of Applied Sciences (NORWAY)

Abstract
This project has two main goals; the first is how turn a traditional subject structured education, (meaning 1 course per subject matter) into a cross-disciplinary courses based education. The second goal is to gamify individual processes in each course, and breaking down the subject matters into small manageable pieces and show through a series of games, how these are linked together. For this example, we have based the project on a higher Ed. program focusing on computer game development. While goal one would need adjustment if used in other educational programs, goal two is non-program related. The design builds on research and development done at IgA (see below).

Only one cross-disciplinary course is taught at the time, and each course is structured over a 5-week period. Each following course will build on the content taught in the previous. The students start by developing mini-games and gradually increase the complexity. The first games they will develop will be platform game, which give the students a broad area to learn from, not only basic programming, but also math and physics.

All teaching will be based on practical game development, “learning by doing”, and the theoretical topics will be taught “just in time”; that is when they appear naturally in the game development process. The courses will be Problem Based (PBL) and divided into 6-10 cases, and the students will work in groups with teachers and higher-grade students as tutors. The assignments will be both individual and group based. Writing blogs will invite informal learning into the classroom and is therefore a key process in each course. Another key issue is to choose the right games, with the right blend of topics and level of challenges.

The gamification aspects will be maintained by making a competitive environment where students compete locally, both individually and between groups. They will also compete on the Internet with other university classes and peers in game forums and online societies. These competitions will be supported by a system of points, badges and leaderboard, with larger rewards for the students, groups or classes that win. In this way, informal learning will also be brought into the classroom; the students will be rewarded for using online peers in online forums and societies to comment on their projects.

In Interactivity, Games and Visual Arts (IgA) in Inland Norway University of Applied Sciences (INN) we have used game development in teaching for 10 years. INN has had complete game development tracks since 2003, and is now one of the biggest and most complete game educations in Scandinavia with 250 game students.

Keywords: Learning, Game Development, Gamification, Games and Learning, Pervasive games, Digital Games, Concurrent Design.

1 INTRODUCTION
While we have seen a huge change in children’s habits and play in the last years due to the rapid development of the Internet and technology, traditional education has not changed much.

In his book about the net generation [1], Tapscott claims that having fun with a product or service is more important than what the thing actually does. He points out that young children of this generation wants creativity and innovation, collaboration and relationships, they want speed and they want to customize everything and do things their own way.

Faster computers and internet have made it possible to make digital games and virtual environments, and more and more young children are playing games from an early age. World wide the video and computer game industry is generating $105 billion dollars[2], and in the US 67% of the households play computer and video games [3]. This market was in 2014 predicted to increase to $100 annually by 2017 [4]. Also the social gaming is increasing rapidly, the massive multiplayer online role play
game (MMORPG) is valued at over $4 billion, with well known titles like World of Warcraft, Club Penguin, Eve Online. Disney Toontown etc. A more recent example of this is Minecraft that now has more than 100 million users, mostly from the younger generation[5].

The use of game playing or game development to learn, is well documented. For example James Paul Gee sums up the most important principles that is built into games [6] - among other things Gee points out that good games are systems, and that students learns best when they see how problems and skills are parts of this bigger system, and that games can give a simplified version of the system that makes it a more transparent and easier understandable view of the context. When done right, learning in games can also be pleasantly frustrating and fun.

The use of gamification techniques is growing fast in industry and marketing, and we don't believe it is a trend that will vanish; it is rather growing and getting more momentum in more and more fields. As learning professionals, we see the need to understand and eventually adapt these techniques also to the education.

In our University we have had a constantly growing game development track for 14 years (Interactivity, Gamification and Visual Arts at Inland Norway University of Applied Sciences) This is now one of the biggest and most complete game educations in Scandinavia with 250 students. We will also by fall of 2017 offer a new post graduate education in Virtual Reality and Augmented Reality together with EON Reality (US).

What would be more natural than to gamify these tracks and make that an example for other educations? This is what we want to do and is the basis for this article.

2 SCOPe AND OBJECTIVES

The project has two main goals; the first is how to turn a traditional subject structured education, (meaning 1 course per subject matter) into a cross-disciplinary course based education. The second goal is to gamify individual processes in each course, and breaking down the subject matters into small manageable pieces and show through a series of games, how these are linked together. While goal one would need adjustment if used in other educational programs, goal two is non-program related.

Each year have 5 to 6 of these cross-disciplinary courses, each of 5-6 weeks length.

Our example will be a higher education program (bachelor), focusing on computer game development. Below this is sketched out:

![Diagram](image)

*Fig 1. This drawing illustrates the overall layout of the education.*
3 METHODOLOGY AND THEORETICAL ASPECTS

All courses will be Problem Based (PBL). Problem based learning originates from medical school, and has since 1960 established itself as an independent pedagogical concept or tool in higher education. We largely implement the 7 step Maastricht method [7], though with a few simplifications and additions. The problems build on practice related descriptions, in this case 5-10 smaller problems or cases that when solved will result in a complete game. The students will work in groups of 6-10 students, led by a tutor or a teacher. In each case the students will first discuss the case and a set of learning goals, then study and develop a solution individually and then meet again to discuss the individual solutions in the group. The assignments can be both individual and group based.

The environment in which the students work together and the game development process, can be seen as a “community of practice” [8] [9] where the students work towards creating a game. For them this will be the overall goal, although they learn the important school topics along the way. This is situated learning, and the situation described here is discussed in more detail in [15]

The students learn by experiencing this environment where they are creating the game, so we tap into experiential learning theories [10]. The game development is also clearly “learning by doing” [11], and they study theory when they need it to solve a game development problem. This is “learning just in time” [6].

The main goal of game development in class is not only to teach the students specific school topics, but also make them interested in learning more about these topics. Interest can mainly be caused by two things, individual interest that is learned through the individual’s experiences and preferences [12], and situational interest [13] that is gained from interacting with the environment. Interest gives motivation; individual interest gives intrinsic motivation [3] while situational interest gives extrinsic motivation [3]. Intrinsic motivation is in psychology seen as deeper and more sustainable, so a point will be to try to stimulate this rather than extrinsic motivation. They can also influence each other though, and extrinsic motivation can even kill intrinsic motivation if used too much. The student’s motivation depends on the response they get when presenting their games for other students in the class and in the PBL groups, as well as peers that the students meet in forums and game discussion groups on the internet.

4 CASE DESCRIPTION

The Bachelor requires no special competence other than K12. All students develop the same tasks in the first 6 months and after that they specialize in 3 directions: Game Programming(GP), Game Art and Animation(GA) and Game Design(GD).

A key issue will be to choose the right games with right blend of topics and level of challenges.
Using a System Thinking Causal Loop Diagram (CLD) we can illustrate the learning loop.

![System Thinking Causal Loop Diagram](image)

**Fig. 3.** The System Thinking Causal Loop Diagram shows the overall learning loop. The arrows show cause and effect, and a + means that the effect is increasing when the cause is increasing, while a - means the effect is decreasing when the cause is increasing.

### 4.1 How to turn a traditional subject structured education (one course per subject matter) into a cross-disciplinary course based education?

Each school year this education have 5 to 6 courses in total, each of 5-6 weeks length. The courses are always about games; in each course the students develop a game, play some games or analyze some games. Each course is problem based and are divided into 5-10 cases which the students have to solve to finalize the course. Together all cases forms a complete game project [6].

The courses are cross-disciplinary in that they contain all topics shown in fig 2, and only one course is taught at the time. Each following course will build on the content taught in the previous.

The students start by developing mini-games and gradually increase the complexity. The first games they will develop will be platform game, which give the students a broad area to learn from, not only basic programming, but also math and physics.

**Course 1**
- Design game environment
- Design game play
- Design main character
- Make character sprite
- Animate character walk
- Animate character jump

**Course 2**
- Programming move graphics
- Programming arrow keys
- Programming walk, cycles
- Programming ground
- Programming enemy
- Programming enemy follow

**Course 3**
- Design platform
- Programming jumps
- Programming jumps
- Design and Program Start screen
- Design and Program win and lose screen
- Tune and test the game

**Fig. 4 Shows the 3 first courses in the first semester of the first year.**
Course 1 Design the environment for a platform game
   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. Design game play.
   4. Divide the character into layers.
   5. Animate a walk cycle.
   6. Animate jumping cycle or other related animation cycles.

Course 2 Make the characters move!
   1. Start working on programming by making simple movement of the game graphics.
   2. Connect the triggering to an external controller or arrow keys.
   3. Trigger walk cycles on the main character.
   4. Make the characters follow the ground.
   5. Make a second character (an enemy) and make it move.
   6. Make the enemy follow the main character.

Course 3 Make the jumps and enemies and tune the game play.
   1. Design platforms and move the screen with programming.
   2. Implement a simple jump for the main character.
   3. Tune the jumping and scroll the screen.
   4. Make a starting screen and menus.
   5. Make a win and lose screens.
   6. Tune the game.

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.

In course 4 and 5 the students move to a 3D environment (usually Unity3D or Unreal). Also here they will design a platform game. In addition the students have now chosen specialization, and teaching is done by several teachers in different specializations.
In the second semester the students are still taught mostly the same topics as in first semester, but on a more advanced level.

The games they will make are not limited to any special type, although some limitations are made in the environment, an example of this can be that the game shall have only one camera, so the complete game is seen only from one position. The students will typically make asteroid-like games or 3D platform games.

In second and 3rd year the model will be the same but they gradually dig deeper into the following topics:

<table>
<thead>
<tr>
<th>Design</th>
<th>Animation</th>
<th>3D</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design principles</td>
<td>Basic animation theory</td>
<td>Modelling</td>
<td>Basic programming</td>
</tr>
<tr>
<td>Basic art skills, eg.</td>
<td>Storytelling</td>
<td>Texturing</td>
<td>Algorithms and</td>
</tr>
<tr>
<td>drawing and anatomy</td>
<td>Storyboarding</td>
<td>Lightening</td>
<td>datastructures</td>
</tr>
<tr>
<td>Color theory</td>
<td>Animation techniques</td>
<td>Rigging</td>
<td>Databases</td>
</tr>
<tr>
<td>Interactivity design</td>
<td>Game engine interface</td>
<td>Game engine interface</td>
<td>Graphic programming</td>
</tr>
<tr>
<td>(GUI)</td>
<td></td>
<td></td>
<td>Simulation</td>
</tr>
<tr>
<td>User interface (UX)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game engine interface</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2 How to Gamify individual processes in each course, and breaking down the subject matters into small manageable pieces and show through a series of games, how these are linked together

The gamification aspects will be maintained by making a competitive environment where students compete locally, both individually and between groups. The competition will not only be in game development but also in how much they attend lectures and labs, how often and how much they blog, how much they participate in activities, how they study curriculum texts etc. We will try out tools like Nearpod, Socrative and Kahoot to make statistics that can be displayed in leaderboards and in for example the classroom or virtually on web pages, Facebook groups etc. There can also be a number of badges that the students can earn and put on their own web pages.

They will also compete on the Internet with other university classes and peers in game forums and online societies. These competitions will also be supported by a system of points, badges and leaderboard, with larger rewards for the students, groups or classes that win. In this way informal learning will also be brought into the classroom; the students will be rewarded for using online peers in online forums and societies to comment on their projects.
5 DISCUSSION

In this section we aim at discussing some experiences gained from running single courses with similar design to what we have described in the Case study above. This basis material for this are collected reading through a large amount of feedback material collected by the students and tutors who participated in the courses, as well as experiences collected by the writers. These notes goes back to 2006 when we started this experimentation[15].

Firstly, creating a “community of practice” seem to works well. The students puts a lot of energy into creating their games. The class, the student groups and the peers they find in forums on internet provide the students with an environment to collaborate and participate according to their skills. More skilled students proved to be very helpful, and the newcomers can learn from them. The students also quickly take on identities such as game developers, programmers, designers or animators etc. The fact that they really made a game, and not only learned how to make a game, seemed to increase their motivation, and they were rarely scared to jump into hard topics to solve a problem they really wanted to learn. It is important though that the tutors and the teachers monitor not only the class and the groups, but also the activity that goes on outside the class. Ex. when the students present their games to the outside world in Internet forums and online discussion groups. This is so these environments are not allowed to kill intrinsic motivation.

Another observation is that if the students are allowed to form their own groups, the students that are skilled and has the most confidence, often ended up in the same groups. Leaning on Lotta Darsoe [14] the groups will be most creative if we make the groups as heterogeneous as possible; we put students with different skills, gender, age etc... in the same group. This is described in more detail in [15]. We therefore strongly recommend that the teachers put the groups together and don't leave this to the students.

To be able to keep the students on the same cases, we recommend that the cases are open ended, so the more skilled students can dive deeper into the given case, rather than rush on to the next. This will keep the students on the same cases and theme, and make it easier to persuade the students that had more experience to help the students that is not so experienced.

Another experience is that learning through game development is bit ad hoc. The students needed help to structure and categorize the theory after the course is finished. We found that a good way to do this was to sum up and help them structure their knowledge after the course was done, or in the start of the next course. Doing it along the way, tended to interrupt the creative process, which is something we didn't want to do. This is also found by Darsoe in[14].

Problem based learning and self-studies, doesn’t fit all kinds of students. We remedied this by providing much help (we were many teachers and tutors), and sometimes gave traditional lectures and labs for selected groups that struggled with PBL [15].

We also learned a lot about how to create the cases to make them fun. One key issue is of course to make the cases so that the learning interval between them is right; the next case is not to difficult and not to easy [15]. This is hard and have for us required much thought and experimentation.

6 CONCLUSION

Having used game development as a tool for learning in separate courses, it has taught us a great deal about how this works. In general we have found that it can be a very motivating method, provided it is done right, as commented on above.

We have not done this to a whole education yet, but hopefully can very soon. And what place could be better to do this than in a game education?

REFERENCES


