

# THE POWER OF GAME DEVELOPMENT IN LEARNING: WHY IS GAME DEVELOPMENT GOOD LEARNING MACHINES?

**Anders Nordby**

*Hedmark University of Applied Sciences (NORWAY)*

## **Abstract**

Many researchers, for example J.P. Gee, have been pointing out for years that game playing can give good learning. When school topics is Gamified we tap into the most important and established learning theories such as, Problem Based Learning, Situated Learning, Social Identity, Experiential Learning (learning by doing and learning just in time) and more. While this alone argues for using game playing in learning, we can add to this list.

These days, children not only play games, but also modify them from early age, and this can be beneficial also in school. Letting the children do their own game development broadens the learning perspective even further and opens up new possibilities. Game development underpins creativity and let the children work with arts, design, animation and storytelling. It also develops technical skills such as programming, system thinking and requires basic understanding of mathematics and physics. And teaching the theoretical science while they are about to code it for their game is way more interesting and effective. And of course, it does so while the children have fun – the children hardly notice the learning because they are making games.

Game development also invites informal learning into the classroom. If the children present their game ideas on the Internet and in game forums, they also get new highly skilled peers, from all over the world. This will also teach them language and cooperation, digital skills and competition, tradition and culture... Internet these days reflects all aspects of the real world.

The children build on skills they already have [1] - most of them have been playing games or developed them in their spare time or even before they started in school. Which is mandatory for good learning [2].

In Hedmark University of Applied Sciences (HUAS) we have used game development in teaching for almost 10 years, both for game students and teachers, and this presentation will give some examples on how to implement this in practice.

HUAS has a complete game development tracks since 2003, and is now one of the biggest and most complete game educations in Scandinavia with 250 game students. HUAS also have various teacher educations and have the latest years also given teacher education in using and development games for learning.

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

## **1 INTRODUCTION**

Minecraft from Mojang is one of the most popular and most talked about online game or virtual world these days. According to Gamespot [3], Minecraft reached 14.3 million sales and 100 million users in February 2014. Microsoft then bought Mojang in September the same year, and later in October announced that over 60 million copies were sold on all platforms. 18 millions were sold on the PC alone, which makes it the best selling PC game ever. Minecraft is all about building and making games, and the numbers above illustrates how popular such virtual online games can be.

This interest for building and game development is not new; we have seen high sales figures in pc or console games like Sim City or Civilization where players develop cities and societies, or Spore where players design new species and develop them over time. Then came online games or virtual worlds like Secondlife, Wow, Club Penguin, Stardoll etc. Lastly there is a long history of game engines that was firstly developed as engines for specific game titles, but later was developed into engines the public could modify and use to make new games (Valve, Microsoft, Unity, Unreal etc). All of these virtual environment and games has drawn lots of young people to game development, and taught them game development skills like programming, design, arts creation and much more.

While all of these games were very popular, they never reached the fast growth we have seen with minecraft. The reason can be many, but surely the simplicity and how fast you can start playing is an important factor. Minecraft has only one building block, cubes. What we first would think is a limitation has proven to be a force; younger people find it utterly exiting to build large structures from this simple tool. Further, the technology to implement virtual reality (VR) is finally here; in minecraft you can soon use glasses that give the players a 180-degree vision, and gloves or cameras that follow your movements in the real world and translate it into movements in the game. This greatly enhances the immersion.

Having all these young kids with new technical, artistic and social skills, it would be logical to have a closer look at how we can use these to create better learning in schools. How can we design building environments in traditional school topics, so that these skills can be utilized and the kids put the same energy into learning this as they do in gaming? This article take a closer look at this, and outlines a way to implement game development to learn traditional school topics.

Before we do that though, we need a closer look at what these kids really want, and how their game playing can be utilized for learning.

## 2 THE NET GENERATION

Marc Prensky [1] argues that the “net generation” or “game generation” is different from other generations. He claims that due to their heavy use of Internet and digital media not only their way of thinking, but also their brain have changed. Prensky point out that these kids, and then the kids that builds in minecraft these days “requires multiple streams of information, prefer inductive reasoning, want frequent and quick interactions with content and have exceptional visual literacy skills – characteristic’s that are all matched in digital game based learning”. They tend to make meaning in an active way, work in teams and play roles and participate, and they combines fantasy and reality and want to have fun while doing it. Also Tapscott [5] elaborates on what these young kids demand from life, education and future employment in interviews about his book “The Net Generation”; “They want to have fun. In fact, 58% 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”.

These kids have a top down approach to learning and they just dive in and use trial and error to solve the challenges. In a learning situation they will be good at making sense from multiple streams of information, but will have little patience for things that don’t pay off right away. When they learn they like working with others, also outside their class; being connected to the world is a must. But most of all, learning must be fun.

These are the kids that the schools expect to sit still and patiently listen to lectures on theory they cant use for years, and accumulate knowledge step by step in just one topic at a time.

We have seen from minecraft that having fun, don’t mean inactivity of just meaningless play. Building things is also seen as fun, as long as they can do it in the same way they play games. But to build stuff often requires more knowledge than just playing games. They need to know more about how things work when they want to construct them instead of just using or modifying them; they need to know how physics, mathematics, programming affect and move things as well as how design, art, animation etc. is used to make the environment look good and immersive. Where they build the things is less important; they can have as much fun building in the real world as in a virtual world, provided they can do it their way.

However, there are a few things the real world can’t provide in the same way a virtual environment can. Firstly, a virtual environment can combine a mix of real world behavior with fantasy behavior. The physics engine in the virtual worlds can be altered to give a different response to behavior than the real world does. For example, gravity can be changed so that the environment behaves as it does on the moon or another planet. Or the environment can be changed to look different from the real world. Only imagination is the limitation. Gee [2] points out this this is important; these kids want to be able to mix fantasy with reality.

Secondly, creative solutions can be tried out without being afraid of being hurt or take any risks. Only the virtual environment will be damaged, what they do have no consequences in the real world.

Thirdly, you can take on new identities when you talk with peers outside the classroom, or if you are inside a virtual world you can let your avatar communicate directly with the peer's avatars. Your peers knows nothing about your real looks or abilities/disabilities, they will learn to know you from the inside. This enables you to take on any personality you want and gives you the possibility to appear as what you aspire to be, and not what you seem to be from your looks or from what your friends know about your background.

All these "extras" obviously have an effect on the learning. You also get a 'second chance' as often as you like to contribute in a team with the things you are really good at, and are not assessed only by school topics.

So far we have just addressed the kids and what an online virtual environment can give them. Lets take a step further and look at what gaming and game development can add to this and how it relates to learning.

### **3 GAMES AND LEARNING**

Games have been played at all times, but only in the last one or two decades we have been able to play them in a virtual environment. Games are about play, they often mix reality with fantasy, and they are considered fun - just what the doctor ordered for the net generation.

In his book about games and learning, Gee [2] discuss how good games relates to learning. In the following I have recaptured some relevant points from Gees discussion, and made some additional comments to relate them to this context or discussion:

- 1 Good games are systems and people often learn best when they first understand how things fit into a larger meaningful whole. That you can also simplify the models so that elements and relations are more transparent, adds to this. To start with the whole in a simplified way is the top down approach we talked about earlier; first get the overall view, then dive into the details later.
- 2 Good games are a set of well-ordered problems that build on each other, and learning events pops up when a problem has to be solved. A good game provide a step-by-step approach to learning in form of related problems that build on each other and leads to the overall goal.
- 3 Good game is pleasantly frustrating. In a learning context, pleasantly frustrating means that the learning intervals are right; not too hard and not too easy.
- 4 Good games let the player see skills in a bigger perspective; it is easy to see why they are needed. Training skills is a necessary evil in learning too, and to be able to see the training as a part of a whole, or a part of the game, will make it more meaningful to do such training.
- 5 Games let the players feel like active agents. The players can participate, be creative and create and modify in their own style.
- 6 Good games let the players work in teams and take on the identity they want. The players can take on any identity related to the game, and work as a part of a team.
- 7 Good games are a like a 'community of practice'; the overall goal seems to be to create a game, while the learning of school topics are 'invisible' and connected to this goal.
- 8 Games integrate perception and action. The players can immerse into the game, get an overall view and manipulate things at a distance.

If we build on this when we design a game development class, we should not only achieve a fun and interesting environment for learning, but also be able to design the learning so that the students that create the game learn what we want them to learn (read relevant school topics and goals).

### **4 GAME DEVELOPMENT AND LEARNING**

To make a good 'game environment' for teaching we need to define set of interesting problems founded in the curriculum, sequenced them so that the player can and want to solve them, and reward them so they continue to all of them is solved. It must be possible to work in teams, it must be possible to take on different identities, and the world/Internet must be available most of the time.

A Problem Based Learning (PBL) [6] [7] approach will get us started; we can divide the game development process into smaller well-ordered problems that can be solved sequentially to reach the

final game goal. PBL also accommodate team-based work where the problems are discussed and solved together.

Below is a suggestion to how to approach this process, based on experience from the examples given in the next chapter:

- 1 Find the learning goals you want to teach from the overall competency aims or curriculum. There should be both some overall goals and some sub goals.
- 2 Make a set of problems from the goals. If necessary, sequence them so that each problem builds on the previous one. Make sure the learning curve is not too steep. The problems should be open ended so clever students can take each task to a higher level without having to go to the next task. If possible, design the tasks so they can be solved in different ways.
- 3 Divide the class into teams or groups that work together. To make the groups as creative as possible, we suggest dividing the class into heterogeneous groups with 6-10 students in each group [8]. As tutors for these groups we suggest to use higher-level students. The groups can serve as a discussion group only if each student makes their own game, or as a development group where the members make a game together.
- 4 Open the classroom to the world and encourage the class to use the net. This integrates informal learning environments into the classroom. The students should be encouraged to use the net to find facts and theory when they need it. They should also be encouraged to present their solutions and get feedback on their work and progress.
- 5 Encourage competition between students, groups and online in forums. By letting the students or student groups compete both between themselves and as a group or class on internet you introduce more game elements and more fun
- 6 Let the students design the game and test it on another class.
- 7 Sum up the experience with the students and help them structure their new knowledge.

There are many game types that could be used for this. In the examples below on game development classes, we have used both digital games and pervasive games.

It is important that we design the class so that the students can be creative right away, and don't have to read manuals or study topics before they start. They will have to do that in time too, but not before they start!

If the games developed are not only digital, but for example also include pervasive games, a number of additional skills is needed, like acting, developing costumes, designing locations and clues in the real world etc.

Game development can also invite informal learning into the classroom. If the students present their game ideas on the Internet and in game forums, they also meet new highly skilled peers from all over the world. This can teach them language and cooperation, digital skills and competition, tradition and culture etc. since Internet these days reflects all aspects of the real world.

Ordinary teaching methods should not be abandoned entirely. Some students will need more attention than others, and their learning styles are different. If you think they or just some need an ordinary lecture, give it to them. You will get an attention and response that is different from what you are used to, because it is given just in time and the students are eager to implement their game. Teaching the theoretical science while they are about to code it for their game is way more interesting and effective.

## **5 THE THEORETICAL PERSPECTIVE**

The main goal of game development in class is not only to teach the children specific school topics, but also make them interested in learning more about these topics. Interest is primarily defined as a phenomena that emerges from an individual's interaction with the environment [9]. Learning on Dohn [10], Interest can be defined as a positively charged experience that relates to an activity [11]. In today's research there is primarily focus on two types of interest; situational and individual interest. Situational interest is about stimuli from the environment, which may or may not last over time [9]. Individual interest is more stable motivation that is associated with increased knowledge, value or positive feelings [12]. Situational interest focuses on individuals' response to external factors that may give interest in a certain context, while personal interest focuses on the individuals experience built from enduring preferences. These two types of interest are not in any way dichotomous, but they can

influence each other's development. When a student is interested in a certain learning situation, it is most likely an interaction between both types of interest. Situational interest can over time have powerful influence over the students' personal interest in a topic [12].

The students' Interest in game development will then depend on both on the social environment (the class as well as and activity in online forums or virtual worlds) and the students' personal interest in game development. We can expect that the social environment will make a difference on the personal interest in game development, but more important, also on the personal interest in the topics the student learn, and how much energy the student put into studying them.

The game development process itself serves as a "community of practice" [14] [15] that invisibly triggers interest in school topics disguised as game development, while they strive to reach a common goal; to learn to create a game. They do this by experience [16], game development is "learning by doing" [17], and they study theory only when they need to solve a game development problem, "which is learning just in time" [2]. With open-ended cases the game development process can accommodate different learning styles [18]. If the game development progress is designed so that it is "pleasantly frustrating" with the right learning intervals, Vygotsky's [19] approximation zone will not be too big, and maintain the players' lust to try again if he fails. A Problem based learning approach facilitates sequenced problem solving where each problem prepares the player for progressing to later parts of the games and finally the final game goal. PBL also facilitates team-based work where the problems are solved together as a team [6].

Teaching through game development then taps into many of the most widely used learning theories such as Problem Based Learning, Situated Learning, Experiential Learning, Identity Construction etc. which in itself should indicate that it is a valuable tool for creating Interest and good learning.

## **6 HOW CAN GAME DEVELOPMENT BE IMPLEMENTED IN SCHOOL? SOME EXAMPLES**

In Hedmark University of Applied Sciences (HUAS) we have used game development in teaching for almost 10 years, both for game students and teachers. Here are some examples on how and where it has been used | practice:

### **6.1 The Programming courses (2006-2011)**

The students attending this course had just started out their education and came from 3 different bachelors in IgA (Interactivity, games and Arts); Animation, Arts and Game Technology. Typically, around 70 students attended the course. Most students were between 18 to 25 years old, and were typical examples of the net generation described in the beginning of this article.

So we gave the students a simple task they loved - to create a game! Our main goal however, was to teach them computer programming and give them an idea of why 'hardcore science' topics were needed in a digital arts education.

We wanted to utilize a programming environment that gave visual feedback to the students right away - an environment where the programming, mathematics and physics would show up visually as behavior or environment changes in the game graphics. At the time Adobe Flash was a highly popular tool to make games and animation movies, and we found that this environment also satisfied our needs for visual feedback as well as giving us a good object oriented programming environment.

We divided the game development process into 6-10 smaller problem based cases that was given to the students as sequenced assignments over the semester. Each case were 1-2 week long, dependent on how difficult the case was. All the cases were designed open-ended so that diligent and clever students could develop them beyond the basic requirements without proceeding to the next case. We started with cases to design, draw and animate the characters, and then moved on to programming cases. When they reached the programming cases they were very motivated to make their design move, which required programming.

The students were divided into groups, with 6-10 students in each group. Each group was given a tutor from a higher-level class. The groups met twice in the case period, each time about 3 hours. First time was in the start of case period, when they needed to define the learning goals and the theory they needed to study to solve the case. After this meeting they separated, studied the theory individually,

and then completed the case in the best way they could. After that they met again and discussed the challenges the case had given them, and how they had solved them.

We choose the Maastericht model with 7 steps [7] as the teaching model. The first 5 steps in this model; clarify difficult terms, define the problem, brainstorm, analyse and formulate the learning objectives was done in the first meeting. Step 6 was self-study period, and step 7 was the post-discussion and sum-up meeting in the group.

Many opportunities to teach mathematics and physics popped up naturally during the cases. For example, when the students had finished their game characters and animated them, they needed to make them jump. They usually did this by applying a force upwards, which made the characters disappear out of top of the screen. To “fix” this they had to study theory about how forces worked in nature, and especially how the gravity force drag everything back down to “earth”. They had to dig deep into this theory; natural looking character jumps and walks required a good understanding of different forces, gravity and friction. Very few of the students found it odd to study forces and gravity in this situation. Another example was when the students needed to implement their characters’ walking cycles and moves around the screen. Then they needed understand about functions, 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 horizontal and vertical components. Students often struggle to understand this kind of theory if it is taught on an abstract level. Here, however, the physics and mathematics became comprehensible because they needed the knowledge to make their game work. In other words, the ‘fun’ and ‘motivation’ aspects of the game development made the students see the ‘hardcore science’ as elements in the overall game development process, and because of that they were willing to put much effort into understanding it.

Throughout the course the students had to present their games online in wiki pages and blogs. This enabled them to present and discuss the project in online forums, and discuss their problems and solutions with online peers. Most wiki pages and blogs were written in English. This was not a requirement, but for most students it was a natural thing and something they had done for years.

Research data was collected from notes, tutor meetings, student reports and interviews. All the data are not yet fully interpreted, that will be done through Grounded Theory and presented in new papers. For further details about this project, see article [20] [21].

## **6.2 The System thinking and Gamification courses (2012-now)**

This project is ongoing and is about teaching mathematics and physics through system thinking and gamification. In the first course the students learn systems thinking and simulation doing 9 sustainability related projects. In all these projects they implement the differential equations generated by the simulation engine using Euler in C# in Unity game engine where they also make simple visual applications to see how the differential equations work.

The Gamification course follows the system-thinking course, and in this course the students gamify a larger sustainability project using the knowledge from the previous course. This class is manly focused on game development (in unity so far, but minecraft has been discussed for the next years course).

In both courses we use a problem based learning approach as outlined earlier in the article. More details on these projects can be found in [20]

## **6.3 The Pervasive game project (2013)**

This project was done in 2013 and was conducted in an IgA class on system thinking, where they developed a pervasive game to learning sustainability. The pervasive game also included 3 digital games. We wanted the game to be tested in a real environment, so we invited elementary schools in the district to volunteer. The response was massive, 15 schools called within two days, and we selected Tangen elementary school for the project. The game was tested in one day each for the 5<sup>th</sup> grade and the 7<sup>th</sup> grade classes. After the game we let the children fill in a diary that was about what they have learned, and lastly we conducted a one-hour system thinking session for the whole class. These two events plus notes from the research team and reports written by the students served as research data. These data is not analyzed yet, but will be in a future article through grounded theory.

The pervasive game was based on competency aims in the Norwegian curriculum related to sustainability. We wanted both the game students and the children to understand how they could support a long-term ecological balance, and to teach them about ecosystems, pollution, the social

aspects of sustainability and point out how sustainable systems could be used to live wisely. We also wanted to apply system thinking, collaboration and problem-based learning (PBL) both in the game development process done by the students, and later when the children was playing the game.

By observing and analyzing motivation and learning when the game was created and later when it was played, we hoped to gain new knowledge on how good pervasive games was suited for learning applications. We also wanted to measure other outcomes such as collaboration skills, creative problem solving, and even physical activity. Finally, we wanted to examine how well the children were able to understand systems thinking, and how important systems thinking was in their understanding of the sustainability issues presented in the game. More details on this project can be found in [22].

#### **6.4 The Teacher courses in Games, Animations and Learning (2013-now)**

These courses were given to teacher students that were in their last year of their teacher education (4<sup>th</sup> year). They then had a thorough teacher and pedagogics background, but hardly any digital experience or even less experience in creating games. The challenge here then was to teach them what they needed during their game development process.

We threw them into deep water right away; the assignment was to develop a game for their own education. We used the same Problem based model as described above; the development process was divided into problem-based steps and the students were divided into working groups. Before we started we also let them develop simple web pages and Facebook groups where they could document the game development process

Since the teachers were familiar with pedagogics and learning goals, the first part of the development process went smoothly. The brainstorming process for game ideas also went good; they had a lot of ideas for the games. However, they did not know much about developing games, so at this point we needed to teach them some game design theory. We also in this step showed them how to design a storyboard and how to apply system thinking in their game design. These classes were a mix of lectures and working sessions where the groups and the supervisor worked together to help them develop their own games. In this way they managed to develop pretty interesting pervasive games.

Previously to these sessions we there had been classes in making Kodu games, and many of the teacher groups choose to develop Kodu games as a part of their pervasive games. During the whole process they blogged and documented their developments. When they had developed their games they took it to their practice or work to test them in their classes. Pretty much all of them worked well and the children had fun playing them.

I general we had no more challenges developing games with the teacher students than we had with the game students. They did great games too, and teaching them about game design for the pervasive games and even the digital games was not hard. Of course the technical level of the game development was much lower than for the game students, but the tools they used proved to create be very useful and easy, and the children loved to play them. Examples on games and implementations are given in [23]

## **7 CONCLUSIONS**

In all our projects in game development for learning during the last 10 year, game development has proven to be a valuable tool to create interest in the school topics. This is not so surprising, considered the interest we see for building in online forums and virtual worlds.

There are some challenges though, specially in tuning the cases so that they get the right learning intervals. It took us several interactions to get this right. It is also hard to synchronize the game development process with the learning process; game development processes tend to naturally sequence what the developers need to learn. Further, these types of projects are also a bit chaotic; we found that the students needed help on structuring what they learned after the game is done.

Games are also very cross disciplinary; there are very few topics game development or game playing will not touch. Therefore it can also be possible to involve a whole class in just one game development process; some do art stuff like design, animation and digital effects, while others do technical stuff like programming, system thinking, math and physics. If the game is a pervasive game, it will also need actors, administrators, helpers etc.

If you want to use game development in learning we encourage you to read [20] to [23] more closely to get some more details on what challenges we struggled with along the way.

We can assure you though, game development and testing in a class situation is fun, and as a teacher you will be amazed about how creative these young kids can be.

## REFERENCES

- [1] Prensky M. (2007), *Digital Game-Based Learning*, Paragon House, the United States
- [2] 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.
- [3] Gamespot: Minecraft passes 100 million registered users, 14.3 million sales on PC (2014, Feb 26), Retrieved from: <http://www.gamespot.com/articles/with-minecraft-microsoft-can-build-broader-horizon/1100-6422330/>
- [4] Dewey, J. (1916). *Democracy and education: An introduction to the philosophy of education*. New York: Macmillan.
- [5] Tapscott, D. (2008). *Grown up digital: How the net generation is changing your world*. New York: McGraw-Hill.
- [6] Barrows, H. S., & Tamblyn, R. (1980). *Problem-based learning: An approach to medical education*. New York: Springer.
- [7] Maastricht University. (2013). *Problem based learning preparatory website*. Retrieved from <http://www.umpblprep.nl/>
- [8] Darsø, L. (2001). *Innovation in the making*. Frederiksberg: Samfundslitteratur.
- [9] Krapp, A. (2002). Structural and dynamic aspects of interest development: Theoretical considerations from an ontogenetic perspective. *Learning and Instruction*, 12, 383 – 409.
- [10] Dohn et al.(2010), Situational Interest of High School Students Who Visit an Aquarium, DOI 10.1002/sce.20425, Published online 18 October 2010 in Wiley Online Library ([wileyonlinelibrary.com](http://wileyonlinelibrary.com)).
- [11] Rheinberg, F. (2008). Intrinsic motivation and flow. In J. Heckhausen & H. Heckhausen (Eds.), *Motivation as action* (pp. 323 – 348). New York: Cambridge University Press.
- [12] Schiefele, U. (1991). Interest, learning and motivation. *Educational Psychologist*, 26(3 – 4), 299 – 323.
- [14] Lave, J., & Wenger, E. (1991). *Situated learning. Legitimate peripheral participation*. Cambridge: Cambridge University Press.
- [15] Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge University Press.
- [16] Kolb, D. A. (1983). *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall
- [17] Dewey, J. (1916). *Democracy and education: An introduction to the philosophy of education*. New York: Macmillan.
- [18] Overview of Learning styles (2016, May 16), Retrieved from: <http://www.learning-styles-online.com/overview/>
- [19] Vygotsky, L. (1978). "Interaction between learning and development." (pp. 79-91). In *Mind in Society*. (Trans. M. Cole). Cambridge, MA: Harvard University Press.
- [20] Nordby, A. (2015) Teaching 'hardcore science' to arts and design students from the gaming generation - Reflection on how to use game development to teach programming, system thinking, mathematics and physics. *Proceedings EDULEARN15, Barcelona*.
- [21] Nordby, A., Karlsen, S. (2014). Teaching 'hardcore science' to arts and design students: Reflections on the development of a basic programming course. *InFormation - Nordic Journal of Art and Research* 2014 ;Volum 3.(2) s. 129-142



- [22] "The best school day ever' – Teaching Sustainability and System Thinking by Pervasive Game playing" (Nordby, Øygardslia, Sverdrup)
- [23] TEACHERS AS LEARNING GAME DESIGNERS: Can elementary school teachers with no background in game design really Gamify their own teaching?