Postprint

This is the accepted version of a paper presented at ECGBL 2018, the 12th European Conference on Games Based Learning.

Citation for the original published paper:

Factors to consider when using learning games for learning programming in K-9 education
In: Mélanie Sciussi (ed.), In proceedings of the 12th European Conference on Games Based Learning (pp. 447-452). Sophia Antipolis, France: Academic Conferences and Publishing International Limited
European Conference on Games Based Learning

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-34645
Factors to consider when using learning games for learning programming in K-9 education

Peter Mozelius and Jalal Nouri
Mid Sweden University and Stockholm University, Sweden
peter.mozelius@miun.se jalal@dsv.su.se

Abstract: Game-based learning is an emerging field with learning games played for various learning objectives in different educational contexts. One context where several research studies have reported on positive outcomes is for programming education. In the wide variety of games for learning to program this study reviews games and gamified concepts that have been developed for K-9 students to learn basic programming concepts. The aim of the study was to identify important factors to consider in the use of learning games for K9 programming education.

The study was designed and carried out inspired by the six step method described by Machi and McEvoy (2016): The six steps were: 1) To define a topic, 2) To search and select literature, 3) Organising the literature, 4) Reviewing the literature, 5) Discussion and critique of literature, and finally 6) Writing the review. The type of literature study could be described as an exhaustive with selective citation, with the aim of considering all the relevant sources, but only describing a selected sample.

Findings show that there is a wide variety in the use of games on basic programming concepts and also different game types. In a thematic analysis best and worst practices in recent research have been grouped into categories. Found main categories to consider in the implementation of learning games for programming in K9 setting were: Computational thinking and problem solving, Challenges, Immersion and Flow, Gender differences, Game types, Teacher involvement, and Game construction with puzzle programming.

Keywords: Game‐based learning, GBL, Learning games, Programming education, K-9 education

1. Introduction

Several recent studies have highlighted the positive effects of using games in K9 education (Yildirim, 2017) and the importance of engaging students in developing computational thinking at an early age (Ternik et al, 2017). There is an abundance of studies showing that games are motivational (Malone, 1981; Garris et al., 2002; Erhel & Jamet, 2013), and that K-9 students are eager to play digital games (Wang et al, 2008). There have also been reports on children playing too much and how to handle gaming disorders (Gentile et al, 2017).

Learning to program has been classified as a problematic subject with learning problems for students also at university level (Gomes & Mende, 2007; Watson & Li, 2014; Iqbal Malik & Coldwell-Neilson, 2017). Today when computational thinking and programming in many countries should be introduced at an earlier age there is a need for new didactic methods. As pointed out by Popović et al., (2017) it appears that programming games that require algorithmic thinking have had tendency to become a mainstream didactics for programming education in primary school. Many educators have presented the idea that starting out early with programming could lead to early inclusion and to increase the low pass rates on programming courses in higher education.

Furthermore, there seems to be a consensus that an early introduction of programming is beneficial and that learning games is a main alternative. On the other hand, a careless use of learning games and unplanned programming education might lead to an early exclusion instead of the opposite. Few research studies can be found about how the learning games best should be used and which factors that are the most important to consider in the use of learning games. The aim of the study was to identify important factors and didactic concepts to consider in the use of learning games for programming education in K-9 settings.

2 Method

This study was de was carried out as a literature study inspired by the six step method described by Machi and McEvoy (2016): The six main steps were: 1) Defining a topic, 2) Searching and selecting literature, 3) Organising the literature, 4) Reviewing the literature, 5) Discussion and critique of the literature, and 6) Writing it up. Main focus was on articles from 2014 – 2018, but older literature that has had a central role in the discussion
has been included. Step 3 to step 6 were carried out iteratively and sometimes with backward searches on interesting references in the selected articles. Backward search could be defined as "reviewing older literature cited in the articles yielded from the keyword search" (Vom Brocke et al., 2009).

The type of literature study could be described as an exhaustive with selective citation, with the aim of considering all the relevant sources from given search strings, but only describing and discussing a selected sample. The searching was conducted in Scopus, in Google scholar and in proceedings from Academic Conferences International Limited Main search string "game-based learning" AND programming AND "games" AND K9 OR "primary school". Many found articles did not discuss programming in the sense of programming education and was removed in the first scan, but with a search string modified with "programming education" relevant articles were lost. Main selection criterion was to choose articles with a discussion on how to use learning games in K-9 programming education and which factors and didactic ideas that are important for teachers to consider. Authors focus has also been to identify factors and didactic principles that could support the idea of inclusion and to support the idea of getting all students involved in the learning of programming.

Findings have been analysed in an inductive data driven way according to the thematic analysis described by Braun and Clarke (2006). Identified patterns and themes were reviewed and grouped in to categories. The categories that are the most relevant to meet the research aim are presented and discussed in the next chapter.

3. Findings and discussions

In the thematic analysis the following categories were found: 1. Focus on computational thinking and problem solving, 2. Challenges, Immersion and Flow 3. Game types, 4. Gender differences 5. Teacher involvement, and finally 6. Game construction with puzzle programming. The first five categories are both crucial factors and didactic concepts, while game construction with puzzle programming is mostly a didactic concept.

3.1 Focus on computational thinking and problem solving

In tertiary education programming education always have had a main focus on fostering professional programmers and system developers. In these contexts, it is natural to have assignments and learning games where students mainly write and optimise programming code. To transfer this model to K-9 settings would not make sense and learning objectives must be different for different age groups. A crucial factor, as well as an important didactic idea in K-9 settings is to shift the focus to problem solving activities that fosters computational thinking.

What seems like a promising approach for lower primary school is to combine playful unplugged activities with learning games on computational thinking instead of writing code (Allsopp & Ejsing-Duun, 2016; Tsarava et al., 2017). Five topics to focus on that were highlighted by Oluk and Korkmaz (2016) are creativity, algorithmic thinking, collaboration, problem solving and critical thinking. The actual programming environment often is Scratch (Oluk and Korkmaz 2016; Ternik et al, 2017), but before puzzling and coding in Scratch it appears to be a good idea to first train students on algorithmic thinking and problem solving (Hsu & Wang, 2018; Oluk and Korkmaz 2016).

An example of how very fundamental algorithmic thinking can be learnt in a game is when students practice to divide a problem into sub-problems like it is done in the game 'Aladdin and his flying carpet' (Jakoš & Verber, 2017). Another example is the game 'LightBot', not a tool to learn how to code, but to learn basic programming concepts (Giannakoulas & Xinogalos, 2018). Several studies also point out the importance of problem solving (Kanemune et al., 2017; Tsarava et al., 2017; Wong et al., 2018). It was also highlighted in a study on computational thinking by Denning (2017) that: "the goal of computational thinking is to solve problems".

3.2 Challenges, Immersion and Flow

As for all other types of learning games there is a need for challenges and fantasy (Lepper & Malone, 1987) and to design for immersion and flow (De Freitas, 2006). If the games are to poorly designed, they would more be electronic books and miss the identified potential for motivation and engagement (Lepper & Malone, 1987). A study conducted by Hamari et al. (2016) confirmed that games creating engagement have a positive
effect on learning. Furthermore, the study found that challenge affects learning both directly and indirectly via increased engagement with perceived challenge as an especially strong predictor of learning outcomes. Considering immersion in the game no significant relation was found. (Hamari et al., 2016)

In a study on online gaming platforms Combéfis et al. (2016) stressed that games for learning to program must have aesthetics to engage and stimulate intrinsic motivation. Another highlighted design feature is to involve music to enhance the gaming experience. Their recommendation for selecting games is to choose: "Games with multiple modalities, adaptive or personalized, based on real world sensory data". (Combéfis et al., 2016)

3.3 Game types

The vast majority of the programming educational games today are designed with a visual interaction in 2D settings using keyboard and/or mouse. Many games, but far from all, are played in online environments (Yang et al., 2015; Combéfis et al., 2016), and they can be single player games or multi-player games. Like all other games, learning games for programming can be divided into the three categories of combat, competition and collaboration, and that these categories can be combined like it has been done in 'World of Warcraft'. Sanchez (2017) has suggested a 'coopetitive' approach for GBL, where competition is combined with collaboration.

Contests has been presented as a part of game-based learning and that students in contests have the possibility to compare their abilities and learn from others. Another finding is that collaborative educational games can raise the participation of students in programming education. A recommendation is also to use multi-player collaborative games since they are more motivating and engaging than the single-player ones. (Combéfis et al., 2016)

A type of game that is frequently used are the so called puzzle games where LightBot (2018) might be the most well-known. The graphical user interface could give a childish impression, but the game is promoted as 'all-ages friendly' and involves advanced programming concepts as recursion and overloading. The recommendation is otherwise to start using games on variables, sequences, iteration and boolean conditions (Tsarava et al., 2017). A typical LightBot puzzle can be seen below in Figure 1.

![Figure 1. A LightBot puzzle (LightBot, 2018)](image)

Another type of game is the 'maze game' and that maze games also can be built quite easily in the Scratch programming environment (Ternik et al., 2017). Finally, GBL is a transmedial discipline and as suggested by Tsarava et al., (2017) analogue board games could be used to improve students' computational thinking. Two suggestions of board games could be Pandemic (Leacock, 2008) and RaBit EscApe (Apostolellis et al., 2014).
3.4 Gender differences

In a study by Papagisterou (2009) the learning outcomes did not vary between boys and girls when their computer science knowledge were tested. Furthermore, the investigated learning game was found to be equally motivating for boys and girls. That the motivational effect is the same has also been suggested from other studies on GBL (Koivisto & Hamari, 2014; Mekler et al, 2017). The idea of ‘one learning game fits all’ seems to have been widespread and frequently practised.

On the contrary, the studies by Osunde et al, (2015) and Osunde et al. (2018) report the opposite. Found design elements where girls and boys have different preferences were: Game graphics, Game colours, Game violence, Game characters, Player interaction and Backstory. The recommendation is to use an inclusive approach towards the design of digital educational games for young learners (Osunde et al, (2015; Osunde et al., 2018). These new findings are for several reasons worth considering and one idea with introducing programming early is to increase the percentage of girls in IT-related programmes in higher education. Even if the percentage seem to slowly increase, there still exists a clear gender difference (Mozelius, 2018).

3.5 Teacher involvement

As highlighted by Popović et al., (2017) learning games could be seen as an established standard tool in programming education in primary school. Basic programming concepts can be understood and learnt, but all game features are not self-explanatory, especially for the younger students (Gomes et al., 2018). For the older students in K-9 settings, games can be complex and that also the teachers have problems to understand the game features. To achieve good learning outcomes, it is important that teachers are familiar with the games and a recommendation is to include GBL in teacher training (Sánchez-Mena et al., 2017). Students should play the games themselves but teachers should, like it is pointed out in constructionism, be able to coach and facilitate (Papert, n.d.; Stelter, 2007). In programming education some students learn slowly, while others have a much faster progression. For both these groups the teacher’s guidance is coaching is crucial if students should keep their motivation.

A concept that looks promising is to develop instructional materials for teachers and to combine game-based learning with project-based learning and the principles of constructionism (Tsarava et al., 2017). In the same way as students need coaching and answers to their question, teachers might have the same need. Five frequent questions from unexperienced teachers that have been identified are: 1). Why are we teaching coding? 2). Should we actually be teaching coding to young children at all? 3). How should we be teaching it?; 4). How to best use tangible user interfaces? and 5). Are there gender issues to overcome? (García-Péñalvo et al., 2016). There are of course more topics than these five to discuss and there seem to be a need to sort out all these why, what and how questions. Programming in K-9 settings is a relatively new discipline and hopefully there will be a number of books on didactic principles published during the coming years.

3.6 Game construction with puzzle programming

The described recommendations of a focus on playful activities without necessarily writing code seems sound and worth using. However, to meet the learning objectives in most courses there is sooner or later a need to introduce the actual programming. This has not necessarily to be done in a traditional text based programming environment. In the K-9 context the standard choice is to use a puzzle programming environment and in a paper on game-based learning it makes sense to include the didactic idea of building games together with the children. These games do not have to be learning games and they can be just any kind of amusing games with the game construction as the learning process.

Several studies have pointed out the potential of learning fundamental programming concepts by game construction (Maloney et al., 2008; Denner et al., 2010; Wilson et al., 2013; Fokides, 2017). In primary school frequently used environments for game construction are the Scratch puzzle programming environment (Wilson et al., 2013; Ternik et al., 2017) and the Kodu Game Lab (Fokides, 2017). It has been suggested that game design fosters systematic and computational thinking (Salen, 2007; Hayes & Games, 2008). The study by Wilson et al. (2013) reports that programming concepts like selection, iteration and user interaction can be learnt from game construction.
Selection and iteration with Boolean conditions was identified by Fokides (2017) to be the most problematic concepts for students to understand. Finally, students that learn design thinking before they learn the concrete programming might learn to program more rapidly and more effectively because they have a context in which to understand coding concepts and coding issues (Claypool & Claypool, 2005; Fokides, 2017).

4. Conclusions

To first develop computational thinking before actually writing any code seems essential in K-9 settings. The use of learning games is today an established standard tool for the teaching and learning of computational thinking and programming. However, without teachers' guidance and integration of didactic principles the gaming might not necessarily lead to programming proficiency. Computational thinking and programming are both aligned to problem solving and a suitable didactic model could be constructionism with teachers as active coaches. Analysed literature has brought up a wide assortment of game types, and the recommendation is variation and to avoid the idea of 'one game fits all'. Diversity seems to support inclusion, variation is more important than repetition.

Games must be games and not just digital exercises, game mechanics, graphics and backstories are all important to stimulate flow and immersion. Challenges are important in games in the same way as they are in all forms of education. The recommendation is an inclusive design that do not exclude girls by stereotypical use of, graphics, colours and game characters. Well-designed learning games is a way to motivate students for learning computational thinking and fundamental programming techniques. Finally, the teachers' role is a crucial factor, with active teachers coaching students’ learning process most programming concepts could be learnt in a joyful way.

5. Future work

What is missing was excluded in the study is the more detailed analysis of the actual learning games. The focus has been on identifying crucial factors in the use of learning games for programming. A natural next step would be to further explore the found game types and to analyse game mechanics, motivational models and the integration of learning content.

Another relevant branch for future research would be to follow up and elaborate on the category of building learning games. To support active learning in K-9 schools the idea of not only playing games but also creating games looks interesting. Game construction also seems to be a didactic idea that is a part of most environments for puzzle programming.
References


Mozelius, P. (2018). It is getting better, a little better–female application to higher education programmes on Informatics and System science. In proceedings of the *International Conference on Gender Research* (p. 249).


