Thesis no: urn:nbn:se:bth-15324

Video Games and Software Engineers

Designing a study based on the benefits from Video Games and how they can improve Software Engineers

Srdjan Cosic Prica

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor in Software Engineering.

Contact Information:
Srdjan Cosic Prica
sadde.95@gmail.com

University Advisor:
Kennet Henningsson
kennet.henningsson@bth.se

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Context: This is a study about investigating if playing video games can improve any skills and characteristics in a software engineer. Due to lack of resources and time, this study will focus on designing a study that others may use to measure the results and if video games actually can improve software engineers.

Objectives: The main objectives are finding the benefits of playing video games and how those benefits are discovered. Meaning what types of games and for how long someone needs to play in order to be affected and show improvements. Another objective is to find out what skills are requested and required in a software engineer. Then it is time to design the study based on the information gathered.

Methods: There is a lot of literature studying involved. The method is parallel research which is when reading about the benefits of playing video games, then also reading and trying to find corresponding benefits in what is requested and required in software engineers.

Results: There are many cognitive benefits from video games that are also beneficial in software engineers. There is no recorded limit to how long a study can go on playing video games that it proves to have negative consequences. That means that the study designed from the information gathered is very customizable and there are many results that can be measured.

Conclusions: There is a very high chance that playing video games can result in better software engineers because the benefits that games provide are connected to skills requested and required by employers and other expert software engineers that have been in the business for a long time and have a high responsibilities over other teams of software engineers.

Keywords: Video Games, Software Engineers, Study, Cognitive skills.

1 Introduction

This thesis is about discovering the positive effects from playing video games and how long you need to play them in order to be affected positively. Later on it will be decided on a positive effect and see how it can be incorporated into software engineering. Finally, a study will be designed to be performed by other with enough resources to find out if these positive effects can help in software engineering. The designed study can also be performed in a different way by employers or bosses in software engineering companies to test if productivity is increased on certain individuals.

Being born in the 90's, you are automatically growing up with the era of video games and as a child, you will most likely love this futuristic entertainment in one way or another. The downside is that the legal guardians do not have an understanding of this. Seen as they grew up in far more strict living standards and had nothing to do but to go outside. This makes them think that all children should only be going outside and that video games are ruining their focus in school. It makes them behave badly from violent video games and treating others with disrespect, and the media is always trying to find a way to connect a problem with video games, because it is new.

Even though it can be entirely possible that video games cause negative effects. Studies have also shown that there are benefits from video games. In [1] researchers from the University of Rochester discovered for the first time that action video-game experience does indeed improve probabilistic inference.

This thesis will focus on the benefits and discover if they can be used to improve productivity in Software Engineering.

2 Research Questions

1. What are the positive effects from playing video games?

In order to be able to design this study, it must be known what the positive effects are from video games. Because later, the study needs to be able to describe what skills are plausible to be enhanced in software engineers and why the study is designed the way it is.

2. What types of video games contribute to different positive effects?

During a small pre-research, almost every research paper or article that discussed the benefits from video games was from the action genre or not at all mentioned. In order for the study to give a more specific set of prerequisites, the study needs to know what games contribute to what skills. If not, then the person trying to perform the study in the future, might not know what type of games to pick in order to test certain skills.

3. How can positive effects from video games be beneficial in Software Engineering?

Perhaps the most important question for this study because this is what this research is about. In my own opinion, there are not a lot of ways of improving as a software engineer without practice with coding, reading documentations and keeping up to date with frameworks and coding languages. This question helps discover if the activity of playing video games can improve software engineers in any sort of way.

4. How long does someone need to play a video game and how often, to achieve positive effects?

In order to be able to design a study that is appealing, a time estimate is required and a huge part of the time it takes to perform a study like this is the duration it takes to play a video game in order to achieve positive effects.

3 Method

The method contains a small interview study with random subjects and a big literature study that are meant to cover all the research questions listed above.

For the interview study, random subjects are picked out, interviewed in person and asked two questions. The interview is a means of discovering what software engineers are according to average people and what their opinion is about the engineer's skills.

1. What skills do you think are necessary for a software engineer to have in their work?

This question will show what the average person thinks a software engineer is. This will later be analyzed with the results from the literature study in order to verify if the skills listed by random people actually fit in with professional opinions such as employers and long time software engineers with high responsibility of other software engineers. This question is also a good setup for the next question.

2. Do you think that video games can improve the skills you just listed?

This question is an interesting follow-up to the previous one. It will give a small percentage of the people's opinion about video games and it's effect on software engineers. It will be an interesting point to reflect back on after the results from the literature study.

The literature study will follow a certain iterative approach. It will be about finding what skills are expected, needed and/or even wanted in a software engineer and what skills can be improved by video games. The results from the interviews will be mapped into this part about finding the skill in a software engineer. The iterative part is going back and forth between discovering skills in a software engineer and skills that are improved by video games. By doing this, the skills can be mapped together very early and enables a deeper discovery of specific skills. This will also result in a more precise design of the study.

The study will be designed based on the results from the literature study. It will account for the safety and health precautions as well. Certain steps will be described what is needed for a result that is to be expected based on the information from the literature study.

4 Literature Review

4.1 The Benefits Of Video Games

Video games help improve the cognitive skills within an individual. In [6] it was summarized from several research articles that action video games improve what is known as visual selective attention. Here, the definition of attention is not "paying attention to something" but instead it is the process of which aspects should receive additional neural processing and which should be filtered out. This ability to select relevant and filter out irrelevant information for a specific task improves with action video games in distinct domains such as across space, across time and in relation to objects. This means that anyone who played action video games developed a stronger **attention to detail**. However, the fact that the ability to filter out irrelevant information is enhanced, indirectly it means that anyone who plays action video games also becomes more effective in making decisions towards what is relevant to any task they are working on. It must be clarified that this assumption of **decision making** is not scientifically proven but a conclusion of the author.

There was a functional magnetic resonance imaging (fMRI) study by [2] (Bavelier, Achtman, Mani, & Föcker, 2012) where they found that the mechanisms that control attention allocation were less active during a challenging pattern-detection task in regular gamers than in non-gamers, resulting in the researchers to suggest that action game players allocate their attentional resources more efficiently and filter out irrelevant information more effectively. In other words, the gamers developed a stronger **attention to detail**.

Cognitive skill is defined as “the ability of an individual to perform the various mental activities most closely associated with **learning** and **problem solving**. Examples include verbal, spatial, psychomotor, and processing-speed ability.” [3]. Another research discovered improvements in cognitive flexibility when test subjects played forty hours of training on a real-time-strategy game. The results were significant increases in in **task switching** and **working memory**[4].

It should be made clear that enhancement in cognitive performance is not documented for all video game genres. The most robust effects come from action shooter games and not from, for example, puzzle and role-playing games[1].

4.2 The Skills Needed In A Software Engineer

There was a study about finding out what makes a good software engineer[5]. The authors wanted to gather information from a wide-range of software companies but as an initial effort to approximate their ideal, they interviewed experienced engineers at Microsoft. It was face-to-face interviews in order to get a detailed and contextualized understanding of the meaning and importance of the list of attributes they gathered. The authors needed a way to determine whose opinions about software engineers could be considered credible. So they decided to base their definition of expertise on people who had achieved some degree of software engineering experts. More specifically, they selected engineers at or above the Software Development Engineer Level 2 (SDEII) title, which meant that the engineers were confirmed experts by other engineers through the hiring or promotion processes.

The researchers got more than 60 hours of interviews on tape and 380,000 words on transcript. After their analysis, the results identified a diverse set of 53 attributes. Quoting the article’s summary of what makes great software engineers, as described by their informants: *“At a high level, our informants described great engineers as people who are passionate about their jobs and are continuously improving; who develop and maintain practical decision-making models based on theory and experience; who grow their capability to produce software that are elegant, creative, and anticipate needs; who evaluate tradeoffs at multiple levels of abstraction, from low-level technical details to big-picture strategies; and whom teammates trust and enjoy working with.”* (p 3. Ch. IV - Findings). The authors organized the attributes into 2 categories. Internal and external. Internal was the software engineer’s personality and ability to make effective decisions while external represented the impact the engineer’s had on people and product. The two categories were given two groups, respectively, to further distinguish the attributes. Internal had **Personal Characteristics** and **Decision making**, external had **Teammates** and **Software Product**. Because there were 53 attributes that were revealed from the analysis of the interviews, the authors noticed that many of the attributes were applicable to other professions and some to simply being a good person. Their objective was now to identify the attributes that experts deemed important for software engineers.

In the **Personal Characteristics** section, there was a total of 18 attributes and the informants felt that many of them were intrinsic to the engineers and were difficult, maybe impossible, to change. There were 4 attributes that the authors found interesting and these attributes are also interesting for this study. The attributes are: *Improving, Passionate, Open-minded* and *Data-Driven*. The open-minded attribute will be mentioned in the **Decision Making** section as it is a more suitable fit. **Improving** was described as not being satisfied with the current state and constantly wanting to improve themselves, their product and/or their surroundings. The informants stated that an engineer who was **passionate** in the area that they worked in, not just interested in extrinsic rewards such as money, was due developing products of high quality. **Data-Driven** was described as being able to take and evaluate measurements of their own actions and of the product. The informants stated that an engineer’s decision should be, when possible, be made using data and not intuition or arguments. One informant was quoted about data-driven engineers and they said that software engineers believe that they are driven by data, but when it is shown to them, they find a way to ignore it. These three attributes can be interpreted as something intrinsic and that it has to do with the engineers personality and passion to their career, workplace and project. By evaluating these explanations, there are two corresponding attributes. **Learning** and **memory**.

The section of **Decision Making** had 9 attributes where many of them were fairly similar but there were 3 attributes that the authors found interesting that also suits this study. The first one was having knowledge about people and the organization. The "people" here were mainly described as coworkers but there was a mention of ownership and key stakeholders, which could mean a customer that ordered a product and end users of that product. The knowledge about the coworkers were being informed of their knowledge, responsibilities and tendencies. With this knowledge, engineers could communicate with the right people to align their work, find the right people for help and, for engineers in leadership positions, take correct course of actions to address knowledge gaps. The second attribute was, as described by the authors and informants, *"Sees the Forest and the Trees"*. The engineers should be able to consider the multiple levels of abstraction including technical details, business/customer needs, industry trends and company vision. The informants believed that this ability allowed engineers to look at things from different perspectives and then make a set of globally optimal decisions. The last one is updating the mental models. Now the *Open-minded* attribute from the **Personal Characteristics** section will be discussed. A great software engineer, as described by the informants, was constantly updating their mental models by explicitly evaluating contextual changes ranging from technical details to industry trends. Fundamental shifts in common long-held understandings of software are critical for engineers to adapt their mental models to. In relation to being open-minded, this is about taking in information from around the world and not being set on the understanding that the engineer has become accustomed to. The last attribute is about handling complexity with ease, by grasping and reasoning about complex and intertwining ideas. An engineer must be able to connect things and solve problems when the engineer has no understanding of what is happening. Informants described great engineers as being able to effectively augment their natural abilities using tools and processes such as writing the problem down and studying the problem. This section had, like mentioned before, very similar attributes and it all seemed more or less the same with complex descriptions. In conclusion, **Decision Making** is a given attribute that a software engineer needs. Reading a bit more into it and understanding what these 3 different attributes actually describe, there is the ability to solve problems and interpreting the world and taking every piece of information to update one's mental model. From this we get **problem-solving** and the interpretation of the world corresponds to **learning**, **open-minded** and **memory** (understanding the changes, allow oneself to new information and updating and remembering the mental model).

Now will the external section be discussed and starting with the **Teammates** category. Going through the analysis, a conclusion was made that, almost all of the 17 attributes that the informants mentioned about teammates, the attributes were connected to a person's personality and common courtesy. A few will be mentioned as they have valuable attributes that are either new or the same in previous sections. The attributes from the informants are: *Creating shared context, Creating shared success, Creating a safe haven* and *Honest*. By reading the titles, it is understandable that one may think that these attributes intrinsic and are difficult to change in a person but they will be discussed in order to try and find corresponding explicit attributes. For creating a shared context, informants believed that it was important for effective communication that an engineer could mold another person's understanding of a situation. This was seen as essential for success by the informants since software development involves many people and getting everyone to have a shared understanding is a great attribute. This communication was also critical towards people who had little to no understanding of the situation such as partner teams, customers and management. Informants believed that software engineering is a collaborative process with people with different motivations and organizational objectives. That is why the informants believed that great engineers needed to create shared success and get everyone into making decisions aligned to one shared goal. In one example, managers often had a broad perspective of a situation while engineers often had a broader understanding of the details. This attribute would help avoid crises where there was a lack of understanding about status and tasks. A safe haven is an attribute where engineers can learn and improve from mistakes without negative consequences. Informants deemed this attribute necessary because in situations, usually associated with leaders, where the engineers would be afraid of mistakes, their improvement would slow down. Even though the informants believed the safe haven was necessary, they also believed that there should be a balance between the safe environment and the feeling of pain of mistakes. The reason was, if an engineer was hurt by something, they would quickly learn to avoid it. The last attribute is honesty and the informants believed that this was the most important attribute related to trust and providing credible information. Informants also expressed that engineers should focus on addressing the problem rather than wasting time shifting the blame. As mentioned before, these attributes can be interpreted as hard to change and related to one's personality. But the **open-minded** attribute still correlates to these with another attribute called **social competence**. These two work together to form an environment with shared context, success, safe haven and honesty.

The last section is **Software Product** which has the last 9 attributes of the total 53. 2 attributes were interesting regarding the software that engineers produced. The first one is creativity. The software that the engineers developed was described, by the informants, as creative with novel solutions based on the understanding of constraints of the context, existing solutions and the limitations of those solutions. The informants divided the attribute into two parts. Understanding of the constraints and requirements of a problem and not being creative by knowing when to apply existing solutions. The second attribute is attentive to details. The informants believed that it was necessary for an engineer to be structured when writing code, understanding the performance, knowing how many bugs it has and how it handles exceptions. For attributes concerning creativity, this paper will relate that to **problem-solving** because creativity also needs solutions and the informants made references that creativity needed understanding of constraints and knowledge of when to use, and not use, existing solutions. The last one needs no explanation and is simply **attention to detail**.

4.3 The Time It Takes To Improve Attributes From Video Games

There are two aspects of the time it takes to see an improvement of attributes (research question 4). The first aspect is playing a game without breaks and pauses. The second one is the duration of all the combined sessions of playing video games and the periods of not playing a video game at all.

In a video with Daphne Bavelier[7] she explains a research she was part of, how they measured subjects who played video games and what the results were. The subjects were faced a few tests and results were measured. Then Daphne and her associate researchers forced the subjects to play **10 hours** of video games. These hours were spread out over several days in the span of 2 weeks and the subjects played about **40 minutes** and then took a break or completely stopped and waited for the next day. When they came back to the lab they did the same tests again with small modifications so they were not completely identical to previous tests and it showed improvement. Daphne asks a rhetorical question of what the results would be like if the subjects played 20 hours or 40 hours or even more. Daphne was part of another study[6] where they did similar tests but with two groups and more focus on fast paced action shooter games. This is explained in the article *"One group receives extensive training on an action video game (depending on the study, this may be **ten, twenty, or even fifty hours** of experience spread out over the course of many weeks). Another group receives the same amount of total video game experience; however, group members play a nonaction game instead. "*

So both groups received the same amount of time playing a videogame but the groups were measured against each other i.e. they did not test if the group who played non action video games improved themselves from before. It is also mentioned that successful training studies have all used distributed practice that required the participants to play between **30 and 75 minutes a day** and no more than **5 days a week** spread over several weeks. According to [8], the amount of weeks it would take for this training to complete would be about

$$\frac{number\ of\ hours}{4\ \mathit{to}\ 5}$$

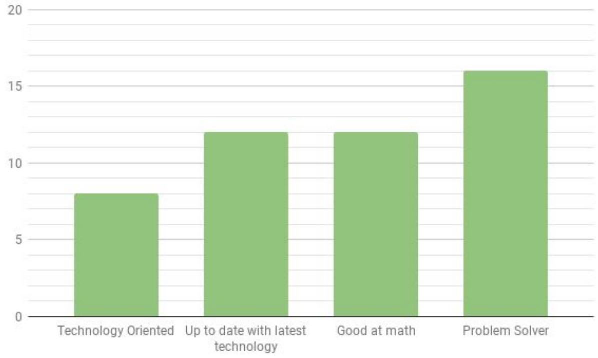
Meaning 10 hours would result in 2 weeks and 50 hours would result in about 10 to 12 weeks. In [9] the author interviews several scientists and researchers about their studies about screen time. In one part of the book, the author interviews Mai-Lis Hellénius. A lifestyle professor at Karolinska Institutet and chief physician at the heart clinic at Karolinska University hospital in Solna. The author brings up a discussion about how long in front of a screen is too much and how much is beneficial. She talks about a study that was done on children between the ages of 5 and 18. The results from that study revealed that anyone who spent over 2 hours a days in front of a screen, had an increased chance of developing depression symptoms and with over 4 hours the chance was 50% higher and with 5 hours the increase chance was 90%. But it also showed that spending one hour a day was stimulating. Mai-Lis explained that it was also possible that the causes of depression was substantial. Meaning that the subjects chose to sit in front of a screen because they were depressed and given the physical outcomes of sitting still for longer periods of time, it was very plausible that there was more to the depression symptoms other than the screen time itself. However, she pointed out that **1 hour** showed that it was stimulating for the child.

5 Analysis and Results

5.1 Results from the interviews

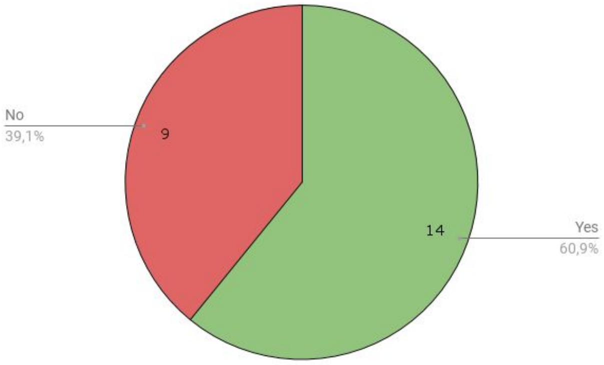
There were a total of 23 interviews and the results from both questions have been added together. The people that were interviewed were able to give multiple answers to the questions and that is why in the following results and summaries, it is summed up to more than a total of 23.

From the first question *"What skills do you think are necessary for a software engineer to have in their work?"*, there were many similar answers. The answers have been grouped together to give a summary of all the results and this is it.

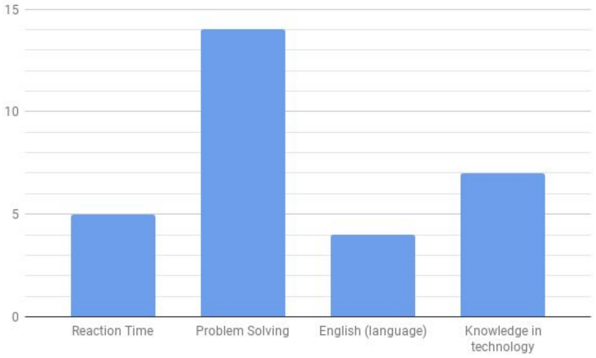


The results are somewhat abstract and similar. But it does give a small sample of what a software engineer looks like in the eyes of common people. While "Technology Oriented" and "Up to date with latest technology" seem like they mean the same thing. Technology oriented is defined as knowing how to handle different types of technologies while up to date is having the latest knowledge about certain technologies and frameworks. From the literature study, both what is wanted in a software engineer as researched at a large Microsoft company on subjects that have been in the software engineering business for a long time and are in charge of a lot of responsibilities, and what are the benefits from video games, there is a category that appears in both of those subjects and the results from this question. That category is **Problem Solving**. This seems to be a very common thing that comes to mind when thinking about software engineers and will be a core skill in the design of the study.

From the interviews, when answering question number two *"Do you think that video games* *can improve the skills you just listed?"* there were several answers that included skills which they thought were improved and helpful for software engineers. First, the results based on whether or not the interviewee thought video games could improve software engineers.



The majority of the interviews yielded a result that showed most of the common people said yes but not by a significant margin. The interviewee's then continued the answer by giving examples on why this would be or would not be possible. By those who voted "No" the majority answered simply that they did not see a connection other than it is electronic and that does not improve software engineers at all. The one's voted "Yes" gave again a variety of answers that have been grouped together in order to make it more easy to get a summary.



Some interviewees thought reaction time would be improved but did not develop further on how this would help software engineers. **Reaction time** was a skill that appeared in the discovery of benefits from video games but due to the method of discovering skills parallelly with what is required in a software engineer, the skill was not included in the literature review and is therefore not of interest from the results from the interviews. Like in the results from question 1, **Problem solving** makes an appearance and this corresponds directly to the skills that are improved by video games. This is interesting since the common people believe both that problem solving is necessary for software engineers and that the skill can be improved by video games and is exactly what this study is about. The reasoning can be that the people who were interviewed have seen proof of benefits from video games from older research studies and the rise of employed software engineers in today's society with complex technology, they can make an educated guess on what is necessary on what is necessary in a software engineer and a simple guess on what video games can improve since they also are being more complex by every year. While the skill **English (language)** is a very good skill to have in communication and from the literature study it showed that it was required that software engineers were good team workers, it does not exactly mean that this skill is a necessity for a software engineer and is therefore ruled out. **Knowledge in technology** is about being critic about the games that are being played and in turn think how oneself can improve on such a game. That leads to research and testing by oneself. In the literature study it is discussed that the software engineer is **Data-Driven** but it might not be the same as what the people in the interviews thought. While it is a perfectly correct skill for a software engineer to have, it is also very abstract and will therefore not be accounted for.

6 Conclusion

In order to design a study with the purpose of revealing if video games can improve skills in a software engineer, the research of finding the skills desired in a software engineer and the research of what genre of games, how long one needs to play and what the benefits are from playing video games, need to be mapped together.

6.1 The Skills And Attributes Of Focus

The results from the study that researched the desired skills in software engineers at Microsoft, showed some skills that were the same from the results of the studies of the benefits from video games. These skills were the following:

- Problem Solving
- Memory
- Learning
- Attention to Detail

6.2 The Video Game Genre

Many, close to all, of the studies used fast paced action video games and some tested with resource management strategy games but did not show any similar promising results as action games.. So the choice of genre is fast paced action video games purely because of the great amount of results and the lack of results from other game genres.

6.3 The Duration Of Playing Video Games

While there are results and discussions on how long a video game should be played without a break. There is almost no data about how long the period of playing video games, including breaks and other activities, reaches it's max and starts showing negative effects. That is why this study found a simple measurement for future studies to use when estimating how long a study will take. If the amount of hours to be played is known, then it should be divided by 4. If the number of hours exceeds 10, then it should be divided by 5 based on the reference in the 5.3 section.

Now, to the duration of playing a video game without breaks. Many of the references showed a similar method or opinion in this aspect. The amount of time is between 30 minutes and 1 hour a day.

7 The Study

7.1 Prerequisites

1. It is required that the test subjects are provided with great comfort when playing the video games. The ideal approach to provide the best comfort for each test subject is to allow them to choose it themselves. I.e standing at a desk or sitting in a couch.
2. Choosing the game genre. This study found research about fast paced action video games, so choosing the same could yield similar results. It is however encouraged to perform a study with another video game genre, i.e strategy and resource management games.
3. Find an evaluation test. Make the test subjects take a test before playing the video game. This test will be used again, with slight modifications, at the end in order to measure the results. Some examples could be to distinguish a defect in code, write a small program with certain functions, pair coding. A suggested example would be to pick a test from https://www.testdome.com/tests. There are many tests for different languages and technologies and even a difficulty factor. The tests are approximately 20 minutes.
4. Decide the time budget of the study. More importantly, decide how many days are in your budget and how many hours you have available for the test subjects to play the game. For instance, if you decide that you want the test subjects to play a total of 10 hours of video games, then you will need approximately 2 weeks and the same goes if you only have a 2 week budget.

7.2 Execution

1. The duration of playing a game should be max 1 hour and preferably, at least 30 minutes. Then the test subjects should do something else, before playing a game again. Ideally they will have to wait for a day before going through another session of playing video games.
2. (Optional) It is encouraged to have the test subjects play together, especially if they are co-workers as this will enhance the social competence and chemistry amongst the test subjects in their working environment.

7.3 Evaluation

1. Have the test subjects perform the same tests as before they started the game with some modifications and measure the results. The options of what to measured are highlighted in this study in chapter 5.

8 Future Work

For future work, not considering researchers that will perform the study designed based on this particular study, there are some areas that can be further researched.

1. What are the benefits of playing video games genres other than fast paced action? Many of the studies referenced here almost only used fast paced action video games in their study and it would be interesting if other genres unveil other benefits or even the same.
2. What daily activities, that do not involve software engineering related work, can increase the skills in a software engineer?

References

[1] C.S. Green, A. Pouget, and D. Bavelier Improved probabilistic inference as a general learning mechanism with action video games

[2] Bavelier, D., Achtman, R. L., Mani, M., & Föcker, J. (2012). Neural bases of selective attention in action video game players. *Vision Research*, 61, 132–143

[3] http://www.ncme.org/ncme/NCME/Resource_Center/Glossary/NCME/Resource_Center/Glossary1.aspx?hkey=4bb87415-44dc-4088-9ed9-e8515326a061#anchorC

[4] Glass, Brian D., W. Todd Maddox, and Bradley C. Love. 2013. Real-Time Strategy Game Training: Emergence of a Cognitive Flexibility Trait.

[5] Paul Luo Li, Andrew J. Ko, Jiamin Zhu. “What makes a good Software Engineer?” https://faculty.washington.edu/ajko/papers/Li2015GreatEngineers.pdf

[6] A. Eichenbaum, D. Bavelier, C.S. Green. Play That Can Do Serious Good. American Journey of Play, V.7 #1.

[7] https://www.youtube.com/watch?v=FktsFcooIG8&

[8] Stafford, Tom, and Michael Dewar. 2014. “Tracing the Trajectory of Skill Learning with a Very Large Sample of Online Game Players.” *Psychological Science* 25:511–18.

[9] Björn Solfors (2016). Skärmtid. Stockholm: *HarperLife.*