Viability of a computer game level creation tool – To facilitate design prototyping

Christopher Johnson
Carl-Oscar Bäckdahl

Systems Sciences, bachelor's level
2019

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Foreword

We wish to extend our thanks to those around us that have been of great help, not limited to the colleagues at Neava Technologies AB who have provided both valuable input as well as good coffee breaks. This extends to our friends and finally our families.

We also wish to extend an additional thank you to Spotify for providing us with the soundtrack to which we have performed our work.
Abstract

The purpose of this study is to investigate how prototyping of video game levels can be made more accessible by developing a user-friendly game level creation system, evaluating it and producing a number of design principles to be followed when designing such systems. The research question for this study is thus: How can video game level creation be made more accessible? There were several key areas at the center of this study such as Usability, GUI Design, Prototyping, and Game Development. This study utilized an adaption of the Design Science Research Methodology, and evaluation of the resulting artifact was done via user testing and surveys. Survey responses revealed that 95% of the participants successfully completed the main task of the trial, to create a completable video game level within 25 minutes. In addition, all participants answered a 3 (neutral) to 5 (best) on ranked survey questions relating to the usability of the artifact. This study resulted in design principles which were derived from the survey responses via a qualitative decoding process. The conclusion drawn is that computer game level creation can be made more accessible via the development of tools abiding by the design principles proposed by this study.
# Table of Contents

1. Introduction ................................................................................................................................. 1  
   1.1. Background of Research ............................................................................................................ 1  
   1.2. Problem .................................................................................................................................... 2  
   1.3. Delimitations ............................................................................................................................. 3  
   1.4. Research Environment ............................................................................................................. 3  

2. Theory ........................................................................................................................................... 4  
   2.1. Game Development .................................................................................................................... 4  
   2.2. Game Design ............................................................................................................................. 5  
   2.3. Interface Design ........................................................................................................................ 5  
   2.4. Game Usability .......................................................................................................................... 5  
   2.5. Prototyping ................................................................................................................................ 6  

3. Methodology ................................................................................................................................. 8  
   3.1. Research Approach ................................................................................................................... 8  
   3.2. Purpose of the Artifact ............................................................................................................. 8  
   3.3. Objectives of the Artifact ......................................................................................................... 9  
   3.4. Evaluation and Data Collection Method .................................................................................... 9  
   3.5. Ethical Considerations and Quality Assurance .......................................................................... 10  
   3.6. Data Analysis Methods ........................................................................................................... 11  
   3.7. Method Reflection .................................................................................................................... 12  

4. Result ........................................................................................................................................... 14  
   4.1. Development and Implementation of the Artifact ...................................................................... 14  
      4.1.1. Development Methodology ................................................................................................. 14  
      4.1.2. Game Mechanics ................................................................................................................. 14  
      4.1.3. Editor Mechanics ................................................................................................................ 15  
      4.1.4. GUI Development .............................................................................................................. 15  
      4.1.5. GUI Sketches ..................................................................................................................... 16
4.1.6. GUI Design Decisions ..........................................................16
4.1.7. Internal Demonstrations - Pilot Studies.................................20
4.2. Artifact ..............................................................................21
4.3. Evaluation of Artifact .........................................................32
5. Analysis ..................................................................................36
5.1. Quantitative Data Analysis ....................................................36
5.2. Qualitative Data Analysis ......................................................38
5.3. Analysis Overview ..............................................................39
6. Conclusion .............................................................................40
7. Discussion ...............................................................................42
7.1. Relevance of Findings ........................................................42
7.2. Recommendations for further research ..................................43
8. Glossary ..................................................................................44
9. References ..............................................................................46
10. Appendix ...............................................................................48
10.1. GUI Sketches .................................................................48
10.2. Survey .............................................................................50
10.3. Raw Data Table ...............................................................51
1. Introduction

In this section the purpose of the research will be introduced. It covers the background of this research as well as its purpose. The problem area is also accounted for along with the research question for this thesis. Delimitations are lifted, and the context of research is presented.

1.1. Background of Research

When Pong was released in 1972 it was hard to imagine the market that would eventually evolve from its release and how this one product would build the foundation for one of the largest industries that exist today (Luenendonk, 2015). While the market is still booming with no signs of stopping the technical expertise that has been required in order to enter the Game Market has always seemed insurmountable to the inexperienced, and to that end there exists a need to simplify the development process to simultaneously open the field to more people as well as make it easier to work for those that are already in it.

There exists in all faucets of software development a fascination and need for abstraction. Abstraction in software development refers to simplifying something to the point where its usability is maximized, and through personal experiences within the field of game level creation it was possible to identify areas which seemed to be lacking in said aspect. The development process for levels in video games has always been linked to either games that are already created or has had a dependency on the user’s ability to code. However, not everyone is able to code, and a developer could be lacking the skills needed to create depending on who it is that is creating the level. With these problems in mind an artifact has been developed to explore the possibility of creating a piece of software that is meant to speed up prototyping a game level for 2D platformers.

A survey performed during the Global Game Jam in 2013 by Alexander Zook and Mark O. Riedl (2013) shows that programming is the biggest problem area according to 108 people out of 419 people when developing a video game. 24 people had also mentioned that time was a limiting factor, and this is what shows that there exists a need for a quick and easy way to use prototyping tools. This is supported in turn by Jonathan Blow (2004) who also claims that “the hardest part of making a game has always been the engineering.”
1.2. Problem

A game level is at its very base the playfield upon which the user creates the enjoyment they desire, and it is therefore fair to argue that without a level there is nothing to play. Be it a console window for a text adventure, a foldable piece of cardboard for tabletop games such as Monopoly, or fully three-dimensional environments present in most video games today, the level is arguably the essence of a game. To an extent it is also worth arguing that the field of level design has been the root from which other games have grown. A notable example of a game having grown out of a level is Defense of the Ancients which is more commonly referred to as DotA. DotA started out as a player modification and game level for the game Warcraft III by Blizzard Entertainment which is significant because Warcraft III, released in 2002, was a so-called Real Time Strategy game that provided a map editor for users to build custom maps. Some map creators took the tool to the next level and DotA was the result. The first DotA map was created in the Warcraft 3 map editor but had its very own game mechanics and was for all intents and purposes, its own game.

Within Game Development, level design has often relied upon paper prototypes and draft boards in order to try out a conceptual draft for the flow of the game (Manker & Arvola, 2011). However, in its current state a big part of level creation includes an understanding of scripting which is not something everyone has competence in or even an interest for which can hinder the user’s desire and ability to create games. In an attempt to alleviate this, the artifact created for this thesis is meant to look into alternative means in which prototyping for video games could be made reality without programming needs. The question posed in this thesis is therefore as follows:

How can video game level creation be made more accessible?

With all that said, the purpose of this research is to investigate means to make game level creation, or game level prototyping more accessible, or usable to those who are not game programmers. How might development of hi-fi game prototypes be abstracted to the point that they can be created with the same ease as making a low-fi sketch prototype? This is what this thesis aims to investigate and provide an answer to.
1.3. Delimitations

As three-dimensional games add a layer of complexity that would fall outside of the scope of what this artifact tries to accomplish, an executive decision was made in which the artifact was limited to two-dimensional platform games for the sake of simplicity and for the sake of having the ability to create a fully-fledged proof of concept.

There were some features cut as they were considered less important for a working proof of concept such as the ability to Undo. This was done since it was important to keep a scope in a rapid development process of the artifact. For a project of the size such as the one within this report it was particularly important not to let feature creep occur as it would undermine the entire workflow. A clear and concise picture of what was to be included in the artifact allowed for a workflow that was as smooth as possible.

The demographic of the data collection was limited to people in or close to Luleå as it would not be viable to study people remotely due to the chosen data collection method. The goal was to acquire twenty people of varying backgrounds in order to have a broad spectrum of people with varying experience when using software.

1.4. Research Environment

This work was done in partnership with Neava Technologies, a software company based in Luleå, Sweden. Resources were provided by them throughout the work. Such resources included guidance, learning materials and an office from where to work and collaborate. Learning materials included online courses for learning game development with Unity3D as this was our first experience using it.
2. Theory

This section will outline the theory areas that form the foundation for this thesis. The various areas include Game Development, Game Design, Interface Design, Game Usability and Prototyping.

2.1. Game Development

Zook et. al. (2013) investigated the motivation as to why individuals participated in the 2013 Global Game Jam as well as what problems they encountered during development. With the help of a survey distributed to willing participants they gathered insight from 419 out of 16,705 of the game jam participants who answered nine open-form questions summarized into four topics by the authors. These topics were: what had inspired them to make a game, what they had set as design goals for this jam, what their development process looked like, and how the game changed during development.

Not all questions got answered by every participant in the survey, but the results showed that a majority of people found that the greatest struggles in development was programming and that they had to cut features during development due to time constraints or because they weren’t received well during testing. Additionally, when asked about prototyping it was implied that a majority of participants in the game jam did not prototype at all out of the belief that they didn’t have the time for it or considered their end product to be a prototype in itself.

This shows the need for tools that could allow developers to prototype development at a higher pace when the time constraints and troublesome programming experiences is put into the limited time scope of a Game Jam.
2.2. Game Design

In order to establish the basic concept and idea of what a video game is this thesis relies upon the work of Djaouti, Alvarez, Jessel & Methel (2008) who propose that an easy way to identify what makes a game is to look at how it is structured and built with the idea of having three different kinds of rules built by what they refer to as metabricks. Metabricks can most easily be compared to LEGO blocks which are stapled next to each other to form a line in which a game’s rules are created. These blocks, or rather rules, are the play rules, game rules, and world rules which dictate what the game actually does. In broad terms the different rulesets can be summarized as the following:

- **Play bricks** — rules which dictate what happens based on player input.
- **Game bricks** — rules which observes the player and judge their performance
- **World bricks** — rules related to running the simulation of the game; engineering rules.

A provided example of the usage of the metabricks concept was the Driver ruleset which combined the Play brick called ‘Move’ with a Game Brick called ‘Avoid’ which when stapled together creates the very basic concept of what a driver in a racing game does: they move in order to avoid other cars and reach first place (Djaouti et al., 2008).

2.3. Interface Design

Wilbert O. Galitz (2007) provided guidelines and frameworks with which Graphical User Interfaces (GUIs) was constructed to allow the greatest degree of usability. The concepts were related in part to how certain elements are to be used and how they are not to be used as well as weighing the consequences of using one item over another. The most important GUI elements considered in this study as well as how they are utilized in the artifact are outlined in section 4.1.6.

Another author (Arvola, 2016) provided a firm foundation for using these examples in a book which focused heavily on what interaction design as well as what a good User Experience Design is. In short Arvola provided a more detailed explanation of why certain elements trigger a response in a user whereas the book from Galitz (2007) provided a balanced view between theory and practical application on the subject.

2.4. Game Usability

A literature review found on Game Usability (K. Isbister & N. Schaffer, 2008) provided a magnitude of articles which reflected upon the field which was still relatively unexplored at the time of its publishing according to Isbister. The review claims that in order to differentiate regular usability
testing and game usability they bring forth the concept of regular usability testing as well as playtesting in which playtesting focuses on explicitly tapping a player’s opinion and attitude towards a game. This is done in order to reveal what the subjective experience of playing a game is like which is invaluable for game designers.

Playtesting is, according to the review, “one way for us to get reliable attitudinal data from people. That is, if we ran the same Playtest (with different groups of participants) on the exact same game multiple times, our findings would be relatively consistent across tests.” (Isbister et. al., 2008, p. 42)

The review (Isbister et. al., 2008) also contains a piece by Henriette C.M. Hoonhout who propagates the use of the Think Aloud protocols to get an idea of what the user is thinking, and why they are doing what they are doing in active use of a designed artifact. Said section also warns that think aloud has a chance of killing the experience or changing it since participants might simply claim they like the end product and focus on what they like more than what could have been better. For this reason, it was not deemed an ideal testing method of the artifact in this thesis.

### 2.5. Prototyping

The importance of prototyping in Game Design is not as straightforward in Game Design as it would be in areas such as Web Design or Software Design because the changes made in a game is often very difficult to predict (Manker & Arvola, 2011). It is therefore important to do frequent iterations of prototypes which can be a costly effort as sketches can take somewhere between minutes to hours to make.

Paper prototyping is not one of the most used prototyping methods, but it is widely regarded as a valid source of insight regardless. Prototyping is also used as a springboard that tested if an idea’s initial execution could work or not (Manker & Arvola, 2011).

The purpose of a prototype is to give shape to a concept’s intended purpose, feeling, look, or implementation (Arvola, 2016). A prototype is important in creating a product that matches the user’s expectation.

Esposito (2018) mentions the concept of design fidelity and how it relates to prototyping. There are two main types of prototypes, being Low-fidelity and High-fidelity prototypes (low-fi and hi-fi for short). Common types of low-fi prototypes include sketches and wireframes from which testable artifacts are then developed. Paper prototypes were mentioned above and fall under this category.
There is no barrier to entry for the creation of low-fi prototypes as one only requires a pen and paper and are of a low-tech nature (Esposito, 2018). On the other side of the spectrum, hi-fi prototypes are interactive and are of a high-tech nature. Esposito (2018) provides the pros of using hi-fi prototypes, those being that they are more familiar as they look like real software, help in the testing process and are generally more presentable to stakeholders.

Low-fidelity prototyping allows you a speed and flexibility that high-fidelity prototyping typically does not allow to the same extent. This is because the nature of low-fi, low-tech prototypes allows one to erase and iterate until the desired result is reached.
3. Methodology

This section will outline the entire procedure of this research in detail. It begins with a description of the research design for this thesis, such as its purpose and objective. The method of evaluating the artifact is described as well as other considerations as they relate to ethicality and integrity.

3.1. Research Approach

In order to properly evaluate the extent to which 2D video game level prototyping can be made more accessible, an IT artifact was developed utilizing the Design Science Research Methodology as a basis. In order to develop a successful artifact one can utilize the discipline that Design Science provides. The Design Science process as proposed by Peffers, Tuutnanen, Rothenberger, & Chatterjee (2007) provides six steps. These are as follows: Identification and motivation of problem, Definition of objectives for a solution, Design and Development, Demonstration, Evaluation and Communication. An interpretation of these steps was used to facilitate the conceptualization, implementation and evaluation of the artifact.

3.2. Purpose of the Artifact

To ensure a successful artifact the six steps proposed by Peffers and contributing authors were followed. The first step is to motivate the problem. That is to say, to identify a gap and to attribute a purpose to the work. The purpose of this work was to investigate the viability of allowing persons to effectively and easily prototype and communicate video game level design ideas without engineering. In other words, to find a way to strip away any technical barriers to game level design implementation. While there are methods of prototyping video game levels without technical skills, such as sketches, to have the levels be playable is an important aspect to this work and its purpose.
3.3. Objectives of the Artifact

The next step is to define the objectives for the solution. From day one, the artifact was conceptualized with a limited scope in mind. A list of features to ensure a minimum viable product was conceived. This list was drawn from prior experience developing and playing computer games in the 2D platformer genre. Such features include a player controller (movement and input system), coin collecting (acts as a score), death counter, goal, obstacles, and decorative props. The above list of features describes the experience of the player in “Play Mode” whereby the user plays a level that has been created.

A list of Editing features was also decided upon. It was important that the user had the freedom to express their ideas with the building blocks provided. An object placement system was required in order to allow the building of levels. This includes all placeable objects in the game, such as tiles, decorations, level logic (spawner, goal, checkpoints), effects, and enemies. In addition, a configurable brush size should exist as well as a deletion tool to delete parts of the level to correct mistakes. Finally, a saving and loading system should exist so that work can persist between sessions. The combination of all these features results in an implemented artifact with the goal of allowing users to prototype playable level design ideas quickly and easily.

3.4. Evaluation and Data Collection Method

Due to the nature of this research design, it was required that each participant be physically accessible. The pool of participants included fellow students of The Luleå University of Technology and colleagues at Neava Technologies. Two children of said colleagues even participated. The ages of the participants ranged from ten to forty-five years old. The participants required a minimum viable experience of using personal computers. This was defined as possessing the ability to use a mouse, keyboard, and of having intuition of universal GUI elements such as the X button to close a window, or how common controls such as buttons checkboxes, input boxes etc behave. It was not a challenge to find people with these skills.

To evaluate the artifact and reach a conclusion, a structure similar to the pilot study was used. Twenty participants participated in total. Each trial was around thirty to forty-five minutes in length and began with an introduction of ourselves and the research. After that a short demonstration of the artifact was done to show the participant exactly what it is used for. In this demonstration, the editor features were briefly introduced. A premade level was then loaded to demonstrate potential, that is to say, give the participant an idea of what the editor is capable of producing. This level was then played by the participant if they desired. This gave them the chance to get a feel for the control of the player character. The participant was then given the task of creating a
level within twenty-five minutes that satisfied a number of requirements. These requirements were as follows:

- **Must contained required objects**
  - Player Spawner
  - Goal
  - Blocks
- **Must be able to be completed**
  - No impossible jumps
  - Not too difficult

The participants would be left alone to build their level. This was to ensure there was a minimal influence due to the anxiety of being watched. After the participants completed their level or the twenty-five minutes had passed, they would carry out a survey of questions to give insight into their experience. The survey in question is shown in the appendix in section 10.2 of this article.

This study was of both a qualitative and quantitative nature. Ranked questions are of interest because they give a good insight at a glance of the overall impressions made on the participants. Qualitative data was collected in the form of free questions in the survey. The purpose of the qualitative data was to support the insights given from the quantitative data by providing in-depth reasoning as to why the participant answered the way they did. The qualitative data is the most valuable type of data in this study, while the quantitative side gives a solid high-level overview the general impact the software had on the participants.

### 3.5. Ethical Considerations and Quality Assurance

When dealing with participants there are several considerations to keep in mind. How to ensure that there is a high degree of transparency? How to ensure the participants’ data is handled in a secure way? To ensure transparency each trail began with a presentation of the research and its purpose. This was to let each participant know how their input would contribute and how the data would be used. Each participant was anonymous and would be referred to by an ID number internally. An ID was used to correlate a survey response to the level created by that participant.

A policy of informed consent was utilized in which the participant is fully aware of what they are participating in and for what purpose. In addition, each participant had their privacy kept intact.

During the data collection process, steps were taken to ensure there was minimal effect imposed on the participants that may have affected the final result. During the trails it was important that
the participant did not feel pressured or scrutinized. The participant was therefore never directly observed as it was thought this would make them behave less natural than otherwise. The participant carried out the survey after the trial on their own time, or they could carry out the survey at once if they so desired. Pilot studies revealed that participants may feel pressured if they did the survey on site.

The survey was carefully designed as to not include leading questions and to be as objective as possible. So rather than have a question formed as “How difficult was the software to use?”, it is instead worded as “Please rate the difficulty of the software.”. This applies to all the questions in the survey, most importantly the open response questions where the participant cannot simply choose an alternative that suits them. This ensured that a subconscious bias as a result of leading questions was eliminated and that the results had a higher degree of integrity. The goal was to complete this research with a high degree of reliability.

3.6. Data Analysis Methods

In order to extract meaning from the survey responses two distinctive methods had to be utilized. As a blend of quantitative and qualitative data was collected, each type of data required its own method of analysis.

The purpose of the quantitative data was to provide a general overview of the impressions the artifact had on the participants. The quantitative data was compiled into bar graphs in order to show the frequency of responses to the multiple choice ranked questions. A statistical analysis was then done to show the percentages of how participants responded to certain questions. This allowed insight into the overall impressions the artifact had on the participants. The overall goal of this analysis was to identify patterns in the participants’ impressions of the artifact, its usability, difficulty and their own level creation.

Qualitative data was collected in order to get more depth of reasoning from participants regarding the artifact. Qualitative decoding was performed from a thematic perspective in which keywords were determined before the work began. The keywords were Usability, Prototyping, Game Development, new ideas, and ideas we had already put into our delimitations. By utilizing this method, it was possible to more easily associate phrases to relevant topics for the thesis. This method was chosen for its simplicity. By being able to get a good idea of topics and themes at a glance by highlighting text in different colors it was possible to expedite the data analysis process.
The quantitative data provided the “What” meanwhile the Qualitative data provided the “Why”. Based on the results of the quantitative dataset the relative success of the artifact can be inferred, but with the qualitative the reasons why participants responded the way they did are highlighted.

3.7. Method Reflection

The work in this thesis relied upon a framework constructed by Ken Peffers (2007). As a major focus of this thesis was the development and evaluation of an artifact, a framework for such work was required. The DSRM as provided by Peffers et. al. seemed to be the best suited to this purpose as it provided clear steps for the creation of successful artifacts. DSRM was also introduced by supervisors at LTU as a viable framework when the research process was first getting off the ground.

There is an additional methodology that could have been used called Action Research, however AR suits better when it comes to the researcher solving practical problems in an organizational context (Peffers et. al. 2007) which was not the purpose of this particular study. That is not to say AR cannot be used to design artifacts because that simply isn’t true. Peffers et al (2007) mentions that the research design chosen mostly affects the structure by which the research takes place as well as the way the result is presented. Investigating both, however, Design Science provided the structure deemed most applicable to this thesis.

There exist other sources on Design Science, one of which was used by Peffers in his own methodology that proved to be an interesting read. The source was a similar framework set up by another set of authors (Hevner, March, Park & Ram, 2004).

The Hevner framework was an older framework with seven guidelines. Simplified for this report, the guidelines were as follows:

1. Design an artifact.
2. Motivate your problem relevance.
3. Evaluate your design.
4. Be clear about how you contribute to the research.
5. Be clear about what rigorous methods used when you performed research.
6. The design process is a search process.
7. Communicate your research.
When put into comparison with Peffers guidelines a comparison between the two can be made in their similarities, the difference being that Peffers can be answered better as questions that needs to have an answer:

1. Is this a solution to a problem? (Problem relevance)
2. Is there an objective with the solution?
3. How was the solution designed and developed? (Be clear about rigorous methods)
4. How was the solution demonstrated to others?
5. How did testers evaluate the solution? (Evaluate your design.)
6. How was the result communicated to others? (Comm
7. How did the research contribute to solving the problem? (Be clear about how you contribute)

It was certainly possible to claim that we held ourselves to the guidelines listed by Hevner et. al. because of their similarities. However, due to Peffers having a clearer and more concise framework it was chosen as our preferred method of evaluation.

When it came to the selection of the demographic it was decided to use people physically accessible within the Luleå area as it was required of them to physically be present for the trial. It is possible to have participants take the trails remotely, but this would have involved extra steps to ensure a reliable result. For example, there would have to be a way to enforce the time limit remotely which would be tricky to coordinate. In terms of ages of the participants, it is possible this had an effect on the result. Say that the average ages of participants were 50 instead of 25 and there was a larger frequency of older, non-technical participants. It is likely the results would be quite different in that case.

Another aspect in which the design of the trials may have affected the result was the first step - Demonstration of the artifact. Due to the fact each participant received a demonstration in order to learn the purpose of the artifact, they did not go in blind. Keep in mind, the purpose of the demonstration was not to teach the user how to use the software, it was more to communicate what the software is for so that they might have some direction. If each participant went in completely blind it is possible we might have seen lower scores on aspects such as difficulty or usability.
4. Result

In this section the development stages of the artifact will be described. The artifact itself will then be presented with the help of screenshots. The quantitative results of the evaluation will also be presented in the form of charts.

4.1. Development and Implementation of the Artifact

This section covers the development stage of the artifact. The development methodology utilized to produce the artifact will be described, decisions made in the design phase, and how by what means that artifact was iterated on during development will also be explained.

4.1.1. Development Methodology

The Kanban development methodology was used to help keep track of tasks and allow an agile approach to development. This allowed clear definition of what features needed to be added, which features were being worked on, and which features were finished. A clear structure meant that work was more efficient as there was never any doubt of what to do next. The Unity 3D game engine was used to implement the artifact. Unity 3D was chosen as it is the game engine of choice at Neava Technologies, thus support was available. In addition, Unity 3D is free to use. Since Unity 3D is one of the more popular game engines (Instabug, 2017) for consumers there is also lots of support and well written documentation online. Work was divided between the authors in terms of expertise, that being front-end (GUI) and back-end (Game Systems).

4.1.2. Game Mechanics

Implementation of the artifact involved several key decisions. One of the most important ones was the type of 2D platformer that would be realized. There are four major ways in which platform games can be implemented. Tile-Based (pure), Tile-Based (smooth), Bitmask, and Vectorial. (Monteiro, 2012) The Smooth Tile-Based approach was used in this case. The Smooth Tile-Based means that collision between the player and the world are determined via a tilemap but the player is not locked to the grid, as is the case with a Pure Tile-Based approach. (Monteiro, 2012)

It was determined that this approach would be the best due to its relative ease to implement and because a fluid, non-restricted character control system (akin to Super Mario Bros) was of desire. This implementation also allows the use of slopes and one-way platforms which would allow additional flexibility in designing game levels.
When the metabricks concept by Djaouti et. al (2008) is applied to the play mode of the artifact in this thesis it can be argued that the play rules ‘Move’, and game rules ‘Avoid’ and ‘Score’ are used to create the play mode of the artifact which has the player move in order to avoid traps, pick up coins, and reach the goal. It was with these rules in mind that the basis of both graphical assets and coded scripts was able to take its shape. The player needed to be able to jump and run in order to avoid the editor’s traps, and they also needed to be able to count their coins before reaching the goal.

4.1.3. Editor Mechanics

Implementing the level editor involved many decisions also. Such decisions were important for keeping the scope of the artifact in check, in accordance with the delimitations defined. There are several features that did not make the cut due to being out of scope for the purpose of this project. Examples are world backgrounds, moving platforms, more enemy types, ability to configure prop scale and rotation, configuration of level parameters (such as gravity and movement speed) and more. It was decided to have a relatively static level creation system in this proof concept, but these are areas that could be developed and implemented at a later time. Sacrificing these features meant relinquishing user control and narrowing flexibility, however it was determined that while good to have, these features were not key, but rather complements to the system as a whole.

With the theory presented by Djaouti et. al. (2012) once more in mind, the metabricks that can be used in order to describe the editor mode would be “create” and “select” which is added to the game bricks “destroy” and “block”. The concept here is simple, the user of the software selects and creates different blocks in order to destroy or block the player. This is to say place traps and terrain which ultimately is the intended use of the editor. Much like with the play mode it was important to structure the work after the metablocks as it played part in understanding which core features and elements was needed. An editor needed to be able to find their blocks and select them fast in order to create a level that blocked the player in some way in order to create a challenge.

4.1.4. GUI Development

The development period involved several iterations. Based on a couple of pilot studies the graphical user interface of the artifact was updated. The final version of the GUI was first prototyped via paper and shown to supervisors for initial feedback. Upon being greenlit, work was done to implement it.
Design of the final version of the interface involved following the guidelines provided by Wilbert O. Galitz (2007). He provides a framework in which to approach user interface design, such as proper typography, element organization, toolbar design, palettes and so on.

The final design of the interface was implemented after the backend of the system was in place. Therefore, the functionality of the interface was well defined. There was one exception. The final version of the interface demanded layer functionality, that is to say, the ability to edit and delete objects depending on the layer they belong to. So, if the user has selected the foreground block layer, the palette will display foreground tiles and the deletion tool will only delete foreground tiles and ignore all other objects. In other words, the deletion tool will only delete objects that belong to the layer the user currently has selected.

4.1.5. GUI SKETCHES

The earliest version of the GUI revealed itself to be quite unusable during early testing. Peers at Neava were often confused and did not understand the meaning of the icons. Back then the UI was heavily icon based and had no tooltips. It was decided to take a few steps back and redesign the GUI from scratch. The reason the GUI was so unusable was due to negligence to hold to existing and proven GUI design guidelines. The first version was improvised and not thoroughly thought through. This was why a redesign was necessary.

The first step of this redesign was to sketch the new design on paper and iterate before implementing. Arvola (2016) mentions that sketches are done after the functionality, content and structure have been defined. With these aspects completed work began on conceptualizing a new GUI design with paper and pencil. To demonstrate, scans of two of the iterations are provided in the Appendix in section 10.1. These sketches serve as a means illustrate the various widgets (GUI Components) and their intended functionality, following the guidelines for the platform the software is designed for, i.e. Windows, MacOS, etc. (Arvola, 2016). The sketches were created following the guidelines provided by Galitz (2007).

The importance of the GUI sketches is not to diminish, as it allowed rapid iteration. It made it easier to receive input and feedback from peers that resulted in further iteration and therefore an improved design. Physical sketches meant that time could be saved in the implementation phase, as it was not necessary to reimplement every iteration.

4.1.6. GUI DESIGN DECISIONS

As usability is an important aspect of the interface, steps were taken to address some of the most common usability problems as listed by Mandel (1994). Ambiguous menus and icons are
combated by using a tooltip system that tells the user exactly what the button’s function is. In some cases, the content of the tooltip would change depending on various factors. For example, the “Play” button is grayed out until a player spawner has been placed somewhere in the level. If the user hovers their mouse over the button in this state, the tooltip lets the user know they need to place a player spawner in order to play. If a player spawner does indeed exist, the tooltip content changes and lets the user know that the application state will change into “Play Mode” or “Edit Mode” depending on which mode the user is currently in.

Unclear step sequences were addressed by having a list of layers, where each layer acts as the next “step” to be taken to build a level. I.e. beginning with foreground blocks, moving on to background blocks, adding decorations, etc. The user does not have to follow this sequence if they do not wish, but the layers were designed with this in mind.

A large step in designing a usable interface is to organize elements clearly and meaningfully. Galitz (2007) shows us that there are elements of two types, noise and signals. Noise refers to elements that serve no purpose or have no meaning to the user, whereas signals are elements showing meaningful information. By ensuring the GUI contains no noise and elements are organized clearly by sticking to a consistent design pattern, logical order, and alignment, a sense of familiarity can be established which results in a higher degree of intuition on the user’s part.

Nielsen (2006) did an eye tracking study in which he found that users generally scan web interfaces in an F-shaped pattern, and sometimes in an L or E pattern. The user typically begins scanning the top left side of the screen, where in the case of the artifact is where the file controls are located (new, save, save as, load). This would be the top bar of the F. The second movement people usually make is a short horizontal movement from the left side, or the F’s shorter second bar. The user then scans down the left side of the screen where the toolbox is located. This is represented by the F’s stem.

What we learn from this study is that the eye tends to move in a certain pattern across the screen, so it is prudent to place important elements along that path. With this logic, when the user has completed their F-scan, and learned of the existence of the toolbox, file controls and mode button, they can move onto the right panel and be introduced to layers, palette and currently selected object.

Galitz (2008) provides guidelines for ensuring components are emphasized as they should be for maximum usability. The most important elements should be emphasized. Important elements in
this case are the mode button, selected layer, currently selected object and icons. There are various methods that were utilized to emphasize these elements which are presented below.

The icon of the mode button was colored a light green to grab the user’s attention. Colors provide the user with an idea of the relative importance of on-screen elements (Koyani, Bailey & Nall, 2004). Lighter colors are the first to grab the user’s attention. It was determined that the mode button was one of the most important elements due to its intention to be heavily used. The icons on the toolbar were also colored white to contrast with the dark background color of the panel.

The layer the user currently has selected uses a similar approach of using color to emphasize it. It does, however darken rather than lighten. The objective was to give the user the impression that the button is “pressed” and will only return to an un-pressed state when another layer is selected.

The approach used to give emphasis to the currently selected item was Size and Isolation. The object the user currently has selected appears as an image on the bottom of the right panel. This image is relatively large in order to catch the user’s attention. It appears with a short description of the object and is isolated via a header text that separates it from the rest of the right panel.

In the case of dialog pop-ups such as save as, exit, load, etc., these were placed in the center of the screen. By doing this it instantly gets the attention of the user.

The editor uses a few keyboard shortcuts to help users more familiar with standard keyboard accelerators to take advantage of a faster workflow. These include CTRL+S to save and CTRL+O to open. Escape asks the user to exit the program and warns that unsaved edits will be lost. Page up/down allow changing the brush size, B is for the brush tool, and DELETE is for the eraser tool. The tooltip also conveys the shortcuts when the user hovers over the respective buttons in the interface.

A key element in the GUI of the artifact is the toolbar. A toolbar is a list of buttons grouped together. These buttons are wrappers for commands in the software. Toolbars have a plethora of advantages. (Galitz 2008) mentions these as being continuously visible, having ease and speed of use and consuming a small amount of space. There are various disadvantages also, such as unlabeled buttons having to be learned, which is addressed with a tooltip system. Due to the software having a number of critical commands it was deemed best to have a toolbar supported by tooltips.

The final key component of the GUI was the palette. This is the scrollable panel in which all game objects are displayed and from which the user may select one to paint onto the level canvas.
Galitz (2008) defines a palette as being a control consisting of a series of graphical alternatives. In this case, it consists of a series of placeable game objects. The contents of the palette update depending on the layer that is currently active to reflect the object types associated with that layer. Because the user is intended to only have one game object selected, a palette was perfect because palettes are used for mutually exclusive choices. Game Objects in the palette were represented as the graphic they would appear as in-game. Upon selecting an object from the palette, a section beneath the palette displaying the currently selected object would update with the newly selected object’s information.

The final aspect of the UI were tooltips. Tooltips were used quite liberally in the final design. Galitz (2008) describes tooltips as being a small pop-up window containing text that describes the functionality of the UI element from where the pointer is located. Tooltips in the artifact are used on the toolbar buttons as well on the layer buttons to describe the purpose and functionality of the various layers. These still need to be discovered by the user of course, but always serve as a reminder if the user were to forget how something works.
4.1.7. Internal Demonstrations - Pilot Studies

Development of the artifact was carried out as an iterative process, so it was important to receive feedback and update as required. Throughout development stage, several demonstrations were held for supervisors in order to receive feedback. The structure of these meetings was typically as such:

- Author briefly demonstrates editor features, or demonstrates features added since last meeting.
- One of the authors load a pre-created level to demonstrate potential.
- Supervisor or author play the level.
- Supervisor uses the editor to build a simple level and experiment with features.
- Supervisor reflects and provides feedback to authors.

A similar structure to this was used during the evaluation, but with a test subject in place of the supervisor. These meetings were not as formal as the actual evaluation. The purpose of these pilot studies was to receive feedback from supervisors and colleagues at Neava Technologies in order to iterate towards an artifact that is as usable as possible for the evaluation.
4.2. Artifact

From this work an artifact was produced. The artifact takes the form of a computer game level editor with the purpose of making it easier to prototype playable level designs. The functionality of the artifact will be demonstrated using figures to highlight different features. Each of the layers will be described in terms of its functionality and purpose. In this demonstration a very simple, short game level will be constructed.

**Fig 4.2.1 - Editor Interface**

The Editor interface, shown in fig 4.2.1, is what the user is greeted with upon opening the software. It is an empty canvas in which the user may begin constructing their level. There are various GUI elements that provide functionality. To the top left there are various buttons for creating a new level, saving, saving as, and loading an existing level. To the left is a toolbar with a control for the brush size, activating the brush tool, activating the eraser tool, panning, and toggling the grid on and off. The right panel contains the layers in which the user interacts with to build their level. The palette is a context sensitive box in which the contents change based on the layer the user is working from. The bottom right shows the user which game object they currently have selected. The top middle of the screen contains a button for entering play mode. The mouse also controls the zoom level using the scroll wheel and the user can also pan by clicking and dragging with the right mouse button.
From this point the user can begin painting blocks of various types to build an environment. Fig 4.2.2 demonstrates the Foreground Blocks layer. Foreground blocks act as the skeleton of the level and collide with the player character. Foreground blocks are the most fundamental pieces of the game level. Editing of the foreground level is done via a brush tool with a configurable size to allow easy painting onto the level canvas. In prototyping this can be seen as the initial stage related to conceptualization. The sketch artist would draw a skeleton of the level in order to test a certain flow before giving it any real detail.
Fig 4.2.3 - Background Layer

By using the background blocks in the Background Blocks layer, the user is able to add a sense of depth to their level, see fig 4.2.3. These blocks do not collide with the player and so are useful for adding the illusion of being in an interior or that there is more to the world than the foreground. The purpose of these blocks is to give the user a higher degree of control in the design of their level. As work with prototypes go, the background layer is part of the conceptualization phase in terms of providing a bridge between something that is bare boned and something that is highly detailed. That is then to say it is meant to allow a better visual sketch of the end product.
To add even more detail and sense of life to the level, the user can add decorative props. These are demonstrated in fig 4.2.4. Decorations serve as another way to customize a game level in order to make it feel more unique. These do not interact with the player in any way and are purely visual in their function. The decorative props are what turns a prototype from a sketch to a highly detailed hi-fi prototype. This is where the bare boned wireframes of foreground and background layers become something that the eye and mind can comprehend as a complete product.
Fig 4.2.5 - Particle Effect Systems

Effects give the user the ability to breathe even more life into their level. In fig 4.2.5, rain effects have been placed under the clouds to give the impression that the clouds themselves are raining. A magical swirl effect has also been placed on top of the cauldron on the stone platform to make it livelier. This is once again part of the phase where details are etched out. Particles catch the eye of the player and they might get curious about what a certain item in a level is or what it does.
**Fig 4.2.6 - Obstacle Layer**

The Obstacles layer shown in fig 4.2.6 provides the user with various ways to hinder the player’s progress in the level. This allows the user to change the degree of difficulty in the level. In this example, an auto turret has been placed on the birch tree left of the pit, and spikes have been added to the pit. If the player falls into the pit they will die and have to restart the level or respawn at the last touched checkpoint. The auto turret will aim and fire upon the player as long as the player is within a certain radius of it. Much like the foreground and background this is part of the conceptualization. The obstacles are what make the level a challenge and creates the flow mentioned in the background.
Fig 4.2.7 - Level Logic Layer

Fig 4.2.7 demonstrates the Level Logic Layer. The Level Logic contains game objects that did not fall under any of the previous categories. These are essentially miscellaneous objects, usually containing some special functionality. This layer gives access to the player spawner, which defined the location in which the player will be initialized in play mode. In this layer is also where collectible coins are located, the level’s goal, checkpoints, invisible colliders (transparent blue block), and a killzone (red block, kills the player on touch). In this example a player spawner has been placed on the left side of the level. Coins have been placed around the level. The goal has been placed on the door of the house and a perimeter of invisible colliders has been placed to ensure the player character cannot fall off the level.

Now the level has all the components required to be considered complete. It is now playable. Note that background blocks, decorations, effects, and obstacles are not required for a playable level. It is possible to build a level just out of foreground blocks, a player spawner and a goal. Control is within the user’s hands. The single real requirement to enter play mode is that a player spawner exists somewhere within the level.
Fig 4.2.8 - Play Mode

The level is now playable. In fig 4.2.8 the user is in play mode, has collected three out of five coins and is jumping to the stone platform to collect the final two coins. The platform system being used is the Smooth Tile Based Method. This means the player character is not locked to the grid and has free movement between tiles. This results in gameplay similar to Super Mario Bros. There are other benefits as well, such as slopes (as demonstrated).
**Fig 4.2.9 - The Goal**

Upon touching the goal, the level is won. Fig 4.2.9 shows how this looks like. The popup shows the player how many coins were collected and how many times they died. From there the player can either replay the level or return to editor mode to make changes. To follow are some examples of some larger levels to demonstrate the full abilities of the editor.
Fig 4.2.10 - Spikeland

An example of a larger level with a higher level of detail is shown in Fig 4.2.10. This level was used to demonstrate the editor potential during the trails. The participant was given the option of playing this level before using the editor.

Fig 4.2.11 - Evil Castle

Fig 4.2.11 demonstrates a very large level with multiple pathways a player can take to reach the goal. It also demonstrates different kinds of environments, i.e. dungeon areas as well as caves.
Fig 4.2.12 - Village

Fig 4.2.12 demonstrates a flatter, urban environment. Houses are constructed using foreground and background blocks. Decorative houses are also used to flesh out the illusion of being in a village. There is also a forest environment using different types of trees to increase immersion.

These figures demonstrate the extent to which users can express themselves with the tools provided by the artifact to produce unique levels easily.
4.3. Evaluation of Artifact

As stated previously, a survey response was used to evaluate the artifact. The survey served as a source for both qualitative and quantitative data. A total of twenty people participated in this study. Graphs summarizing the quantitative survey data will be presented. The quantitative data provides a general statistical overview of the thoughts and impressions participants had on the artifact. This ensures that conclusions can start to be drawn based on the following responses while the qualitative responses support and add additional depth to the overall result.

The raw quantitative data is shown in the Appendix in section 10.3. It is from this data that the following charts were created.

**Fig 4.3.2 - Spread of participant ages**

As one can see on Fig 4.3.2, participant ages spanned from ten to forty-five years of age. The demographic of this work was not limited in terms of age, the idea being that it would be interesting to investigate how people of varying ages would handle the challenge of creating a level with the artifact. The average age of the participants is twenty-five.
Fig 4.3.3 - Percentage of participants were completed the task

Fig 4.3.3 shows the ratio of participants that were able to create a playable, completable level within twenty-five minutes, verses those who failed to do so. The graph shows that 95% of participants completed the task, while 5% did not. This question was important to establish a baseline. For those who were unable to complete the task, they were prompted to give the reason as to why.

The next four graphs will display the results of the ranked survey questions.

Fig 4.3.4 - Participants’ satisfaction with their level

Fig 4.3.4 shows the satisfaction participants had with the level they created during their trial. Six participants were neutral regarding their level. Eleven participants thought they created a good level, and three thought their level were great. This question was asked because the ultimate
The purpose of this study is to investigate how to increase accessibility to game level creation. The extent to which the participant is satisfied with their level could indicate the degree to which the user was able to express themselves with the tools provided by the artifact unimpeded by its systems.

**Fig 4.3.5 - Usability of artifact**

The responses given in Fig 4.3.5 give valuable insight into how the participants perceived the usability of the artifact. It is an indicator of the extent participants felt hindered by the interface and systems that make up the artifact when carrying out the task. One participant was neutral on this regard while fourteen rated the usability as good. Five participants thought the usability was great.
**Fig 4.3.6 - Impressions of the artifact**

Fig 4.3.6 shows the overall impression the artifact had on the participants. This question serves as an indicator of the level of quality of the artifact. Once again, one participant had a neutral overall impression. Thirteen had a good impression and six had a great impression.

![Difficulty of Software](image)

**Fig 4.3.7 - Difficulty level of the artifact**

The final graph shown in Fig 4.3.7 presents the difficulty of the artifact as expressed by participants. This question gives direct insight into the extent to which participants experienced difficulty with the artifact. Similar to Fig 4.3.5, it is an indicator that the artifact was usable and functioned in a manner expected by the participants and was intuitive. One participant had a neutral view on the difficulty. Nine participants thought the artifact was easy to use and half of the participants thought the software was very easy to use.

The free-form questions from the survey were collected into a document which were coded according to theme. With all but one participant providing answers in the survey, these were sorted in tables by question and then taken to be analyzed and decoded.
5. Analysis

In this section the data from the previous section will be interpreted and meaning will be extracted from it. It is here the main evaluation of the artifact will be presented and discussed.

5.1. Quantitative Data Analysis

Upon first analysis of the quantitative data results, it is clear that 95\% of all participants were able to create a level that satisfied the requirements of the trial which was to create a completable level. The remaining 5\% had not completed this task but for reasons that are outlined in section 5.2. On the subsequent ranked survey questions, not a single participant answered anything less than a 3, or neutral. To better illustrate the survey question responses, the bar graphs from section 4.3 will be processed into pie charts.

![Satisfaction with Level Creation](image)

Fig 5.1.1 - Participant satisfaction with level creation

From fig 5.1.1 it is noted that the majority (70\%) of participants had a high degree of satisfaction with their level creation. 30\% of participants were neutral. This result indicates the participants are proud of the level they created, even given the time restraints of the trial. This also indicates that the software allowed them to express themselves creatively unimpeded by the software.
Fig 5.1.2 - Usability of the software

Fig 5.1.2. shows the overall usability experienced by participants. This study had a major focus on usability. To ensure that game level prototyping can be made as accessible as possible it was important that the artifact was as usable as possible. For this reason, the question regarding the artifact’s usability was a very important indicator to the success of the artifact. The responses show 70% of participants rating it as good. 25% thought the usability was great and only 5% thought it was neither good nor bad.

Fig 5.1.3 - Participant impression of the software
The participants’ overall impressions of the artifact, shown in fig 5.1.3 are an indicator of the overall quality of the artifact as a whole. Both in terms of presentation and features. 65% of participants had a good overall impression which is interpreted as the artifact having a high quality but could be more in terms of features and refining. 30% had a great impression and only 5% had a neutral impression.

**Fig 5.1.4 - Difficulty of the software**

Fig 5.1.4 illustrates the participants’ view on the overall difficulty of the software. In other words, how difficult or easy was the software to use. This question relates highly to the question on usability. If the software is difficult, it will result in a lower score of usability. This study finds that 95% of participants thought the software was easy or very easy to use, with 5% thought it was neither easy or difficult. Exactly half of the participants thought the software was very easy to use. This is an indicator that the software was easy to get started with to build a level without much hindrance.

**5.2. Qualitative Data Analysis**

Once decoded, the data from playtesting (evaluation) and the attitudinal data indicates that the artifact is usable. As stated by Isbister et. al. (2008) on the topic of playtesting and Game Usability the responses in the survey were consistent throughout the tests with a response that was generally interpreted to have been favorable from the way answers were formulated.

Eleven out of twenty participants mentioned the ease of use in the free-form questions provided with the most common issue being a lack of features that were already part of the delimitation of the artifact. The ability to Undo although absent was a feature that many associated as a crucial part of software. Four out of twenty participants outright mentioned it in their desired features and
to that end it is fair to assume that the usability of the software was affected for it and most likely gave a worse experience.

In the end some of the participants’ experienced a sense of lesser usability which seemed to stem from a lack of features more than a lack of actual usability. Two participants felt restricted by the game assets provided by the artifact and would have wanted the ability to add their own. One participant brings up the speed at which a prototype can be created, citing “Easy to use and prototype levels even without programming knowledge. For example, you can build a little bit, test it, and then continue creating. It’s iterative and prototyping in its best form.” The same participant also mentions their fondness for the concept of utilizing the system for getting people interested in game development, saying “I like the program as a means to increase interest for learning more about game development.”

In terms of what was missing, participants were very creative with their suggestions for new features although some of them had already been considered during the creation of the artifact such as non-player characters and ladders. Undo was a recurring desired feature that participants felt to be missing.

For the participant that did not manage to complete the test, the question why they did not manage to complete the test was posed, to which the answer was that they had simply forgotten the time limit which resulted in them not playtesting the level enough to ensure it was able to be completed. To quote the response directly when asked why they did not complete the task, they said: “Not understanding how much time I had left, I neglected play testing my level, resulting in an impossible challenge.”

5.3. Analysis Overview

Upon cross-referencing the results and analysis data from the quantitative and qualitative questions and taking the feedback of the responses that rated at a 3 or lower it was possible to draw a connection between what it was that made software usable. Participants 4, 7, 10 and 18 provided detailed responses that were especially insightful. What made a usable game level editor turned out to be the size of hitboxes, the amount of features the software had, whether it was able to be seen by colorblind people and automation. With these aspects in mind, a conclusion can be drawn.
6. Conclusion

The usability of an artifact is partially determined by its features. The most prominent comment in those who felt that the usability was lower than the average participant brought up a lack of features as their primary concern for continued use.

Automation was found to be a very important feature in the development of video game level editors. This point came from participants using slopes and spikes. In the artifact, slopes and spikes appear as several individual objects, each one representing a different orientation. For example, there are four spikes, each facing a different direction. Users expected the editor to automatically align objects depending on their surroundings, i.e. to have a single spike object that will adapt its orientation depending on the level geometry around it. Same with slopes. It was expressed that the lack of said feature made them feel as if there was a lack of polish.

Hitboxes (Collision Detection) were also proven to be a very important feature. Improper hitboxes on game objects caused frustration amongst several participants. Game objects that had a trigger boundary that went beyond the expected area of effect made the user experience frustration as they behaved in a way different than what the user expected. An example several participants pointed to were the spikes, which appeared to have a detection area that was larger than expected and thus would kill the player unfairly.

Lack of support for the colorblind was another issue. One participant made the point that the software was less usable as they had trouble differentiating between where certain buttons were due to the contrast being too similar to that of another object in the interface.

Results indicated that keyboard shortcuts were not implemented to result in the highest degree of usability. Despite being a feature that was unused by most, those who used it felt that they had to move around the keyboard too much to be viable and wished for more ergonomic shortcut placements. This hurt their perception of the usability as it made using the keyboard more cumbersome than anything else.
<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Undo and Redo</td>
<td>Certain staple features are more desired, if not essential, than others. The lack of ability to Undo and Redo certain actions while creating will result in a product that is perceived to be less usable than other alternatives.</td>
</tr>
<tr>
<td>2. Property sized colliders</td>
<td>If handled improperly, collision detections will cause a user to experience a lack of usability. Things to look out for:</td>
</tr>
<tr>
<td></td>
<td>● Improper eraser behavior</td>
</tr>
<tr>
<td></td>
<td>● Improper collisions with obstacles</td>
</tr>
<tr>
<td>3. Support for colorblindness</td>
<td>Always keep the contrasts and colors of your interface in mind. Depending on what kind of colorblindness the user have they will not be able to differentiate certain features if handled incorrectly. Alternatively, include a colorblind mode option.</td>
</tr>
<tr>
<td>4. Automate the boring stuff</td>
<td>The ability to automate parts of the creation is important in making an editor usable. The system should have objects change shape or direction according to its position in the environment.</td>
</tr>
<tr>
<td>5. Ergonomic Hotkeys</td>
<td>The purpose of hotkeys is to allow users to quickly access functionality in the system without having to manipulate the mouse. These keys should be logically and ergonomically placed to be viable.</td>
</tr>
<tr>
<td>6. Instant Feedback</td>
<td>The system should allow the user to view and test their changes instantly to allow them to easily iterate on their designs.</td>
</tr>
<tr>
<td>7. User Generated Assets</td>
<td>The system should contain support for users to create their own game objects. Users expect the ability to create their own assets, such as tiles, decorative props, custom behaviors, etc.</td>
</tr>
<tr>
<td>8. Quality of life features</td>
<td>Features that are crucial to ensuring higher degrees of usability are so-called Quality of Life features. Such features could include:</td>
</tr>
<tr>
<td></td>
<td>● Selection &amp; Copy/Paste</td>
</tr>
<tr>
<td></td>
<td>● Hiding of UI elements</td>
</tr>
<tr>
<td></td>
<td>● Configurable Inputs</td>
</tr>
</tbody>
</table>

**Fig 6.1. Table of Design Principles**

Fig 6.1 presents the final list of design principles created. These are meant to act as a set of guidelines to abide by to ensure a higher degree of success for the development of video game level creation tools.
7. Discussion

In this section the findings of this research will be discussed. This includes an answer to the research question for this thesis. Ideas for further research are also provided.

7.1. Relevance of Findings

Our study has resulted in eight key design principles that we have determined to be essential when designing video game level editors. These design principles have been conceptualized from the perspective of usability and accessibility. When we began work on the artifact, we had a general idea of some key design principles that originated from various literature sources, such as guidelines in GUI development as provided by Galitz (2007), Game Mechanic design as provided by Djaouti et al (2008) and several others.

The evaluation of the final IT artifact produced by this study revealed many more aspects that we either missed or had decided to skip in the delimitation. It turns out features such as Undo/Redo are essential for the highest degree of usability and it was a poor decision to not implement it in the artifact. However, the fact that most participants missed that feature indicates that it is an important principle to abide by despite not existing in our artifact. It simply means it was a more important feature to prioritize than we initially thought. Such is the case with other unfinished, or perhaps non-existent features in the artifact such as proper key bindings, better collision detection and user generated assets. It goes to show that a list of design principles can be compiled based on what users missed in addition to what they praised about the artifact. We argue that the features that were missed and were repeated throughout participant responses were the most important to consider when compiling the final list of design principles.

The purpose of this research was to find ways to make video game level creation more accessible. By creating a prototype for a level creation tool, evaluating it and putting together a list of design principles for development of such tools, we do believe we have answered the question we sought to answer: How can video game level creation be made more accessible? One method of making video game level creation more accessible is to abide by the design principles proposed by this study when designing and implementing a video game level creation system.

The artifact that was produced as a result of this study proved to be enjoyed by participants who often expressed disappointment that the time was already up. Many expressed their enjoyment in the open response survey responses. Many also felt that the software allowed them to express their creativity in a new and interesting way.
An important part of this study’s purpose was to investigate how game levels can be prototyped and be playable in an accessible way. There are two types of prototypes, Low-fidelity and high-fidelity prototypes (Esposito, 2018). Paper sketches of a game level would fall under low-fidelity prototypes. These levels aren’t playable but serve as a basis from which to develop artifacts. Esposito (2018) also mentions that high-fidelity prototypes are more familiar to users and generally more presentable to stakeholders. For this study, a concern of ours was how to meet these two fidelities in the middle and create a system that has a low technical barrier to entry, allows easy and fast iteration but still delivers high-fidelity results. We argue that the artifact does show that one can develop systems to facilitate the creation of high-fidelity game level prototypes, and that the design principles resulting from this study can be applied to other such tools. We base this statement from the results we have received from our evaluation.

7.2. Recommendations for further research

This study focused primarily on the user experience of video game level creation. The scope of the artifact was heavily limited to support only static 2D platform game levels. This means that the eight design principles that this study resulted in were designed based on this delimitation and may or may not be fully applicable to game level editors of other scopes and natures.

For example, for an artifact that was developed further to be more dynamic to support multiple game genres and perhaps a scripting system, there might be a need for further design principles. What if there is the addition of a third spatial dimension? In the case of an accessible dynamic game creation engine that requires no knowledge of engineering, but still the ability to define game mechanics, inventive thinking would be required to create a model to support it. It would require one to take several steps back, more steps than we took and question what a computer game really is, building off the concepts provided by Djaouti et. al. (2008) to a greater extent than we did in this study.

In addition, how might such principles be applied to game creation systems? In the case of our study, we investigated this problem using a homemade game level creation tool but how might these principles apply to systems built for creating entire standalone games, and what additional design principles might be necessary when tackling that challenge?

Another point for further research would be to investigate how one might fit this way of prototyping into the game development process. That is to say, using user friendly tools to produce testable high-fidelity prototypes.
## 8. Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Market</td>
<td>The market upon which videogames are sold.</td>
</tr>
<tr>
<td>Game Development</td>
<td>“Development in relation to computers means computer programming. In the game industry development refers to all of the tasks that the game creators accomplish.” (Pulsipher, 2010)</td>
</tr>
<tr>
<td>Game Design</td>
<td>“A combination of problem-solving and creativity used to create the framework, structure, and mechanics of games.” (Pulsipher, 2010)</td>
</tr>
<tr>
<td>Game Object</td>
<td>Refers to an object within a video game that exhibits behavior.</td>
</tr>
<tr>
<td>Game Level</td>
<td>“In video games, level usually means a stage or episode or mission that a player completes on the way to completing the entire game.” (Pulsipher, 2010)</td>
</tr>
<tr>
<td>Metabricks</td>
<td>A way of describing how different rulesets interact with each other within the scope of a video game.</td>
</tr>
<tr>
<td>Spawner</td>
<td>A Game Object that defined the location the player begins the level.</td>
</tr>
<tr>
<td>Prototypes</td>
<td>“A prototype is an original model, form or an instance that serves as a basis for other processes. In software technology, the term prototype is a working example through which a new model or a new version of an existing product can be derived.”</td>
</tr>
</tbody>
</table>
| Low-Fidelity Prototypes       | “Low-fidelity prototypes, for example, are simple and low-tech concepts.... The goal is to turn your ideas into testable artifacts that you
<table>
<thead>
<tr>
<th><strong>High-Fidelity Prototypes</strong></th>
<th>“High-fidelity prototypes are highly functional and interactive. They are very close to the final product, with most of the necessary design assets and components developed and integrated.” (Esposito, 2018)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Real-time Strategy</strong></td>
<td>A real-time strategy game is a game in which the player assumes command of several units and buildings on a battlefield, seeking to destroy the other players’ units and buildings in real-time as contrast to turn-based strategy games in which each player waits for the others to finish their moves to act.</td>
</tr>
<tr>
<td><strong>Tilemap</strong></td>
<td>Refers to a grid of tiles, tiles being square shaped geometry that collide with the player and stop their movement.</td>
</tr>
<tr>
<td><strong>Platform Game</strong></td>
<td>“Video game genre in which a principal activity is running, leaping, and jumping, often from one platform (like a ledge, but sometimes in the middle of the air) to another.” (Pulsipher, 2010)</td>
</tr>
</tbody>
</table>
9. References


10. Appendix

10.1. GUI Sketches

Fig 10.1.1 Sketch Iteration 1
Fig 10.1.2 Sketch Iteration 2
10.2. Survey

Section 1 - Let’s lay down the groundwork

- What is your participant ID? *
  - ID Number
- How old are you? *
  - Age as a number
- Did you complete the task? *
  - Yes
  - No

Section 1.2 - Please Elaborate

- Why were you unable to complete the task? *
  - Open Response

Section 2 - Rate your experience

- Please rate your satisfaction with your level creation. *
  - Very Poor (1)
  - Rather Poor (2)
  - Neutral (3)
  - Good (4)
  - Great (5)
- Please rate the usability of the software. *
  - Very Poor (1)
  - Rather Poor (2)
  - Neutral (3)
  - Good (4)
  - Great (5)
- Please rate your overall impression with the software. *
  - Very Poor (1)
  - Rather Poor (2)
  - Neutral (3)
  - Good (4)
  - Great (5)
- Please rate the difficulty of the software *
  - Very Difficult (1)
  - Difficult (2)
  - Neutral (3)
  - Easy (4)
  - Very Easy (5)

Section 3 - Let’s wrap it up

- What was your overall impression of the software? *
  - Open Response
- What features (if any) did you feel were missing?
  - Open Response
- Any final comments for us?
  - Open Response
### 10.3. Raw Data Table

<table>
<thead>
<tr>
<th>Participant</th>
<th>Age</th>
<th>Satisfaction</th>
<th>Usability</th>
<th>Impressions</th>
<th>Difficulty</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4.6</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4.6</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3.6</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4.6</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4.3</td>
</tr>
<tr>
<td>8</td>
<td>39</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4.3</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4.3</td>
</tr>
<tr>
<td>10</td>
<td>28</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4.3</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4.3</td>
</tr>
<tr>
<td>13</td>
<td>36</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>24</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4.3</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3.6</td>
</tr>
<tr>
<td>17</td>
<td>24</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4.3</td>
</tr>
<tr>
<td>18</td>
<td>23</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3.6</td>
</tr>
<tr>
<td>19</td>
<td>30</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4.6</td>
</tr>
</tbody>
</table>

**AVG** 25.2 3.85 4.2 4.25 4.45

Fig 10.3.1 Raw data results table