# Welcome to UWE Bristol - University of the West of England ...

Student Name: Harry Marsh

Student Number: 18016017

UFCFXK-30-3: Digital Systems Project

# The skills, knowledge, and methodology for the development of current-day video development.

by

Harry Marsh

A Dissertation

Submitted to:

The University of the West of England

In Partial Fulfillment of the Requirements

for the Degree: BSC(Hons) Computer Science

# Acknowledgements

Acknowledgements.

# Abstract

In this paper

The abstract is a summary of your entire dissertation. It concisely reports the aims and outcomes of your research and is generally around 150-300 words long, or no more than one page. Write the abstract last when you have completed the rest of the dissertation.

State your research problem and objectives:

• Briefly describe the problem or question your dissertation addresses.

• State your specific objectives.

• Use the present or simple past tense.

Describe your methods:

• Describe how data was gathered and analysed.

• Use the simple past tense.

Summarise your key results or arguments:

• Highlight the most important findings here.

• Use the simple past tense.

Present your conclusion:

• State the answer to your primary research question.

• Mention any significant limitations or recommendations.

• Use the present tense.

The abstract is the **first thing** that someone reads in a report to get a general idea of the project. Therefore, it plays a vital **role**.

The abstract should provide a **short description** of your project. It usually is no more than one or two paragraphs **summarising your work**. A good strategy is to write this at the end of your project.

# Table of Contents

Contents

[Acknowledgements 2](#_Toc70176913)

[Abstract 3](#_Toc70176914)

[Table of Contents 4](#_Toc70176915)

[Table of Figures 5](#_Toc70176916)

[List of abbreviations 5](#_Toc70176917)

[Introduction 6](#_Toc70176918)

[Literature Review 8](#_Toc70176919)

[Requirements 13](#_Toc70176920)

[Methodology 17](#_Toc70176921)

[Design 18](#_Toc70176922)

[Implementation 19](#_Toc70176923)

[Project evaluation 20](#_Toc70176924)

[Conclusion 21](#_Toc70176925)

[References/Bibliography 22](#_Toc70176926)

[A) Appendices 23](#_Toc70176927)

# Table of Figures

# List of abbreviations

* SDM – Software Development Methods
* FPS – First Person Shooter

# Introduction

The contents of this report outline and demonstrate the development of an application with educational intent, focusing on teaching programming logic. The intended targets for this project are identified as being between the age of 8 – 14. They focus on providing an informative and exciting method of teaching the foundation of programming.

Within this report, the answers to many questions are discussed and researched. These questions include:

* What is the impact of educational applications on the current young generation?
* How can education take place through interactive multimedia such as games and videos?
* How can these mediums benefit in creating an understanding of topics and information?
* How can interactivity and design affect focus and attention with younger users?

*This chapter should provide a* ***short description of your project****, including the* ***aims and objectives*** *and the* ***scope*** *of the project, your* ***research questions,*** *and an outline of your* ***report's chapters****. Ideally, your introduction chapter should answer these questions:*

* *What is the real-world problem you are looking at?*
* *Why is it important?*
* *What are your research questions?*
* *What did you learn from the literature review? (provide an outline of your findings)*
* *How did you approach the problem?*
* *The outcome of the project?*

*At the end of the Introduction Chapter, you should provide an outline of your report's chapters. For instance:*

*Chapter 2 provides a critical review of relevant work/tools/software blah blah blah. Chapter 3 includes … In Chapter 4 we provide the … etc.*

# Literature Review

In this chapter, the review and analysis of relevant literature are shown. The literature discussed in this chapter focuses primarily on the technical requirements of this project. It outlines the educational impacts of the project relating to its significance within its field. The development of this application is based on the research that is currently taking place for the impact and attitude to skill developed through video games. The literature that is being reviewed focuses on younger age groups and how video games can influence children into developing interests in skill-based careers and providing an engaging method to learn these methods.

The article (Barr, 2017) uses qualitative interview data to analyse commercial video games used to develop valuable skills. The article uses a collection of interview data to judge whether or not many skills referred to as 'graduate attributes' (Barr, 2017) are exercised and developed. These skills were listed as 'communication, resourcefulness and adaptability. This report is created to develop these and implement other necessary skills that a younger age group would benefit from developing, such as puzzle-solving and understanding how to solve logical issues.

The article uses a data set of a collection of undergraduate students playing a collection of games for 2 hours over 8 weeks. This collection of data was performed on various games in a variety of genres with a focus on their ability to develop specific skills. This literature supports the development of video games and the skills gained by playing and developing. Due to the articles' clear response proving through the data collected in the interview, the development of video games to develop and exercise skills is proven to have the potential to be effective and positively impact children (Barr, 2017).

In addition to the previous article, the paper (Barzilai and Blau, 2013) discusses the impacts on learning achievements concerning experiences through games. This discussed achievement can be associated directly with the development of games. The paper focuses on the challenges of making connections between knowledge learned in games compared to knowledge learned in school and their relations to engagement and attention. Like the previous article, the data set used to support their point is collecting qualitative interview data on an average age group of "10.10 years". The difference between the papers is the relation to games and studying and their impacts on each other. For example, the first paper discusses the relation to be learning directly from games. In contrast, the paper (Barzilai and Blau, 2013) discusses learning with a study and play, play and study and play only structure to learning. The relation of this material to the topic discussed in the report supports and justify why video games have become an important topic and debates for the relativity of learning and attention.

The previous topic discussed solidly supports and justifies the reasons behind this paper and the developed software alongside it. The software developed alongside this paper outlines the skills, methodology and knowledge required for the development of video games and the suggestion of introducing game development at a young age in an educational environment. The article (Kafai 2016) discusses game-making approaches and the skills required to make games as a creator, and the skills and abilities learnt from the games. The software development for this paper takes both the skills and methods discussed in this paper to produce and covers the skills that are learned through the development and their relation to other fields of software engineering. Within the paper, the conclusion of "*It is clear from our analyses that video game making can provide a rich context for learning programming*" (Kafai 2016) was made with the consideration of the cultural participation of video games.

The discussion of which development method to perform while producing the software was supported by the paper (Bygstad, 2007). When developing a project of this scale, research for appropriate SDM's (software development methods) is needed to ensure maximum productivity. The article discusses two development types, one agile method and the other known as a rational unified method. Both methods discuss different disciplines that are further explained in the dissertation. In addition to these two methods, the paper discusses the use of personal development methods tailored to developers by themselves what quantitative data of the SDM usage. The paper showed that 68% of people who were utilised to make the data sets used their SDM process to develop software.

The design of a video game is a crucial portion of the required research when planning and preparing for game development. Depending on the game's intent, for example, many components need to be considered for personal use or distribution. In the perspective of commercial development, the intent to sell the product, the design, and the category must be carefully planned. The paper (Moore, 2016) discusses an essential feature for consideration when developing a game. The paper details essential game mechanics and features such as the maths and logic of games concerning the inside mechanics such as movement, cameras, and other game objects.

With the game category in development, a puzzle platformer Moore's paper (Moore, 2016) can be utilised to provide structure and consistency with design. In chapter 4, the article discusses the importance of movement and the player's interactions with the character within the game. It is stated that limitations must be implemented into the game to fit into the users' playthrough so that characters perform how the user would expect. The information collected about movement and scale, including speeds of objects to create an immersive environment, can be used and implemented into the development of the program that this literature is based on. Overall, the paper expands into extensive detail about almost every aspect of the design of video games of one specific category. Thus, only a particular aspect of this paper can be utilised in the development of the code. Still, the practice and methodology discussed can be modified and manipulated to fit any category of the game, providing a good standard of design.

The paper (Moore, 2016) proves to be beneficial in guiding developers with the standard of design of internal functions and extensively covers level development and User interface design. Concerning level design Moore (Moore, 2016) States, "Whether in 2D or 3D, the playing field for a game has to be appropriate for the game world and enjoyable to explore." This quote is about the design of worlds, also known as levels. The video game under development has a procedurally generated level. According to the article, this is important to have a large variety of challenges for the player to "explore" to keep engaging their interest. Also, making the levels increasingly challenging can be used as an incentive to increase engagement with the game.

The incentive for companies to develop video games and the incentive for developing this project is lightly discussed by (Fencott, Clay and Massey, 2012) with relation to types of games and their genre. The paper, specifically the chapter about genres, discusses the community of computer games and their separation based on the genre of games. This paper helps outline the market of games with relation to the genre of game. For example, the popular FPS genre consists of the larges group of players in that category with titles that include Call of Duty and Battlefield. This is one of the most popular genres, including open-world games. The market is full of constant new releases. With this knowledge, developing a small puzzle platform comes from the ideology of the small game creator community. As discussed, (Fencott, Clay and Massey, 2012), this community involves the majority of less popular game genres such as puzzle, platformer, adventure, and more. This is the target audience that the project is developed to reduce the impacts of competition if the project were to enter the gaming market.

In addition to the article previously, the paper (Aguzzoni, 2014) studies the effects of a merger between two large game development companies and its effect on the market of video games. As this project is developed to the stander of commercial distribution, the importance and understanding of the market justify the type of game developed and the platform that the game is developed, e.g., mobile, pc or console. Although the article being about a merger between two game distribution companies, Game and GameStation, the paper discusses the current value of the gaming market being around " £2 and £3 billion in 2006" (Aguzzoni, 2014) with a collection of online sources stating that the gaming market is an estimate of "USD 151.55 billion in 2019". This is important to the project as it justifies the financial initiative to produce games with the intention of commercial distribution due to the apparent growth in the gaming market, which leads to an increased demand for content.

There are many different development methods that are utilisable for the development of any software including video games the paper (Aleem, Capretz, and Ahmed, 2016) out lines some of these methods as well as good programming practises that should be implemented into any program. The discussion of development types in the paper includes object-oriented programming and event related programming. The method used in this project is object-oriented programming due to how useable and practical it is for the development of video games as stated by the paper. The focus for this paper is towards the development factors that are needed for the development of a successful object orientated program.

Aguzzoni, L. (2014) A Retrospective Merger Analysis in the UK. Videogame Market. Journal of Competition Law & Economics [online]. 10 (4), pp. 933-958. [Accessed 26 March 2021].

Barr, M. (2017) Student Attitudes to Games-based Skills Development Learning from Video Games in Higher Education. Computers in Human Behaviour [online]. 80 (1), pp. 283-294. [Accessed 24 January 2021].

Barzilai, S. and Blau, I. (2013) Scaffolding Game-based Learning: Impact on Learning Achievements, Perceived Learning, and Game Experiences. Computers & Education [online]. 70, pp. 65-79. [Accessed 18 March 2021].

Bygstad, B., Ghinea, H., Ghinea, B. and, (2007) Software Development Methods and Usability: Perspectives from a Survey in the Software Industry in Norway. Interacting with Computers [online]. 1 (20), pp. 375-385. [Accessed 14 January 2021].

Fencott, C., Clay, J. and Massey, P. (2012) The Theory and Understanding of Computer Games. *Game Invaders* [online]., p. 240. [Accessed 14 March 2021].

Kafai, Y.B. (2016) What Making Video Games Can Teach Us About Learning and Literacy. Connected Gaming [online]. 1 (1), pp. 224-225. [Accessed 26 January 2021].

Kangas, M. (2009) Creative and Playful Learning: Learning Through Game Co-creation and Games in a Playful Learning Environment. Thinking Skills and Creativity [online]. 5, pp. 1-15. [Accessed 23 March 2021].

Moore, M. (2016) Basics of Game Design. *Basics of Game Design* [online]. 1 (1), p. 361. [Accessed 24 March 2021].

Aleem, S., Capretz, L.F. and Ahmed, F. (2016) Critical Success Factors to Improve the Game Development Process From a Developer’s Perspective. Journal of Computer Science and Technology [online]. 31 (5), pp. 925-950. [Accessed 28 March 2021].

# Requirements

This chapter will include the requirements of the project with analysis. The requirements will involve both functional and non-functional requirements in relation to the project.

Functional Requirements.

* The software must have an interactive and easy to use menu for navigation.

The design and usability of the software is defined by the users first interaction with the software. This interaction for the projects first time interaction is the menu. For this reason, it is required that the menus be instantly recognised and able to use from the start of the application to ensure that the user does not get confused or annoyed by the system. In addition, the applications menus must flow seamlessly between each menu such as main, game or options to ensure the most usability possible for the application.

* The software must consist of levels that are achievable.

The game is required to be fully playable and completable in relation to its levels. This means that the user should not fail at a level due to an impossible senari9o for example the user should be able to reach every platform that is presented to them and be able to dodge all obstacles. This is to ensure that the user does not fell like the game is unfair and there for unplayable. This can be achieved by ensuring that the procedurally generated terrain does not create impossible routes.

* The software must have a scorekeeping system including a high score on all of the levels

A score keeping system is required within the software to ensure that the user is able to see that progress is being made. It is also required as it provides a sense of competition in the user to best their own high scores. This requirement is crucial to the game as it assists in the repeatability of the game. Leader board could possibly be implemented to share high scores between friends increasing the competition between players.

* The software must have the ability to control the character through the use of user input such as a keyboard and mouse

The ability to interact with the software is a crucial requirement to the system to ensure that the game is playable. This interaction includes the interactions with the previously discussed menus but also extends to include the player-controlled character. The control method can very between multiple devices and methods. For example, the game being on mobile requires to be functional through the user of a finger tap and for a pc version through the use of either a mouse click or the pressing of the space bar on a keyboard.

* The software must have procedurally generated terrain for the user to navigate.

Procedurally generated terrain is a term used to describe level design that is randomly generated as the player continues through the game. This is required as it ensures that the game is unpredictable and different for every playthrough implementing luck into the game play. This is required as it increases the playability and the repeatability of the game.

* The software must have sound and music to complement game play

A compliment of sound and music is required to immerse the player into the game. The sound of the game must be controllable in the settings allowing the user to adjust volumes. The audio added to the game is required to simulate interaction between the controllable character and other objects and terrain within the game.

* The software must include a level end point

The requirement for an end point in the game is made due to the scoring system within the game but also by ensure the game is not endless. An endpoint within the game can be achieved in multiple ways. The first method that is required is a pause menu. Other examples include the death of the playable character through the collision with objects, or falling off the map including other boundaries of the play area.

* the software must have the ability to pause and play the program

A pause ability is required within the game so that the user is able to take a break from the game. This pause must consist of a menu to either exit the game to the main menu as well as the ability to resume the game. When the pause is enabled all actions within the game must pause in place so that the player does not loos the game by pressing pause.

* The software must have an adaptive resolution for devices so that it can be played in all screen sizes and shapes

Having the ability to run on all sizes and shapes of displays for a system is crucial to a system. All attributes and objects within the game must be able to scale depending on the device screen that is being used. This is a requirement as without this feature the game may not be playable on devices that have too large of a screen such as ultra-wide monitors or too small of a screen such as some mobiles.

* The software must contain realistic collisions between multiple objects

Physics within the game is required to ensure that the game is realistic and playable. The collision physics are a large requirement within the game as it allows for the playable characters to interact with objects and more importantly the floor and use the interactions within the game to give the player interaction to the character. In addition, the physics of the game includes the gravity and weight of characters and objects that is used to justify how fast the character falls.

Non-functional Requirements.

* The software must be able to run on low specification devices to ensure good availability

The ability to run on low spec systems will increase the size of the user group for the game allowing for more people with a large variety of devices and specifications to enjoy the game. This requirement can be achievable by implementing the ability to change the graphical quality of the game as well as ensuring that all textures and assets that are used within the game are only as detailed as necessary to save on system usage.

* The software must be stable and contain no bugs that cause a significant effect on user experience.

This requirement is to ensure that the game is always playable. This includes ensuring that the game does not crash during gameplay or hinder the gameplay to cause annoyance to the player. This includes any bugs that will make the levels unplayable or unfair forcing the player to lose due to no other option.

* The software GUI must be easy to understand

The requirement for an easy to understand and quick to learn GUI is very high although not considered one of the most important parts of any video game it has the most important job of first-time interaction. An overly complex GUI can throw off any new users from wanting to attempt to learn a game. With the context of this project the simpler the GUI thew more beneficial and easier it is to read it is for the user. This requirement is what introduces the users to the game and provides an idea or theme to what they are about to play.

* The software must fit into the universal category due to the target demographic being 11-16 years old.

This requirement was put in place due to the requirements of the University of the West of England as well as market research for the type of game being made. In addition to ensuring it works well on all devices its important that the game can be played by all age groups and ethnicities to ensure the maximum number of potential engagements for the game.

* The software must have adaptable settings to increase usability

In addition to the requirement above availability is key when producing a multi-platform application. With the requirement of adaptable settings, the game is able to function on all platforms that it can be exported to which includes PC/ISO/Android. With the adaptable settings platforms such as mobiles which have smaller processors, and less memory will be able to have an adaptive video setting to ensure that there is a balance between quality and performance.

* The software must have a tutorial to describe how the software operates

The program requires a method for a new user to learn or understand the game that they are plying. With this in mind a tutorial that will be used to not only show how the player can control the playable character but also how the game mechanics will function. In addition, the use of a setting menu or graphic that can be added to the game to achieve the same requirement within the program for example a small tap icon could be implemented to suggest the player to tap the screen and explore the reaction in game.

* The software should be access able from a reliable source

A secure place of download is required for any piece of software when ready for distribution. Depending on the platform the game can be released onto popular app stores such as the apple store or google play. Putting the software on these distributions mediums allows for security scanning and verification to ensure that your app is safe for distribution. In addition, if being released as a project like the software being create4d the use of ?GitHub repository could be beneficial as it will provide a platform to access the application directly from the source.

* The software should have compatibility with multiple platforms

Compatibility with multiple platforms is required to ensure that max engagement can be achieved. For the game to run successfully on multiple platforms a large number of features and requirements need to be implemented. Most of these implementations that are required have been discussed in this chapter but include, adaptive graphics and GUI elements as well as its ability to run on low spec.

* The software should have a visual appeal in relation to design

This requirement focuses on the engagement and interests of the user. A fun colourful game with playful characters and objects will be required within the game to make it visually appealing. In addition, it is required that the visual style or theme of the software is consistent.

* The software should be engaging and fun for the user

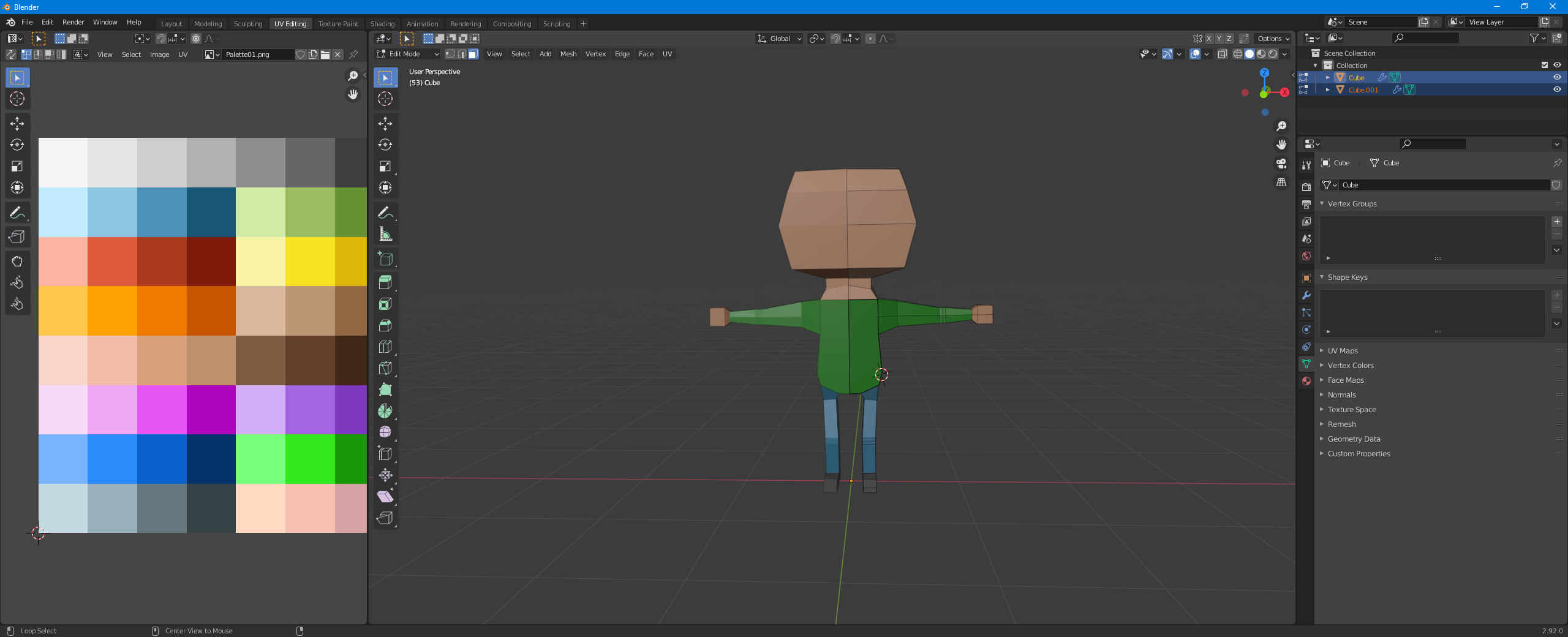
This requirement means that the software should be playable and enjoyable for the user to achieve the software’s purpose of being an entertaining video game. This requirement can be achieved by ensuring that all of the requirements in this chapter are completed to the best of the developer’s ability.

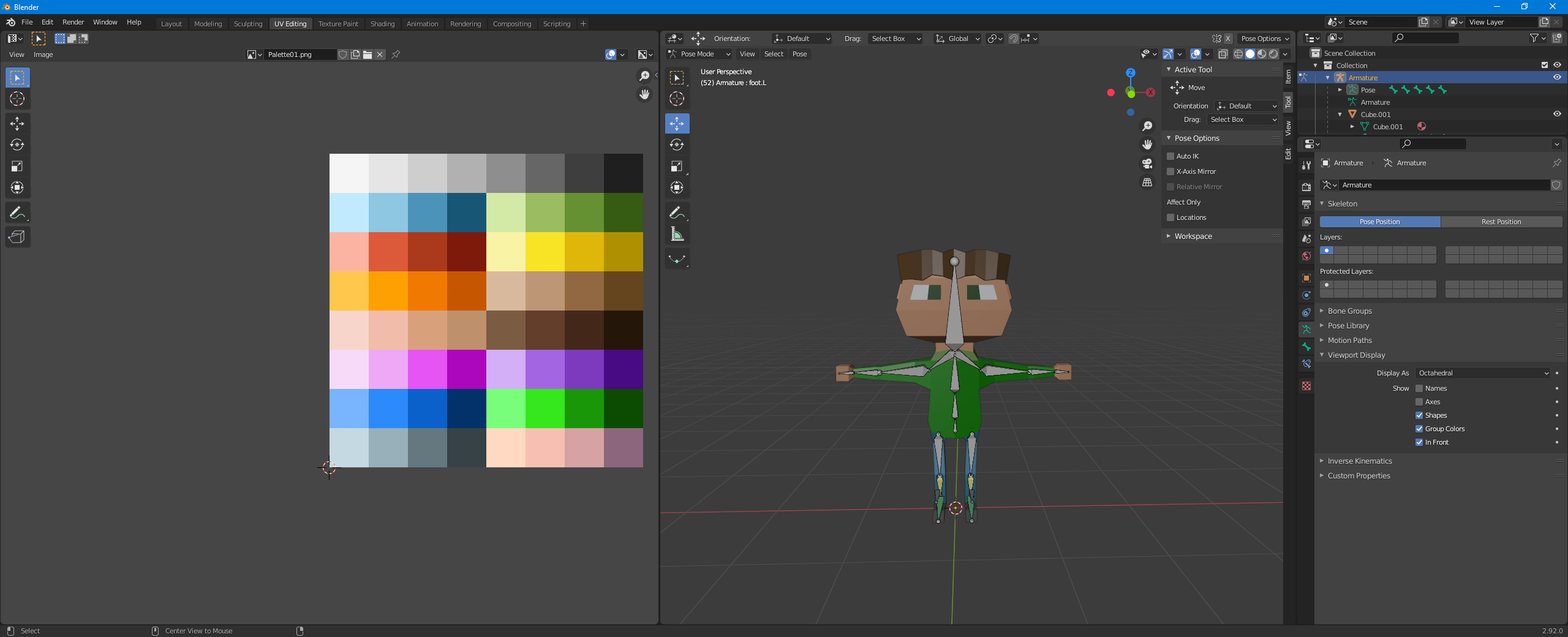
# Methodology

A short description of the methodology you followed. For more information about the various methodologies please read Chapter 2 "Software Processes" and Chapter 3 "Agile Development Software" from the Software Engineering book written by Ian Sommerville (2016) – see the reading list section on Blackboard.

# Design

Description and presentation of the design. For more information, please check the "design" documents on Blackboard (in the Learning material section).





# Implementation

Description and presentation of the implementation. In this chapter, you need to demonstrate your technical skills and identify how/whether your product meets the aims and objectives of the project.

A short description of the implementation. This chapter should answer the following questions:

* How did you implement the project and why? (demonstrate technical skills; no need to provide code; could provide some screenshots of the product)
* Have the aims and objectives of the project been met?
* Software testing
* Reflection on the test's results

# Project evaluation

You should use this chapter to reflect on all aspects of the project.

Reflect on the project's outcome, the difficulties you faced and how you overcame them.

* Discuss **limitations**. Reflect on the tests' results.
* Any room for **improvement**? Further work?
* **Reflect** on all parts of the project (research, requirements, implementation etc.)
* **Reflect** on the way you used the **supervisor's feedback** (both from the Project in Progress day and your meetings).

# Conclusion

The chapter concludes your report. It should include a summary of your work focusing on its outcomes (e.g., the final product).

Further work can also be included in this section (instead of the evaluation section)

# References/Bibliography

# Appendices

Appendices should go at the end of your report. Appendices should be numbered to easily identify them and also point to them from within your main report.