Bald Man’s Adventure

A 2D Rogue-Like Videogame

Daniil Kazakov

20380923

Final Year Project – 2024

B.Sc. Single Honours in

Computer Science and Software Engineering

****

Department of Computer Science

Maynooth University

Maynooth, Co. Kildare

Ireland

A thesis submitted in partial fulfilment of the requirements for the B.Sc.

Single Honours in Computer Science and Software Engineering.

**Supervisor**: **Dr Wu Hao**

Table of Contents

[Declaration i](#_Toc161142903)

[Acknowledgements ii](#_Toc161142904)

[Abstract iii](#_Toc161142905)

[Chapter one: Introduction 1](#_Toc161142906)

[Summary 1](#_Toc161142907)

[1.1 Topic addressed in this project. 1](#_Toc161142908)

[1.2 Motivation 1](#_Toc161142909)

[1.3 Objectives 2](#_Toc161142910)

[1.4 Game Design Principles. 2](#_Toc161142911)

[Chapter two: Technical Background 4](#_Toc161142912)

[Summary 4](#_Toc161142913)

[2.1 Why Godot Engine? 4](#_Toc161142914)

[2.2 Why C#? 4](#_Toc161142915)

[2.3 Comparison of Godot to other engines. 4](#_Toc161142916)

[Chapter three: The Problem 4](#_Toc161142917)

[Summary 4](#_Toc161142918)

[3.1 Project UML documentation 4](#_Toc161142919)

[3.2 Problem analysis 5](#_Toc161142920)

[Chapter four: The Solution 6](#_Toc161142921)

[Summary 6](#_Toc161142922)

[4.1 Analytical Work 6](#_Toc161142923)

[4.2 Architectural Level 6](#_Toc161142924)

[4.2 High Level 6](#_Toc161142925)

[E.g. Packages, Class Diagrams, etc. 6](#_Toc161142926)

[4.2 Low Level 6](#_Toc161142927)

[E.g. Method specifications, Algorithms, etc. 6](#_Toc161142928)

[4.2 Implementation 6](#_Toc161142929)

[Chapter five: Evaluation 7](#_Toc161142930)

[Summary 7](#_Toc161142931)

[5.1 Solution Verification 7](#_Toc161142932)

[E.g. use your equations to verify the correctness of your solution 7](#_Toc161142933)

[5.2 Software Design Verification 7](#_Toc161142934)

[5.3 Software Verification 7](#_Toc161142935)

[5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing) 7](#_Toc161142936)

[5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.) 7](#_Toc161142937)

[5.3.3 Your test results 7](#_Toc161142938)

[5.3.4 An interpretation of the results 7](#_Toc161142939)

[5.4 Validation/Measurements 7](#_Toc161142940)

[5.4.1 Results 8](#_Toc161142941)

[5.4.2 Explanation of Results 8](#_Toc161142942)

[5.4.3 Analysis of Results 8](#_Toc161142943)

[5.4.4 Comparison with previous solutions (if relevant) 8](#_Toc161142944)

[References 10](#_Toc161142945)

[**Appendices** 11](#_Toc161142946)

[Appendix 1 Schematic of the hardware associated with this project. 12](#_Toc161142947)

[Appendix 2 Code developed for this project. 13](#_Toc161142948)

[Appendix 3 UML Class, Use Case and sequence diagrams for this project. 14](#_Toc161142949)

[Appendix 4 Screen shots of the project implementation 15](#_Toc161142950)

## Declaration

I hereby certify that this material, which I now submit for assessment on the program of study as part of Computer Science and Software Engineering qualification, is *entirely* my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of my work.

I hereby acknowledge and accept that this thesis may be distributed to future final year students, as an example of the standard expected of final year projects.

Signed: Daniil Kazakov Date: 11/03/2024

## Acknowledgements

I would like to thank my supervisor, Dr. Wu Hao for providing great support, feedback and motivation throughout the duration of this project.

Also, I would like to thank my fellow students who have helped me to test my application and provided great feedback that was used in the evaluation of the developed game.

## Abstract

How difficult is it to develop a game that would meet the industry standards and player expectations without any prior experience and knowledge? This is the goal that was set for this Freedom Project. The development of a roguelike game is described in this paper, from the original concept to a functional prototype with numerous features. The game features the fundamental components required for a fully developed product, even though it is far from becoming a finished, marketable product. With a little more development and polish, these essential components have the potential to make the game something that might be sold. The process of creating the game was highly instructive, providing a practical look at game design through tasks like creating pixel art and using game engines from scratch. By emphasising strategy and character development through gameplay, the game embodies the fundamental elements of the roguelike genre, making it entertaining and compelling to play again. Let’s see how this was done.

# Chapter one: Introduction

## Summary

The goal of this project is to discover the basics of developing a game, starting from choosing the game engine and the genre of the game, to having a playable product that resembles other products of the genre on the market. Developing a video game involves many different features. Large scale games by world-renowned studios might have an entire team of developers working on a single feature. Since this project was conducted by a single person, this paper will try to discuss all the challenges and steps taken to resolve them in a concise but informative way on the high and low levels.

## 1.1 Topic addressed in this project.

This paper talks about but not limited to the following topics:

* Intricacies of developing videogames using Godot Engine.
* Developing game logic using C# and why you might want to use this language.
* Creating/ finding art, sprites and sound effects for the visuals and the audio aspects of the game.
* Developing the game logic that combines all the above into a good videogame.

## 1.2 Motivation

Gaming industry is one of the most rapidly growing industries in the world. It generates more revenue than the movie and music industries combined. Although it takes a lot of effort and commitment to develop a video game, it could be very rewarding. The experience you gain along the way helps you to discover different aspects of Computer Science and how all of them can be linked together into a single product. If such product comes out to be good, combined with some advertisement might improve one’s financial situation greatly. Combination of all the above is the best way to describe the motivation behind this product.

## 1.3 Objectives

At the start of the project, many objectives were made, but they all can be summarised into the following four:

* Game Development Experience: Acquisition of hands-on experience with game development processes and procedures, utilizing Godot Engine and C# programming language.
* Engaging Gameplay: Creation of engaging and compelling gameplay that would help the players to distract themselves from the real world while having an easy learning curve.
* Learning and enhancing existing knowledge: Using the relevant knowledge gained throughout the Computer Science course and enhancing it with new technologies and techniques.
* Product Viability: Development of a prototype that has all the basic game mechanics, simple visuals and the audio covered. As the prototype scales, all of these can be improved / changed easily.

## 1.4 Game Design Principles.

What are the Game Design Principles? Game design principles focus on players’ experience from start to finish. It lays out how the player will play the game, from its core mechanics to its overall story arc. Here are the five fundamentals that were followed during the development:

* Game designed around one core mechanic: Having many features to your game is great, but you must focus on a single core mechanic e.g. not turning a shooter game into a visual novel.
* Game being easy to learn but challenging to master: If you make a game too hard from the start, it might scare the player off. Hence beginning of many games is usually easy and welcoming. After the player learns the basics, you must increase the difficulty, otherwise it will be too easy and boring.
* Balanced gameplay: Having a balanced gameplay in all aspects is essential. If you have many options to go with but only one of them is working well/ too good compared to other options, it will make the player go with that every time and your game will become stale.
* Regular feedback and rewards for the player: When player invests their time into your game, they expect to get emotions/ feeling of accomplishments back. After achieving a hard goal, getting a good reward can release endorphins (hormone of happiness) into the system. This further increases players’ immersion with the game.
* Meaningful core mission: You must give players an end destination to which they will strive throughout their time with your game. It sets a goal and gives an incentive to play the game through until the very end.

# Chapter two: Technical Background

## Summary

This chapter will describe how the project was started, why the Godot engine was chosen as the Game Engine for this project and other successful projects made on this engine. It will also talk about other technologies that were used in this project.

## 2.1 Why Godot Engine?

## 2.2 Why C#?

## 2.3 Comparison of Godot to other engines.

2.4

# Chapter three: The Problem

## Summary

The purpose of this chapter is to clearly explain the technical problem and/or identify the user requirements.

## 3.1 Project UML documentation

Provide any model(s) of the problem (e.g. equations, ERD’s, UML Use Cases & Scenarios, Activity Diagrams, etc.)



Figure 3‑1 UML class diagram overview for this project.

## 3.2 Problem analysis

Provide any analysis of the problem, leading to a greater understanding. There should be no decisions made in this chapter.

# Chapter four: The Solution

## Summary

The purpose of this chapter is to clearly identify, discuss, and justify the decisions you make

Depending on your type of project, you may not need to include all of these:

## 4.1 Analytical Work

E.g. Equations, etc. that describe your solution

## 4.2 Architectural Level

E.g. Implementation Diagrams

## 4.2 High Level

## E.g. Packages, Class Diagrams, etc.

## 4.2 Low Level

## E.g. Method specifications, Algorithms, etc.

## 4.2 Implementation

Discuss anything interesting here; put full source code in an appendix or attachment

# Chapter five: Evaluation

## Summary

Chapter 5 describes……..

## 5.1 Solution Verification

## E.g. use your equations to verify the correctness of your solution

## 5.2 Software Design Verification

How did you show that your design worked properly?

Using a model of your solution. E.g. use UML interaction diagrams to verify each scenario.

## 5.3 Software Verification

How did you demonstrate your software worked properly?

If you have not tested your software, then you cannot rely on your results. Clearly describe:

### 5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing)

### 5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.)

### 5.3.3 Your test results

### 5.3.4 An interpretation of the results

## 5.4 Validation/Measurements

How did you measure how well your solution solved the problem.

### 5.4.1 Results

### 5.4.2 Explanation of Results

### 5.4.3 Analysis of Results

### 5.4.4 Comparison with previous solutions (if relevant)

Chapter six: Conclusion

Summary

Chapter 5 identifies and discuss the implications of your work.

5.1 Contribution to the state-of-the-art

If you made a contribution to the state-of-the-art, clearly identify it here.

5.2 Results discussion

Discuss whether your results are general, potentially generalizable, or specific to a particular case. Identify threats to the validity of your results (e.g. limitations, risks introduced by your approach, etc.)

5.3 Project Approach

Discuss your project approach

5.3 Future Work

Discuss future work, based on what you have done (and not done)

# References

Action Research : A definition . (2015). Retrieved February 25, 2016, from http://valenciacollege.edu/faculty/development/tla/actionResearch/ARP\_softchalk/ARP\_softchalk\_print.html

Hammersley, M. (1993). On the teacher as researcher. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 211–231). The Open University.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Source: Administrative Science Quarterly Qualitative Methodology*, *24*(4), 602–611. Retrieved from http://www.jstor.org/stable/2392366

Kemmis, S. (1993). Action Research. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 175–190). The Open University.

Kolb, D. (1984). *Experiential learning*. New Jersey: Prentice Hall.

McNiff, J., Lomax, P., & Whitehead, J. (2003). *You and Your Action Research Project* (2nd ed.). London & New York: London & New York.

**Appendices**

Include here all extra material, e.g. your source code, project management (optional) including: the task list, Gantt Chart diagrams (or equivalent), discussion of any significant deviations from plan, and how you managed them, discussion of what you would do differently if you repeated the project.

## Appendix 1 Schematic of the hardware associated with this project.

## Appendix 2 Code developed for this project.

## Appendix 3 UML Class, Use Case and sequence diagrams for this project.

|  |
| --- |
|  |
| Appendix 4 Screen shots of the project implementation |
|  |