OCR GCE A

COMPUTER SCIENCE PROJECT

H446-03

Name : Bryan Dissanayaka

Candidate Number : <INSERT CANDIDATE NUMBER>

<Institution Name> : <INSERT CENTRE NUMBER>

Title of Project : <INSERT PROJECT TITLE>

H446-03 – Project CONTENTS

Table of Contents

[A. Analysis 3](#_Toc77297534)

[Stakeholders<See H446-03 Project Advice Booklet for help and guidance of what must go here.> 3](#_Toc77297535)

[OUTLINE 3](#_Toc77297536)

[B. Design 3](#_Toc77297537)

[Systems diagram 3](#_Toc77297538)

[C. Developing the coded solution (“The development story”) 3](#_Toc77297539)

[D. Evaluation 3](#_Toc77297540)

[Project Appendixes 4](#_Toc77297541)

# A. Analysis

## Stakeholders

As the game is made to be easy to be played by anyone, I will give a demos of the game as I develop it to both developers and non-developers.

For developers, I can get feedback from friends and online game development communities which will be more objective and critical, and will comment on both the experience of the game and on the technical side of it. From the proposed solution, the developers will have a game they can analyse and gain knowledge from it. After I complete the project, I will be uploading the final coded solution to a public GitHub repository, along all the assets, so that they can download it and use my code in their projects.

For the non-developers I will ask friends and family which can describe whether or not the game is actually fun and engaging or not.

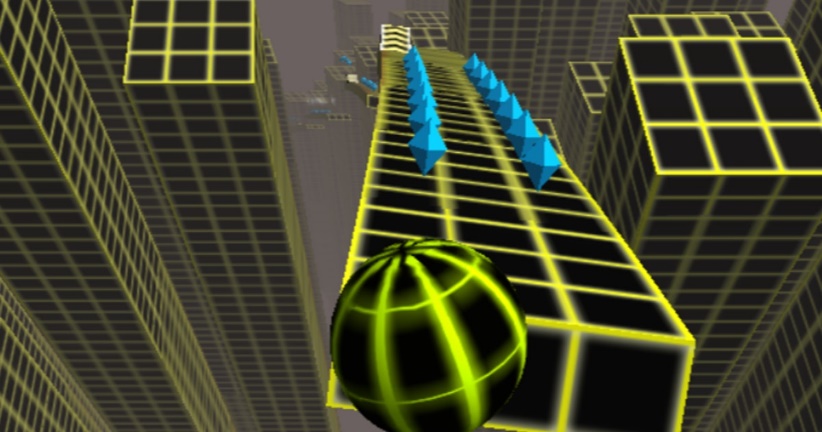
## PROBLEM

Most casual games like this do not give enough choices to the player in order to progress through a level in their own way. There is always only way to complete a level and everyone that plays the game will have the same experience and because of this.

To solve this problem, I will give the player a variety of things they can use to complete a level so that they can come up with their own unique strategy. This will also allow them to replay the same level but using a different strategy, therefore it will make the game automatically longer to be completely finished.

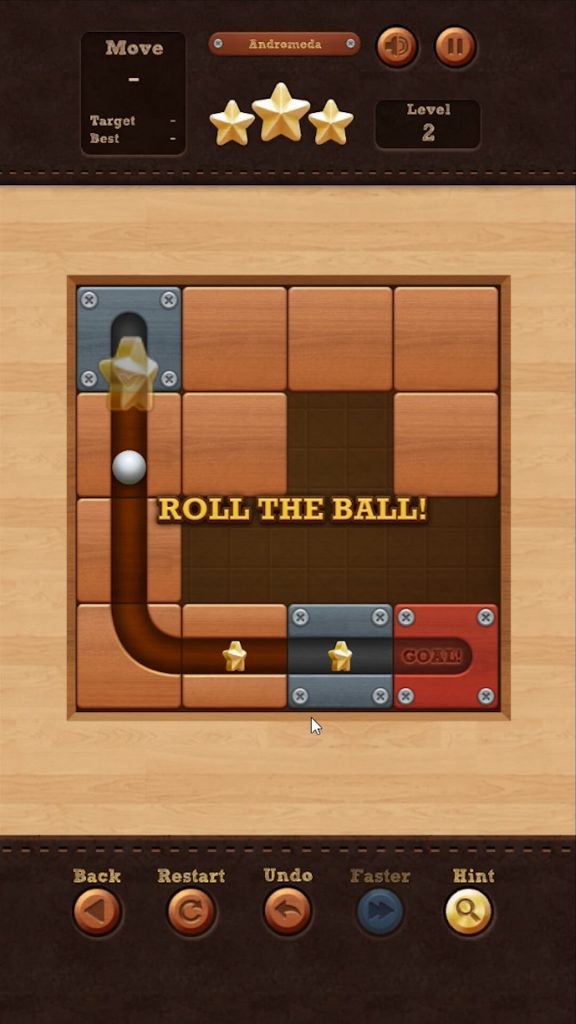
## RESEARCH

Plenty of roll-a-ball games have 3D graphics to impress the player and make them think that they will able to fully utilise the 3 directions, while only featuring 1 or 2 directions of input.



An example is a game called Crazy Roll 3D, where the player can only move the ball along 3 lanes and the ball automatically moves forward and the game stops when the balls collides with an obstacle. The input is 1 directional.

My game will have 2D graphics and the ball will only be moved by gravity acting downwards.



A game that comes closest to my game’s type of gameplay is Roll The Ball. The concept is similar: shape the level in order to complete it, with no control over the player, which is the ball.

My game differs from it as Roll The Ball features only one mechanic, which is to slide the tiles. As mentioned in the problems section, every single player playing this game will complete all the levels in the same way, and none of them have replay value because of this.

It is a fun casual game to play on a mobile device, but once the player completes all the levels, they will have no reason to keep playing the game.



An interesting mechanic my game will feature is inspired by a minigame from Super Mario 64 DS on the Nintendo DS. The players were able to draw trampolines on the touch screen in order to keep Mario from falling off. The angle of the trampoline mattered because to score points Mario would have to go through the rainbow coloured hoops. The size of the trampoline determined how bouncy they were. A smaller trampoline would make Mario bounce higher but it easier for him to miss it and fall down.

I want to create a similar mechanic where players will be able to draw slopes to make the ball roll in the direction they want.

## SYSTEM REQUIREMENTS

-Desktop

OS: Windows 10

CPU: Intel Core 2 Duo E8400 CPU

RAM: 4GB

GPU: NVIDIA GeForce 6200 or higher, compatible with OpenGL3

STORAGE: 1GB

-Mobile

OS: Android

RAM: 2GB

GPU: Compatible with OpenGL3

STORAGE: 1GB

\*\*These are the generic Godot system requirements and can change later

## OUTLINE

The game will feature a non-controllable player character. The player will move automatically and the user must

change the environment and objects of the level in order to get the player to the end goal.

The main inspiration where the game’s fundamental goal comes from is a minigame from Wii Party, where the players tilt their controllers to rotate the stage in order to roll a ball to the bottom of the level.



The main (and only) controls of this minigame are tilting the controller left and right. This minigame is also multiplayer, and the player reaching the end first wins. My game will not feature motion controls but mechanical and touch controls. To make the game more interesting I will add mechanics like enemies, lasers, different terrains that affect the ball’s speed, drawing shapes on the screen to create slopes etc. The player will be offered a variety of methods to complete each level of the game, and they will able to replay the same levels with less power-ups and mechanics to impose themselves with a more challenging experience.

The game will feature an infinite mode as well, where there is no end goal, but a high-score system where the score is based on how far down the player’s ball has travelled.

## LIMITATIONS

As the game is targeted to mobile devices, memory management and processing power need to be carefully assessed. The game will have 2D graphics as a 3D game with high polygon counts for the assets will consume the device’s battery much quicker.

For the infinite game mode, instead of instantiating new assets as the player goes down, the no longer used assets from above will be shifted down to reduce memory consumption. This creates a limitation in terms of game design as the game can seem to be repeating itself quite frequently.

Another method that could be used to solve this problem is to simply delete the unused assets when instantiating the new ones, so that memory consumption is constant. In this case the problem would be to load in the new assets required as the player moves down in a fast and seamless way. Sometimes it may cause stutters when large more complex assets are loaded in such as multiple enemy with specific behaviour and respective real-time pathfinding.

The game engine I am using, Godot, lets me select between OpenGL2 and OpenGL3. The first one will make the graphics of the game less demanding and it is the one recommended for porting the game as a web application (game will be run on a web browser). The second one will make the game look nicer and support higher resolutions, but it may cause performance issues if played on a web browser.

I will be using OpenGL3 as I did not intend to upload the game for web browsers. The advantage I would have by porting it as a web app would be that it would not require an installation for it to be ran. It would be easier to distribute as if I compiled the game to an .exe file format, most computers would recognize it as malware. This however is not a problem because I will publish the game on a website called itch.io so that it can be downloaded by anyone. It is a free platform for developers to share their game without paying a cost upfront (like Steam, which is also on desktop). All games uploaded to the website are scanned for potential malware that they can contain, therefore players feel safer to download games from it, rather than me distributing it myself (for example hosting a website for the game’s installation).

The required hardware is minimal for both desktop and mobile. Desktop will only need a mouse to play, but there will be keyboard shortcuts implemented. Mobile will only need touch controls. This is essential because as it is a casual game, users should not be required to plug in extra peripherals.

## SUCCESS CRITERIA

|  |  |
| --- | --- |
| **Requirement** | **Justification** |
| Vertical scrolling for infinite mode. | Since the ball is moved by gravity, this is the most intuitive perspective. |
| Drawing on the screen to create slopes. | Makes the player able to direct the ball in a certain direction or block a path. |
| Key to unlock doors. | The player cannot finish the level by going to the exit but must collect a key first. |
| Level is never bigger than the screen. | Player sees everything they need to know without needing to scroll to find the end of the level. |
| No controls require a keyboard. | Only a mouse is necessary to play the game. There will be keyboard shortcuts for adjusting settings for example, which can still be done using a mouse. |
| Mouse cannot go off the screen. | This is to not allow to draw slopes outside of the screen, which can possibly create bugs with the generation of next levels. |
|  |  |

# B. Design

<See H446-03 Project Advice Booklet for help and guidance of what must go here.>

## Systems diagram

# C. Developing the coded solution (“The development story”)

<See H446-03 Project Advice Booklet for help and guidance of what must go here.>

# D. Evaluation

<See H446-03 Project Advice Booklet for help and guidance of what must go here.>

# Project Appendixes

Insert as many project appendixes as you need for your project.

These might include, but are not limited to:

* Complete Code Listing (ESSENTIAL)
* Interview Transcripts
* Meeting notes
* Observation notes or questionnaires