Abstract

This document is will outline the idea of my game for the CPD assignment.

The Good Robot  
Cross Platform Development

Author: Logan Ryan

# Change Log

Updates made to the document should be described below.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date of change | Description |
| 0.0 | AIE | 31/08/2020 | Initial Template created |
| 0.1 | Logan Ryan | 31/08/2020 | Filled in most of the template. |
| 0.2 | Logan Ryan | 4/09/2020 | Added feedback from today’s session. |
| 0.3 | Logan Ryan | 10/09/2020 | Added feedback from today’s and yesterday’s session. |
| 0.4 | Logan Ryan | 11/09/2020 | Updated Asset table, game objects and their logic, level structure. |
| 0.5 | Logan Ryan | 14/09/2020 | Updated Asset table, contents page. Added more interface screens to the interface section. |
| 0.6 | Logan Ryan | 16/09/2020 | Updated game objects and their logic, interface screens, and asset table. |
| 0.7 | Logan Ryan | 17/09/2020 | Added console controls. |
| 1.0 | Logan Ryan | 18/09/2020 | Updated Asset table. |

Contents

[Change Log 1](#_Toc50980491)

[1.0 Development Environment 3](#_Toc50980492)

[1.1 Game Engine 3](#_Toc50980493)

[1.2 Source Control 3](#_Toc50980494)

[1.3 Third-Party Libraries / assets 3](#_Toc50980495)

[2.0 Game Overview 3](#_Toc50980496)

[2.1 Genre 3](#_Toc50980497)

[2.2 Camera Perspective and Movement 3](#_Toc50980498)

[2.3 Platform 4](#_Toc50980499)

[2.4 Technical Goals 4](#_Toc50980500)

[2.5 Game Objects and Logic 4](#_Toc50980501)

[3.0 Controls 4](#_Toc50980502)

[3.1 Windows / Web 4](#_Toc50980503)

[3.2 Android / Touch 4](#_Toc50980504)

[4.0 Mechanics 5](#_Toc50980505)

[4.1 Hazards 5](#_Toc50980506)

[4.2 Obstacles 5](#_Toc50980507)

[4.3 Items / Collectables 5](#_Toc50980508)

[5.0 Graphics 5](#_Toc50980509)

[6.0 Audio 5](#_Toc50980510)

[7.0 Game Flow 5](#_Toc50980511)

[7.1 ‘Mission’ / ‘Level’ structure 5](#_Toc50980512)

[7.2 Objectives/Goal 5](#_Toc50980513)

[10.0 Interface 6](#_Toc50980514)

[10.1 Menu 6](#_Toc50980515)

[10.2 UI/HUD 7](#_Toc50980516)

[10.3 Game Over 8](#_Toc50980517)

[10.4 Pause 9](#_Toc50980518)

[10.5 Victory 10](#_Toc50980519)

[11.0 Progress report and feedback Meeting Minutes 12](#_Toc50980520)

[Friday 4th September 12](#_Toc50980521)

[Wednesday 9th September 12](#_Toc50980522)

[Thursday 10th September 12](#_Toc50980523)

[Friday 11th September 13](#_Toc50980524)

# Development Environment

## Game Engine

I will be using Unity version 2019.3.6f1 for this project. The reason I am using this version of unity is because this is the version that was provided to me from the college.

## Source Control

Link to github repo: <https://github.com/loganryan01/AIECPDAssignment>

## Third-Party Libraries / assets

|  |  |  |
| --- | --- | --- |
| Asset Name  License | Url | Reason for use |
| **Tiny Robots Pack** | https://assetstore.unity.com/packages/3d/characters/robots/tiny-robots-pack-98930 | Character asset use for main player and enemies in game. |
| **Snaps Prototype | Sci-Fi / Industrial** | https://assetstore.unity.com/packages/3d/environments/sci-fi/snaps-prototype-sci-fi-industrial-136759 | Asset use for environment |
| **20 pence piece put down on surface** | https://www.zapsplat.com/music/20-pence-piece-put-down-on-surface/ | Sound effect used when player collects a coin |
| **Game music – action, mildly aggressive, electro breakbeat, growling bass synth, electronic rhythms** | https://www.zapsplat.com/music/game-music-action-mildly-aggressive-electro-breakbeat-growling-bass-synth-electronic-rhythms/ | Music used for the main level of the game |
| **Game music – repetitive dark dramatic action music bed, pounding drums and a chugging bass** | https://www.zapsplat.com/music/game-music-repetitive-dark-dramatic-action-music-bed-pounding-drums-and-a-chugging-bass/ | Music used for the main menu |
| **Game music – repetitive dark mysterious music bed with an airy, lightly tense style** | https://www.zapsplat.com/music/game-music-repetitive-dark-mysterious-music-bed-with-an-airy-lightly-tense-style/ | Music used for the victory scene |
| **Science fiction robot death** | https://www.zapsplat.com/music/science-fiction-robot-death/ | Music used for when the player or enemies die |
| **Designed electricity spark or arc** | https://www.zapsplat.com/music/designed-electricity-spark-or-arc/ | Music used for the lightning on the electric fences |
| **Joystick Pack** | https://assetstore.unity.com/packages/tools/input-management/joystick-pack-107631 | Joystick used to allow player to move on mobile. |

# Game Overview

## Genre

The genre for this game will be a 3D Platformer game

## Camera Perspective and Movement

The camera will follow behind the player throughout the game. The camera will not move up, down, left or right. It will only move forwards or backwards.

## Platform

The platforms that the game will be deployed to is Windows, Web, Android. To deploy to windows, I will need to set the build settings of unity to target Windows platform, same with web but with Android I will need to download the necessary tools so Unity can build a .apk file for Android to launch the game. Once the files are downloaded, I will need to go into unity preferences and set the tools that are used to build the .apk file to the tools that were downloaded.

## Technical Goals

* Implement basic player movement.
* Implement basic platformer gameplay mechanics.
* Implement sound effects and particle effects.

## Game Objects and Logic

* Evil Robots – These robots are the main enemy in the game. They will move left and right in this game and they can be destroyed if the player either jumps on them or punches them.
* Boxes – The boxes are what the player can destroy to earn coins.
* Walls – These will stop the player from going off screen.
* Lights – These will light up the level so the player can see.
* Teleporter – This will be the finish line for the level.
* Electric Poles – These will shoot electric sparks back and forth every 5 seconds to provide the player a challenge with this game.
* Moving platform – These platforms will move forward and backwards, left and right.
* Dropping platform – These platforms will drop after 2 seconds when the player touches the platform.
* Checkpoints – These will let the player respawn at their location if the player touches it.

# Controls

## 3.1 Windows / Web

* The arrow keys will control the player movement.
* The spacebar will allow the player to jump.
* The left shift button will allow the player to attack
* The escape button will allow the player to pause the game

## 3.2 Android / Touch

* To move the player, their will be a joystick for the player to use.
* There will be two buttons that player can push to jump and attack.
* There will be a pause button in the top right corner for the player to pause to pause the game.

## 3.3 Console

* To move the player, the player will use the left joystick.
* The A button will let the player jump.
* The B button will let the player attack.
* The Start button will pause the game.

# Mechanics

* **Jumping**

When the player presses the spacebar, the player will jump until it’s head just touches the top edge of the player’s view.

* **Attack**When the player presses the left shift key, the player’s arms will spin 180o and back to destroy enemies or boxes if the player is close enough to them.

## Hazards

* Jump gaps
* Electric poles

## Obstacles

There will be evil robots throughout the game that will destroy the player if the player doesn’t jump or attack them. There will also be platforms that move side to side that the player has to jump on and that will drop after 2 seconds when the player touches it.

## Items / Collectables

There will be coins for the player to collect in this game. Coins can be collected by player collision or boxes being destroyed.

# Graphics

The graphics used in this game is third person 3D because that is a popular graphic feature used in platformers.

# Audio

There will be sound effects when the player dies, an enemy has been destroyed, box has been broken, when a coin is collected and when the player reaches the teleporter.

# Game Flow

## ‘Mission’ / ‘Level’ structure

The level will be a straight line with obstacles in the way to try and stop the player from completing the game. The camera will follow behind the player when they move forward or backwards.

## Objectives/Goal

The player’s goal is to reach the teleporter that is at the end of the level.

1. Levels

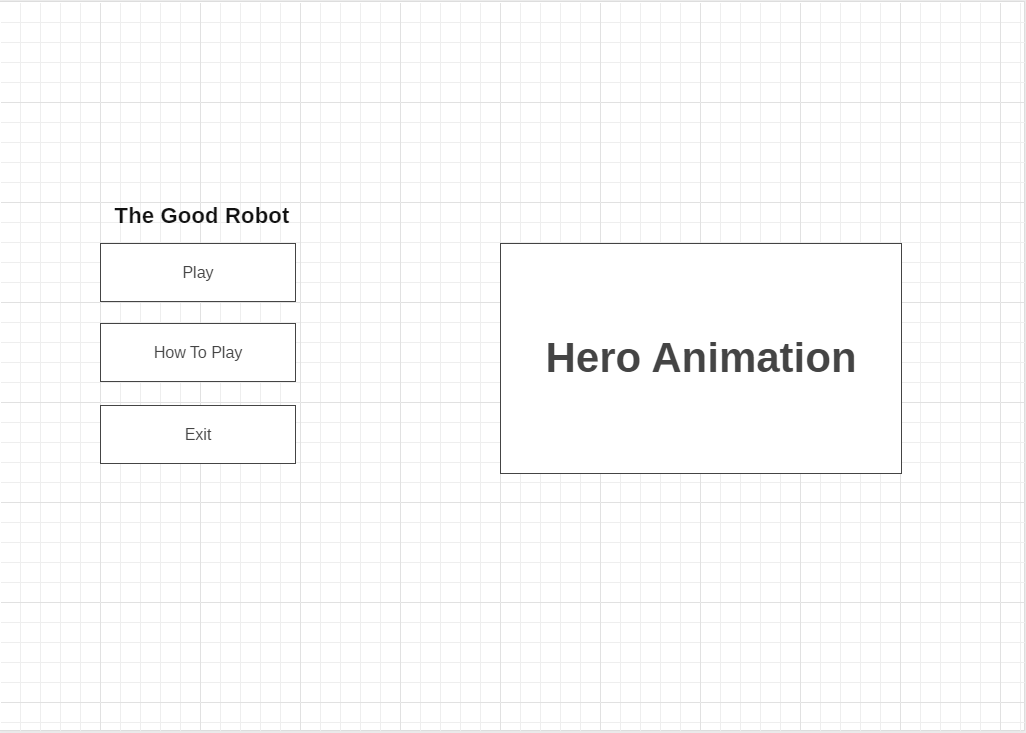
The level was built manually by me with close reference to the 1st level of the first Crash Bandicoot game.

1. Items

The coins will increase the player’s score by 1.

# Interface

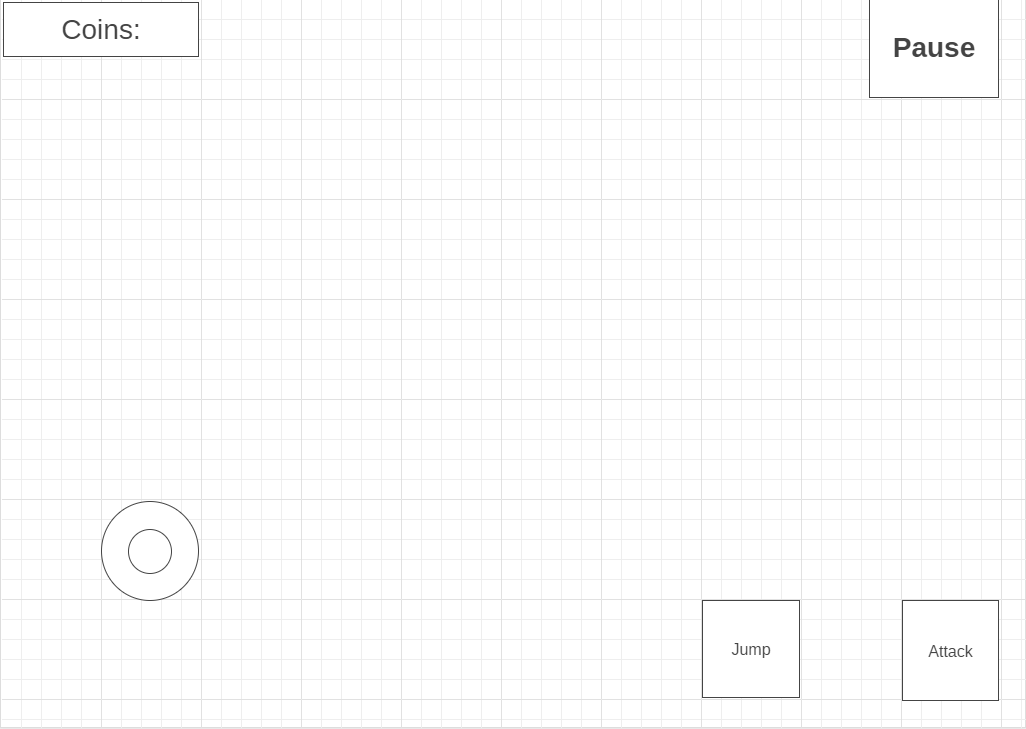
## Menu



Menu buttons

* Play – starts the game
* How to play – shows the player how to play the game
* Exit – Quits the game
  + Not visible on web

## UI/HUD



Action buttons

* Pause button
  + Pauses the game.
* Jump button
  + Hidden on windows/web
  + Visible on mobile
  + Activates player jump
* Attack button
  + Hidden on windows/web
  + Visible on mobile
  + Activates player attack

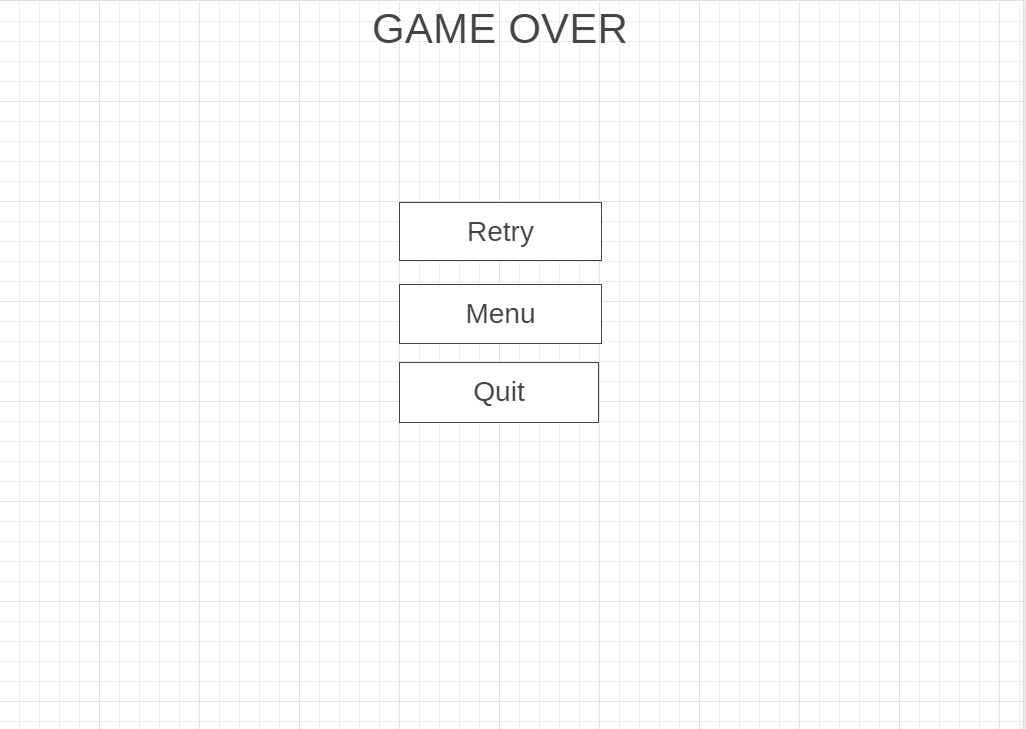
Coins

* Display number of collected coins

Joystick

* Move player
* Hidden on windows/web
* Visible on mobile

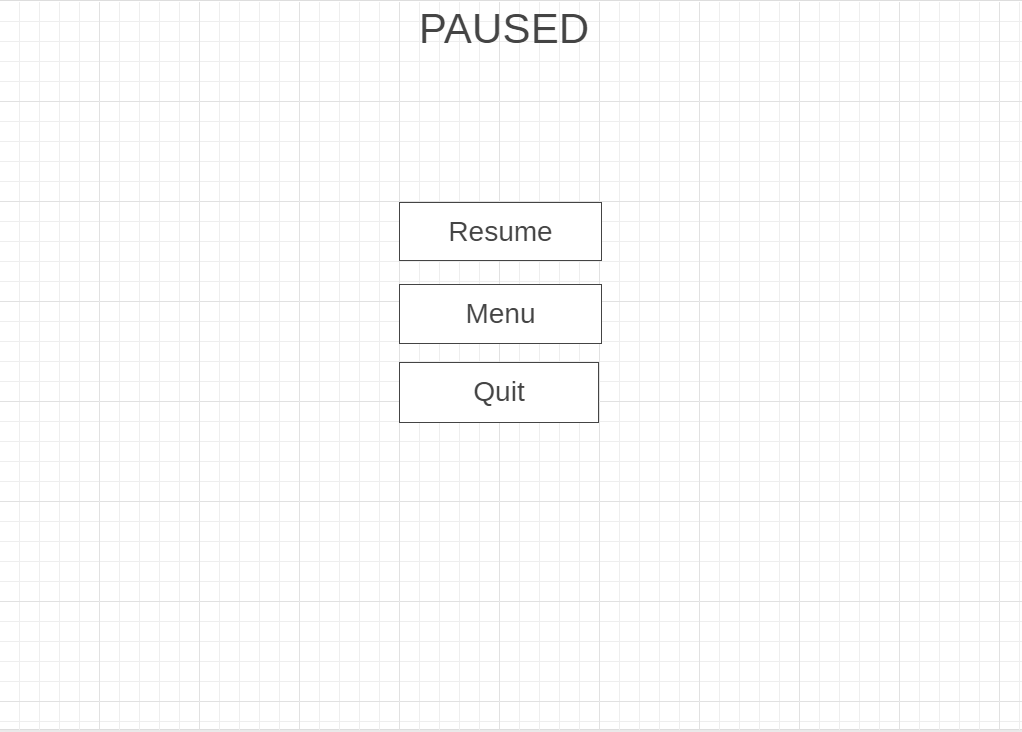
## Game Over



Action Buttons:

* Retry
  + Reloads the player at the starting position or from the last check point they touched.
* Menu
  + Returns the player to the main menu.
* Quit
  + Quits the application.

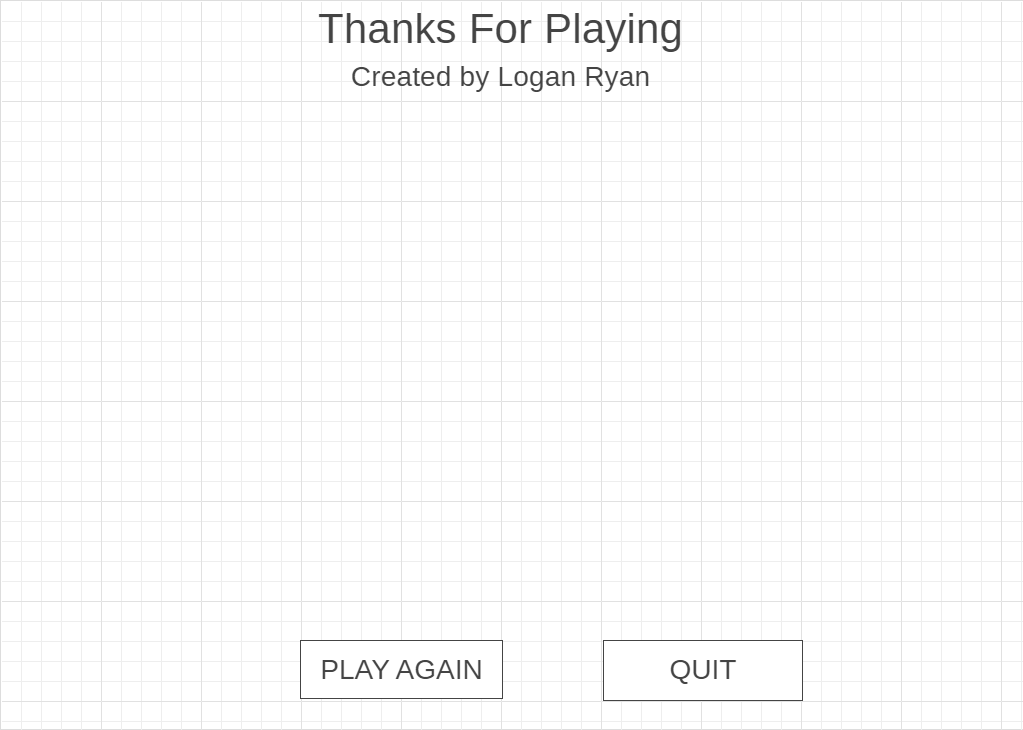
## Pause



Action Buttons:

* Resume
  + Resumes the game.
* Menu
  + Returns the player to the main menu.
* Quit
  + Quits the application.

## Victory

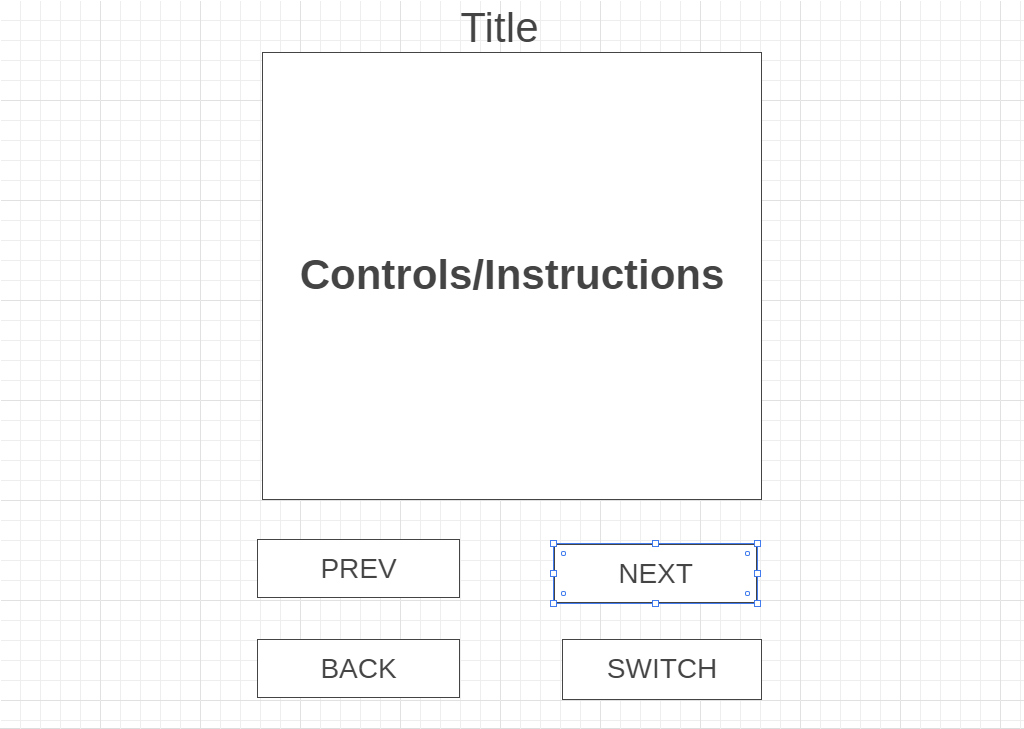


**HERO ANIMATION**

Action Buttons:

* Play again
  + Reloads the level from the start
* Quit
  + Quits the application.

## How to Play



Action Buttons:

* Back
  + Go back to the main menu
* Switch
  + Switch between the controls and instructions menu
* Prev
  + Visible in the instruction’s menu
  + Go back one page on the instruction’s menu
* Next
  + Visible in the instruction’s menu
  + Go forward one page on the instruction’s menu

# Progress report and feedback Meeting Minutes

## Friday 4th September

Describe state of project

* Basic controls work
* Collision works
* Particle effects work

Feedback from teacher and peers:

* Try to implement a bouncing box
* Create an electric fence
* Make an auto-run level

Action Items:

* A bouncing box is a good idea as it can give the player more ways to earn coins in the game.
* An electric fence would look cool in my game and can be used to give the final obstacle a bit more of a challenge.
* I may not implement an auto-run level due to it changing the current level design, but I may consider doing an auto run level another time

## Wednesday 9th September

Describe state of project

* Textures on characters
* Main Menu
* Game Over
* Pause

Feedback from teacher and peers:

* Good pick of textures for characters
* Good demonstration of basic UI

Action Items:

* Try to implement controller input for menus
* Get a level working
* Change electrical obstacle from broken wire to electric fence

## Thursday 10th September

Describe state of project

* Victory menu
* Improved collision detection
* Wider area

Feedback from teacher and peers:

* Great game idea
* Good functionality

Action Items:

* Create a how to play menu
* Add lightning to the top of the electric fence to let the player know that they cannot jump over the electric fence

## Friday 11th September

Describe what has been done since last time

* How to play menu
* Moving and dropping platforms

Feedback from teacher and peers:

* Are you going to use textures for the environment?

Action Items:

* Fix the collision on the platforms
* Implement the textures on the environment.