# **SCRIPT DESIGN – Simple Ball Game Matthew Wakely**

YELLOW TEXT IN PURPLE = INSTRUCTIONS, REMOVE ENTIRELY ONCE FOLLOWED.

BLACK TEXT IN YELLOW = STUFF TO FILL IN, REPLACE WITH ANSWERS/INFO.

Contents

[SCRIPT DESIGN – Simple Ball Game 1](#_Toc69075429)

[Document overview 4](#_Toc69075430)

[Project Script Requirements 4](#_Toc69075431)

[Description 4](#_Toc69075432)

[Feature & Mechanics List 4](#_Toc69075433)

[Key Scripts to Design 4](#_Toc69075434)

[Script Design 5](#_Toc69075435)

[Script 1 - Goal 5](#_Toc69075436)

[Required Functionality & Outcomes 5](#_Toc69075437)

[Pseduocode 5](#_Toc69075438)

[Flowchart 5](#_Toc69075439)

[Script 1 Plan feedback 5](#_Toc69075440)

[Pseudocode feedback notes 5](#_Toc69075441)

[Flowchart feedback notes 5](#_Toc69075442)

[Script 1 Revised Plans 5](#_Toc69075443)

[Final pseudocode 5](#_Toc69075444)

[Final flowchart 6](#_Toc69075445)

[Script 2 – Game Manager 6](#_Toc69075446)

[Required Functionality & Outcomes 6](#_Toc69075447)

[Pseduocode 6](#_Toc69075448)

[Flowchart 6](#_Toc69075449)

[Script 2 Plan feedback 6](#_Toc69075450)

[Pseudocode feedback notes 6](#_Toc69075451)

[Flowchart feedback notes 6](#_Toc69075452)

[Script 2 Revised Plans 6](#_Toc69075453)

[Final pseudocode 6](#_Toc69075454)

[Final flowchart 7](#_Toc69075455)

[Script Implementation & Iteration 7](#_Toc69075456)

[Script 1 - Goal 7](#_Toc69075457)

[Script(s) generated 7](#_Toc69075458)

[Functionality review 7](#_Toc69075459)

[Implementation feedback notes 7](#_Toc69075460)

[Response to feedback notes 7](#_Toc69075461)

[Script 2 – Game Manager 8](#_Toc69075462)

[Script(s) generated 8](#_Toc69075463)

[Functionality review 8](#_Toc69075464)

[Implementation feedback notes 8](#_Toc69075465)

[Response to feedback notes 8](#_Toc69075466)

Document overview

The purpose of this document is to show the planning, design, implementation of at least two scripts in a single project. The document is broken up into phases to go through in order.

Project Script Requirements

### Description

## This is an arcade game where the aim is to score goals in the quickest time possible with more points given the quicker the player scores a goal. The goal posts will reset their location after each goal.

### Feature & Mechanics List

**Main Menu**

* **Start/Exit Buttons** - Two separate buttons, one to get into the game level, and one exit the game

**Game Level**

* **Player Movement** – Controlled by script in Unity Standard Asset Pack
* **Goal** - A collision of the ball and the goal space, this will update score and reset the game level after a brief pause
* **Goal Spawn** – After the level resets the goal moves location within the game space, always facing towards the centre of the level

**Game Level GUI**

* **Score** – Displays the total score the player has achieved so far
* **Timer** – Displays the time the player has left to get a goal

### Key Scripts to Design

* **Goal** – OnTriggerEnter event that runs a function from the Game Manager script
* **Game Manager** – Manages the GUI including the main menu, starts the game, exits the game, gives the player a score \* the amount of time left, and resets the level while spawning the goals in a different location, runs the timer that counts down and displays the current player score carried over from the last goal

## Script Design

### Script 1 - Goal

#### Required Functionality & Outcomes

Detect if the ball has been kicked into the goal space, if it has then run the Reset function from the Game Manager script.

#### Pseudocode

If the game object tagged ‘Ball’ enters the collider

Run the Game Manager Reset function

#### Flowchart

Diagram

Description automatically generated

### Script 1 Plan feedback

#### Pseudocode feedback notes

* <Second note>
* <Third note>
* <etc>

#### Flowchart feedback notes

* Have a start state
* <Second note>
* <Third note>
* <etc>

### Script 1 Revised Plans

#### Final pseudocode

SHOW THE FINAL PSEUDOCDE HERE.

#### Final flowchart

SHOW THE FINAL PSEUDOCDE HERE.

### Script 2 – Game Manager

#### Required Functionality & Outcomes

Exit the game & start the game.

Deactivate and activate the main menu canvas.

Start gameplay within the game level.

Update the GUI to increase the score and reset the timer, spawn the goalposts in a different location, move the player back to the centre of the game world with the ball in front of them.

#### Pseudocode

{Exit game button pressed

Close application}

{Start button pressed

Deactivate the menu canvas

Activate Timer/Score Canvas

Run Game Coroutine}

{Game Coroutine

Activate Ready Canvas

Wait for seconds (~3), deactivate Ready canvas and then allow the player to move and start the timer

If the timer runs out

Game over screen and return to main menu}

{Reset Game

Add score and reset timer

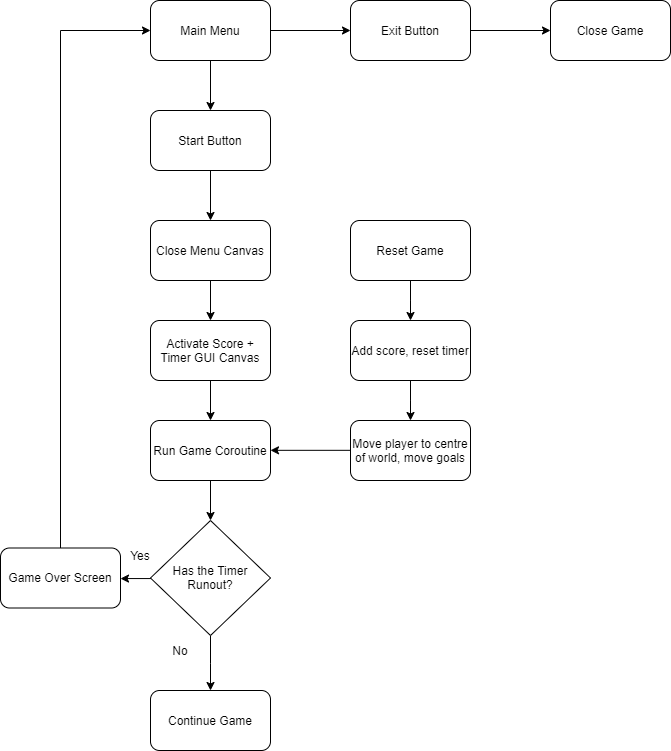
Move player to centre of world

Select location from array

Place goals in selected location facing central game object

Run Game Coroutine}

#### Flowchart



### Script 2 Plan feedback

#### Pseudocode feedback notes

* <First note>
* <Second note>
* <Third note>
* <etc>

#### Flowchart feedback notes

* Reset function must occur outside normal functions
* <Second note>
* <Third note>
* <etc>

### Script 2 Revised Plans

#### Final pseudocode

SHOW THE FINAL PSEUDOCDE HERE.

#### Final flowchart

SHOW THE FINAL PSEUDOCDE HERE.

## Script Implementation & Iteration

IN THIS SECTION YOU ARE SHOWING THE OUTCOME OF YOUR ATTEMPTS TO CODE THESE THINGS. SHOW YOUR FIRST WORKING OUTCOME THE

### Script 1 - <Name of script/function>

#### Script(s) generated

SHOW THE CODE THAT WAS GENERATED HERE (PASTE ENTIRE CODE OF SCRIPT)– IT NEEDS TO HAVE AT LEAST MOSTLY ACHIEVED THE GOAL. IF THIS IS OVER MULTIPLE SCRIPT FILES IN IMPLEMENTATION, SHOW BOTH AND MAKE SOME BULLET NOTES ON HOW THE SCRIPTS INTERACT.

#### Functionality review

<State if the script functioned as desired, and also if there were any issues or shortcomings apparent to the developer. Bullet points if this becomes a list.

#### Implementation feedback notes

AFTER SHOWING THE OUTCOME TO OTHERS, COLLECT FEEDBACK NOTES AND LIST THEM HERE

* <First note>
* <Second note>
* <Third note>
* <etc>

#### Response to feedback notes

START BY LISTING YOUR RESPONSES AND THE REASONING BEHIND THEM. THEN PASTE THE FINAL SCRIPT(S) BELOW.

* <First response (“Did X to achieve Y”)>
* <Second response>
* <etc>

FINAL SCRIPT(S) PASTESD HERE .

### Script 2 - <Name of GUI script/function>

#### Script(s) generated

SHOW THE CODE THAT WAS GENERATED HERE (PASTE ENTIRE CODE OF SCRIPT)– IT NEEDS TO HAVE AT LEAST MOSTLY ACHIEVED THE GOAL. IF THIS IS OVER MULTIPLE SCRIPT FILES IN IMPLEMENTATION, SHOW BOTH AND MAKE SOME BULLET NOTES ON HOW THE SCRIPTS INTERACT.

#### Functionality review

<State if the script functioned as desired, and also if there were any issues or shortcomings apparent to the developer. Bullet points if this becomes a list.

#### Implementation feedback notes

AFTER SHOWING THE OUTCOME TO OTHERS, COLLECT FEEDBACK NOTES AND LIST THEM HERE

* <First note>
* <Second note>
* <Third note>
* <etc>

#### Response to feedback notes

START BY LISTING YOUR RESPONSES AND THE REASONING BEHIND THEM. THEN PASTE THE FINAL SCRIPT(S) BELOW.

* <First response (“Did X to achieve Y”)>
* <Second response>
* <etc>

FINAL SCRIPT(S) PASTESD HERE .