Kimberley Collins

MSD 6.1B

Abstract

Game Documentation

Assignment 2 – Runner Game

Object Oriented Programming

Contents

[Section 1 – Game Design 2](#_Toc93853810)

[Section 2 – Functionality Requirements 6](#_Toc93853811)

[Section 3 – Code Requirements 7](#_Toc93853812)

[Section 4 – The Final Game 7](#_Toc93853814)

[Git Hub Link 9](#_Toc93853815)

## Section 1 – Game Design

1. Prepare a Game Design Document, which outlines the following:
   1. Game Title and Description
   * Game title for the game is Runner. The game concept of this game is to create a player that needs to collect coins as much as possible in a few seconds and while facing the obstacles without touching them so the player lives and health won’t decrease and the player will end up losing. Throughout the game, the player is going to face obstacles, the player needs to run away from the obstacles which they will consists of different strengths. If the player hit a coin, the score will increase. At the end of the game, the game will show the player how much score and high score does the player have. To be able to move the main character who is a man, the user need to click on the four-arrow key of the keyboard which are the top, bottom, left and right keys and the user need to click on the space bar to make the man jumper higher to collect as many coins as possible in a few seconds because a timer will be applied to the game.
   1. Choose and describe a Game Type (Single Screen, Platformer, Scroller, Side-Scroller, Adventure)
   * The genre of this game is going to be a running and jumping game which is going to be developed for computer devices. It is going to be a 2D platformer game which contains a scrolling background which is intended for kids who are six years old and more.
   1. Game Loop
      1. What will the player be doing in your game?
      2. How will they do it?
      3. How does the player progress through the game?
      4. How is the narrative delivered?
   * For the first level the user will have eight seconds of chance of playing the game while the second level the user will have ten seconds of chance playing the game. The objective of this game is to avoid the enemies and collect many coins as much as possible. In the first level in the second level the player will also need to collect coins as much as possible in few seconds and try to avoid the obstacles in level 2. If the user hits the obstacles the health lives and points will be deducted because each obstacle will have a different power and so according to the enemy’s power, the points will be deducted. To collect the coins and show the points according to the coin collected, the player will have a box collider2D and the coins will have a circle collider 2D, which they will be triggered. As soon as the player and the coin, touches each other, the coin will be destroyed because it is being triggered and the points are increasing according to the coin. If the score is 0 and the health lives are 0 the game scenes will finish, and it will go to the end scene which it will show the player’s points and highscore. As soon as the timer finishes in level 1, the user will proceed to level 2 by using the singleton pattern. And when the timer finishes in level 2, the player will proceed to the end scene, so the user can see how many coins the user collected and the user’s highscore. As soon as the user clicks on the application, the user can be able to see a UI menu with three multiple options Play, End and Quit buttons. The Play button will take the user to level 1 of the game, the End button will take the user to the end scene of the game and show him a UI menu with buttons and the current score and highscore and the quit button, it will exit from the application. And in the end scene, the user can also make use of the UI menu, which it consists of three buttons Welcome, Level 1 and Quit buttons. The Welcome button will take the user to the Welcome page and can use to play, see score and highscore or play the game again or quit the game. Level 1 button will take the user to play the game again and quit button, it will exit from the application. The narrative is delivered by it keeps the audience to continue playing until, they collect every coin without touching the enemies and before the timer ends.
   1. Scope
      1. How long is the game?
      2. How many levels are there?
      3. What is the average play time?
      4. What are the objectives?
      5. How many playable characters?

* The game is about 25 seconds long which consists of a welcome scene, two levels, and the end scene. The Runner Game consists of two levels, which the second level is a little bit difficult than the first one. Average play time is 20 seconds. The objective of this game is to avoid the enemies and collect many coins as much as possible. In the first level in the second level the player will also need to collect coins as much as possible in few seconds and try to avoid the obstacles in level 2. If the user hits the obstacles the health lives and points will be deducted because each obstacle will have a different power and so according to the enemy’s power, the points will be deducted. There will be four different characters, which they are the player itself, and three enemies. One of the enemies will be fixed and the other two enemies will move according to the position of the player. To collect the coins the player, need to use the keyboard because this game is designed for computers and so the user needs to press the four arrow keys and the space bar to make the player jump higher. The left and right arrow keys will be used so that the user can be able to move the player to right and left side of the game scene without exceeding the camera main points. They way the characters are designed, they can be manipulated at any time by making an instance of the original characters, making them prefabs or change the characters from their settings. Especially the enemies, the enemies are created so that the developer can change their power from the enemy’s inspector or from the script. The three enemies will be three different spaceships.
* Game Mechanics
* The rules are that the player need to collect as many coins as possible without touching the enemies in a few seconds. The player will have a rigidbody2D of body type dynamic which it had a gravity scale 0, so the player can move and jump between the camera positions and a box collider so when the player collides with the coin which it will have a circle collider2D, the coin will be destroyed. Three enemies will also have physics, one of the enemies will have a rigidbody2D of type static which it will have a fixed position at the centre but it will rotate in its place according to where the player is, if the player is at the top left side, the enemy face at the top left side, the other two enemies will also have a body type of rigidbody2D dynamic of gravity scale 0, so when the game starts these enemies will not fall and so they can move and rotate towards the player in the camera main positions, so the effects can be seen realistic. The moving enemies will be inherited from the fixed enemy, to avoid duplicates of code, and so they can use the function LookRotation() to move towards the player. In the welcome scene and in the end scene, a UI menu is going to be shown to the user. which it consists of three buttons Welcome, Level 1 and Quit buttons. The Welcome button will take the user to the Welcome page and can use to play, see score and highscore or play the game again or quit the game. Level 1 button will take the user to play the game again and quit button, it will exit from the application. Each button is going to consists an event listener and a trigger, the event listener is going to be the AddListener, add each button is going to have a trigger with the function onClick(), so when the user clicks on a particular button this will take the user to a particular page. To be able to go to different scenes a library from the unity management will be need which is the using UnityEngine.SceneManagement;. If the user is going to hit an enemy in level 2, his lives will be reduced and he will only have two chances, if the live is 0, then it will proceed to the end scene to show him the highscore, score and a menu to choose from. The game economy is the virtual economy that figures all the game loops in the game such as time loops, levels, and currencies. It will structure different character’s behaviours within the game and will balance on how much the user need to invest in every element, so every user can be rewarded accordingly when winning or losing. The game options are that the player can choose whether he starts playing, quit the game or see the previous score if he has already played the game. As soon as the users clicks on the application, the user can be able to see the welcome page, from it he can be able to choose from a menu which conists of three buttons. If the user presses the play button it will take the user to the level 1 game and then to the level 2 game and then to end scene which also has a menu with three buttons. The player will consist of two animations which are using state machine system and with a sprite sheet. The state machine system will have an idle animation then, the player will start jumping and it will continue to repeat the idle and jumping animation. As an interface HUD is going to be created which is going to be consisted of score, lives and highscore. The type of camera that I am going to use is orthographic projection of size 5 throughout the whole the game. When the player is going to collide with a coin, a sound effect is going to be heard, so the user will know that he collides with a game object that it is going to earn points from it. The Runner game is going to be created for a computer device, which is going to be developed through unity and the scripts through visual studio.
  1. Art Style / Assets
     1. Include a description of your art style and supplement with art concepts or inspirational concepts. Describe your style for the environment, characters, UI, etc. You can also link to a different area/scene it lives in.
* The art of style that I used was like a cartoon style because this game is intended for children who are 6 years old and more. The colours were a little bit vibrant so, the children will enjoy playing the game, which therefore it makes them feel happy. The player who is a man and the enemies which are spaceships are designed for children, so the game could be more engaged with them and can be easily understandable by children and not make them confuse or scared when they see certain elements. The UI menu in the welcome and end scenes is designed simple so, kids will know where to go and when they hover on a particular button, this will change the background colour to white and the text to black.

A group of hot air balloons in the sky

Description automatically generated with medium confidence

* Inspirational References are Temple Run (2014) and Subway Surf (2012).
* Inspirational Art References are the below images

A screenshot of a video game

Description automatically generated

Figure2: Subway Surf (2012)

Figure1: Temple Run (2014)

## Section 2 – Functionality Requirements

Design one Animation using state machine system.

Diagram

Description automatically generated

Graphical user interface

Description automatically generated

Sprite animation diagram for the coin

Diagram

Description automatically generated

## Section 3 – Code Requirements

## 3.4 Research and describe Exception Handling

In Object-Oriented Programming, exception handling is a dominant mechanism for centralized processing of issues and unusual situations which let the exceptions to be thrown and caught. This type of mechanism is controlled by the Common Language Runtime (CLR) which changes the procedure-oriented method or error handling in which each method or function returns a code that suggests a successful execution or an issue. Exception handling will answer to the exceptions when the program starts running. The exception will occur when the unpredictable event appears that it needs certain processing. For instance, a developer will produce an abnormal input, when the method is trying to attempt an answer to divide by zero or else a file system error is experienced throughout the write or read a file. Exception handling can happen when the developer is trying to save the score, lives and the highscore of the player in a file during the Running game and the file has not been created to load and save the data in it.

## Section 4 – The Final Game

4.1 Design a class diagram to describe your OOP application.

Diagram

Description automatically generated

4.3 Write a short report detailing:

a. The code design pattern and OOP concepts used in your game.

Code Design Pattern

I used code design patterns because these are solutions to minimize the common problems and extra coding in many scripts, which makes the project more reusable, clean, maintainable and easy to use. Two different types of patterns that I used were structural and real-world. Structural code when using types of names as explained in the UML class diagrams. While real-world provides situations where theses patterns are going to be used. Code design patterns that I used were GameData script, which is an abstract class, GameManager script was used to take care of the game states and handle the game data, singletons and inheritance.

Game Data script which is an abstract class was used so the other scripts can be derived from this class, so the object, especially the player when is moving, the player will remain between the camera main points and so it will not exceed them. Even when the player is in Level 2 and it is moving, the enemies which are moving along the player will stay between the camera main points. To avoid duplicates of code, the game data is going to save the score, the highscore and health of the player, which from this class, so later on these can be called from the GameManager script, so a method for each one of them can be created accordingly, which later on these methods can be called from another script like the Coin script, so when the user hits the coin, the score will be added and in the enemy script, so when the user hits an enemy the health will be reduced.

Singletons were used to make sure that a class has only one instance which provides access to that instance to every script of the project else if there are more than one instance it will destroy the other instance, so the project can run smoothly without any issues, so the instance can be used according to the developer needs. This was used in the GameManager script in the private void Awake() method, to make sure that only one instance is being used, if not then destroy the other instances. Also, the single pattern was used to make the player visible from level 1 scene to the level 2 scene from the private void Awake() method, so only one player will be visible, therefore this reduce the amount of code which needs to written in a script and attach it to another player.

Inheritance was also used to avoid duplicates of code. It is used in the Enemy script which inherits from the ConstantEnemy script. To make the enemies moving with the player, the code is only written in the ConstantEnemy and then from the Enemy script make the methods public override and in these methods in the Start and Update method call the Start and Update method of the ConstantEnemy with the base.Start() or base.Update().

OOP Concepts

OOP concepts that I used were access modifiers, inheritance, polymorphism, abstract class and the getters and setters. Certain access modifiers were implemented to be public, private and protected. Public, so that I can call and amend certain attributes from other scripts, for instance public static float XMin() from the GameData script to the Player script, so the player will not exceed the camera x and y positions. Private so that, I can use them within the same class only, so I will make sure not to amend them from any other scripts and protected, I used it for the methods when a script is being inherited from another script. Inheritance was used in the Enemy script which inherits from the ConstantEnemy script. To make the enemies moving with the player, the code is only written in the ConstantEnemy and then from the Enemy script make the methods public override and in these methods in the Start and Update method call the Start and Update method of the ConstantEnemy with the base.Start() or base.Update(). Polymorphism was used in the GameManager script to change the method PlayerScore which maintains the score of the player to return a string. So, the score can be changed into text, so the user can see how many points he has. Getters and setters were used, so certain attributes can be overwritten or can be read only. For example, get { return \_PlayerLives; } set { \_PlayerLives = value; }. To store the current live of the player when he won’t hit the enemy and when the user hit the enemy, his lives will reduce, and it will be shown to the user.

Abstract class was used in the GameData script so the other scripts can be derived from this class, so the object, especially the player when is moving, the player will remain between the camera main points and so it will not exceed them.

Graphical user interface

Description automatically generated with low confidence

b. Suggestions for improvement.

To improve my game, I could make the player move his legs while he is moving on the grass, and while he is jumping to collect the coins, I could make him jumping position instead of running. Another improvement could be that while the background is scrolling and the player is moving and jumping, I could create a script which it will take care of the grass, so as soon the last grass is showing, I could keep looping and showing the grass automatically throughout the game and could add more enemies with different health. If the player touches an enemy, a health bar will be shown with and the health of the health bar will be decreased according to the enemy’s health. I could add more levels, by making them more difficult. I could add a menu with different players, so the user can choose which player the user wants to play with. Also, I can make the game 2 versus 2, so two users can play their main characters, while the other two users can play as the enemies. As soon as the user clicks on the application a background music will be played until the user exits the game.

## Git Hub Link

My github link: <https://github.com/KimColl/OOP_Kimberley_Level6_FirstYear.git>