Saint Martin's Institute of Higher Education

Computing Diploma

Assessment Brief

ASSESSMENT DETAILS	
Assessment Title	**King Kong Side-Scroller**
Programme	Diploma in Computing
Course Code	SMc10330
Course Name	Maths and Science for Computer Games
Type	Home Coursework
Sequence	1 of 1
Aim	To be able to create a Physics based game within a Game Engine
Workload	15 hours
Issue Date	Monday, 11th November 2019
Deadline Date	**Sunday, 5th January 2020**

ASSESSMENT OUTCOMES AND REQUIRED RESOURCES	
Learning Outcomes	**As a result of undertaking this coursework, the student will be able to:** • *Complete a gaming software development task within a set deadline* • *Understand the essential mathematics necessary to handle the kind of programming required to develop video games, especially within a 3D environment.* • *Familiarise with the geometry and manipulation of objects in 3D and their associated physics* • *Apply mathematical techniques to solve rendering and physics problems* • *Design basic game logic and level progression*
Required Resources	**The following list of resources are required to complete this assessment:** • **Unity Scripting Documentation:** *Rigidbody: Adding Force:* https://docs.unity3d.com/ScriptReference/Rigidbody.AddForce.html • **Unity Scripting Documentation:** *Physics: Physics Material:* https://docs.unity3d.com/Manual/class-PhysicMaterial.html • **Unity Scripting Documentation:** *Collider: OnCollisionEnter* https://docs.unity3d.com/ScriptReference/Collider.OnCollisionEnter.html • **Unity Scripting Documentation:** *Camera: World to Screen Point* https://docs.unity3d.com/ScriptReference/Camera.WorldToScreenPoint.html • **Unity Scripting Documentation:** *Camera: Screen to World Point* https://docs.unity3d.com/ScriptReference/Camera.ScreenToWorldPoint.html • **Unity Scripting Documentation:** *Animator: SetTrigger* https://docs.unity3d.com/ScriptReference/Animator.SetTrigger.html
ASSESSMENT INSTRUCTIONS (Read the following instructions carefully)	
Instructions	This assignment forms part of the graded coursework, and counts towards the final mark in this unit. Only softcopies of the final coursework are to be accepted. A link will be provided on this unit's Moodle page for students to upload their work. Failure to submit by the deadline will result in the assessment not being graded. Presentation is important and students may be penalised for lack of adequate presentation. All relevant working must be shown. The final uploaded document needs to be in **PDF format** for the documentation part of the assignment and a **ZIP file** containing all of the Unity project and a build of the game. Students are to answer **ALL** the questions set in this assignment. The respective distribution of marks for this coursework is included next to each question.

	This assignment involves programming work, research as well as self-criticism. Students will be penalised for lack of adequate referencing and citation techniques in the research.

WARNING!

Plagiarism will get your coursework annulled earning you 0 marks.

Plagiarism can be one of the following:

- *Copy and pasting someone else's work*
- *Paraphrasing someone else's work*

To avoid claims of plagiarism:

Copy and pasting someone else's work should be limited to definitions and should be enclosed in double quotation marks and cited immediately afterwards.

Paraphrasing someone else's work involves replacing words here and there with synonyms. You should either quote it as per above or write it in your own words. How would you explain it to your younger sibling? Even then, you still need to cite the source (Author, Year) and have a reference to it listed at the back of the report using the following format (not necessarily in CAPS):

AUTHOR, YEAR, TITLE, PUBLICATION, PUBLISHER, PAGES

Background

"A side-scrolling game, side-scroller, or horizontally-scrolling game is a video game in which the gameplay action is viewed from a side-view camera angle, and the onscreen characters can generally only move to the left or right. These games make use of scrolling computer display technology. The move from single-screen or flip-screen graphics to scrolling graphics, during the golden age of video arcade games and during third-generation consoles, would prove to be a pivotal leap in game design, comparable to the move to 3D graphics during the fifth generation. Although side-scrolling games have been supplanted by 3D games, they continue to be produced."[1]

Part A – Programming

Section 1: Basic game structure

The objective of this King Kong side-scroller shooter is to move the helicopter from one side of the level to the other whilst dodging or shooting down projectiles being thrown at the helicopter. When the helicopter eventually reaches Kong, the helicopter must then shoot at Kong whist avoiding his melee attacks. The level is complete once the player manages to topple Kong. Download the Unity package available on Moodle Implement the following tasks:

The Helicopter

1. One of the elements included in the Unity Package provided is that of an animated Helicopter, which can be found in the Heli1 folder (or Heli2 folder which happens to be a subfolder of Heli1). Drag this helicopter onto the scene and play the game. You should notice that the helicopter animates, however, for the purposes of this game, it is way too big. It is also facing in the wrong direction.
    - Scale the helicopter to an appropriate size and change its orientation so that it is facing the right.
    - Parent this helicopter to an Empty Game Object to preserve the world orientations.
    - Give the Empty Game Object a name which makes sense.

    [3 marks]

2. Give your parented helicopter the ability to move around the screen however it should not be able to move beyond the screen border. Hint: you would need to make use of Screen To World Point:

    https://docs.unity3d.com/ScriptReference/Camera.ScreenToWorldPoint.html

    and Wold To Screen Point:

[1] https://en.wikipedia.org/wiki/Side-scrolling_video_game

https://docs.unity3d.com/ScriptReference/Camera.WorldToScreenPoint.html

In essence, your helicopter should:

- Move up
- Move down
- Move left and
- Move right

[8 marks]

3. Give your helicopter suitable colliders. Note, you can make use of multiple colliders if you want.

[3 marks]

4. Give your helicopter the ability to fire bullets at a rate which is controllable from the inspector.

[5 marks]

The screen movement

5. Parent your helicopter with your main camera, then provide behaviour to your camera such that it would move towards the right at a specified speed for a given distance. That is, if the distance specified to travel is that of 10m and the speed of the camera is that of 1m/s then the camera (and the parented helicopter) should move together towards the right for 10s.

   Note:

   - The helicopter should still be able to do all of the movements specified earlier.
   - The distance specified is essentially going to denote the length of the level.
   - You might want to alter the skybox of your camera in order to be able to that there is some form of movement. Watch:

     (https://www.youtube.com/watch?v=VCW_T3ZDcBc)

[3 marks]

Level generation/ Game Logic

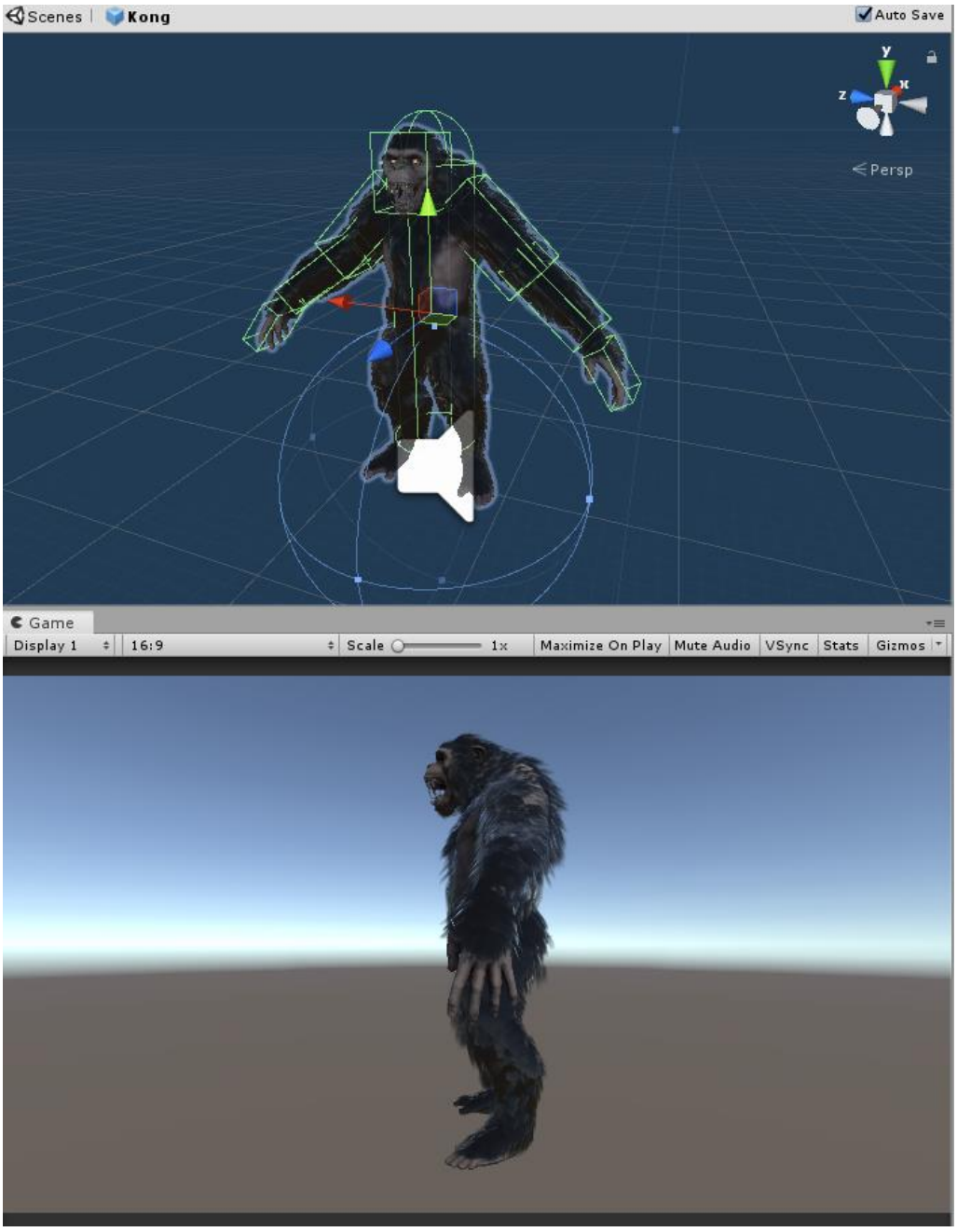
6. Use the distance value set for your screen movement (above) in order to generate a basic level. The level should be composed of a ground element with Kong placed at the end of the level.

a. Make sure that the ground generated has a collider and when the player collides with the ground, he/she should lose a life and then respawn at the side of the screen. You might find this documentation handy:

   https://docs.unity3d.com/Manual/InstantiatingPrefabs.html

   Note that when the player respawns, he/she should have one less life than before.

b. When the player collides with Kong, he/she should also loose a life. Note that the colliders for Kong have already been added as can be seen here.

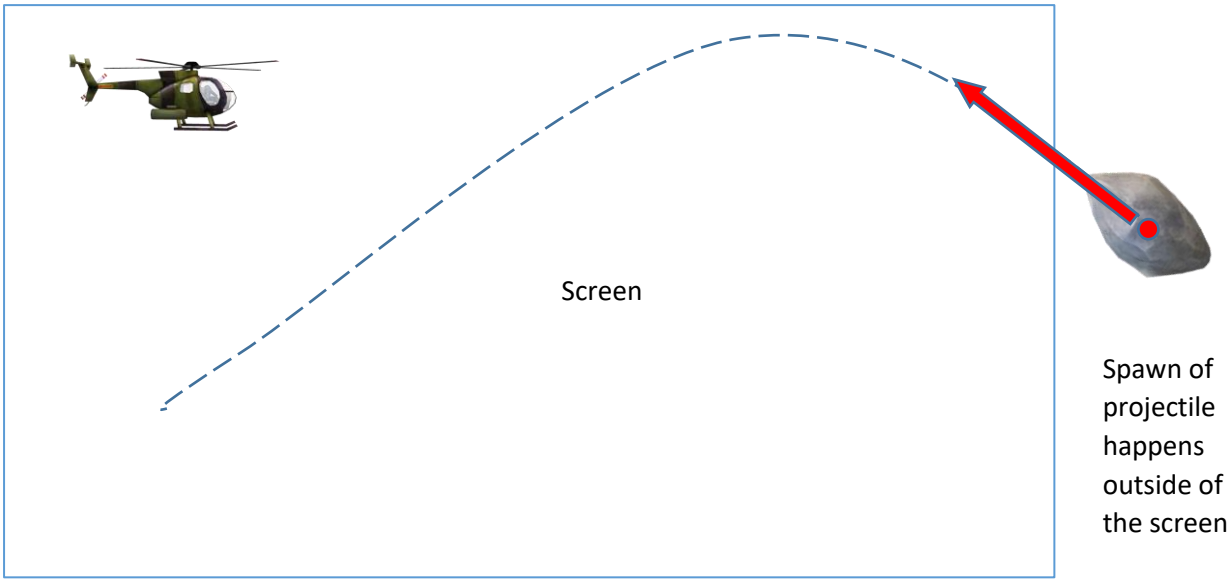


c. If the player ends up having 0 lives, then the game should end.

[10 marks]

Projectiles

Whilst Kong is off screen, one should see projectiles coming towards the player. This behaviour is depicted in the diagram below:



1. Using a controllable rate, spawn projectiles from the side of the screen. Make use of forces in order to make these projectiles move. (Hint use impulse forces).

[2 marks]

2. If the player collides with these projectiles, then he/she would lose a life and subsequently respawn if he/she has more than 0 lives.

[2 marks]

3. The projectile should destroy itself when:
    a. It hits the ground and
    b. It is hit a number of times by the player bullets.

    In either case, the player should be awarded a number of points for successfully dodging/shooting down the projectile

[4 marks]

Kong boss battle

When the player reaches the end of the level, the projectiles should stop being generated and one would enter a boss battle with Kong. Kong has a number of animations which are played when one triggers:

- Attack 1
- Attack 2
- Attack 3
- Attack 4
- GetHit 1

- Die

7. Implement a sequence with which Kong is going to attack the player. Note you should find that this https://docs.unity3d.com/ScriptReference/Animator.SetTrigger.html would help you.

[4 marks]

8. Kong should only be vulnerable to player bullets when he is in an idle state. This link should help you out https://docs.unity3d.com/ScriptReference/Animator.GetCurrentAnimatorStateInfo.html

[2 marks]

9. Kong should only show the Get Hit animation after a certain number of hit points. If Kong still has life left, he should immediately go on the attack. Otherwise he should die. If Kong dies, the level is won and a large number points is awarded to the player.

[4 marks]

Note that during this boss battle, if the player is hit by Kong at any time, then he should lose a life. If he has no more lives to lose, then the game would end there.

[Section 1 total: 50 marks]

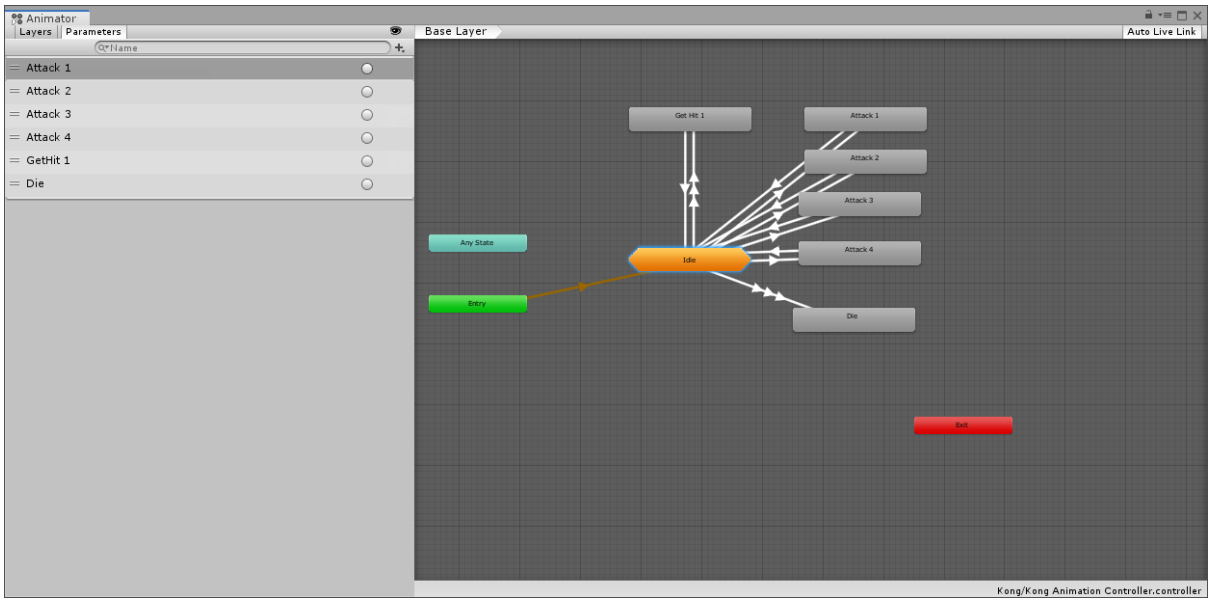
Section 2: Additional Polish … Making the game your own

Add your own personal twist and flavour to the game that you are going to create. Add as many elements that you want to the game in order to make it feel complete and polished. Here are a few ideas that you might want to implement:

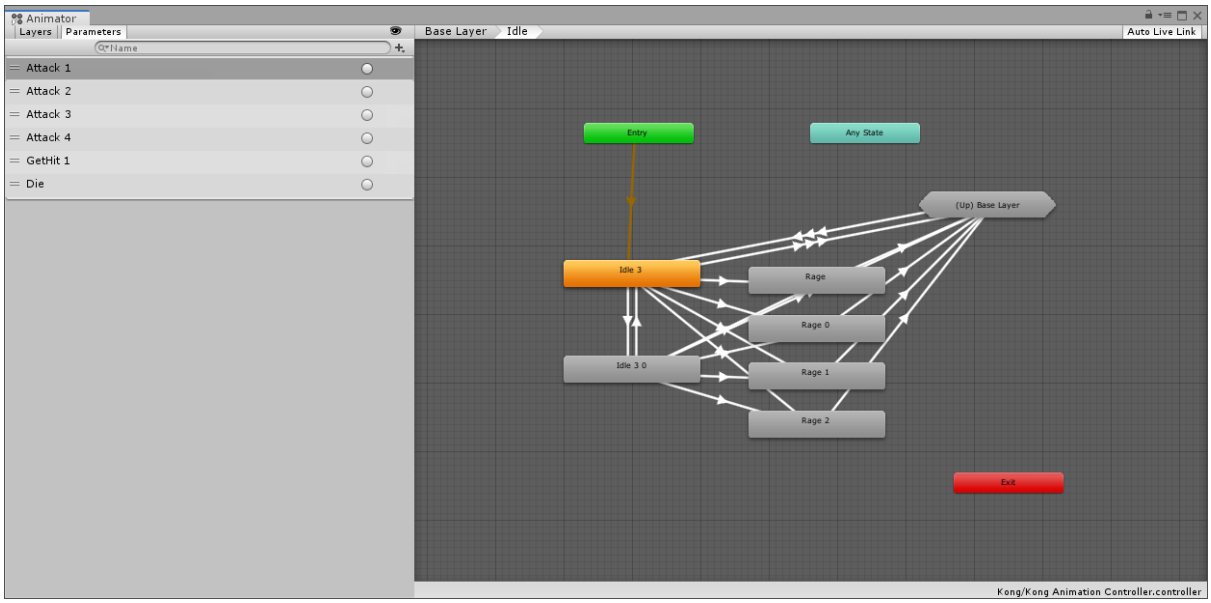
- Game UI including menu screens
- Add particle systems – such as blood effects whenever Kong registers a hit
- More obstacles to avoid when scrolling from one side of the level to the other such as buildings or different types of projectiles
- Different types of projectiles being thrown by Kong
- Add levels
- Add parallax
- Have power-ups:
    - Increasing the fire rate
    - Change the fire pattern (spread shot, double shot, triple shot etc…)
    - Add a life
    - A shield that can protect the player for a short amount of time
- Include debuffs
    - Slow down the helicopter for a short amount of time

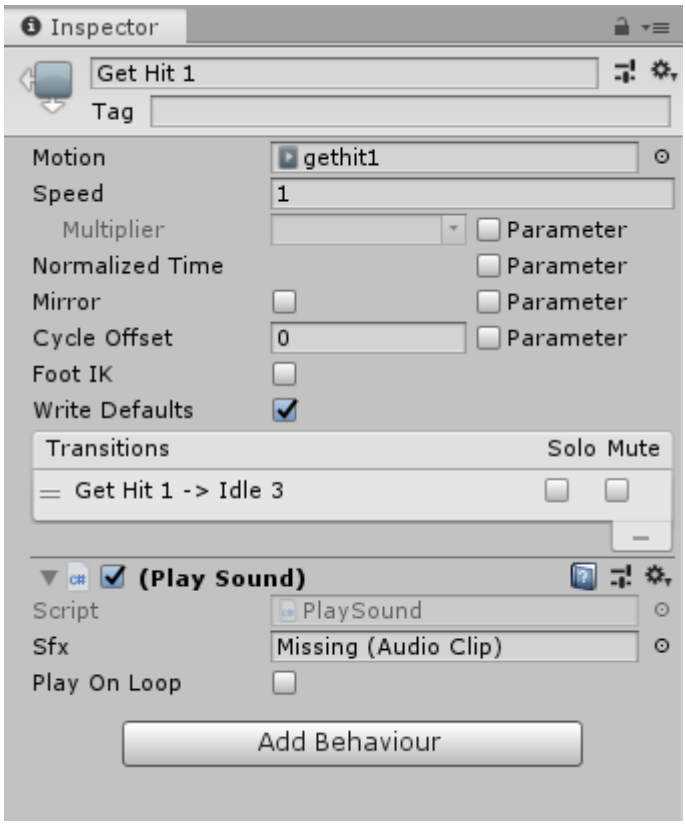
- Add music and sound effects
  - ***Note when adding sound effects to the animations provided***:

    You should have a Kong Animator Controller object within your project folder. If you open this file you should see the following:



And if you double click on the Idle state you should also see:





If you click on any state (i.e. on any rectangle), you should notice that there happens to be a "Play Sound" behaviour in the inspector. This behaviour would allow one to add sound effects so that they are played once an animation has been triggered. The behaviour also allows one to play a sound effect on loop just by selecting the check box "Play On Loop", however, if this option is not selected then the sound effect would only play once. Make use of this behaviour to add sound effects to your Kong Character.

It might also be a good idea to record your own sound effects, modify them to suit the feel of the game and then apply them to the character using this script. A useful asset that one may choose to use is this:

https://assetstore.unity.com/packages/audio/sound-fx/dehumaniser-monster-voice-rage-monster-105546

This asset allows one to modulate one's own voice to make it sound like that of a monster. You may choose to use this however it is not a requirement.

Some useful resources might include:

- https://www.assetstore.unity3d.com/en/#!/
- http://tf3dm.com/
- http://www.freesound.org/
- https://www.turbosquid.com/

Notice that the more that you add to your game the more likely it is that you would be able to obtain a good mark for this portion of the assignment.

[Section 2 total: 30 marks]

What to submit

- You need to submit a copy of your Unity Project that you have created. This should include **ALL** of the assets, scripts, art files, sound files, scenes, prefabs etc... that you have used within your game. This folder should have the same name as the project name that you have given when you created the project in the first place so make sure that you know where you have created your project.

- You also need to submit a build of your game so that one can play it without opening unity. Please take a look at this video to see how you can do so: https://www.youtube.com/watch?v=7nxKAtxGSn8

- An upload area will be created on Moodle to allow you to upload your work. It could be the case however that your game would be too large to submit via this portal. If this is the case then make sure to upload your work onto google drive (provided by Saint Martin's) and send me a shared link. The shared link should point to a folder having your Name and SM number as the folder name (e.g. **Joe Borg SM1234**). Within this folder one should find a folder containing your build files and another containing your Unity Project files. Please also include a README text file which states
    1. What unity version you made use of in order to create your game
    2. How to play your game (i.e. control scheme)
    3. The PC specs used in order to create and test this game.

[Part A total: 80 marks]

Part B – Progress documentation and self-evaluation

1. Progress documentation

Document your progress as you are developing your game. Make sure to include screenshots showcasing how your work has evolved as you were working on it.

Consider this part of the documentation to be a form of portfolio of the work that you have created. Please write a short description of what is being depicted so to give context to the reader.

Make sure to list the Unity version that you made use of to create your game.

[10 marks]

2. Self-evaluation

Write a self-critical assessment of the work you have submitted. In this write up you need to mention:

- What effects have you tried to implement
- Any difficulties that you had when working on this project and how you solved them
- A critical analysis of the work that you have provided and how you think you could improve your work given more time and/or resources
- At the end of this self-evaluation, cite all of the assets that you have made use of in the creation of your game. Make sure to use proper citation techniques. It is important to differentiate your work from that done by others. It does not mean that you cannot make use of other people's work, it just means that you should give credit where it is due.

[10 marks]

What to submit

For this portion of the assignment you are required to provide a **PDF** or **Word document** showcasing your *documented progress* and *self-evaluation*. Please make sure that your document layout makes sense and formatted in a professional manner. You should not make use of fancy fonts.

Please make sure that you provide all of the links to external assets used in your game. You can make use of Word's in-built referencing system in order to aid you. (https://support.office.com/en-us/article/add-citations-in-a-word-document-ab9322bb-a8d3-47f4-80c8-63c06779f127)

[Part C total: 20 marks]