

ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

Course Title	Advanced Diploma	Lecturer Name & Surname	NEIL AQUILINA		
Unit Number & Title	Programming for Computer Games				
Assignment Number, Title / Type	Simple 2D Car Game - Home				
Date Set	09/12/2020	Deadline Date	19/01/2021		
Student Name		ID Number		Class / Group	

<input type="checkbox"/>	<i>Student's declaration prior to handing-in of assignment:</i> <i>❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy</i>
<input type="checkbox"/>	Student's declaration on assessment special arrangements (Tick only if applicable) <i>❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.</i>
<input type="checkbox"/>	<i>❖ I declare that I refused the special support offered by the Institute.</i>
Student Signature: _____ Date : _____	

Assessment Criteria	Maximum Mark	Mark Achieved
KU2: Describe asset types	5	
KU5: Identify suitable resolution for images of various types	5	
KU6: Select asset types and settings for a range of media assets for a game design	5	
KU7: Identify resolution issues for the development platforms	5	
KU8: Identify appropriate input devices and methods for development platforms	5	
KU9: Show and explain how to deploy a game to multiple devices	5	
AA1: Examine and apply basic programming techniques for a simple game	7	
AA2: Develop a game with graphical and audio assets	7	
AA3: Demonstrate ability to resolve issues and input devices in the development environment	7	
AA4: Apply coroutines for a more interesting gameplay	7	
AA5: Examine and solve gameplay problems	7	
SE2: Resolve programming issues related to sound, graphics and gameplay	10	
Total Mark	75	

Assessor's feedback to student
(If necessary, use reverse side of page for IV feedback on assignment brief / sample of assessment decisions)



	Name & Surname	Signature	Date
Internal Verifier : Approval of <u>assignment brief</u>		For approval signature, please refer to electronic audit trail	
Lecturer / Assessor : Issue of results and feedback to student		For approval signature, please refer to electronic audit trail	
Internal Verifier : Approval of <u>assessment decisions</u> (Sample)		For approval signature, please refer to electronic audit trail	
Learner's signature upon collection of corrected assignment.			

Assessment Criteria
<i>KU2: Describe asset types</i>
<i>KU5: Identify suitable resolution for images of various types</i>
<i>KU6: Select asset types and settings for a range of media assets for a game design</i>
<i>KU7: Identify resolution issues for the development platforms</i>
<i>KU8: Identify appropriate input devices and methods for development platforms</i>
<i>KU9: Show and explain how to deploy a game to multiple devices</i>
<i>AA1: Examine and apply basic programming techniques for a simple game</i>
<i>AA2: Develop a game with graphical and audio assets</i>
<i>AA3: Demonstrate ability to resolve issues and input devices in the development environment</i>
<i>AA4: Apply coroutines for a more interesting gameplay</i>
<i>AA5: Examine and solve gameplay problems</i>
<i>SE2: Resolve programming issues related to sound, graphics and gameplay</i>

IICT4016-

Programming for Computer Games

Create a Simple 2D Car Game

Scenario: 2D Car Game

You are to create a simple 2D Car Game where your goal is to move the car left and right to avoid the obstacles found on the way, while having a scrolling background.

The game:

You should be able to control the car by moving only left and right. Obstacles will spawn and move from top to bottom at different speeds and having different damage levels. Obstacles can be other cars, stones, barriers, etc. 2 types of obstacles should also fire.

For every obstacle that your car avoids you get 5 points. When you reach 100 points you win the game.

If you get hit by obstacles, your health is reduced. If your health ≤ 0 , Game Over.

N.B.: Different tasks have different deadlines so make sure that you keep up with your deadline dates.

Always upload your project on GitHub repository and share it with your lecturer.

Task 1: Setup Play Area, Player Car, Enemy Path and Obstacle Waves

(Deadline date: Friday 8 January 2021)

- a. To avoid issues regarding development platforms, create a new Github public Repository with the name *NameAndSurnameYourGroupAssignment* and send the link to your lecturer.
Example: ***JoeBorgMSD42YAssignment***.
Create a folder *HomeAssignment* inside and put all your project files inside this folder. From the start of the assignment till the end you should have at least 12 Commits spread over at least 25 days.
If Github is not correctly used for Assignment submission, you will end up NOT SUBMITTED.
- b. Use suitable images (Sprites) for: 2D Player Car, Road background, and 5 different types of Obstacles (example: bicycles, cars, trucks, barriers, stones, etc.). Set suitable resolutions and scaling for the above images. Import all in Assets folder. Set all suitable assets and GameObjects as Prefabs.
(KU5 – 5 marks)
- c. Correctly import in the Assets folder different Audio files for:
 - Player health reduction
 - Points gained when obstacle is avoided
 - Background Music(KU6 – 1 mark)
- d. Setup the Camera to a 10:16 Aspect Ratio resolution
(KU6 – 1 mark)
- e. Using `ViewPortToWorldPoint()`, create a border around your camera
(KU6 – 1 mark)
- f. Set the background as repeated scrolling using Quad, Materials and Offset. The scrolling speed should be in line with the racing car speed.
(KU6 – 2 marks)
- g. Arrange the Assets folder into subfolders having a folder for each type of file used in your game
(KU2 – 5 marks)

- h. Create a correct Path for each type of Obstacle in (a) above (5 Paths in total). Paths should be made up of Waypoints. Paths should be set as Prefabs in a *Paths* folder. Use different Lists to save the Waypoints and the Paths for easier access.
(AA1 – 3 marks)
- i. Create an Obstacle Wave Scriptable file which contains:
- Obstacle Prefab to spawn
 - Path Prefab on which to move
 - Obstacle movement speed
 - Number of obstacles per wave
- (AA1 – 2 marks)
- j. Create a DamageDealer class. This class should be implemented with different damage output for different obstacle prefabs as follows:
- Wave 1 Obstacles: 1 damage
 - Wave 2 Obstacles: 2 damage
 - Wave 3 Obstacles: 3 damage
 - Wave 4 Obstacles: 4 damage
 - Wave 5 Obstacles: 5 damage
 - Bullets from obstacles: 1 damage
- (AA1 – 2 marks)
- k. Using coroutines and a timer your game should:
- Spawn all Obstacles in wave
 - 2 Obstacle Waves should fire a bullet
 - Spawn all Obstacle waves using Lists and foreach loop
 - Make Obstacle Waves restart from the beginning using Lists and foreach loop
- (AA4 - 7 marks)

- l. Make the player car move on the x-axis only within the border of the camera. Use Unity built-in methods to make the game frame-rate independent and to stop the car from going out of border.
(KU8 – 3 marks)

- m. Give proper Colliders or Triggers to Player Car, Obstacles, Obstacle Bullets and Obstacle Destroyer
(KU8 – 2 marks)

Task 2: Implement gameplay, sound, graphics and UI objects in game
(Deadline Date: 20 January 2021)


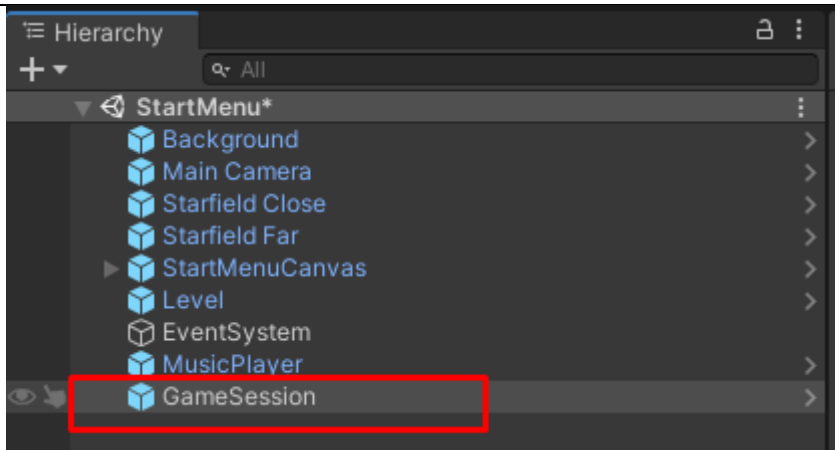
Write proper code to implement the following features in your game:

- a. Implement a *Level* script which takes care to load levels as indicated in the following features.
(SE2 – 1 mark)
- b. Start the Game from a Menu Scene with 2 Options, Play and Quit using UI. Background Music should start playing and keeps on going till the game quits. When clicking *Quit* the Game ends. When clicking *Play*, go to the Game Scene and start the Game
(AA2 – 2 marks)
- c. Player starts with 50 Health Points. If an Obstacle or a bullet collide with the Player Car, the Player should have its Health reduced depending on the Damage done by the Obstacle and a proper Sound Effect is played.
(AA2 -3 marks)
- d. When an Obstacle hits the Player Car, the Obstacle should be destroyed, an Explosion Particle effect applied and a proper Sound Effect is played.
(SE2 – 2 marks)
- e. Use Layers and Layer Collision Matrix to reduce Collision issues.
(SE2 – 1 mark)
- f. If Player Health ≤ 0 and Player Game Points < 100 , Player Car should be destroyed, an Explosion Particle Effect and a proper Sound Effect applied and load Game Over Scene with total points obtained shown.
(SE2 – 3 marks)
- g. Obstacles which are avoided by the Player, should be destroyed when they hit an Obstacle Destroyer at the bottom of the Scene. For every Obstacle avoided, the Player is awarded 5 Game Points. Avoiding bullets does not gain the Player any points.
(SE2 – 2 marks)

- h. If Player Game Points ≥ 100 , the Game should stop and a Winner Scene should be loaded
(SE2 – 1 mark)
- i. Health Points and Game Points should be shown and updated in a UI Text object
(AA2 – 1 mark)
- j. Both the Winner and Game Over Scene should have options to go to Start Menu Scene.
(AA2 – 1 mark)

Task 3: Review, Build and Deploy the Game (Deadline Date: 20 January 2021)

- a. During Development you will encounter programming issues and gameplay problems. You are to document 14 **different** errors shown by the Console, the line error where you had the problems and a short explanation on how you solved the problem. Use the Game Test Document provided to document such information. Save the Game Test Document as a PDF and upload on Github. An example is shown below:

Test Case 1	 [10:58:37] NullReferenceException: Object reference not set to an instance of an object Level.LoadGame () (at Assets/Scripts/Level.cs:21)
Line Error	<i>FindObjectOfType<GameSession>().ResetGame();</i>
Error Explanation	The Game cannot find the <i>GameSession</i> in the Scene
Error Correction	<i>GameSession</i> prefab was moved in the Hierarchy
Error Correction ScreenShot	

(AA5 – 7 marks)

b. Make sure that you have the following Modules installed in Unity Hub:

Platforms			
<input checked="" type="checkbox"/>	Android Build Support	Installed	1.1 GB
<input checked="" type="checkbox"/>	Android SDK & NDK Tools	Installed	3.0 GB
<input checked="" type="checkbox"/>	OpenJDK	Installed	70.5 MB
<input checked="" type="checkbox"/>	iOS Build Support	Installed	1.6 GB

After finishing the game, make sure that it is fully working. Then build the game to:

- PC, Mac, Linux StandAlone (.EXE file and other folders). Save the Build in the project **outside the Assets folder** in a folder named ***BuildEXE***
- Android (.apk file). Save the Build in the project **outside the Assets folder** in a folder named ***BuildAndroid***
- iOS. Save the Build in the project **outside the Assets folder** in a folder named ***BuildiOS***

(KU 9 – 5 marks)

c. As explained in Task 1a, upload everything on Github Repository (including Build folders, Game Test Document) and share the link with your lecturer. You should have at least 12 commits over a span of at least 25 days to avoid development issues. The last commit should be before the deadline date and time, as otherwise it will not be considered.

(KU7 – 5 marks)

d. After submission, you will have an interview scheduled with your lecturer and a good amount of marks is assigned to good answers based on your work (AA3 – 7 marks)

Assignment Rubric:

Criteria and tasks	Marks
KU2: Describe asset types	
Arrange the Assets folder into subfolders having a folder for each type of file used in your game	5
KU5: Identify suitable resolution for images of various types	
Use suitable images (Sprites) for: 2D Player Car, Road background, and 5 different types of Obstacles (example: bicycles, cars, trucks, barriers, stones, etc.!).	2
Set suitable resolutions and scaling for the above images.	1
Set all images as GameObjects and Prefabs.	2
KU6: Select asset types and settings for a range of media assets for a game design	
Correctly import in the Assets folder different Audio files for: <ul style="list-style-type: none"> • Player health reduction • Points gained when obstacle is avoided • Background Music 	1
Setup the Camera to a 10:16 Aspect Ratio resolution	1
Using ViewPortToWorldPoint(), create a border around your camera	1
Set the background as repeated scrolling using Quad, Materials and Offset. The scrolling speed should be in line with the racing car speed.	2
KU7: Identify resolution issues for the development platforms	
Upload Project on Github Repository (including Build folders, Game Test Document) and share the link with your lecturer. You should have at least 12 commits over a span of at least 25 days to avoid development issues. The last commit should be before the deadline date and time, as otherwise it will not be considered.	5
KU8: Identify appropriate input devices and methods for development platforms	
Make the player car move on the x-axis only within the border of the camera. Use Unity built-in methods to make the game frame-rate independent and to stop the car from going out of border.	3
Give proper Colliders or Triggers to Player Car, Obstacles, Obstacle Bullets and Obstacle Destroyer	2



KU9: Show and explain how to deploy a game to multiple devices	
Research how to build the game to:	1
PC, Mac, Linux StandAlone (.EXE file and other folders). Save the Build in the project outside the Assets folder in a folder named BuildEXE	
Android (.apk file). Save the Build in the project outside the Assets folder in a folder named BuildAndroid	2
iOS. Save the Build in the project outside the Assets folder in a folder named BuildiOS	2
AA1: Examine and apply basic programming techniques for a simple game	
Create a correct Path for each type of Obstacle (5 Paths in total). Paths should be made up of Waypoints. Paths should be set as Prefabs in a <i>Paths</i> folder. Use different Lists to save the Waypoints and the Paths for easier access.	3
Create an Obstacle Wave Scriptable file which contains: <ul style="list-style-type: none"> • Obstacle Prefab to spawn • Path Prefab on which to move • Obstacle movement speed • Number of obstacles per wave 	2
Create a DamageDealer class. This class should be implemented with different damage output for different obstacle prefabs as follows: <ul style="list-style-type: none"> • Wave 1 Obstacles: 1 damage • Wave 2 Obstacles: 2 damage • Wave 3 Obstacles: 3 damage • Wave 4 Obstacles: 4 damage • Wave 5 Obstacles: 5 damage • Bullets from obstacles: 1 damage 	2



AA2: Develop a game with graphical and audio assets	
Start the Game from a Menu Scene with 2 Options, Play and Quit using UI. When clicking <i>Quit</i> the Game ends. When clicking <i>Play</i> , go to the Game Scene and start the Game	1
Background Music should start playing and keeps on going till the game quits.	1
Player starts with 50 Health Points. If an Obstacle or a bullet collide with the Player Car, the Player should have its Health reduced depending on the Damage done by the Obstacle	2
Obstacle collision Sound Effect is played.	1
Health Points and Game Points should be shown and updated in a UI Text object	1
Both the Winner and Game Over Scene should have options to go to Start Menu Scene.	1
AA3: Demonstrate ability to resolve issues and input devices in the development environment	
An interview is held with your lecturer where you are tasked to answer any practical questions related to your work	7
AA4: Apply coroutines for a more interesting gameplay	
Using coroutines and a timer your game should: Spawn all Obstacles in wave	2
2 Obstacle Waves should fire a bullet	2
Spawn all Obstacle waves using Lists and foreach loop	2
Make Obstacle Waves restart from the beginning using Lists and foreach loop	1
AA5: Examine and solve gameplay problems	
During Development you will encounter programming issues and gameplay problems. You are to document 14 different errors shown by the Console, the line error where you had the problems and a short explanation on how you solved the problem. Use the Game Test Document provided to document such information. Save the Game Test Document as a PDF and upload on Github.	7



SE2: Resolve programming issues related to sound, graphics and gameplay	
Implement a <i>Level</i> script which takes care to load levels	1
When an Obstacle hits the Player Car, the Obstacle should be destroyed, an Explosion Particle effect applied and a proper Sound Effect is played.	2
Use Layers and Layer Collision Matrix to reduce Collision issues	1
If Player Health ≤ 0 and Player Game Points < 100 , Player Car should be destroyed, an Explosion Particle Effect and a proper Sound Effect applied and load Game Over Scene with total points obtained shown.	3
Obstacles which are avoided by the Player, should be destroyed when they hit an Obstacle Destroyer at the bottom of the Scene. For every Obstacle avoided, the Player is awarded 5 Game Points. Avoiding bullets does not gain the Player any points.	2
If Player Game Points ≥ 100 , the Game should stop and a Winner Scene should be loaded	1
TOTAL MARKS:	75