23.09.2019

Assessment 4
Complete Project

Due Dates:

Video and Unity Project Files:
Due on UTSOnline before Monday 14th November (Week 12) before 9am

Demo to Tutor:
During your lab in Week 12

Deliverables:

Unity Project Files:
<student_number>_A4_project**.zip** (NOT .7zip or .rar)

This is your entire Unity project folder, zipped up (as a .zip, no other format), with the specified naming convention. This will be submitted to UTSOnline separate to the below video file. The contents of this project for Assessment 4 are specified below.

Video File:
<student_number>_A4_video**.mp4 or .mov** (NOT AVI, MKV, or any other video format)

This is your pre-recorded demo video that is a maximum of 10 minutes and is in a .mp4 or .mov format, with the specified naming convention. This will be submitted to UTSOnline separate to the above project .zip project file. The contents of this video for Assessment 4 are specified below.

Demo:
During your lab time, you will demo your project to your tutor or the lecturer in a one-on-one conversation. Your demo time will be a **maximum of 10 minutes**. You will need to talk about the work you have done and show your Unity project, code, and Git repository. You do not need to prepare presentation slides.

Created by: Dr William Raffe
UTS CRICOS PROVIDER CODE 00099F

Task and Grade Overview:

In Assessment 4, you are expected to deliver on the recreated game that you chose in Assessment 3 and the design iteration that you promised to deliver. The purpose of this assessment is to showcase your ability to develop clean and optimized code to produce a smooth player experience and to demonstrate your knowledge by describing the work you have done.

Criteria	Details	Marks
Code Understanding	Your tutor will randomly pick sections of code that they want you to explain to them. You will be graded on the accuracy and depth of your explanation and your ability to communicate this.	**5**
Code Completeness	When the game is played, the Console Window will be examined. Marks will be deducted for any red exception errors resulting from your code and any yellow warning messages that your tutor believes you should have been able to fix (such as unused variable warnings.)	**3**
Code Optimization	Code will be examined to check for obvious optimization issues such as: • Using GameObject.GetComponent() or GameObject.Find() in every frame of the game when the result could be easily cached in a member variable. • Large loops in every frame of the game or very large loops occasionally that could cause performance issues (unless they are split across multiple frames using coroutines). • Excessive use of computationally expensive processes such as Raycasting (a small number of raycasts every frame is ok, but should be minimized where possible)	**3**
Code Commenting	Code must be commented such that it could be handed to another person and they could easily understand the logic of your code. Each method and major code block (such as significant loops and if statements) should have a detailed comment block describing their purpose and how they work.	**3**

Created by: Dr William Raffe

Keeping to Scope	The scope of the assessment must be kept to – especially in regard to using frame-rate independent motion, not using rigidbody physics or the default Unity character controller, and not using existing code libraries other than those officially provide by Unity (e.g. Unity's networking and pathfinding API's may be used in the Design Iteration level).	**3**
Git Usage	The Git repository should show significant and consistent use across multiple branches with sensible commit comments.	**3**
Quality of Experience – Menu and Recreated Level	The recreated level of the game should feel similar to the original game. It should be easy to control, provide a smooth experience, provide all needed feedback to understand what is happening in the game, and have no oblivious visual or audio bugs. The main menu scene of the game should work well, be visually appealing, and allow the player to select the Recreated Level, the Design Iteration level, or exit the game.	**4**
Quality of Experience – Design Iteration	The design iteration level of the game should provide an interesting and smooth experience beyond the original game. The complexity of the design iteration shows that students have put significant effort into developing their knowledge further in a way that would help with future game development projects. This is relative to the complexity of the game chosen to be recreation.	**5**
Art and Audio Completeness	The visual and audio assets in both the Recreated level and the Design Iteration level should feel complete, with no obvious missing visual or audio feedback in the game and only royalty free audio used. You should aim to improve on any feedback given during Assessment 3 regarding these assets.	**3**
Pre-Recorded Demo Video	You must provide a pre-recorded version of your demo (screen capture that is spoken over). This is a maximum of 10 minutes (similar to your in-class demo) and should show aspects of the above, including: • The game being played with the console window open.	**3**

	• What your Design Iteration level is and how it improves your knowledge. • A discussion of complex bits of code that you are proud of. • Showing your Git repository tree and talking about how it was used during development. Recommended tools to record a screen capture (though you can use any other tool you may prefer): • Windows 10: Search “Game bar” in the Start Menu or press WindowsKey+G. Click “Broadcast and Capture” to open the recording panel. Make sure the microphone is turned on. • Mac: Search for QuickTime Player. Go to “File->New Screen Recording”. Mac sure that your microphone is also recording.	