Practical Gaming Project

# Assessment

100% Continuous Assessment.

5% Has already been allocated based on your project proposal/presentation at the start of the module

10% Will be allocated for a presentation to be held before the Easter Break (Possibly online)

10% Will be allocated for the final presentation, which will be on the last week of lectures

75% Based on the quality of the project, detailed further below

## Pre-Easter Presentation

The pre-Easter presentation will most likely be online. You will be expected to give a 5-minute presentation of your project outlining the following

* Introduction/description of the game
* Work done to date
* Work to do
* Outline the schedule to allow the completion
* Address the core requirements (See Project Marking Description) i.e. I have implemented X which is an example of Y or in this game I will implement W which will be an example of Z

## Final Presentation

The Final presentation will be a showcasing of the completed game. As with the Pre-Easter presentation you will have to address the core requirements (See Project Marking Description) i.e. I have implemented X which is an example of Y or in this game I will implement W which will be an example of Z. Obviously, at this stage we would expect that most of these would be implemented. You will be expected to highlight the code which satisfies the given criteria.

### Submission

GitHub links will have been shared at this stage, so the final project will be downloaded by me on Friday 8th of May.

### Project Marking Description

The marking of a game project must allow for the many differing genres of games. So, the following are guidelines that will be followed when marking the projects.

The project should include implementation of the following,

* Character and Camera Movement
* Frame Rate independent code throughout
* Artificial Intelligence for NPC’s
* List/Index/Live Instantiate based object/characters
  + What is to be avoided is having a specific variable for each object.
* Implement the Observer Pattern
* Implement the State Pattern
* Script Communication methodologies
* Integration and Animation of object(s) and character(s) from 3D Animation (if doing)
* Abstract Classes
* Interfaces
* Polymorphism

Each of the above will be graded, taking into consideration the following

* Quality, complexity, placement and organisation of the code
* Own code versus Imported code (with clear delineation between the 2) i.e. include comments and references (avoid plagiarism!!!!)
* Engagement/Attendance
* Independent Progress

As stated previously, these are guidelines, and marks can and are allocated for worthy code/implementations outside of the above, but ideally these should be discussed with me, or at the very least highlighted at presentations.