UFCF9M-30-2 Game Engine Programming

Final Feedback

TEAM NAME: Mouldy Ghost

**15% Implementation & Discussion of Pipeline & Game Data Files**

This is a bit of an odd mixture, it’s naïve in some ways, i.e. attack hard coded to pulling in a set order when you could have read in the variable name as well as the value, whilst generally it is well constructed and a great deal of behaviour and content is pulled in from the files. A nice touch is your own version of UNITY’s tag system.

The documentation could perhaps have had a bit more of lead in, but for the most part very well describes the form and function of each file required.

Slight further nick-pick would be that the players in the demo scene are hardcoded and the same every time! The animation editor whilst quite simple, is a nice touch in the right direction and a nice touch that it is only accessible in debug.

11 /15

**40% Implementation of Game Engine**

Whilst lacking a bit in polish in places and just a tad glitchy, this is a solid first stage implementation of Smash Brothers. The essential game play is all there, and the game is relatively playable.

The correct spirit has been used in the various sub-systems in sharing the load of the game, and code within individual functions is fairly well structured for the most part. That said there is a rather sparse amount of commenting in the code pretty much through out! You could also have cleared out a bit more of the base engine functions that you aren’t using.

That said some solid polish has been given in so far as the AI and attract state, as well as the zooming play focused camera.

26 /40

**10% TDD & GDD**

Rather sparing in detail, but two essentially bullet pointed lists going through all the basic points.

5 /10

**10% Implementation for Arcade Machine**

Whilst possible to the detriment to the base PC version, you have ensured that the ARCADE build works exactly the same as the PC one, appropriate use of the builds types and compiler pre-processors have been made to ensure the two builds share as much code as possible.

6 /10

**5% Use of repository and other collaboration tools**

Appropriate use of the GIT Hub project system have been made, the commits to the repo are suitably atomic and whilst well titled are often lacking in comments to clear indication of what commits do in detail.

4 /5

Total: 52/80

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student Name | Student ID | Weight /20 | W. Mark / 80 | Alpha/10 | Beta/10 | FINAL/100% |
| Thomas Sylvester | 16015433 | 20 | 52 | 7 | 8 | 67 |
| Filip Skacanyi | 16015204 | 20 | 52 | 7 | 6 | 65 |
| Arthur Muddiman | 16028552 | 20 | 52 | 7 | 6 | 65 |
|  |  |  |  |  |  |  |

**Group mark distribution**

Each group will have a number of points to distribute amongst team members, according to their perceived overall contribution to the project. The overall mark for the project will be scaled according to this distribution of points, to make up each student’s individual mark for the module. The number of points allocated for a group will be 20 \* number of students in the group.

Individual student marks are determined based on the formula:

Ms = Ps / 20 \* Mg

Where Ms is the student’s mark, Ps is the points given to the student by the team, and Mg is the overall mark given to the group.

**For example:**Group A consists of 5 students, who will have 100 points to distribute amongst the team members.

Students 1, 2 and 3 are perceived to have contributed equally to the project, while student 4 has put in much more work, and student 5 much less. The team distribute their marks as follows:

1. 20 points

2. 20 points

3. 20 points

4. 30 points

5. 10 points

When marked, the project receives an overall mark of 65%. This mark is scaled as follows, for each student:

1. 20 / 20 \* 65% = 65%

2. 20 / 20 \* 65% = 65%

3. 20 / 20 \* 65% = 65%

4. 30 / 20 \* 65% = 97%

5. 10 / 20 \* 65% = 32%

**Please note:** Group weightings are intended to allow teams to reflect the reality of their development practice throughout the project. However, the module leader reserves the right to adjust or otherwise moderate the metric and/or weightings submitted in the event of exceptional group circumstances occurring.