UFCF9M-30-2 Game Engine Programming

Final Feedback

TEAM NAME: Super Smash Potatoes

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

Whilst brief and to the point your discussion of your pipeline gives the general outline of your range of files which can be loaded in to your game. The in-game level-editor is a really nice touch. Whilst simple it links well into the game, even with saved levels automatically being selectable, although obviously being able to edit already saved levels would be useful to make this fully functional.

12/15

**40% Implementation of Game Engine**

Whilst a little rough round the edges in a few places, it has to be said this is a really good rip off of Smash Brothers! The entire game loop is there, with a range of additional polish and you’ve made really good use of “acquired” art and still managed to make it seem consistent. The gameplay is pretty much on point! The particle system is also a very nice data driven system and works really quite well.

The code lacks a bit in commenting but is otherwise very well structured, both data driven and with appropriate “manager” sub-systems and you’ve even made good use of the filter system in Vis Studio to help with locating files. Slight nit-picks would be that things like your animation system is perhaps a little hardcoded to produce the players for just this game, but frankly that is the sort of middle ground I expected for an outcome for this project.

32 /40

**10% TDD & GDD**

GDD: whilst breaking up into bullet point lists in places, this is still a good general overview of the design idea of the game and really does match the final product.

TDD, on GIT Hub wiki: Once again quite sparing but does give the basic starting point on the design and requirements for the majority of the systems produced.

7 /10

**10% Implementation for Arcade Machine**

Suitable Arcade build created and essential control function rewritten for the Arcade machine, allowing the majority of the code to be shared.

6 /10

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

Suitable atomic commits, sensible titled if lacking comments as to what they more fully do. A few built file seem to have crept into the repo, but generally well used.

4/5

Total: 61/80

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student Name | Student ID | Weight /20 | W. Mark / 80 | Alpha/10 | Beta/10 | FINAL/100% |
| Benji Campbell | 16007799 | 24 | 73.2 | 6.5 | 6.8 | 86.5 |
| Huw Tingley | 16028409 | 19 | 58 | 6.5 | 6.5 | 71.0 |
| Samuel Hibbert | 16011283 | 16 | 48.8 | 6.5 | 5.9 | 61.2 |
| Adam Fort | 16009754 | 21 | 64.1 | 6.5 | 6.8 | 77.4 |

**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.