UFCF9M-30-2 Game Engine Programming

Final Feedback

TEAM NAME: MetroBlade

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

A selection of aspects of the game are read in from file, although not entirely clear how much this is actually used for the level information. Appropriate descriptions of these are given on the GIT Hub wiki. Nice touch is the key bindings as a file, could potentially mean that they could just be read in from a file (and written out to with the configuration is changed).

7 /15

**40% Implementation of Game Engine**

Some quite involved systems appear to have at least in part been implement, however it is hard to gauge how well as I was unable to navigate to actually playing the game. The FSM being apparently unable to get to the play state, no directions.

The supplied game play video shows that the bare bone elements of Smash Brother are there somewhere.

20 /40

**10% TDD & GDD**

On GIT Hub wiki: GDD is brief and to the point does highlight but does manage to highlight some aspects of the game to be produced, sadly the language in places makes it clear this was written after the fact about what ought to have been done, which if even the case is the style / verb tense that should be used for a GDD.

The TDD has a number of TODOs left in place (and empty sections), also lacks a bit in technical detail for the most part, but in other places has much more detail. A range of aspects of the engine are at least touched upon to some level.

5/10

**10% Implementation for Arcade Machine**

A novel approach of using a separate class top capture the control system for the ARCADE machine.

5 /10

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

Appropriate use of the GIT Hub project system has 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: 41/80

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student Name | Student ID | Weight /20 | W. Mark / 80 | Alpha/10 | Beta/10 | FINAL/100% |
| Elliot Martin | 1502353 | 16 | 32.8 | 5.5 | 5.1 | 43.4 |
| Jack Watson | 16008335 | 18 | 36.9 | 5.5 | 5.1 | 47.5 |
| Nathan Butt | 16013327 | 26 | 53.3 | 5.5 | 7.8 | 66.6 |
|  |  |  |  |  |  |  |

**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.