



**Project Assessment**

ModellingInThreeDimensions

The Team That Does Stuff

|  |  |
| --- | --- |
| **Team Name:** | **The Team That Does Stuff** |

|  |  |  |
| --- | --- | --- |
| **Name** | **Student Number** | **Preferred E-mail Address** |
| **Timothy O’Neill** | 40062164 | Toneill22@qub.ac.uk |
| **Nathan Sullivan** | 40059116 | Csullivan06@qub.ac.uk |
| **Stewart Reaney** | 40038703 | Sreaney01@qub.ac.uk |
| **Christopher Wilson** | 40049073 | cwilson48@qub.ac.uk |
| **Huy Vu Le** | 40075766 | hle02@qub.ac.uk |

|  |  |
| --- | --- |
| **Project Overview** | The Develop And explore The Aspects of Implementing a Game Engine. |
| **Key Outcomes** | The key outcomes of this project were to develop a partial game engine including elements of sound, AI, terrain generation, and advanced rendering techniques such as reflection, refraction, and shadowing. |
| **Languages** | Visual Studio 2012 C++ DirectX 10 |

|  |  |
| --- | --- |
| **Area** | Mark allocation |
| Organisation and Approach | 10 |
| Complexity of Development | 20 |
| Quantity of Development | 25 |
| Code Architectural Quality | 25 |
| Code Performance Quality | 20 |

The ‘Organisation and Approach’ section is fixed at 10 marks. The remaining sections are project dependent, with a minimum of 10 assigned marks and a maximum of 40 assigned marks. The total number of assigned marks across all five sections must sum to 100.

You should discuss the contribution and effort that each member has made towards the project. Following this you should then decide how the marks will be distributed within the project and record the team’s decision in the table shown below.

To help you complete this process, the entire team should sit down and complete the following two tables. To be fair to everyone in the team, it is of the upmost importance that the tables are honestly and accurately completed. To be blunt, someone who has provided a stronger contribution to the project should receive a greater mark as all the team members have benefitted from their contribution in terms of the final awarded mark.

Use the following scale to complete the table: 5 more than average, 3 average, 1 less than average.

|  |  |  |
| --- | --- | --- |
| **Team member 1: Timothy O’Neill** |  | Contribution |
| * Amount of time and effort applied throughout the project * Organisational (i.e. planning) and motivational contribution * Contribution to solve problems, develop complex code, remove bugs, etc. | | * **4** * **5** * **4** |
| **Team member 2: Nathan Sullivan** | | Contribution |
| * Amount of time and effort applied throughout the project * Organisational (i.e. planning) and motivational contribution * Contribution to solve problems, develop complex code, remove bugs, etc. | | * **5** * **5** * **5** |
| **Team member 3: Huy vu le** | | Contribution |
| * Amount of time and effort applied throughout the project * Organisational (i.e. planning) and motivational contribution * Contribution to solve problems, develop complex code, remove bugs, etc. | | * **5** * **5** * 5 |
| **Team member 4: Stewart Reaney** | | Contribution |
| * Amount of time and effort applied throughout the project * Organisational (i.e. planning) and motivational contribution * Contribution to solve problems, develop complex code, remove bugs, etc. | | * **3** * **3** * 4 |
| Team member 5: Christopher Wilson | | **Contribution** |
| * Amount of time and effort applied throughout the project * Organisational (i.e. planning) and motivational contribution * Contribution to solve problems, develop complex code, remove bugs, etc. | | * **4** * **4** * **4** |

The table should provide a rough overview of contribution from each team member (although I stress it is a rough measurement).

On the next page the team must complete the table by listings all the .cpp/.h files contained within your project (including the size of the .cpp/.h file in Lines of Code (LOC)). Against each source file, the team must then agree the contribution of each team member to the design and development of that piece of source code. If needed, extend the table across multiple pages for large projects.

A fictitious example is provided below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source File | LOC | J.Blogs | S.Smith | P.Laverty | Z.Zhou |
| Game.cpp | 258 | 40% | 10% | 20% | 30% |
| Player.cpp | 105 | 0% | 40% | 0% | 60% |
| Level.cpp | 145 | 60% | 40% | 0% | 0% |
| Collectable.cpp | 56 | 0% | 0% | 0% | 100% |
| Splash.cpp | 25 | 0% | 100% | 0% | 0% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Source File | LOC | Timothy O’Neill | Huy vu le | Nathan Sullivan | Christopher Wilson | Stewart Reaney |
| SystemClass.cpp | 342 | 0 | 0 | 5 | 5 | 0 |
| SystemClass.h | 75 | 0 | 0 | 5 | 5 | *0* |
| SoundClass.cpp | 633 | 0 | 0 | 0 | 100 | *0* |
| SoundClass.h | 107 | 0 | 0 | 0 | 100 | *0* |
| GraphicsClass.cpp | 2343 | 15 | 15 | 40 | 15 | 15 |
| GrahpicsClass.h | 152 | 15 | 15 | 40 | 15 | 15 |
| ResourceManagerClass.cpp | 318 | 0 | 20 | 40 | 0 | 0 |
| ResourceManagerClass.h | 86 | 0 | 20 | 40 | 0 | 0 |
| MeshClass.cpp | 1751 | 20 | 5 | 20 | 0 | 0 |
| MeshClass.h | 189 | 20 | 5 | 20 | 0 | 0 |
| Texture2D.cpp | 266 | 0 | 0 | 100 | 0 | 0 |
| Texture2D.h | 47 | 0 | 0 | 100 | 0 | 0 |
| EntityClass.cpp | 251 | 0 | 0 | 20 | 0 | 0 |
| EntityClass.h | 119 | 0 | 0 | 20 | 0 | 0 |
| AI.cpp | 524 | 0 | 0 | 100 | 0 | 0 |
| AI.h | 152 | 0 | 0 | 100 | 0 | 0 |
| EntityPack.cpp | 194 | 0 | 0 | 100 | 0 | 0 |
| EntityPack.h | 46 | 0 | 0 | 100 | 0 | 0 |
| Action.cpp | 30 | 0 | 0 | 0 | 0 | 100 |
| Action.h | 52 | 0 | 0 | 0 | 0 | 100 |
| ActionManager.Cpp | 62 | 0 | 0 | 0 | 0 | 100 |
| ActionManager.h | 21 | 0 | 0 | 0 | 0 | 100 |
| Condition.cpp | 81 | 0 | 0 | 0 | 0 | 100 |
| Condition.h | 67 | 0 | 0 | 0 | 0 | 100 |
| State.cpp | 40 | 0 | 0 | 0 | 0 | 100 |
| State.h | 33 | 0 | 0 | 0 | 0 | 100 |
| StateMachine.cpp | 59 | 0 | 0 | 0 | 0 | 100 |
| StateMachine.h | 29 | 0 | 0 | 0 | 0 | 100 |
| Transition.cpp | 22 | 0 | 0 | 0 | 0 | 100 |
| Transition.h | 26 | 0 | 0 | 0 | 0 | 100 |
| Shadermanager.cpp | 62 | 10 | 90 | 0 | 0 | 0 |
| Shadermanager.h | 75 | 10 | 90 | 0 | 0 | 0 |
| BasicShaderClass.cpp | 205 | 0 | 100 | 0 | 0 | 0 |
| BasicShaderClass.h | 71 | 0 | 100 | 0 | 0 | 0 |
| Basic.fx | 190 | 0 | 100 | 0 | 0 | 0 |
| BumpMapEffect.fx | 188 | 0 | 30 | 0 | 0 | 0 |
| DepthShaderClass.cpp | 185 | 0 | 100 | 0 | 0 | 0 |
| DepthShaderClass.h | 46 | 0 | 100 | 0 | 0 | 0 |
| Depth.fx | 81 | 0 | 100 | 0 | 0 | 0 |
| FoliageShaderClass.cpp | 263 | 100 | 0 | 0 | 0 | 0 |
| FoliageShaderClass.h | 46 | 100 | 0 | 0 | 0 | 0 |
| Foliage.fx | 103 | 100 | 0 | 0 | 0 | 0 |
| GlassShaderClass.cpp | 171 | 0 | 100 | 0 | 0 | 0 |
| GlassShaderClass.h | 68 | 0 | 100 | 0 | 0 | 0 |
| Glass.fx | 133 | 0 | 100 | 0 | 0 | 0 |
| ParallaxShaderClass.cpp | 200 | 0 | 100 | 0 | 0 | 0 |
| ParallaxShaderClass.h | 76 | 0 | 100 | 0 | 0 | 0 |
| Parallax\_normal\_mapping2.fx | 591 | 0 | 100 | 0 | 0 | 0 |
| ReflectionShaderClass.cpp | 319 | 100 | 0 | 0 | 0 | 0 |
| ReflectionShaderClass.h | 57 | 100 | 0 | 0 | 0 | 0 |
| Reflection.fx | 145 | 100 | 0 | 0 | 0 | 0 |
| RefractionShaderClass.cpp | 300 | 100 | 0 | 0 | 0 | 0 |
| RefractionShaderClass.h | 56 | 100 | 0 | 0 | 0 | 0 |
| Refraction.fx | 132 | 100 | 0 | 0 | 0 | 0 |
| RenderTextureClass.cpp | 142 | 50 | 50 | 0 | 0 | 0 |
| RenderTextureClass.h | 35 | 50 | 50 | 0 | 0 | 0 |
| ShadowShaderClass.cpp | 253 | 0 | 100 | 0 | 0 | 0 |
| ShadowShaderClass.h | 58 | 0 | 100 | 0 | 0 | 0 |
| Shadow.fx | 82 | 0 | 100 | 0 | 0 | 0 |
| SkyboxShaderClass.cpp | 140 | 0 | 100 | 0 | 0 | 0 |
| SkyboxShaderClass.h | 62 | 0 | 100 | 0 | 0 | 0 |
| Skybox.fx | 75 | 0 | 100 | 0 | 0 | 0 |
| TerrainShaderClass.cpp | 287 | 100 | 0 | 0 | 0 | 0 |
| TerrainShaderClass.h | 70 | 100 | 0 | 0 | 0 | 0 |
| Terrain.fx | 174 | 100 | 0 | 0 | 0 | 0 |
| TextureOnlyShaderClass.cpp | 201 | 0 | 100 | 0 | 0 | 0 |
| TextureOnlyShaderClass.h | 71 | 0 | 100 | 0 | 0 | 0 |
| TextureOnly.fx | 132 | 0 | 100 | 0 | 0 | 0 |
| TextureShaderClass.cpp | 257 | 0 | 0 | 10 | 0 | 0 |
| TextureShaderClass.h | 49 | 0 | 0 | 10 | 0 | 0 |
| Texture.fx | 91 | 0 | 0 | 10 | 0 | 0 |
| WaterShaderClass.cpp | 341 | 70 | 30 | 0 | 0 | 0 |
| WaterShaderClass.h | 63 | 70 | 30 | 0 | 0 | 0 |
| Water.fx | 185 | 70 | 30 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |

With reference to the above two completed tables, the team should discuss and agree an overall contribution measure for each team member (as covered below).

**Important:** Not being able to come to a clear decision is a valid outcome. If this is the case then talk to me and we can arrange a team meeting where we can explore how the marks should be allocated. As part of the arbitration process I will meet with the entire team and go through all source files to identify the sections that each team member has developed. These sections will then be separately marked and an individual mark awarded. An individual viva will still be held to explore each individual’s contribution.

**Important:** I also have a say in the peer distribution process. In particular, if it is very clear to me from meetings/vivas and my interaction with the team over the module that one individual contributed significantly more than other team members then I will require that the individual contribution is reflected in the peer assessment. I feel strongly that the peer assessment should be fair, and will contact teams if I feel this is not reflected in the peer distribution.

A percentage figure must be entered against each team member’s name. A value of 100 entails that the person contributed as expected by the team to the project. A value of more than 100 entails that the team felt the individual’s contribution is deserved of additional recognition (a value of 110-120 represents a notable additional contribution; a value of 120-125 represents a very significant additional contribution). A value of less than 100 entails that the team felt the individual did not contribute to the team as expected.

Note: In order to calculate each individual score (Si) from the team score (T) and the peer assessment scores (P1,...PN) where N is the number of team members, the following formula will be used:

For example, for an individual with a peer score of 110 (with other peer scores of 100, 100) and a team score of 68% will have a final score of = 110 / (1/3 \* (110+100+100)) \* 68% = 73%

|  |  |  |
| --- | --- | --- |
| **Team member** |  | Peer weighting |
| Timothy O’Neill | | 100 |
| Nathan Sullivan | | 105 |
| Huy vu le | | 105 |
| Stewart Reaney | | 95 |
| Christopher Wilson | | 100 |

To electronically complete the declaration shown below, enter your name, today’s date and within the ‘Declaration’ box type ‘I agree to the terms of the declaration’. For example, if I were to complete the declaration I would enter:

*Name Date Declaration*

*Philip Hanna 18/12/13 I agree to the terms of the declaration*

If this is a team project, then every member of the team must complete the declaration shown below:

Before completing the declaration below, check that the submission:

1. contains full acknowledgement of all secondary sources used (paper-based and electronic)

2. all code is original unless clearly referenced as otherwise

I declare that I have read the Queen's University regulations on plagiarism, and that the attached submission is original work unless clearly referenced as otherwise. I understand that my submission will be subject to an electronic test for plagiarism and will also be subject to the University regulations concerning late submission if received after the deadline.

***If the team has submitted a set of agreed peer contribution weightings***: I declare that the peer review process was conducted in accordance with the guidelines and the specified weightings provide an accurate assessment of individual contribution.

|  |  |  |
| --- | --- | --- |
| **Name** | Date | Declaration |
| Timothy O’Neill | 12/05/2014 | I agree to the terms of the declaration. |
| Nathan Sullivan | 12/05/2014 | I agree to the terms of the declaration. |
| Christopher Wilson | 12/05/2014 | I agree to the terms of the declaration. |
| Stewart Reaney | 12/05/2014 | I agree to the terms of the declaration. |
| Huy vu le | 12/05/2014 | I agree to the terms of the declaration. |

The following sets of assessment criteria assume that a quantity of 30 marks has been allocated to the particular section (this does not apply to the Organisation and Approach section which is worth a fixed number of marks).

Allocating less than 30 marks to a particular section will entail that it is easier to score a high conceptual mark. Allocating more than 30 marks to a particular section entails that it is more difficult to score a high conceptual mark.

This section is concerned with how you approach the project. In other words, the extent to which you approached the project in a manner that helped you succeed and make progress within your project.

|  |  |
| --- | --- |
| Mark Category | Description |
| 1st [70-100] | Worked with exceptional effort through-out sprints. Demonstrated excellent planning skills. Demonstrated excellent reflection within sprints. |
| 2.1 [60-70] | Worked with strong effort through-out sprints. Demonstrated good planning skills. Demonstrated good reflection within sprints. |
| 2.2 [50-60] | Worked with acceptable effort through-out sprints. Demonstrated acceptable planning skills. Demonstrated acceptable reflection within sprints. |
| 3rd [40-50] | Worked with acceptable effort in most sprints. Demonstrated acceptable planning skills. Demonstrated insufficient reflection within sprints. |
| Fail [0-40] | Worked with acceptable effort in few sprints. Demonstrated unacceptable planning skills. Demonstrated insufficient reflection within sprints. |

This section is concerned with the algorithmic complexity of your development. Was the development complex in the sense of being mathematically or computationally rich? Being able to tackle a complex development demonstrates not only that you can conceptually handle the complexity but also that you can develop and debug software that deals with such complexities. Complexity within this section will be measured against the following intertwined metrics:

* Mathematical complexity (does your development involve significant mathematical modelling).
* Algorithmic complexity (does your development involve significant algorithmic computational complexity, this might be in terms of a highly parallelised algorithm, complex internal logic, etc.).
* Hardware complexity (does your development involve complex hardware interaction, e.g. utilising hardware such as a GPU/PPU).

|  |  |
| --- | --- |
| Mark Category | Description |
| 1st [70-100] | Project assessed to have very high complexity |
| 2.1 [60-70] | Project assessed to have high complexity |
| 2.2 [50-60] | Project assessed to have reasonable complexity |
| 3rd [40-50] | Project assessed to have borderline complexity |
| Fail [0-40] | Project assessed to have insufficient complexity |

This section is concerned with the size of your development (crudely as measured in terms of LOC). It is important to stress that a project which offers a small set of features may still be considered a large development (measured used a LOC metric).

It is also important to stress that artificially inflating a project’s LOC count (to include inefficient code and/or irrelevant functionality) will not count towards this section.

The baseline of comparison is that which could be reasonably expected of a Level 3 student (where the student has been assumed to be working on the project design, code development and code debugging for an average of six hours/week over a twelve week period).

|  |  |
| --- | --- |
| Mark Category | Description |
| 1st [70-100] | Project assessed to strongly exceed base expectations |
| 2.1 [60-70] | Project assessed to comfortable exceed base expectations |
| 2.2 [50-60] | Project assessed to match base expectations |
| 3rd [40-50] | Project assessed to fall short of base expectations |
| Fail [0-40] | Project assessed as inadequate against base expectations |

This section is concerned with the quality of your architectural design. In particular, have you designed your game component so that it can be easily integrated with other components, or easily utilised by an end-developer, or easily extended/modified to incorporate new behaviour or change existing behaviour.

|  |  |
| --- | --- |
| Mark Category | Description |
| 1st [70-100] | High extensible and customisable component design that can be readily and flexibly integrated into a wider game engine and offers ease of use to end developers. |
| 2.1 [60-70] | Extensible and customisable component design that can be readily integrated into a wider game engine and offers ease of use to end developers. |
| 2.2 [50-60] | Extensible component design that can be integrated with some effort into a wider game engine. |
| 3rd [40-50] | Component design with limited extensibility / customisation that can be integrated into a wider game engine with some dependencies/assumptions. |
| Fail [0-40] | Component design with inadequate extensibility / customisation that could not be readily integrated into a wider game engine. |

This section is concerned with the performance quality of your code. In particular, have you written your game component in a manner that is likely to be fast/efficient – measured in terms of execution time, memory footprint, cache usage, etc. To score high marks you should be able to demonstrate the coding standards and practices you have adopted to maximise code performance.

To score highly in this section every line of code need not have been written to maximise performance, although, you should have been able to identify execution critical sections which have then been optimised.

|  |  |
| --- | --- |
| Mark Category | Description |
| 1st [70-100] | Comprehensive approach adopted towards identifying critical code regions, coupled with excellent high performance optimisation. |
| 2.1 [60-70] | Comprehensive approach adopted towards identifying critical code regions, coupled with appropriate high performance optimisation. |
| 2.2 [50-60] | Partial approach adopted towards identifying critical code regions, coupled with mostly appropriate high performance optimisation. |
| 3rd [40-50] | Limited (scattergun) approach adopted towards identifying critical code regions, coupled with some regions of mostly appropriate high performance optimisation. |
| Fail [0-40] | No approach adopted towards identifying critical code regions, coupled with some few regions of mostly appropriate high performance optimisation. |

**