**PRODUCT CRITERIA GRID – ASSESSMENT BRIEF**

Name: Miron Bury Student number:s1906443

|  |  |  |
| --- | --- | --- |
| **Grade** | **Content** | **Claim** |
| To achieve 0 | Name only. Copyright violation. | done |
| To achieve <40 | Project code compiles, but contains multiple unaddressed warnings. Project is incomplete or missing key components (e.g. unable to load OBJ models, only functional on a single platform). Level of programming does not meet level outlined in the coding standards document for this subject | done |
| To achieve 40+ | Level of programming meets minimum requirements of assessment in that the project is able to be compiled and run without error. Project is functionally complete with majority of features implemented. | done |
| To achieve 50+ | Meets the requirements for **40+**, and additionally has code meets coding standards, and contains the requirements as outlined in the evidence table found in this document. Project is feature complete with some optimisation of particles being generated, sprite scene contains a good level of user interaction. Demonstration of good programming practice is evident in source code  Loading of multiple models, textures | done |
| To achieve 60+ | Meets the requirements for **50+,** and additionally shows effort has been made to make sure code is as efficient and as clean as possible with thorough commenting throughout. Advanced optimisation has been attempted (e.g. Memory alignment, threading, other optimisations). Sound knowledge of programming principles is demonstrated in source code.  More than one type of model format has been attempted to be loaded. | Partially |
| To achieve 70+ | Meets the requirements for **60+,** and additionally exceeds the requirements outlined in the evidence table in this document. Project contains advanced features (different rendering pipelines DirectX, Vulkan etc). Code is well written and efficient, good knowledge of programming principles is demonstrated in source code. |  |

**CONFIGURE:**

* go to checked-out folder
* run: cmake .

**BUILD on Windows:**

* open RenderingFramework.sln
* build all
* **WARNING: you may need to RenderingFramework as startup project before runnning**

**BUILD on Linux:**

* run: make [-j 8]
* Note: you may need to install missing development packages if prompted

**BINARIES in build.zip**:

* RenderingFramework.exe - Windows 10 64-bit binary
* RenderingFramework - Ubuntu 20.04 64-bit binary

**RUN:**

L = load model

Right Mouse Button = move camera

W = foward

A = left

S = back

D = right

Q = up

E =down

Shift = increase speed