|  |
| --- |
| MHI625659-22-A |
| Coursework |
| Games Programming 3 |

|  |
| --- |
| Young, Bryan (bryan.young@gcu.ac.uk) |

*Session 2023 - 2024*

***Plagiarism***

*Attention is drawn to the University regulations on plagiarism. Whilst discussion of the coursework between students is encouraged, the actual work has to be undertaken individually. Collusion may result in a zero mark being recorded for the coursework for all concerned and may result in further action being taken.*

# Scenario

This coursework will test your ability to develop a 3D game while implementing more advanced games programming concepts.

# Specification

The following **minimum** specification should be adhered to, a game that contains:

* *Three different* models, including at least one *ARRAY* of models.
* A player that interacts with the scene.
* *One* camera with movement, that can track a model as it moves.
* The models should be *skinned* using *appropriate* textures for the game.
* The game must make use of at least *one imported vertex* and *one imported fragment* shader.
* The game must include a skybox.
* The game must contain audio.
* The game must contain collision detection.

## Extension

The above specification constitutes the core of the game. Extra marks (worth up to **65%**) can be gained by implementing the suggested extension work contained within the Lab materials:

* Advanced Cameras
* Timestep management system
* Game Object abstraction
* Component based architecture
* Enhanced model movement and interactions
* Performance optimisation
* Deferred rendering
* Extension of your choice

## Deliverables

The following should be submitted:

* A cover page with a LINK to your full application (with solution), if these have been uploaded to external storage (e.g. GitHub, Dropbox). Ensure your files **ARE NOT** set to private as internal and external moderators also need access.
* The cover page should clearly stating: Name, Matriculation number, Course and the following disclaimer:

*I confirm that the code contained in this file (other than that provided or authorised) is all my own work and has not been submitted elsewhere in fulfilment of this or any other award*.

*Signature*.

## Submission

## Final Submission

Submission of this coursework should be made electronically via GCULearn. Upload your document to GCULearn with a link to your application stored using **GITHUB/DROPBOX** no later than **17:00 on 12/01/23**. Your submission **must also include** a 30 second video of the application running which should be included in your application folder.

Late submissions will not be tolerated without a valid and documented reason.

## Marking Scheme

|  |  |
| --- | --- |
|  | Mark |
| Code: Base application | **Up to** |
| Base Game  Appropriate functionality – Game Audio, collision detection, skybox. | 5 |
| Models  Appropriate functionality – Model class that loads into memory and operates as a component of an Entity | 5 |
| Cohesion  Appropriate functionality – Apply the principles of Cohesion and Abstraction to the “mainGame” functions. | 5 |
| Cameras  Appropriate functionality may include – Camera that can track a game object while **both** the camera and the game objects are moving. | 5 |
| Memory Management  Appropriate functionality – Passing components by reference, arrays, code optimisation. | 5 |
| Frame Buffer Object  Appropriate functionality may include – The game contains and can render from a Frame Buffer Object | 5 |
| Time Stamps  Appropriate functionality may include – Appropriate components must update using DeltaTime. | 5 |
| **Sub Total** | **35** |
| Extension Material & Documentation |  |
| Extension Material  There is an expectation that extension materials will build on and go beyond pervious GP2 and Graphics Programming materials. For example:   * Implementing single, or multipurpose shaders VS implementing forward or deferred rendering. * Applying more advanced rendering techniques, such as instancing. * Building on basic in game physics to make use of broad phase and narrow phase. * Implementing hash maps to optimise shader communication.   The manner in which the extra materials are implemented will also influence the mark, for example:  Does the game make use of memory management?  Have good coding practices been employed?   * Initialisation (member initialiser list) * Data Structures * Pointers * Conventions * Readability * Modularity, Encapsulation, Abstraction, Cohesion   While students will not be penalised for applying an object orientated approach. Extra marks will also be awarded to using a data driven, component-based approach. | 65 |
|  |  |
| **Sub Total** | **65** |
| **Total** | **100** |

# Plagiarism

Attention is drawn to the university regulations on plagiarism. Whilst discussion of the coursework between students is encouraged, the actual work has to be undertaken individually. Collusion may result in a zero mark being recorded for the coursework for all concerned and may result in further action being taken.

University regulations, codes & policies: <http://www.gcal.ac.uk/student/about/regulations/index.html>

Plagiarism and cheating: <http://www.gcal.ac.uk/student/coursework/regulations/plagiarism.html>