A picture containing text, electronics

Description automatically generated

Cardiff School of Technologies

### Assessment

### Brief

|  |  |
| --- | --- |
| Module Code | Module Title |
| CIS5013 | Real Time Computer Graphics |
| Academic Year | Semester |
| 2023/24 | 1 |
| Module Leader email | |
| pnangel@cardiffmet.ac.uk | |

Content

[Assessment Details 2](#_Toc67665314)

[Submission Details 3](#_Toc67665315)

[Assessment Criteria 3](#_Toc67665316)

[Further Information 5](#_Toc67665317)

[Who can answer questions about my assessment? 5](#_Toc67665318)

[Referencing 5](#_Toc67665319)

[Submission problems 5](#_Toc67665320)

[Unfair academic practice 5](#_Toc67665321)

[How is my work graded? 6](#_Toc67665322)

# Assessment Details

|  |  |  |
| --- | --- | --- |
| Assessment title | Abr. | Weighting |
| Developing a Portfolio of Real-Time Rendering Effects | PORT1 | 60% |

|  |  |
| --- | --- |
| Task/assessment brief: | |
| GAME BRIEF:  **You are required to work on a** [**Populous II: Trials of the Olympian Gods**](https://en.wikipedia.org/wiki/Populous_II:_Trials_of_the_Olympian_Gods) **(**[**Bullfrog**](https://en.wikipedia.org/wiki/Bullfrog_Productions)**, 1991) clone.**        This is the successful sequel to Populous the very first ‘God Game’ and heralded as one of the all-time greats.  The player takes on the role of a deity, tasked with encouraging their followers to settle and multiply to increase Mana, which can then be used to access divine powers. The objective of the game is to create a larger number of followers before attempting conquer the computer-controlled deity.        The game was developed by [Peter Molyneux](https://en.wikipedia.org/wiki/Peter_Molyneux)’s Bullfrog Productions and published by Electronic Arts. Notable aspects of the game are its terraforming, which is required to sculpt the land into a place that can be settled and the ability to use Mana to perform divine acts to smite the opposition. Where Populous II built upon the first game was in the number of abilities the deity has (increased from 8 to 30 on PC, 29 on other platforms), a much-improved navigation system, and a more narrative structure that has the player cast as a demigod working their way through a pantheon of other demigods until they reach the ultimate showdown with Zeus. The original PC version of the game is available to play for free on Microsoft Game Pass. A short playthrough is available here: <https://www.youtube.com/watch?v=IBJ35qZ7UgQ>.  TASKS:  For this assignment, you are required to create a portfolio of rendering effects using OpenGL. The effects should be in the context of the given game brief (above). Your portfolio should contain the following effects created using Visual Studio, C/C++ and OpenGL:  1. 3D model rendering: Import and render a number of game assets that would represent static world objects, NPC vehicles and/or characters. This portion of the portfolio should demonstrate your use of Vertex Buffer Objects (VBOs) and Vertex Array Objects (VAOs) for efficient rendering and vertex shaders to geometry processing.  2. Texture mapping: Render a number of objects that make up the game scene using appropriate texture mapping techniques to create the required look of the game. This will demonstrate your use of texture images, texture coordinates and texture access in the Fragment Shader.  3. Lighting: Render a simple scene that makes use of multiple light sources. This can contain NPC and scenery elements for example. This will demonstrate your use of shaders to implement multiple light sources.  4. Transparency: Render a scene showing some transparent element. This will demonstrate your use of multi-pass rendering for transparent objects.  5. Game Camera: Create a suitable camera to represent the view of the player, using appropriate projection matrices to show the isometric world in the correct way.  6. Interaction: Implement basic player controls to allow the user to traverse through and interact with the scene. This will demonstrate your use of 3D matrix transformations. Note: there is no requirement to include collision detection.  Each of the elements can be implemented as separate projects or can be run as distinct demos within a single project. You can also combine them into a coherent scene if you wish, but each element should be clearly identifiable.  Hint: If you want to create a combined project, then start implementing each effect separately and combine later. It would help in your work by recording/writing your progress regularly in a portfolio design logbook.  In addition to your implementation, you are also required to:   * Produce a A2 sized poster showing the output of each effect, how it fits the given game brief. Note: The poster does not have to be printed but can be presented on-screen or on a projector. * Record a small informal code demo walk-through that highlights specific sections of your code / attempts that satisfy the given brief. * Produce a Closing Kit document summarising key APIs, resources and IP used (See Appendix 1 below) | |
| Word count (or equivalent): | 2400 |
| This a reflection of the effort required for the assessment. Word counts will normally include any text, tables, calculations, figures, subtitles and citations. Reference lists and contents of appendices are excluded from the word count. Contents of appendices are not usually considered when determining your final assessment grade. | |

|  |
| --- |
| Academic or technical terms explained: |
| **3D Model** – A 3D mesh composed of vertices and (typically) triangular faces used to represent objects in the game world.  **Textures** – The surface colour and detail of a 3D model. Typically mapped or painted onto the surface of the model using a UV Map.  **Projection Matrix** – A projection matrix is used to map points from 3D into 2D.  **Orthographic Projection** – A projection of points (vertices) from 3D to 2D that preserves the length between points. Objects do not appear smaller if they are further from the player / camera.  **Isometric Projection** – A type of orthographic projection that shows the object from a point of view where the angle of the edges from the horizontal are the same.  **Perspective Projection** – Project points from 3D to 2D, but unlike orthographic projection, perspective projection does not preserve the length between points. The effect is objects appear smaller that further they are from the player / camera. The is used for first-person games for example.  **Closing Kit** – This is a combination of documentation and software artefacts which provides an archive of the game for handover or future resurrection.  **VCS** – Version Control System. Software used in conjunction with good developer practices to track and manage changes to source code. Git is an example of a distributed version control system, and GitHub is a service built around Git. Bitbucket is an alternative also built around Git |

# Submission Details

|  |  |  |  |
| --- | --- | --- | --- |
| Submission Deadline: | 10/01/24 | Estimated Feedback  Return Date | This will normally be 20 working days after initial submission. |
| Submission  Time: | By 4.00pm on the deadline day. |  | |
| Moodle/Turnitin: | **Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.** | | |
| File Format: | The implementation is to be maintained on a suitable repository (ideally GitHub). The video presentation should be stored on your student OneDrive with suitable access setup (anyone with the link can view). You are required to submit the poster presentation as a PowerPoint file through the Turnitin submission point in Moodle. **The footer of the presentation should include links to your repository and video presentation.**  **The title of assessment should include with your student ID number, module code and assessment ID. For example:**  **st12345678-CIS5013-PORT1.pptx** | | |
| Feedback | Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas which you can improve. View the [guidance](https://learn.cardiffmet.ac.uk/mod/glossary/showentry.php?courseid=8107&eid=9581&displayformat=dictionary) on how to access your feedback.  All marks are provisional and are subject to [quality assurance processes](https://outlookuwicac.sharepoint.com/:b:/s/QED/Ec3kYQQeEHdKrCbo_tJnr2kBomIiiLINmPebUgvTUljq9Q?e=a0G2z5) and confirmation at the programme Examination Board. | | |

# Assessment Criteria

|  |
| --- |
| Learning outcomes assessed |
| |  | | --- | | **Learning Outcomes** | | [LO1] Demonstrate a detailed understanding of modern graphics hardware, APIs and their rendering pipelines by modelling, rendering and animating 2D and 3D scenes implemented using industry-standard languages and APIs.  [LO2] Discuss and compare graphical modelling, rendering and animation techniques used in computer game development through the analysis, design and development of solutions to common rendering problems.  [LO3] Demonstrate an understanding of shaders and the part they play in modern game development through the implementation of shader-based effects fundamental to games development. |  |  |  | | --- | --- | | **The marks for the assignment are broken down as follows:** | **100%** | | **3D Model Rendering** [LO1, LO2] | 10% | | **Texture Mapping**[LO1, LO3] | 10% | | **Lighting**[LO1, LO3] | 15% | | **Transparency**[LO1, LO3] | 10% | | **Camera**[LO1, LO3] | 10% | | **Interaction**[LO1, LO3] | 10% | | A2 **Poster**[LO1, LO2, LO3] | 15% | | **Recorded** **Code Demo**[LO1, LO2, LO3] | 15% | | **Closing Kit Document**[LO1] | 5% | |
| Other skills/attributes developed  This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment. |
| |  |  | | --- | --- | | Ethical | Issues in ethical information display and illusion of reality. | | Digital | Graphics media content design and it process using OpenGL. | | Global |  | | Entrepreneurial | A commercial game product brief used for building portfolio. | |

|  |
| --- |
| Marking Criteria |
| |  |  | | --- | --- | | **Grade** | **Criteria** | | **70 – 100%**  **(1st)** | An excellent range of techniques has been used to create a complete portfolio with all elements present. Very good use of OpenGL’s techniques has been made to create an efficient implementation. The poster is clear, well laid out and succinctly describes each effect and the methods used to implement it, with clear reflection on the process undertaken and techniques used. An excellent understanding of the code is evident from the code demo. | | **60 - 69%**  **(2:1)** | A very good range of techniques has been used to create a complete portfolio with all elements present. Good use of OpenGL’s techniques has been made to create the implementation, but some elements could be improved upon in order to optimize the implementation. The poster is clear, well laid out and describes each effect and the methods used to implement it. A very good understanding of the code is evident from the code demo. | | **50 - 59%**  **(2:2)** | A good range of techniques has been used to create a portfolio with all elements present, but the content could be expanded upon. Good use of OpenGL’s techniques has been made to create the portfolio, but the solution is sub- optimal and could be improved upon. The poster is well laid out but could be more succinct. A good understanding of the code is evident from the code demo. | | **40 - 49%**  **(3rd)** | A good range of techniques has been used to create a portfolio with all elements present, but the content needs to be expanded to fit the game brief. Some use of OpenGL’s techniques is evident to create the portfolio, but several key approaches have not been used or incorrectly implemented. The poster contains the key elements but is unclear and could be better laid out and more succinct. Only a basic understanding of the code is evident from the code demo. | | **35 - 39%**  **Narrow Fail** | A basic range of techniques has been used to create a portfolio with most present, but the content needs to be significantly expanded to fit the game brief. Despite this, the key learning objectives are largely addressed. Some use of OpenGL’s techniques is evident to create the portfolio, but several key approaches have not been used or incorrectly implemented. The poster contains some key elements but is unclear and could be better laid out and more succinct. Only a very basic understanding of the code is evident from the code demo. | | **<35%**  **Fail** | A basic range of techniques has been used to create a portfolio with most present, but the content needs to be significantly expanded to fit the game brief. Little to no use of OpenGL is evident to create the portfolio. The poster contains few elements and needs to be significantly expanded. Little to no understanding of the code is evident from the code demo. | |
| Further Information on assessment, referencing and grading can be found in the Module Handbook (on Moodle) |

# Appendices

Appendix 1 – Closing Kit Template

**Note:** Sections highlighted in Red are options for this assignment!

## COVER

(Insert evocative cover image here)

## YOUR GAME’S TITLE

Document version number (keep this current!)

Written by (your name/team name here)

Point of contact (producer or lead designer with contact info.)

Date of publishing

Version number (This is the software version number of the game).

Footer should always have:

Copyright © Team name Date Page Number Current Date

**Closing Kit Outline**

**Table of contents**—Remember to keep this current.

**API Versions** — A list of APIs and version numbers, possibly with links to an archive containing the relevant installers and any required software licence keys.

**Release Version** — A running version of the game built without debugging information, including the required libraries and assets to run. This might be a release folder or a release build tagged in a VCS.

**Instructions** — Brief instructions on how to run and play the game.

**Repository** — A link to the code repository in a suitable VCS (e.g. a git repository on Bitbucket).

**Credits** — Details of those who contributed to the game and if appropriate recognition of their IP.

**Other IP** — Documentation of any IP sourced for the game, for example IP used under licence from another source. The owner of the IP and the type of licence must be listed.

**Issue List** — A list of any outstanding issues known at the time of closing, i.e. any bugs known or features which remain incomplete.

