|  |  |  |
| --- | --- | --- |
|  | **FACULTY OF COMPUTING, ENGINEERING and SCIENCE** | Final mark awarded:\_\_\_\_\_ |

**Assessment Cover Sheet and Feedback Form 2017/18**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  CS2S565 | Module Title:  Computer Graphics | | Module Lecturer:  Nathan Thomas |
| Assessment Title and Tasks:  2D Scene Rendering in OpenGL | | | Assessment No.  1 of 2 |
| No. of pages submitted in total including this page: | | | Word Count of submission: N/A |
| Date Set:  In lecture TW:5 | | Submission Date:  End of TW:11 | Return Date:  Approx. 21 working days after submission |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Centre on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Centre to be consulted).  Work should be submitted as detailed in your student handbook. You are responsible for checking the method of submission. | |
| **You are required to acknowledge that you have read the above statements by writing your student number (s) in the box:** | Student Number:  15051013 |

**IT IS YOUR RESPONSIBILITY TO KEEP A RECORD OF ALL WORK SUBMITTED**

|  |
| --- |
| **Part B: Marking and Assessment**  **(to be completed by Module Lecturer)** |
| This assignment will be marked out of 100%  This assignment contributes to 40% of the total module marks.  This assignment is bonded |
| **Assessment Task:**  You are required to write an application that will render an original 2D scene using OpenGL’s modelling and rendering features discussed in lectures and tutorials. The subject of the scene is left for you to decide. For example, you could develop an outdoor scene, the plan view of a building or a scene from a game or film.  You are required to model the scene as a set of separate components, or objects using the modelling and rendering features that OpenGL provides. Your implementation should make use of the following techniques…   * You should use a variety of rendering modes (GL\_TRIANGLES, GL\_LINES etc.) in your scene to demonstrate your understand what these rendering modes do. Your models should contain at least vertex position data and additional marks will be awarded for the use of per-vertex colour and texture coordinate data. Also, additional marks will be awarded if you use Vertex Buffer Objects and provide your own additional per-vertex data that is then processed in your own modified vertex shader. If you choose to use shaders in your assignment you do not have to use shaders for every object of your scene. * You will be required to apply texture maps to your objects where appropriate. More marks will be awarded for texture mapping more complex objects. You should use the different texture modulation modes to mix the texture image with the per-vertex colour data on your objects to create different effects. * Use OpenGL’s blending feature to create at least one transparent object in the scene. * Your scene should be interactive, allowing the user to control certain objects in the scene with the keyboard and/or mouse. * As well as modelling single, discrete objects, you are also required to create at least one composite object by connecting separate objects, or components together using hierarchical modelling techniques. More marks will be awarded for more complex hierarchies.   You will also be required to explain your design and implementation in a short 5-10 minute code demo which will take place in the tutorial sessions after the assignment has been submitted. As part of the code demo you will be required to discuss how OpenGL’s modelling and rendering features were applied in your application as well as any problems you faced during the development of your application and how you addressed these problems. **The code demo is mandatory. The above sections will also be marked according to how well you demonstrate your understanding of them in the code demo.**  You may use the tutorial or lecture demo code as a starting point for your implementation but this is not a copy code and paste exercise. Any code obtained must be sufficiently modified by yourself. Marks will be awarded for the level of complexity of the scene you are rendering, the quality of the objects, or models that you develop and the use of OpenGL’s modelling and rendering features used to implement each object and the scene as a whole.  **Deliverables**   1. A zip containing the source code and executable of your implementation. This is to be submitted to Blackboard no later than the submission date shown on the assignment front sheet. Please name your zip file with your enrolment number (e.g. 12345678\_cw1.zip). Ensure the file is in accordance with the .zip file format. 2. An electronic copy of this document is to be included in your zip file, with your Student Enrolment Number filled in on the front sheet and the optional Reflection sheet (see Part C below) filled in accordingly. 3. A 5-10 minute code demo discussing your implementation, the results obtained and the problems you faced in implementing the assignment. Date will be announced via Blackboard. Please note that a code demo will need to be carried out and failure to demonstrate your OpenGL scene may result in a mark not being awarded. |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <http://icis.glam.ac.uk>):  LO1. To understand and evaluate the techniques required to model, render and animate a graphical scene. LO2. To use a suitable API to model, render and animate a graphical scene. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Marking Criteria** | **Mark Available** | **Mark Awarded** | | 1. Overall scene quality | **10** |  | | 1. Varied use of OpenGL’s rendering modes, including use of Vertex Arrays or Vertex Buffer Objects | **30** |  | | 1. Use of texture mapping and texture mapping modes | **15** |  | | 1. Appropriate use of blending in OpenGL to create transparent effects | **10** |  | | 1. Use of transformations and keyboard or mouse input to make part of the scene interactive. | **15** |  | | 1. Use of hierarchical modelling | **10** |  | | 1. Code demo\* [Mandatory] | **10** |  | |  |  |  | |  | **100** |  |   **Assessors Feedback** (linked to assessment criteria): |
| **Work on this module has been marked, double marked/moderated in**  **line with USW procedures.** |
| *Provisional mark only: subject to change and/or confirmation by the Assessment Board* |

|  |  |
| --- | --- |
| **Grading Criteria** | |
| **Performance Level** | **Criteria** |
| Fail  (< 40%) | Only a few basic rendering features are employed. No attempt at texture mapping, hierarchical modelling or interaction has been made. The scene contains little detail and the objects are only simple geometric shapes. The student fails to discuss the OpenGL features that are used or the problems faced during implementation. Little understanding of the techniques used is evident. |
| 3rd Class  (40% - 49%) | A number of OpenGL’s rendering features have been applied to create a basic 2D scene. Only basic geometric shapes have been used, but some originality is present. Basic use of texture mapping has been applied and only a basic hierarchy is presented. Little understanding of the use of the transformation matrix stack is evident. No more than one element in the scene can be controlled by the user. The student fails to discuss in detail the OpenGL features that are used or the problems faced during implementation. |
| Lower 2nd Class  (50% - 59%) | Some of OpenGL’s rendering and hierarchical modelling features have been applied to create a 2D scene. The modelling of the objects used is of reasonable quality. Varied use of texture mapping is presented and a basic hierarchy is shown, with correct use of the transformation matrix stack. No more than two elements in the scene can be controlled by the user. The student discusses the OpenGL features that are used, but there is little discussion of the problems faced during implementation. |
| Upper 2nd Class  (60% - 69%) | OpenGL’s rendering and hierarchical modelling features have been applied to create a detailed scene with numerous objects. Most of OpenGL’s rendering features have been employed to create detailed objects and the student has made good use of Vertex Arrays. Good use of texture mapping is also evident and a basic hierarchy is shown, with correct use of the transformation matrix stack. The user is able to interact with the application, with multiple parts of the model being user-controllable. The student provides a detailed discussion of OpenGL’s modelling and rendering features that are used and a good discussion of the problems faced is given. Some attempt has been made to use VBOs and process the data in the vertex shader. |
| 1st Class  (70% +) | OpenGL’s rendering and hierarchical modelling features have been applied to create a highly detailed scene with many different objects. All of the techniques looked at in lectures have been used to good effect. The quality of modelling is high and OpenGL’s rendering features have been used effectively. Excellent use of texture mapping has been made with the use of high-resolution or custom designed textures, and a detailed hierarchy of at least 3 levels deep is given, with correct use of the transformation matrix stack. The user is able to interact with the application, with multiple parts of the model being user-controllable. The student provides a detailed discussion of OpenGL’s modelling and rendering features that are used and an in-depth discussion of the problems faced is given. Good use of VBOs and processing in the vertex shader is also evident. |

|  |  |
| --- | --- |
| **Part C: Reflections on Assessment**  **(to be completed by student – optional)** | |
| **Use of previous feedback:**  In this assessment, I have taken/took note of the following points in feedback on previous work: | |
| **Please indicate which of the following you feel/felt applies/applied to your submitted work**   * A reasonable attempt. I could have developed some of the   sections further.   * A good attempt, displaying my understanding and learning, with   analysis in some parts.   * A very good attempt. The work demonstrates my clear   understanding of the learning supported by relevant literature and scholarly work with good analysis and evaluation.   * An excellent attempt, with clear application of literature and   scholarly work, demonstrating significant analysis and evaluation. | |
| **What I found most difficult about this assessment:** |  |
| **The areas where I would value/would have valued feedback:** |  |

1. University Academic Integrity Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from Campus Advice Centres [↑](#footnote-ref-2)