Major Assignment

Overview

Your major assignment in this course is to write a graphics application that generates a 3D scene that the user can interact with. Your project will be implemented in Java, LWJGL and GLSL. There are a number of criteria that needs to be met when implementing your task. This project will assess your understanding of the 3-dimensional theories taught in lectures, and the implementation of these in OpenGL as covered in the lab classes. Feel free to use your creativity in this project - a portion of the marks will be assigned for interesting and innovative features.

You will be submitting all development files as well as a report outlining and justifying what you have done.

Timelines and Expectations

Percentage of Course Mark: 50%

Due: 5:00pm on Friday of Week 11

Minimum time expectation: 40 hours

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:•

Skills:

S1. Construct a software application to display three dimensional graphics.

S2. Apply input handling techniques to manipulate an interactive video game.

S3. Analyse and debug the functionality and performance of interactive computer games.

S4. Implement computer animation using interpolation and simulation approaches.

S5. Implement data structures and algorithms commonly used in computer game engines.

Application of knowledge and skills:

A1. Create interactive video games to meet supplied specifications.

A2. Create a working game engine using inputs, graphics and animations.

Assessment Details

Your task is to implement a simple scene in 3D using OpenGL with GLSL with the main driver program written in Java. You are welcome to use any code you have developed in your lab classes, but if you use code supplied as part of the course content then this should be acknowledged both in the comments of the code as well as in your report. This assignment contains three (3) sections:

- Core components
- Extras
- Report

Each of these will be discussed in the following subsections.

Core Components

Percentage Value of Task: 50%

Minimum Time Expectation: 24 hours

To accomplish this component you should decide on what your simple scene will be such that it meets the following criteria:

- Must contain at least two (2) different non-hierarchical objects not previously used in lab classes eg you can use something where code has been supplied. Each must consist of at least 10 triangles as a minimum
- Must contain at least one (1) object that uses a hierarchical articulated structure (such as a human body– the lower-arm connects to the upper-arm, which connects to the torso). This must either use a preset animation or be controlled by the user. There must be at least one example of animation of subparts separate from other parts eg a helicopter travelling horizontally while rotors are spinning.
- There must be at least two (2) light sources that objects respond to appropriately.
- At least one object should have a texture applied to it.
- The user should be able to interact with the scene. This may be moving one (or more) objects or the virtual camera around with mouse or key input. The animation created should be independent of the computers speed running the application.
- Your scene should have some meaning, tell a story

It is up to you what your scene contains (adult rated content will not be accepted).

Extras

Percentage Value of Task: 20%

Minimum Time Expectation: 14 hours

To gain additional marks you will need to add more complexity to your scene. It is up to you how much extra you want to implement in your scene. Some tasks you choose will be complex and will be worth more marks. If in doubt have a conversation with the course coordinator. You do not have to limit what you do to the following list. This list is just some suggested extensions:

- Use a complex hierarchical structure with complex animation.
- Supplied are some example BVH files. You could write a loader for these and build your hierarchical articulated structure and animate it with these files.
- Use of shader programs not used in lab classes (eg a cel or toon shader). You will need to be able to explain what your shader does and how it works in your report.
- More than one shader program used in your scene
- Use of multiple light sources.
- Generation of complex shapes such as extruded shapes
- Write an .obj loader and load a more complex mesh.
- Use of advanced texturing techniques such as multi-texturing, bump, cube, displacement mapping.
- Use of techniques such as fog, shadows.
- Implement simple physics-based animation for objects (either those controlled by the user, and/or other objects)
- Implement simple collision detection between graphical/game objects using bounding volumes.
- Implement collision response (caution: this would be an ambitious undertaking as the relevant material will not be covered until Week 11, and is dependent on accurate collision detection also being implemented).

Report

Percentage Value of Task: 10%

Minimum Time Expectation: 2 hours

An important component of this assignment is the submission of a written report. This report should detail and justify what you have implemented. You should have as a minimum the following sections:

- Overview
  - Explain what your scene is about and what the user can do when interacting with it.
- Core Criteria
  - Explain and justify how you have met the criteria. e.g. tell me what objects you have created and how they meet the criteria. etc.
  - Discuss how you have used colour, light and shading in your scene.
- Extras
  - Identify what your extras are and justify how complex they are. Do not rely on the marker noticing what extra you have done.
- Efficiency of implementation
  - Discuss how efficient you implementation is and why you have designed it the way you have. Discuss whether there are things that you have done that you believe could needlessly slow it down.

Submission

You will need to submit all your source code and associated files such as textures, .obj or .bmv files, as well as project files required by eclipse. You will also need to submit your final report as a word document. All of these files should be zipped into a single file and uploaded to Moodle as your submission.

Marking Criteria/Rubric

Due to the open-ended nature of this task there is no marking rubric. Marks will determined by the following:

- Core Components 50%. Factors considered:
  - Inclusion of components
  - Elements all work
  - Implementation is done correctly.
- Extras 20%
  - Correctness of components implemented
  - Elements all work
  - Implementation is done correctly.
- Report 10%
  - Detail and accuracy given around suggested topics
  - Quality of analysis.
  - Quality of formal writing
  - Grammar and Spelling
- Coding quality 5%

- Following appropriate coding styles
- Suitable level of commenting
- Efficiency of code

- Quality of implementation 15%
  - Creativity in Scene
  - Degree of difficulty in the Extras which were implemented

Feedback

Marked assessments will be returned via Moodle prior to the final exam and no later than 2 weeks after submission.

Plagiarism

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found at http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism.

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Friday, 22 October 2021, 5:00 PM
Time remaining	22 days 23 hours
Last modified	-
Submission comments	▸ Comments (0)

Add submission

You have not made a submission yet.

◄ Lab Class Assignment

Jump to...

mocapData ►