Southern New Hampshire University

CS 330 Final Project Guidelines and Rubric

Overview

The creation of computational graphics and visualizations is a skill in growing demand. These skills are applicable to the following scenarios: the game industry for creating graphics and animations, the healthcare industry for medical visualizations, the entertainment industry for computer-generated imagery (CGI) and visual effects, 3D printing for creating physical objects for applied real-world problem solving, and much more. Throughout this course, you will learn how to write computer code that commands any OpenGL graphics processing unit (GPU) to create, texture, light, render, and animate 3D models in 3D space, and control virtual environments relative to a virtual camera.

This assessment addresses the following course outcomes:

- Generate accurate representations of three-dimensional objects using application programming interface (API) libraries and computer graphics development best practices
- Create interactive graphics applications that respond to input devices allowing for successful navigation around three-dimensional objects and through three-dimensional space
- Employ best practices in formatting, commenting, and functional logic that produce reliable computer programs
- Defend computer graphic and program development decisions for their effectiveness in meeting project requirements

Prompt

Your **commented code and reflection** for this project will demonstrate the skills you have gained creating 3D graphics and the principles discussed in the course. To complete this project, you will select a real-world object (personal item, commercial product, etc.) and create a three-dimensional representation of it. You will create a digitally lit, fully textured 3D object that can be orbited using a virtual camera and mouse controls. In addition, you will reflect on this project by providing documentation explaining how the graphics were created and write about how you applied each step in the OpenGL pipeline. You will also reflect on any challenges that influenced your coding/development decisions.

Personal object selection: To minimize complexity and save 3D modeling time, the polygon count for your objects should not exceed 1,000 triangles. While you may want to choose a more complex object, to complete the project within the time constraints of the course, the number of triangles must be limited. Below are examples of objects that can be created using a low polygon count. Please choose from one of these categories:

- Furniture (chair or table)
- Kitchen appliances (spoon, knife, cup, not a kettle)
- Body wash container
- Animal head sculpture (not a monkey head sculpture)

Note: When using images for textures, make sure you are using royalty-free images with resolutions of 1024 x 1024 pixels or higher.

Specifically, your project must address the following **critical elements**:

I. **3D Object**

A. Your 3D object will be assessed visually to ensure it meets the elements below:

i. Utilize **organized geometry**, ensuring that polygons (triangles) on the 3D model are well spaced and connected and give a low-polygon representation of a real-world object.

ii. Utilize **textures**, ensuring that high-resolution textures are projected accurately on the 3D model.

iii. Generate **lighting**, ensuring that all lights are implemented to give a professional-looking presentation and visualization of the model.

iv. Apply **color** to lighting with varying intensities.

B. **Navigation** Through Input Devices: Your applications will be navigated by using the mouse and keyboard input devices to control a virtual camera. The elements below must be met:

i. Create **horizontal orientation** navigation of the 3D object in the application allowing for azimuth orientation of a virtual camera that orbits a lit model when the mouse is moved horizontally.

ii. Create **vertical orientation** navigation of the 3D object in the application allowing for altitude orientation of a virtual camera that orbits a lit model when the mouse is moved vertically.

iii. Create code to **clamp** or gimbal lock azimuth and altitude orientation to prevent irregular camera angles (e.g., a 90-degree camera rotation clamp on the pitch axis).

iv. Create **perspective and orthographic displays** of the 3D object so that the user can change the viewport display of the 3D model from 2D to 3D and vice versa, using the tap of a keyboard key, allowing the user to switch between orthographic (2D) and perspective (3D) views at will.

C. **Syntax** Assessment—Best Practices: These best practices should be evident within your program:

i. Employ **formatting best practices** by providing program code that is easy to read and follows industry-standard code formatting practices, such as indentation and spacing.

ii. Utilize **commenting** best practices, ensuring that project source **code** used is briefly and clearly explained using descriptive comments.

iii. Employ **functional** coding **logic** best practices, ensuring that the program runs as expected.

II. **Reflection**

A. Justify **development choices** for your object. Why did you choose your selected object? Were you able to program for the functionality required?

B. Explain how a **user can navigate** your 3D object. Explain how you set up the virtual camera for your 3D object and the programming syntax you used to control its navigation using the input devices.

C. Explain the **custom functions** in your program that you are using to make your code more modular and organized (what does the function do and how is it reusable?).

Milestones

Your work on the final project is supported by two milestones.

Milestone One: *Project Proposal*
In **Module Three**, you will propose a real-world object to model in your project, submit a photograph of it, and discuss how you will re-create it as a 3D object in modern OpenGL, following the parameters established for the final project. This milestone will be graded with the **Milestone One Rubric.**

Milestone Two: *Project Draft*
In **Module Five**, you will submit a draft of the final project, including all the aspects of graphics development covered up to this point in the course. This is an important opportunity to try out your skills and receive valuable feedback as you prepare for the final project submission. This milestone will be graded with the **Milestone Two Rubric.**

Final Submission: *Commented Code and Reflection*
In **Module Seven**, you will submit your final project. It should be a complete, polished artifact containing **all** of the critical elements of the final product. It should reflect the incorporation of feedback gained throughout the course. **This submission will be graded with the Final Project Rubric.**

Southern New Hampshire University

Rubric

Guidelines for Submission: Submit a commented .cpp file for the 3D object. Also submit a reflection, which should be 1–2 pages using 12-point Times New Roman font and double spacing. Any citations should be in APA format.

Critical Elements	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
3D Object: Organized Geometry	Meets "Proficient" criteria and demonstrates a sophisticated use of geometry in the 3D object	Utilizes organized geometry, ensuring that polygons are well spaced and connected while keeping the polygon count moderate	Utilizes organized geometry, but with errors in ensuring that polygons are well spaced and connected while keeping the polygon count moderate	Does not utilize organized geometry ensuring that polygons are well-spaced and connected while keeping the polygon count moderate	5.94
3D Object: Textures	Meets "Proficient" criteria and demonstrates a sophisticated use of texture on the 3D model	Utilizes textures, ensuring that high-resolution textures are projected accurately on the 3D model	Utilizes textures to ensure that high-resolution textures are projected on the 3D model, but contains some inaccuracies	Does not utilize textures to ensure that high-resolution textures are projected accurately on the 3D model	5.94
3D Object: Lighting	Meets "Proficient" criteria and demonstrates an advanced application of lighting on the 3D object	Generates lighting, ensuring that all lighting is implemented for a professional presentation and visualization of the model	Generates lighting, ensuring that all lighting is implemented for a professional presentation and visualization of the model, but with errors	Does not generate lighting ensuring that all lighting is implemented for a professional presentation and visualization of the model	5.94
3D Object: Color	Meets "Proficient" criteria and demonstrates a sophisticated use of types of color, tone, and intensity in lighting	Applies color to lighting with varying intensities	Applies color to lighting, but with errors	Does not apply color to lighting	5.94
Navigation: Horizontal Orientation	Meets "Proficient" criteria and demonstrates a sophisticated use of horizontal orientation control	Creates horizontal orientation navigation of 3D object allowing for azimuth orientation of a virtual camera that orbits the model when the mouse is moved horizontally	Creates horizontal orientation navigation of 3D object allowing for azimuth orientation of a virtual camera that orbits the model when the mouse is moved horizontally, but with errors	Does not create horizontal orientation navigation of 3D object allowing for azimuth orientation of a virtual camera that orbits the model when the mouse is moved horizontally	5.94

Navigation: Vertical Orientation	Meets "Proficient" criteria and demonstrates a sophisticated use of vertical orientation control	Creates vertical orientation navigation of 3D object allowing for altitude orientation of a virtual camera that orbits the model when the mouse is moved vertically	Creates vertical orientation navigation of 3D object allowing for altitude orientation of a virtual camera that orbits the model when the mouse is moved vertically, but with errors	Does not create vertical orientation navigation of 3D object allowing for altitude orientation of a virtual camera that orbits the model when the mouse is moved vertically	5.94
Navigation: Clamp	Meets "Proficient" criteria and demonstrates a sophisticated use of angle control due to clamping	Creates code to clamp azimuth and altitude orientation to prevent irregular camera angles	Creates code to clamp azimuth and altitude orientation to prevent irregular camera angles, but with errors	Does not create code to clamp azimuth and altitude orientation to prevent irregular camera angles	5.94
Navigation: Perspective and Orthographic Displays		Creates perspective and orthographic displays of 3D object so that the user can change the viewport display of the 3D model from 3D to 2D using the keyboard	Creates perspective and orthographic displays of 3D object so that the user can change the viewport display of the 3D model from 3D to 2D using the keyboard, but with errors	Does not create perspective and orthographic displays of 3D object so that the user can change the viewport display of the 3D model from 3D to 2D using the keyboard	5.94
Syntax: Formatting Best Practices	Meets "Proficient" criteria and demonstrates a sophisticated awareness of industry best practices in formatting	Provides program code that is easy to read and follows formatting best practices as defined by the industry	Provides program code that is easy to read, but follows only some formatting best practices	Does not provide program code that is easy to read or does not follow any formatting best practices	7.92
Syntax: Commenting Code	Meets "Proficient" criteria and demonstrates keen insight into best practices in commenting code	Utilizes commenting best practices, ensuring that project source code used is briefly and clearly explained using descriptive comments	Utilizes commenting best practices, ensuring that project source code used is explained using descriptive comments, but comments lack detail or clarity	Does not utilize commenting best practices to explain project source code	7.92
Syntax: Functional Logic	Meets "Proficient" criteria and demonstrates keen insight into best practices in functional logic	Employs functional coding logic best practices, ensuring that program runs as expected	Employs functional coding logic best practices, ensuring that program runs, but with errors	Does not employ functional coding logic best practices	7.92

Reflection: Development Choices	Meets “Proficient” criteria and demonstrates a keen insight into development choices for this project	Justifies development choices of the 3D object and explains how the required functionality was achieved	Justifies development choices of the 3D object and explains how the required functionality was achieved, but justification and explanation lack detail or clarity	Does not justify development choices or explain how the required functionality was achieved	7.92
Reflection: User Can Navigate	Meets “Proficient” criteria and demonstrates a nuanced understanding of user navigation	Explains how the user can navigate the 3D object, the setup of the virtual cameras, as well as the programming syntax used for the input devices	Explains how the user can navigate the 3D object, the setup of the virtual cameras, as well as the programming syntax used for the input devices, but explanation lacks detail or clarity	Does not explain how the user will navigate the 3D object, the setup of the virtual cameras, as well as the programming syntax used for the input devices	7.92
Reflection: Custom Functions	Meets “Proficient” criteria and demonstrates a nuanced understanding of the program functions	Explains the custom functions used in the program, what they do, and how they are reusable	Explains the custom functions used in the program, what they do, and how they are reusable, but explanation lacks clarity or detail	Does not explain the custom functions used in the program, what they do, and how they are reusable	7.92
Articulation of Response	Submission is free of errors related to citations, grammar, spelling, syntax, and organization and is presented in a professional and easy-to-read format	Submission has no major errors related to citations, grammar, spelling, syntax, or organization	Submission has major errors related to citations, grammar, spelling, syntax, or organization that negatively impact readability and articulation of main ideas	Submission has critical errors related to citations, grammar, spelling, syntax, or organization that prevent understanding of ideas	4.96
				Total	**100%**