SCIT

School of Computing & Information Technology

CSCI336 – Interactive Computer Graphics
Autumn 2019

Assignment 3

Due on Tuesday, 4th June 2019 at 23:55

Task

Write an OpenGL program that displays a simple textured 3D scene with lighting. Use AntTweakBar to create a Graphical User Interface (GUI) for the user to control the various settings. Figure 1 shows an example of the scene taken from two different viewpoints with different properties.

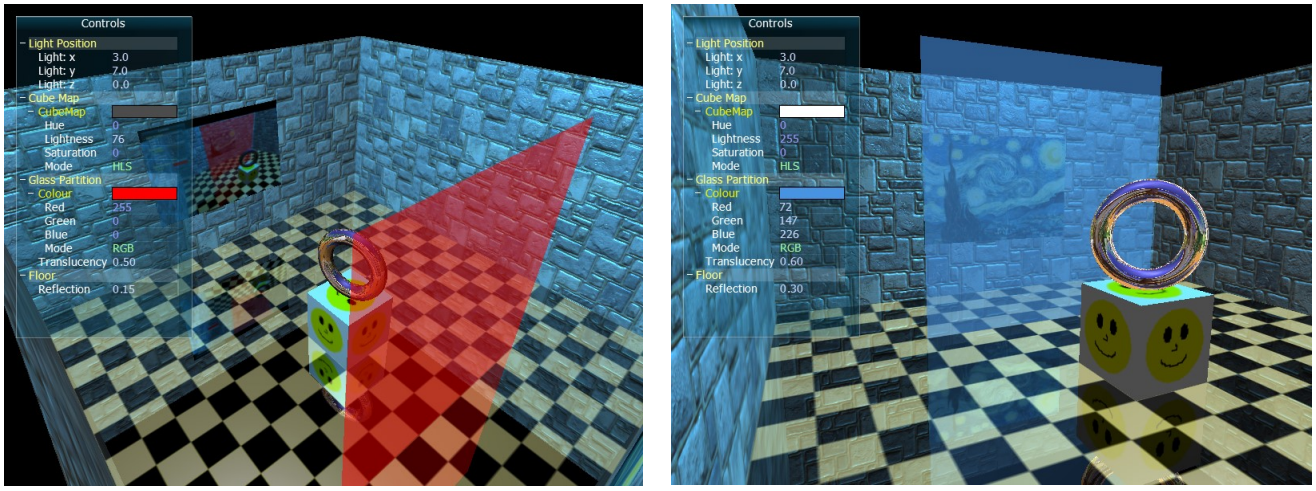

Figure 1: Example scene.

- The 3D scene (3 marks)
  - Construct a room with four walls and a floor (no ceiling). Create the room by specifying vertex positions, surface normals and texture coordinates.

    The room should contain:
    - Two paintings, which are hung on separate walls
    - A rotating ornament (load a mesh from a model file) on top of a cube pedestal in the centre of the room
    - A tinted glass room partition somewhere in the room

Your program should have a movable camera that the user can control using keyboard and mouse input. To keep camera rotation separate from interaction with the GUI, only allow camera rotation when the right mouse button is pressed and held.

- Lighting and textures (2 marks)
  - The room should be lit.
    - A single light source is sufficient
    - Allow the user to adjust the position/direction of the light source
  - Use different images to texture each of the following:
    - Walls
    - Floor
    - Pedestal
    - Paintings
- Normal mapping (2 marks)
  - Render the walls of the room using normal mapping to create the appearance of bumps on the surface.
- Cube environment mapping (2 marks)
  - Render the ornament (on the pedestal) using cube environment mapping.
  - Allow the user to control the intensity of the cube reflection.
- Translucent surface (2 marks)
  - The tinted glass room partition (use a plane with alpha blending) should be a translucent surface.
  - Using the GUI, the user should be able to control the amount of translucency and the colour of the tint.
- Screen capture (2 marks)
  - When the user presses '0', your program should take a screen capture and save it into an image file.
  - Also, when the user presses '9', take a screen capture and display this image as one of the paintings in the room.
- Reflective surface (2 marks)
  - The floor of the room should a reflective surface. Use the stencil buffer and blending for this.
  - Provide an option in the GUI for the user to control the intensity of the reflection.

Screenshots

In your submission, include three screenshots of your program. The screenshots are to show:

- Two screenshots from opposite ends of the room showcasing different properties
- A screenshot from an angle showing the painting that was displayed using a previous screenshot

Save the screenshots using one of the common image formats, i.e. bmp/jpg/png.

Instructions and Assessment

Zip **all your source files** (.cpp, .h, .vert and .frag) and **screenshots** (.bmp/.jpg/.png) into a single file and submit this via Moodle by the due date and time (do **NOT** zip your entire visual studio project file as this can be very large). Assignments that are not submitted on Moodle will not be marked.

You will have to demonstrate your working program during the lab in Week 13. Do not try to fix your code during this lab, your program will be assessed based on what you submitted. Your program must work on the computers in the lab or you must be able to demonstrate that it works on your own laptop.

The assignment must be your own work. If asked, you must be able to explain what you did and how you did it. Marks will be deducted if you cannot correctly explain your code.

NOTE: The marking allocations shown above are merely a guide. Marks will be awarded based on the overall quality of your work. Marks may be deducted for other reasons, e.g., if your code is too messy or inefficient, if you cannot correctly explain your code, etc.

For code that does not compile, does not work or for programs that crash, the most you can get is half the marks (i.e. 7.5 marks or less).

