



ASSIGNMENT 2

Assignment Posted: July 17, 2021

Assignment Due: July 26, 2021 before 11.59pm

Final Deadline with 20% flat Penalty: July 29, 2021 before 11.59pm

Description:

This OpenGL programming assignment will build upon the modeling program, which you have developed as part of Assignment 1. If you have not completed Assignment 1, then you may do it as part of this assignment and continue with completing Assignment 2. In this case we will grade it for 4 of the 10 marks allotted to the first assignment, provided you do the assignment yourself.

More specifically, in this second assignment you will learn about lighting, shading, texture mapping, shadow generation, and model movement.

Please recall that your model mesh is composed of different cube parts hierarchically linked to each other through appropriate modelling transformations.

Implementation Specifications:

1. Extend your OpenGL Assignment 1 with the following functionality and features:

- Illuminate the scene by adding a point light source (white) 30 units above the model using the Phong model.
- Enclose the light in a big cube to showcase physical location of the light (use triangle mesh representations.)
- Render the scene with tile texture on ground mesh, brick texture on wall and shiny metal finish (color of your choice) for the object passing through the wall. (Key X for toggling texture on/off).
- Render the scene with shadows using two pass shadow algorithm. (Key B to toggle).
- Make the model move along the z-axis, (One key of your choice for one small forward movement, and other key for continuous forward movement.)
- Provide functionality for rotating the object around all the axes. (Select keys of your choice for clockwise and counter-clockwise rotations along all axes).

2. Please note that the above are extensions to your program and previous key presses (of assignment 1 should work as before, but with the extended rendering as required in this assignment.

- The application should handle following input for each model which should be selected by pressing a key from 1 to 5:
 - The user can incrementally size up the model by pressing 'U' to scale-up and 'J' to scale-down. Each key press should result in a small size change.
 - The user can control the model position and orientation using keyboard input i.e. A → move left, D → move right, W → move up, S → move down, a → rotate left 5 degrees about Y axis, d → rotate right 5 degrees about Y axis. You may add other rotations about other axis, if you want.
 - The world orientation is changed by using keyboard input i.e. left arrow → Rx, right arrow → R-x, up arrow → Ry, down arrow → R-y. (Rx denotes a small anti-clockwise rotation about positive x axis, R-x about negative x axis, etc.) Pressing "Home" button should reset to the initial world position and orientation.
 - The user can change rendering mode for the model, i.e. points, lines, triangles based on keyboard input, namely, key 'P' for points, key 'L' for lines, key 'T' for triangles.
 - The user can pan and tilt the camera as follows:
 - While right button is pressed → use mouse movement in x direction to pan; and
 - While middle button is pressed → use mouse movement in y direction to tilt.
 - The user can zoom in and out of the scene using just the camera perspective - while left button is pressed → use mouse movement to move into/out of the scene.
 - Window resize handling: The application should handle the window resize events and must correctly adjust the aspect ratio accordingly. This means, meshes should not be distorted in anyway.

3. The application must use OpenGL 3.1 and onwards and must include brief comments explaining each step.

Submission:

Assignment must be submitted only through Moodle. No other form of submission will be considered. Please create a zip file containing your C/C++ code, vertex shader, fragment shader, a readme file (.txt). The zip file should be named Assignment#_YourTeamID. In the readme file document, the features and functionality of the application, and anything else you want the grader to know i.e. control keys, keyboard/mouse shortcuts, etc.

Additional Information

- ❖ You can use the skeleton code provided during the lab sessions to get started.

Bonus Features:

You can achieve an extra 20 points as bonus if you implement a glowing boundary effect or moving border effect for your object.

Evaluation Procedure

You MUST demonstrate your program to the lab instructor during a pre-scheduled zoom session. All the team members must be present during the chosen timeslot. You must run your submitted code, demonstrate its full functionality, and answer questions about the OpenGL programming aspects of your solution. Major marking is done on the spot during the demonstration. Your code will be further checked for structure, non-plagiarism, *etc.* However, ONLY demonstrated submissions will receive marks. Other submissions will not be marked.

Grading Rubric for Assignment 2

Total	100 pts
Model	03 pts
Drawing the Light Source	03 pts
Lighting	10 pts
Point Light Source	10 pts
Texture Mapping	21 pts
Texture for the ground	07 pts
Texture for the wall	07 pts
Texture for the object	07 pts
Shadow Mapping	25 pts
Shadows in the scene	25 pts
Model Movement	18 pts
Translation of the object	3 pts
Rotations of the object	15 pts
Window Resizing Events	03 pts
Handle window resizing gracefully	03 pts
Demo QnA by the Grader	20 pts
Bonus Points	20 pts