UNIVERSITÉ
Concordia
UNIVERSITY

ASSIGNMENT 2

Assignment Posted: October 02, 2019

Assignment Due: October 22, 2019 before 11.59pm

Final Deadline with 20% flat Penalty: October 25, 2019 before 11.59pm

Description:

This OpenGL programming assignment will build upon the car modelling program that you have developed as part of Assignment1. If you have not completed Assignment1, then you may do it as part of this assignment and continue with completing Assignment 2 in which case we will mark it for 3 of the 8 marks allotted to the first assignment, provided you have done the assignment yourself.

More specifically, in this second assignment you will learn about lighting, shading, texture mapping, shadow generation, and wheel motion to move a vehicle.

Please recall that your car mesh is composed of different parts and 4 wheels. These should be hierarchically linked to each other through appropriate modelling transformations. The wheels can be rotated about their center line and also moved forward/backward.

Implementation Specifications:

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

   ➢ Make the wheels cylindrical (use piece-wise linear representations for the circular edges)

   ➢ Illuminate the scene by adding a point light source (white) 30 units above the car using the phong model

   ➢ Render the scene with grass texture on the ground mesh, tire texture on the wheels and shiny metal finish (color of your choice) for the car body (Key X for toggling it)

   ➢ Render the scene with shadows using two pass shadow algorithm (Key B to toggle)

   ➢ Make the car move by rotating the wheels, while also providing lateral movement to the wheels. (One key of your choice for one small rotate + small forward movement, and other key for continuous forward movement. Same for reverse movement).

   ➢ Make the car turn right or left by a small amount (choose your own keys)

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

➢ Pressing the spacebar should re-position the car at a random location on the grid.

➢ The user can incrementally size up the car by pressing 'U' for scale-up and 'J' for scale-down. Each key press should result in a small size change.

➢ The user can control the car 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 the "Home" button should reset to the initial world position and orientation.

➢ The user can change rendering mode for the car, *i.e.* points, lines, or triangles, based on keyboard input *i.e.* 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 - while left button is pressed → use mouse movement to move into/out of the scene.

➢ Window resize handling: The application should handle window resize events and correctly adjust the aspect ratio accordingly. This means, the meshes should not be distorted in anyway.

3. The application should use OpenGL 3.0 and onwards, and 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#_YourStudentID. 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.*

Extra Credit (10% Points)

A more elaborate model of the car with windows and including some interior of the car.

Evaluation Procedure

You MUST demonstrate your program to one of the lab instructors during lab hours or at prearranged times. 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 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.