UNIVERSITÉ
Concordia
U N I V E R S I T Y

| ASSIGNMENT 1 |
|---|
| **Assignment Posted:** June 30, 2022 |
| **Assignment Due:** July 09, 2022, before 11.59pm |
| **Final Deadline with 20% flat Penalty:** July 12, 2022, before 11.59pm |

Description:

This OpenGL assignment will provide an introduction to OpenGL programming. More specifically, you will learn how to create simple virtual scenes consisting of objects, display them as wireframe meshes by drawing triangles, and also how to set up and manipulate a virtual camera to view the scene from varied angles and distances.

A mesh is a set of polygons that share vertices and edges which describe the shape of a geometric object. In this assignment you have to first create a unit cube in 3D at the origin, then using appropriate modeling transformations create different parts of Olaf (a snowman) -body, head, nose, two hands, two legs, and two eyes. (see figure depicted below: side-view [left] perspective-view[right]).
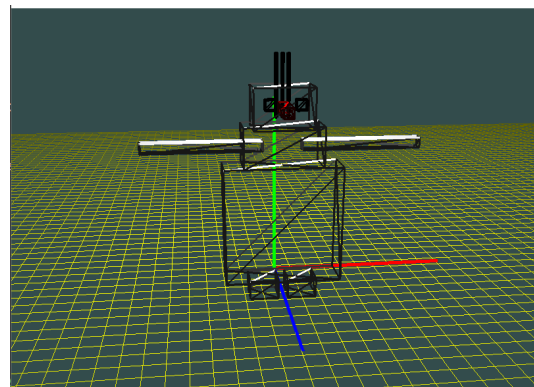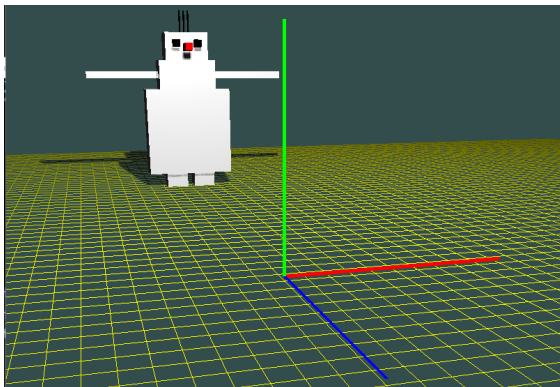
Implementation Specifications:

Develop an OpenGL application with the following functionality and features:

➤ Creates a 100x100 square grid (ground surface) in the XZ plane centered at the origin.

➤ Creates a set of three lines 5 grid units in length, in 3 different colors, representing each coordinate axis in virtual world space, centered at the origin.

➤ Creates a model of the Olaf (a snowman) like the one depicted in the figure by suitably transforming one unit cube to create different body parts. The Olaf is initially positioned at the center of the grid facing along X axis. The body parts should be independent parts so they can be rotated/moved on their own. We recommend that you use hierarchical modelling, so that a single transformation applied to Olaf's origin will apply to all of Olaf.

➢ Places a virtual camera with the world space origin as the point of focus.

➢ For display and animation:

▪ Create a GLFW window of size 1024x768 with double buffering support.

▪ Render the coordinate axis, ground and Olaf in the window.

▪ The application should use a perspective view to display all the objects and enable hidden surface removal.

➢ The application should handle the following input:

▪ Pressing the spacebar should re-position the Olaf at a random location on the grid.

▪ The user can incrementally size up the Olaf 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 Olaf 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 car, *i.e.* points, lines, 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.

➢ The application should use OpenGL 4.1 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#_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.

Additional Information

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

Extra Credit (10% Points)

A more elaborate model of the Olaf (including hair, fingers, *etc.*) which could later enable more realistic movement/animation of the Olaf.

Evaluation Procedure

You MUST demonstrate your program to the lab instructor during lab hours or at some 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 the demonstrated submissions will receive marks. Other submissions will not be marked.