

Lab 9

CSE 165: Object Oriented Programming

Spring 2022

100 points

This programming assignment has two tasks, complete each task as instructed. Write a separate file for each of the following tasks. To submit your assignment, please organize your code in the folder "Lab9" by placing your code in its corresponding sub-folder. For example, store your code for task 1 in the following directory "Lab9/1/". Then, submit the compressed version of folder Lab9 to CatCourses. Submissions must arrive by one minute before the lab section of week 12 (4/04 – 4/08). All of the files you need for this programming assignment are available in a ZIP archive file called "Lab9.zip".

1. Abstract Data Types (40 points)

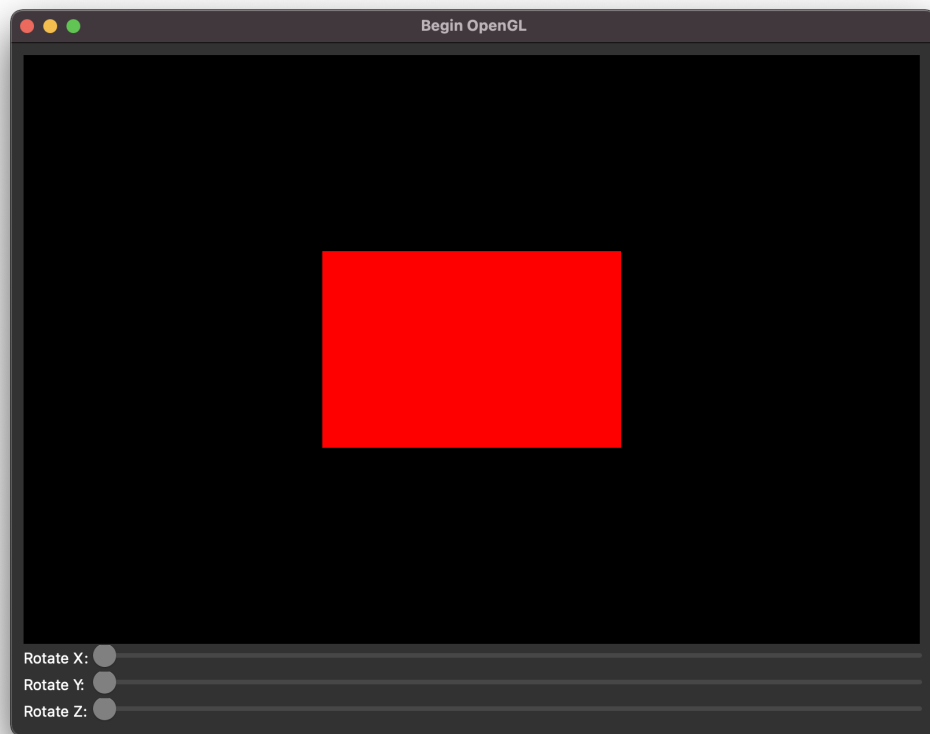
Study the file Object.h. It defines an abstract data type Object with a virtual multiply() function. Complete the implementation of two classes Number and Coord3D which inherit from the Object class. The multiply() function must dynamically create a new object to hold the result of the computation. If the operands of multiply() are both Numbers, then the result should be a Number. If one operand of multiply() is a Coord3D, and the other is a Number, then the Number should multiply each component Coord3D. If both are Coord3Ds, then the result should be a component-wise multiplication. The file Object.cpp provides a basic test.

```
# The expected output is:
x = 4, y = 8, z = 12
x = 4, y = 8, z = 12
x = 1, y = 4, z = 9
num = 16
```

2. OpenGL Windowterface (window and interface)

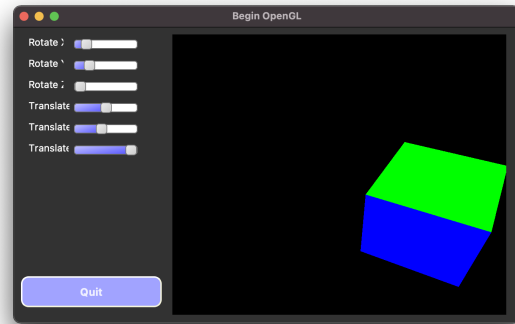
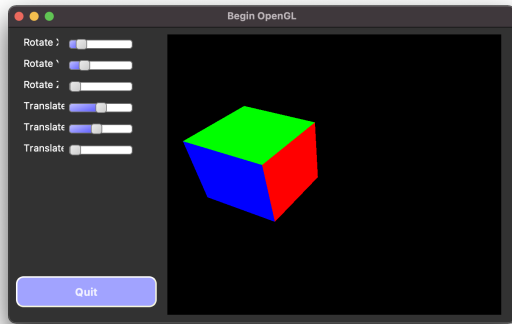
In this problem, you will learn how to create OpenGL window and update OpenGL window using user interface.

Download the zip folder “OpenGL_UI_Example.zip” and Extract the content. Next open Qt Creator and load the OpenGL_UI_Example.pro. Study my_gl.h,my_gl.cpp, mainwindow.h, mainwindow.cpp and main.cpp. Try to change code and see the changes by running the project. Your window should look like something below.



Your task is to use “OpenGL_UI_Example.zip” and implement the following:

- Draw a tetrahedron instead of a cube. Assign unique color to each vertices. (20 points)
- Implement 3 sliders for translating object in x, y and z direction. (20 points)
- Add a button to quit the application. (10 points)
- Add some form of style to your buttons and slider. (10 points)



The two images above show an example application. The example application is a cube, not a tetrahedron (minus 20 😞). There are also three additional sliders for translating the object. In the left image the last slider is all the way to the left, and in the right image, the last slider is all the way to the right. The object's position reflects the change in the value of the last slider. At the bottom left there is a button to quit the application. Finally, the styling of the sliders and buttons differ from the first image (e.g. square instead of circular knobs).