CS 770/870 Assignment 1 (Revised): OpenGL GUI Framework
C++: GLUT/GLUI Interaction Java: AWT/Swing Interaction
September 14, 2014

Due: Friday, 9/26 by 23:59:59
Lateness: Sat/Sun -3, Mon -10, Tue -15

The goal is to learn to implement basic interactive access features useful for future assignments in this course. This will give you some experience with the basic interaction features of GLUT and GLUI (C++) or AWT/Swing functionality.

Extend your solution to P0 by allowing a user to add shapes to the scene interactively and define shape fill and line colors and styles for shapes being drawn. You should define 2 different triangles, 2 different quads, a rectangle, and at least 3 different polygons. Each object should be assigned a single unique letter that will be the start of the id representing each instance of that object. This letter should be part of the label for its button.

1. Use GLUT or Java mouse handling facilities
   - mouse down/up at same location (within a few pixels): place an instance of the **current** object at this location at a default size using the **current** color and attribute settings
   - mouse down/up at different locations: one corner of a shape to be added is the down position, the other corner is the mouse up position; the **current** shape using the **current** color and attribute settings should be added to the scene and located and scaled so it fits in the space specified by the mouse up/down positions. ***The order of the 2 positions should be irrelevant.*** The object should be drawn continuosly **as the mouse is moving.**
2. Build a GLUI or Swing interaction window with
   - a label (static text) that defines the identifier to be assigned to the next object added to the scene; the id starts with the objects unique letter followed by an integer that is incremented by 1 as an instance of each kind of object is added to the scene. When the current shape changes the label should be updated to reflect the type of the current shape and the correct ordinal value for that kind of shape.
   - a button that erases all shapes (and resets all counters)
   - a radio button panel to select which shape is to be drawn next;
   - a glui list box or java JList to specify the color for the boundary of subsequently added shapes;
   - a list box or JList to specify the color to be used for the fill of subsequently added shapes;
   - a checkbox to control whether polygon boundaries should be drawn for subsequently added shapes;
   - a checkbox to control whether polygon interior should be filled for subsequently added shapes;
   - two scrollbars for specifying X and Y translation to be applied to the most recently added shape;
   - a text input window that allows a user to enter the id of a shape to be deleted; the program must store the original id with each shape, so you can search for the correct shape to be deleted; the user can also enter * as the number part of the id, which means to delete all objects of the specified type.

Point allocation for interaction specification and associated functionality

15	Mouse interaction (down/up in same position) places new object of default size
25	Mouse down/move/up defines location/size of object added with continuous update (5 points)
5	Button for erase all
10	Radio buttons for shapes
10	Two checkboxes and associated behavior
10	List boxes for boundary and fill colors (10 for first one, 5 for second)
10	Translation sliders
15	text input for deletion of shapes

Note:

Points are earned by correctly implementing features robustly. Points are deducted for bugs, incorrect implementation, poor style, poor design decisions, etc.

If you are using C++, you may have to install glui yourself from http://glui.sourceforge.net. If you are using Mac OS X 10.9 (Mavericks), you can use the precompiled *libglui.a* available at *~cs770/macosx/lib/libglui.a* and contained in *macosx.tar.gz* from the course web page. I think these will work with any version of Mac OS X, but I haven't tried it. The files in the *macosx/lib* and *macosx/include* directories should be copied to your *cs770/lib* and *cs770/include* files although you probably already have *libGLEW.a*. We'll need *libjpeg.a* later in the semester.