Lab #2 – csc471
Coordinate transforms and event callbacks

Today we will alter an existing program, simple_release.cpp, in order to learn more about coordinate transforms and event callbacks.

Please download simple_release.cpp from the class website:
http://www.csc.calpoly.edu/~zwood/teaching/csc471/csc471.html

Compile and run the program.
Please read what the program does and notice the style and format of the program. Note that my data structures and code are mostly C data structures and code, but you could easily replace them with C++ style code and classes.

In its current form, simple_release.cpp has two drawing modes. It will either draw a multi-colored square or should draw 10 points where a user clicks with the mouse.
Note, however, that the code to draw points is broken!!! This is intentional.

1) Add a reshape function and register the callback so that your program behaves properly (i.e. the square remains a square) when you reshape the window. Recall that the parameters to glOrtho2D will define the left, right, bottom and top parameters for your 'window' into world coordinates. See the slides for an example.

1) Next, you need to fix the code to draw points. You will need to make changes in the mouse function in order to make sure that you can indeed draw 10 points.
These 10 points should appear exactly under where the user clicks the mouse. Please recall what we discussed in class about coordinates and carefully think about what coordinate system the x and y are returned in by the mouse function.
Hint: I recommend writing two simple helper functions something like:
float p2w_x(int) and float p2w_y(int).

3) Using the program 1 resource, add a simple menu to your program that replaces the keyboard "m" switch. In other words, add a menu that allows the user to toggle between drawing a square and drawing points.

Note that this initial program could be a good starting point for your first programming assignment, except it is not a good idea to use a static sized array to store strokes because you do not know ahead of time how many strokes the user many want to draw. You will need to use a collection class like a list or vector from STL, etc. that can be resized (you could in fact use an array and just resize it when needed but please just use an appropriate STL data structure).

This lab will need to be checked off in lab within one week of being assigned.