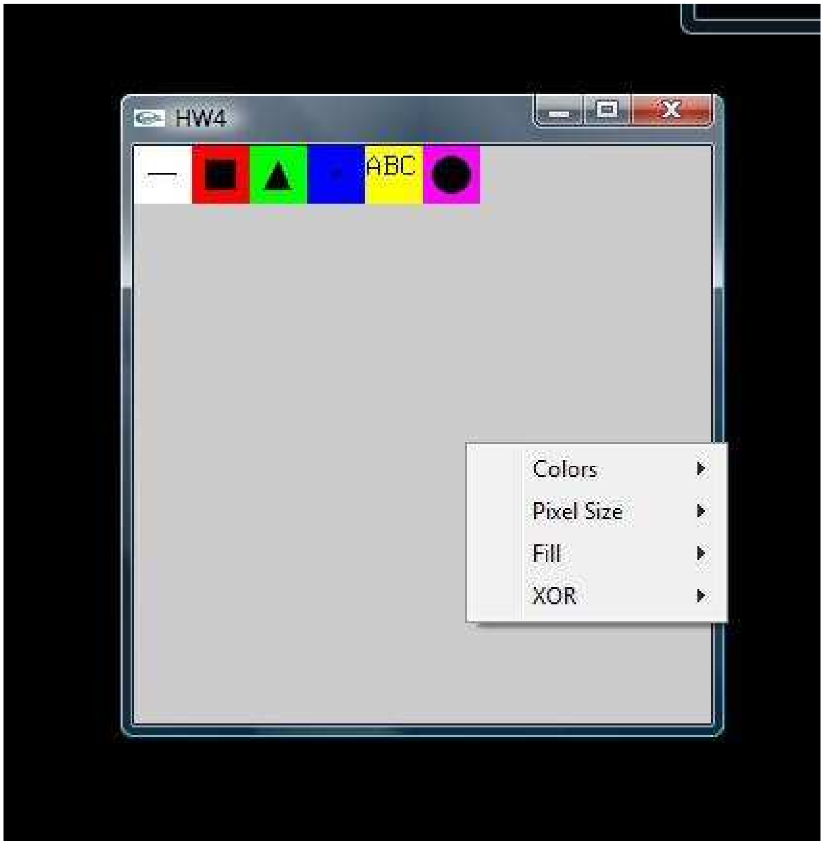
Homework 4, Email to Peter by Sep 30

(110 points total)

1. Extend the paint program presented in class as follows. Hand in the code and screendump(s) showing the new functionality. Document anything that is unclear.
   a. *(10 points)* Add 3 more colors of your choosing.
   b. *(10)* Add a main menu entry and submenu to enable or disable writing with XOR instead of with copy.
   c. *(10*) Add an entry to the permanent menu on the top to draw a circle; the user selects opposite ends of a diameter.
   d. *(10)* Modify the program to keep drawing primitives until you select something else from the top menu. That is, after you click on *line*, each pair of future clicks causes another line. (Now the program ignores left clicks after the first 2 until you again select something on the top menu.)
   e. *(20)* Modify the program to use a display list, so that the objects are redrawn when you redisplay the window. (This may be surprisingly hard.)



2. *(5)* Why does *paint.c* contain this line:

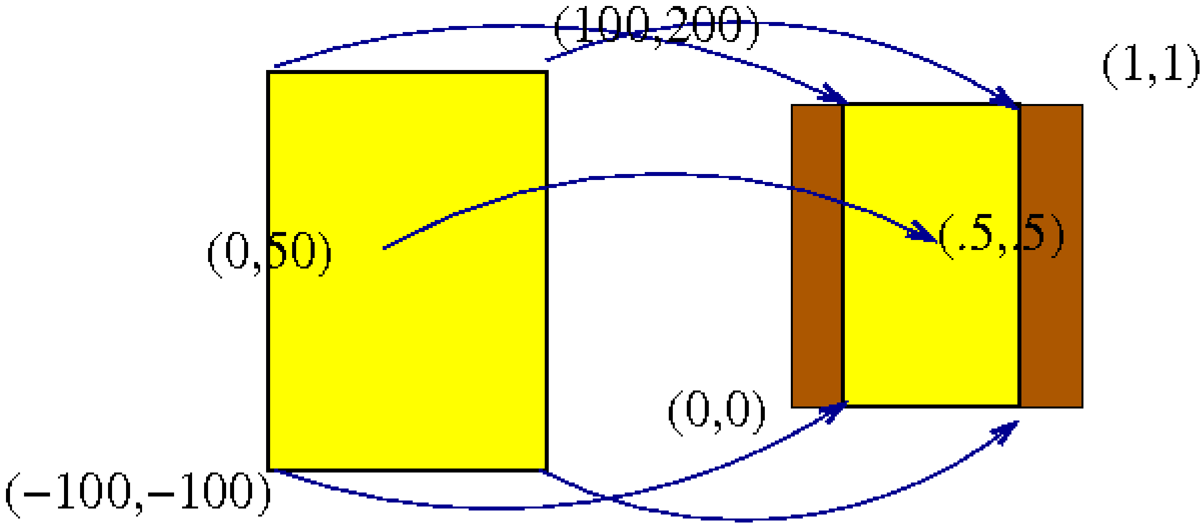
   y=wh-y;

   Mouse returns X and Y from the top, but OpenGL has Y measured from the bottom. You need to subtract the window height from the mouse Y value to get distance from the bottom to draw where you click.

3. *(15)* Assume that you have a model coordinate space with corners (-100,-100) and (100,200). You wish to map points in it to a window with corners (0,0) and (1,1). (0,50) should map to (0.5,0.5). X and Y should scale the same. Compute what the transformations for X and Y should be, in this form:

   X' = s X + dx
   Y' = s Y + dy

   I.e., tell me what s, dx, and dy are.



del(x)=200, del(y)=300. => Possible scaling S=1/200 or S=1/300.
Choosing the lower scaling in order to maintain same scale so, S=1/300
Substituting values of coordinates in equation yields:
0.5=1/300 *0 + dx   and   0.5=50/300 + dy
therefore,   dx = 0.5   and   dy=1/3

4. *(5)* What is the difference between a *Motion* callback and a *PassiveMotion* callback?
   The motion callback for a window is called when the mouse moves within the window while one or more mouse buttons are pressed. The passive motion callback for a window is called when the mouse moves within the window while *no* mouse buttons are pressed.

5. *(5)* How can *glutPostRedisplay* make your program a little more efficient?
   Multiple calls to glutPostRedisplay before the next display callback opportunity generates only a single redisplay callback.

6. *(5)* When creating a menu, when would you need to store the menu id that is returned?
   You would need to store the menu ID if you wanted to alter the menu dynamically or create sub-menus.

7. *(5)* Since a menu has only one callback, which is called regardless of which item you click on, how does the callback know which item you clicked on?
   An integer is returned that identifies the menu item that was clicked.

8. *(10)* Edit *alpha.c* to remove the line that enables GL_BLEND, compile and run. Report on how the output is different.
   The triangles are no longer "fade" from opaque to translucent - they appear opaque throughout.