Important GLUT Functions

(See also the OpenGL Programming Guide, p. 17)

GLUT Window Setup:

*void glutInit(int *argc, char **argv)* runs GLUT internal initialization routines and captures system-specific command line parameters. Goes in main() and should probably be the first line of code executed.

void glutInitDisplayMode(unsigned int mode) tells GLUT what sort of OpenGL window you want and what features you want exposed for this program. NOTE: This sets important OpenGL state! If you specify the wrong features or do not specify everything you need, some features will be undefined (will fail to work silently)! Possible modes include: GLUT_RGBA, GLUT_INDEX, GLUT_SINGLE, GLUT_DOUBLE, GLUT_DEPTH, GLUT_ALPHA, GLUT_STENCIL, GLUT_ACCUM, GLUT_MULTISAMPLE, GLUT_STEREO, and GLUT_LUMINANCE. Default: glutInitDisplayMode(GLUT_RGBA | GLUT_SINGLE)

void glutInitWindowPosition(int x, int y) sets the initial (suggested) window position.

void glutInitWindowSize(int width, int height) sets the (suggested) width and height of the drawing region.

*int glutCreateWindow(char *string)* creates a window with name *string*. It returns an identifier for the window (which allows for manipulation of all windows in multi-window programs). NOTE: the window is not actually created (or shown) until *glutMainLoop()* is called, so GL commands before the *glutMainLoop()* are likely to do nothing, or have undefined results.

GLUT User Interaction Setup: (or Callback Setup)

*void glutDisplayFunc(void (*func)(void))* sets the function that draws to the screen. This display function should basically handle all the GL drawing commands. All GLUT programs should setup a display function!

*void glutReshapeFunc(void (*func)(int width, int height))* sets the function that handles user changes to the GL window size. The inputs are the new user-set window width and height.

*void glutKeyboardFunc(void (*func)(unsigned char key, int x, int y))* sets the function that handles keystrokes occurring while the GL window is active. The inputs are the key pressed and the (x,y) coordinates of the mouse when the keystroke occurred. Valid comparisons include *if (key=='Q') ...* or *if (key=='f') ...*, for instance.

*void glutMouseFunc(void (*func)(int button, int state, int x, int y))* sets the function that handles mouse clicks occurring in the GL window. The parameter 'button' is either GLUT_BUTTON_LEFT, GLUT_BUTTON_MIDDLE, or GLUT_BUTTON_RIGHT. The 'state' parameter is either GLUT_UP or GLUT_DOWN, and (x,y) is the window coordinates of the mouse at the time of the click.

*void glutMotionFunc(void (*func)(int x, int y))* sets the function that handles mouse movement while a mouse button is depressed.

*void glutPassiveMotionFunc(void (*func)(int x, int y))* sets the function that handles mouse movements while no mouse buttons are pressed.

*void glutIdleFunc(void (*func)(void))* sets the *idle* function, which is called whenever no other user interface events are pending. Usually the idle function updates viewer parameters (in the case of dynamic OpenGL content) and calls glutPostRedisplay(). Do not call the display function directly!

Other Important Functions:

void glutMainLoop(void) displays the GLUT windows and starts the user interface loop (which includes drawing to the screen, and handling events though the idle function, mouse function, keyboard function, etc.) This should be the last line of code in your *main()* function, as the function never returns.

void glutPostRedisplay(void) notes that the GL window needs to be redrawn. The display function should *not* be called directly, because multiple glutPostRedisplay() calls can be combined into one, helping eliminate redundant redraws (which would slow interactivity).