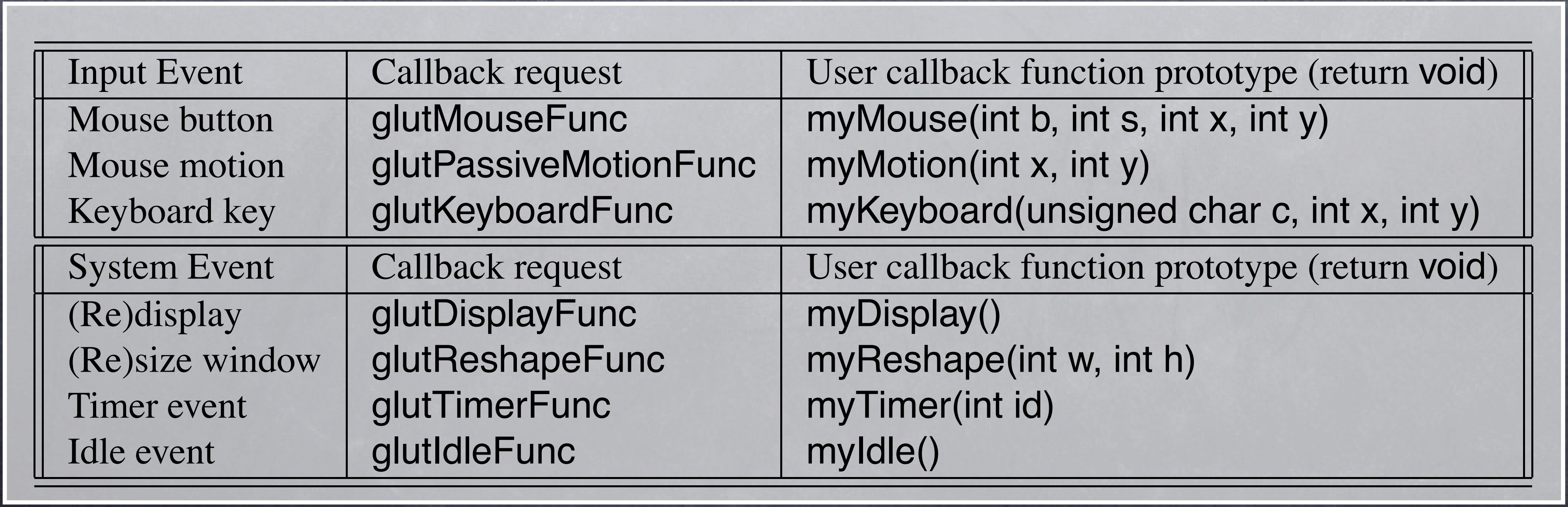
Event Driven Programming

- Allows for interactivity
- “Register” “callbacks” with the graphics system
- Two event types: input & system

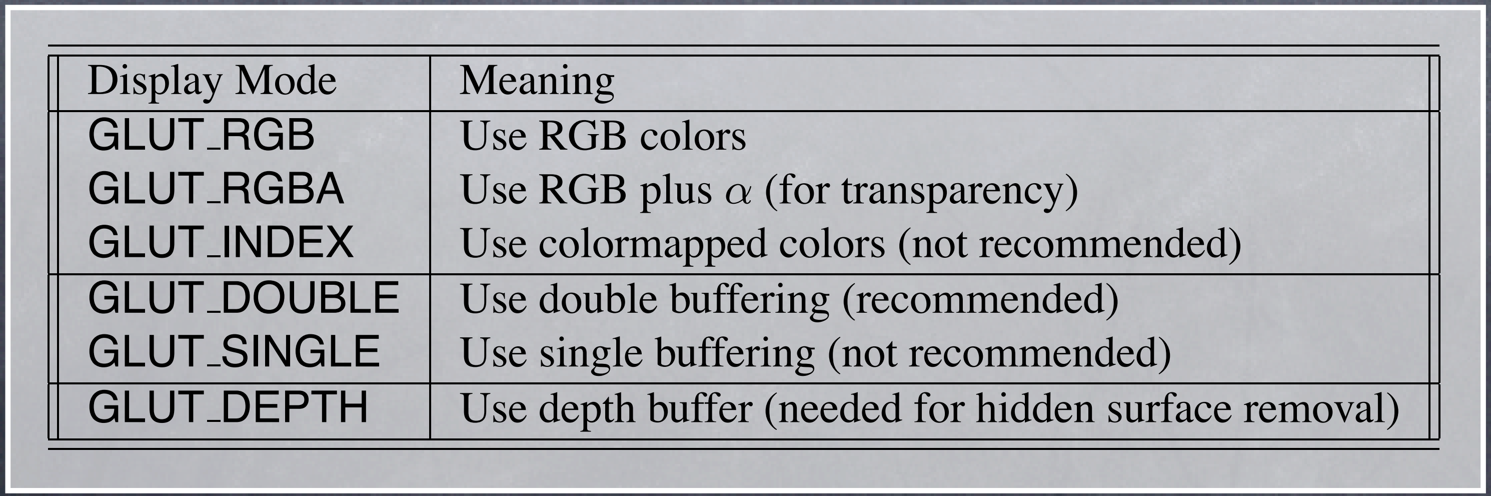


Input Event	Callback request	User callback function prototype (return void)
Mouse button	glutMouseFunc	myMouse(int b, int s, int x, int y)
Mouse motion	glutPassiveMotionFunc	myMotion(int x, int y)
Keyboard key	glutKeyboardFunc	myKeyboard(unsigned char c, int x, int y)

System Event	Callback request	User callback function prototype (return void)
(Re)display	glutDisplayFunc	myDisplay()
(Re)size window	glutReshapeFunc	myReshape(int w, int h)
Timer event	glutTimerFunc	myTimer(int id)
Idle event	glutIdleFunc	myIdle()

Glut Calls

- glutInit(int*, char**): probably the first call of your program, takes argc & argv
- glutInitDisplayMode(): initializes the frame buffer



Display Mode	Meaning
GLUT_RGB	Use RGB colors
GLUT_RGBA	Use RGB plus α (for transparency)
GLUT_INDEX	Use colormapped colors (not recommended)
GLUT_DOUBLE	Use double buffering (recommended)
GLUT_SINGLE	Use single buffering (not recommended)
GLUT_DEPTH	Use depth buffer (needed for hidden surface removal)

Glut Calls

- glutInitWindowSize(int width, int height): “suggests” a particular window size
- glutInitPosition(int x, int y): “suggests” a window location
- glutCreateWindow(char *): requests creation of the window. Can’t start doing stuff till we get notification (via the display callback) that the window has been created.

The Display Callback

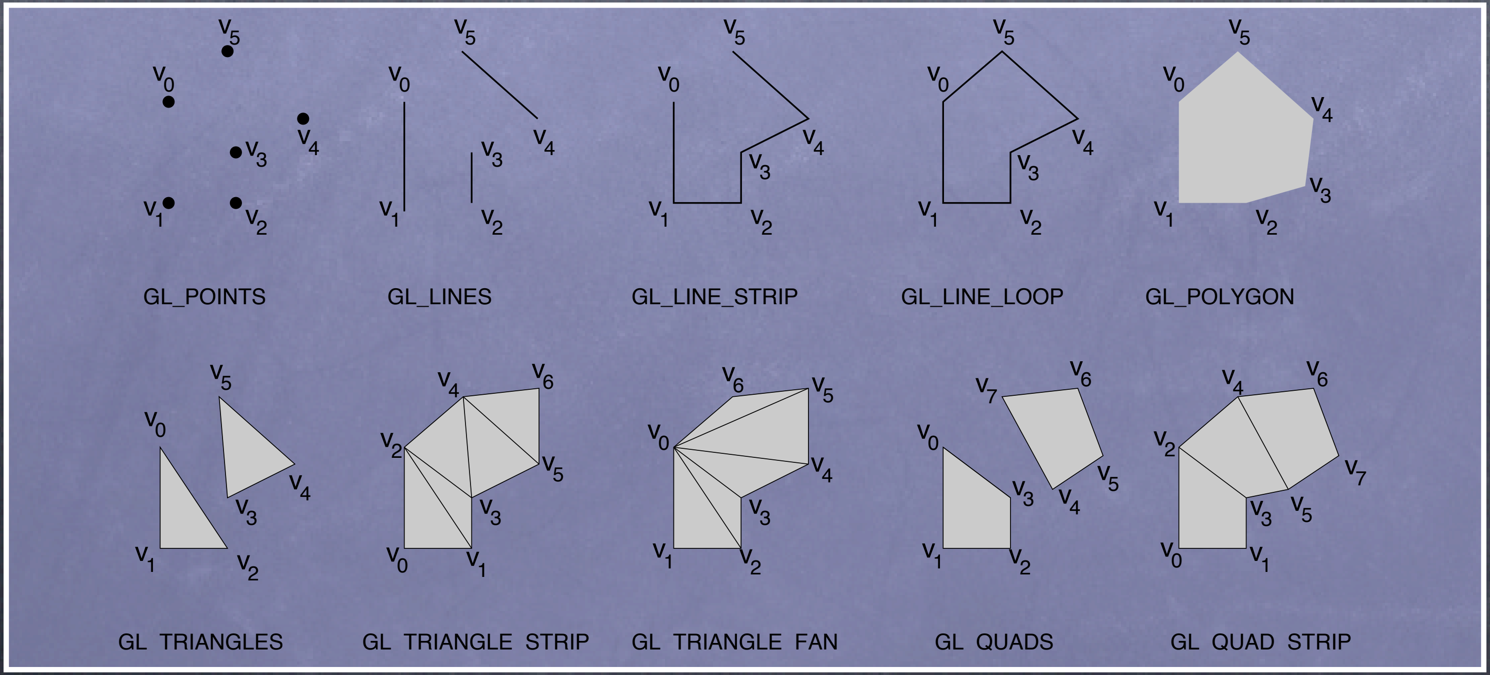
- Called upon:
  - the initial creation of the window
  - whenever the window is uncovered by the removal of some overlapping window
  - whenever your program requests that it be redrawn (via glutPostRedisplay())
- Start by clearing with glClear()
- End by swapping buffers glutSwapBuffers()

Drawing Attributes

- You can draw in different ways (color, line thickness, point size), GL uses whatever is the current style
- Functions you might find useful: glColor3f(), glPointSize(), glLineWidth(), glLineStipple(), glPolygonStipple()
- Other attributes exist and are useful for shading in 3D.

Drawing Polygons

- Specific calls (e.g. glRectf()) exist, but we get more flexibility with glBegin(mode), glVertex(), and glEnd()
- Limited gl calls allowed in between glBegin() and glEnd()



Viewport

- Set using glViewPort(int x, int y, int width, int height)
- Defines the part of the window you'll be drawing in.
- Typically called in the window reshape callback and usually covers the whole window
- Should probably call glutPostRedisplay() after changing viewpoirt

A Few Words of Advice

- Don't reinvent the wheel. Use the Standard Template Library and other tools. If you need to sort a list, use the STL sort. It will save you time.
- Don't try to do the whole project at once, work in small steps so that if you introduce a bug you can find it.
- Get started!!! Maybe try to be able to read the input file and use openGL to draw the polygons by monday.

Writing Images

- Set up your own “framebuffer”
- Fill it
- Dump it to a file (ppms are simple, libraries exist to help you with other formats)