

GLUT

glutInit (int argc, char ** argv)

- before any other GLUT routine
- processes command line options, system dependent

glutInitDisplayMode (unsigned int mode)

- controls color depth - RGB vs indexed
- # of buffers - single or double
 - depth, stencil, accum
- default is GLUT_RGBA / GLUT_SINGLE
- usually want GLUT_RGBA / GLUT_DOUBLE / GLUT_DEPTH

glutInitWindowSize (int width, int height)

glutInitWindowPosition (int x, int y)

int glutCreateWindow (char * name)

- opens the window - so do this after previous
- gives an int value back, can be used to ID window (if using more than 1)

Handling Events

void glutDisplayFunc (void (*func) (void))

- function to be called whenever window needs to be redrawn
- called when
 - initially opened
 - window popped / damaged
 - glutPostRedisplay()

void glutReshapeFunc (void (*func) (int width, int height))

- function to be called when window is reshaped
- usually you call glViewport
- if not provided, default behavior is to call glViewport (0, 0, width, height)

glutMainLoop()

(6)

void glutKeyboardFunc(void (*func)(unsigned int key, int x, int y))
 → when key pressed
 → key == (unsigned int) 'a'
 → x, y are location of mouse

void glutMouseFunc(void (*func)(int button, int state, int x, int y))

button: GLUT_LEFT_BUTTON
 GLUT_RIGHT_BUTTON
 GLUT_MIDDLE_BUTTON

state: GLUT_UP
 GLUT_DOWN

x, y: mouse coordinates, 0,0 is top left corner

void glutMotionFunc(void (*func)(int x, int y))
 - when mouse moves while button pressed.

void glutIdleFunc(void (*func)(void));
 - something to execute when idle
 - animation, force update

⇒ EXAMPLE CODE

lab2.cube.c

→ add to the keyboard handler
 so that cube changes color.

GLUT Menus

`int glutCreateMenu (void (*func) (int))`

→ creates a new pop-up menu & returns an identifier; current menu is set to the new menu. Takes a menu handling function as an argument. This function usually just performs a switch on an int.

`glutAddMenuEntry (char * text, int id)`

→ adds an entry with a specified value to current menu
bottom of

`glutAttachMenu (int button)`

→ attaches menu to a mouse button (replacing the normal mouse button event handler
GLUT_RIGHT_BUTTON

Example

```
#define MENU_RED 1
#define MENU_GREEN 2
#define MENU_BLUE 3
float r = 0; g = 0, b = 0;
void crackMenu () {
    int menu;
    menu = glutCreateMenu (processMenu);
    glutAddMenuEntry ("Red", MENU_RED);
    glutAddMenuEntry ("Green", MENU_GREEN);
    glutAddMenuEntry ("Blue", MENU_BLUE);
    glutAttachMenu (GLUT_RIGHT_BUTTON);
}
```

```
void processMenu (int option)
{
    switch (option) {
```

```
    case MENU_RED; r = 1; g = b = 0; break
    case MENU_GREEN; r = b = 0; g = 1; break
    case MENU_BLUE; b = 1; r = g = 0; break
```

```
    }
}
```

void glutDetachMenu (int button)
→ removes menu from button click

void glutDestroyMenu (int ID)

void glutAddSubMenu (char * name, int menuIndex)
→ adds one menu to another menu

OpenGL Primitives

glBegin ()
glEnd

① Points → GL_POINTS

→ small squares

→ change size with glPointSize (float)

② Lines → GL_LINES → each pair of points is a line

→ GL_LINE_STRIP → connect each point to last with a line

- glLineWidth (float) → GL_LINE_LOOP → same as above, but connect last to first.

③ Triangles → GL_TRIANGLES → 3 vertices = 1 triangle

→ GL_TRIANGLE_STRIP

→ GL_TRIANGLE_FAN

④ Quads → GL_QUADS → 4 vertices = 1 quad

GL_QUAD_STRIP → after 4, every 2 vertices adds a new quad.

⑤ Polygon → GL_POLYGON

Between glBegin & glEnd

glVertex

glNormal

glColor

glMaterial

glTexCoord

glArrayElement

glCullList

glEvalCoord

glArrayElement

glIndex

OpenGL Application

Main

- initiate GLUT / QT (set up window creation)
- call init gl function
- set up display, resize, keyboard, mouse, etc event handlers
- begin rendering loop

Init GL

- things you only need to do once to setup OpenGL
- glClearColor
- lighting / materials setup
- glEnable (GL_DEPTH_TEST)

Display

- as little as possible here
- drawing
- glClear

Resize

- everytime window gets resized
- adjust viewport
- set up projection

- Assignment
- Plagiarism