

You will need to following OpenGL and glut commands for program 1

In order to process menus you will need to know the following GLUT calls:

int glutCreateMenu(void (* f)(int value))

Creates a top-level menu that uses the callback f(), which is passed an integer value for the menu entry. A unique identifier is returned for the menu created.

void glutAddMenuEntry(char *name, int value)

Adds an entry with name displayed to the current menu; value is returned to the menu callback.

void glutAttachMenu(int button)

Attached the current menu to the specified mouse button (GLUT_RIGHT_BUTTON, GLUT_MIDDLE_BUTTON, or GLUT_LEFT_BUTTON).

Since you will need two menus and also need the mouse to register user interaction – I recommend using the middle and left mouse button for menus. Alternatively, you could use submenus to choose the color with the following command:

void glutAddSubMenu(char *name, int menu)

Adds a submenu entry name as the next entry in the current menu. The value of menu is the ID of the submenu returned when the submenu was created.

For example, I added a simple menu with a submenu by making the following changes to my simple opengl program (skeleton.cpp). I added two functions to process the menus, “mymenu” and “process_submenu” function and added the appropriate lines to main:

```
void mymenu(int value) {
    if (value == 1) printf("hi\n");
    if (value == 2) exit(0);
}

void process_submenu(int value) {
    if (value == 1) printf("red\n");
    if (value == 2) printf("blue\n");
}

int main(int argc, char** argv) {
    int submenu, menu1;

    glutInit(&argc, argv);
    glutInitWindowSize(100, 100);
    glutCreateWindow("My First Window");
    glutDisplayFunc(display);
    glClearColor(0.0, 0.0, 0.0, 0.0);

    submenu = glutCreateMenu(process_submenu);
    glutAddMenuEntry("red", 1);
    glutAddMenuEntry("blue", 2);
```

```
menu1 = glutCreateMenu(mymenu);
glutAddMenuEntry("Say hi", 1);
glutAddMenuEntry("exit", 2);
glutAddSubMenu("colors", submenu);
glutAttachMenu(GLUT_RIGHT_BUTTON);

glutMainLoop();
}
```

GLUT function descriptions are from “OpenGL: A Primer” by Edward Angel.

In order to draw the shapes for this program you will need the following calls:

You can either draw a rectangle by using the OpenGL function:

glRecti(int x1, int y1, int x2, int y2);

Or, you could also draw a rectangle using:

glBegin(GL_POLYGON);

There is not a single command to draw a circle – you will need to calculate points on the exterior of a circle of the given radius and create a filled polygon. I encourage you to use GL_TRIANGLE_FAN for example.