

```
1  #include "gwm.h"
2
3  int gWidth = 0;
4  int gHeight = 0;
5  float gAngle = 0.0f;
6
7  bool animate = false;
8
9  void main (void)
10 {
11     // declarations
12     void keyboardFunc(unsigned int);
13     void displayFunc(void);
14     void updateFunc(void);
15     void reshapeFunc(int, int);
16
17     // code
18     gwmKeyboardCallback(keyboardFunc);
19     gwmDisplayCallback(displayFunc);
20     gwmUpdateCallback(updateFunc);
21     gwmReshapeCallback(reshapeFunc);
22
23     gwmCreateWindow("3D Animation", 100, 100, 800, 600);
24     gwmEventLoop();
25 }
26
27 void keyboardFunc(unsigned int key)
28 {
29     switch(key)
30     {
31         // VK_ESCAPE
32         case 0x1B:
33             gwmExitEventLoop();
34             break;
35
36         // 0x20
37         case VK_SPACE:
38             if (animate == true)
39             {
40                 animate = false;
41             }
42             else
43             {
44                 animate = true;
45             }
46             break;
47     }
48 }
49
50
51 void reshapeFunc(int width, int height)
52 {
```

```
53     glMatrixMode(GL_PROJECTION);
54     glLoadIdentity();
55
56     gWidth = width;
57     gHeight = height;
58
59     glViewport(0, 0, gWidth, gHeight);
60     gluPerspective(45.0, (float)width / (float)height, 0.1f, 100.0f);
61 }
62
63 void displayFunc(void)
64 {
65     // code
66     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
67
68     glMatrixMode(GL_MODELVIEW);
69     glLoadIdentity();
70     glTranslatef(0.0f, 0.0f, -6.0f);
71     glScalef(0.8f, 0.8f, 0.8f);
72     glRotatef(gAngle, 1.0f, 1.0f, 1.0f);
73
74     glBegin(GL_QUADS);
75
76     /* Top */
77     glColor3f(1.0f, 0.0f, 0.0f);
78
79     glVertex3f(1.0f, 1.0f, -1.0f);
80     glVertex3f(-1.0f, 1.0f, -1.0f);
81     glVertex3f(-1.0f, 1.0f, 1.0f);
82     glVertex3f(1.0f, 1.0f, 1.0f);
83
84     /* Bottom */
85     glColor3f(0.0f, 1.0f, 0.0f);
86
87     glVertex3f(1.0f, -1.0f, -1.0f);
88     glVertex3f(-1.0f, -1.0f, -1.0f);
89     glVertex3f(-1.0f, -1.0f, 1.0f);
90     glVertex3f(1.0f, -1.0f, 1.0f);
91
92     /* Front */
93     glColor3f(0.0f, 1.0f, 1.0f);
94
95     glVertex3f(1.0f, 1.0f, 1.0f);
96     glVertex3f(-1.0f, 1.0f, 1.0f);
97     glVertex3f(-1.0f, -1.0f, 1.0f);
98     glVertex3f(1.0f, -1.0f, 1.0f);
99
100    /* Back */
101    glColor3f(0.0f, 0.0f, 1.0f);
102
103    glVertex3f(1.0f, 1.0f, -1.0f);
104    glVertex3f(-1.0f, 1.0f, -1.0f);
```

```
105     glVertex3f(-1.0f, -1.0f, -1.0f);
106     glVertex3f(1.0f, -1.0f, -1.0f);
107
108     /* Right */
109     glColor3f(1.0f, 0.0f, 1.0f);
110
111     glVertex3f(1.0f, 1.0f, -1.0f);
112     glVertex3f(1.0f, 1.0f, 1.0f);
113     glVertex3f(1.0f, -1.0f, 1.0f);
114     glVertex3f(1.0f, -1.0f, -1.0f);
115
116     /* Left */
117     glColor3f(1.0f, 1.0f, 0.0f);
118
119     glVertex3f(-1.0f, 1.0f, -1.0f);
120     glVertex3f(-1.0f, 1.0f, 1.0f);
121     glVertex3f(-1.0f, -1.0f, 1.0f);
122     glVertex3f(-1.0f, -1.0f, -1.0f);
123
124     glEnd();
125
126     gwmSwapBuffers();
127 }
128
129 void updateFunc(void)
130 {
131     if (animate == true)
132     {
133         gAngle = gAngle + 0.5f;
134     }
135 }
136
```