

```
1  #include "gwm.h"
2
3  int gWidth = 0;
4  int gHeight = 0;
5
6  void main (void)
7  {
8      // declarations
9      void keyboardFunc(unsigned int);
10     void displayFunc(void);
11     void reshapeFunc(int, int);
12
13     // code
14     gwmKeyboardCallback(keyboardFunc);
15     gwmDisplayCallback(displayFunc);
16     gwmReshapeCallback(reshapeFunc);
17
18     gwmCreateWindow("Viewports", 100, 100, 800, 600);
19     gwmEventLoop();
20 }
21
22 void keyboardFunc(unsigned int key)
23 {
24     switch(key)
25     {
26         // VK_ESCAPE
27         case 0x1B:
28             gwmExitEventLoop();
29             break;
30
31         case '0':
32         case VK_NUMPAD0:
33             glViewport(0, 0, gWidth, gHeight);
34             break;
35
36         case '1':
37         case VK_NUMPAD1:
38             glViewport(0, 0, gWidth / 2.0f, gHeight / 2.0f);
39             break;
40
41         case '2':
42         case VK_NUMPAD2:
43             glViewport(gWidth / 2, 0, gWidth / 2.0f, gHeight / 2.0f);
44             break;
45
46         case '3':
47         case VK_NUMPAD3:
48             glViewport(gWidth / 2, gHeight / 2, gWidth / 2.0f, gHeight / 2.0f);
49             break;
50
51         case '4':
52         case VK_NUMPAD4:
```

```
53     glViewport(0, gHeight / 2, gWidth / 2.0f, gHeight / 2.0f);
54     break;
55
56     case '5':
57     case VK_NUMPAD5:
58         glViewport(0, 0, gWidth / 2.0f, gHeight);
59         break;
60
61     case '6':
62     case VK_NUMPAD6:
63         glViewport(gWidth / 2, 0.0f, gWidth / 2.0f, gHeight);
64         break;
65
66     case '7':
67     case VK_NUMPAD7:
68         glViewport(0, gHeight / 2, gWidth, gHeight / 2.0f);
69         break;
70
71     case '8':
72     case VK_NUMPAD8:
73         glViewport(0, 0, gWidth, gHeight / 2.0f);
74         break;
75
76     case '9':
77     case VK_NUMPAD9:
78         glViewport(gWidth / 4, gHeight / 4, gWidth / 2.0f, gHeight / 2.0f);
79         break;
80     }
81 }
82
83 void reshapeFunc(int width, int height)
84 {
85     glMatrixMode(GL_PROJECTION);
86     glLoadIdentity();
87
88     gWidth = width;
89     gHeight = height;
90
91     glViewport(0, 0, gWidth, gHeight);
92     gluPerspective(45.0, (float)width / (float)height, 0.1f, 100.0f);
93 }
94
95 void displayFunc(void)
96 {
97     // code
98     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
99
100    glMatrixMode(GL_MODELVIEW);
101    glLoadIdentity();
102    glTranslatef(0.0f, 0.0f, -3.0f);
103
104    glBegin(GL_TRIANGLES);
```

```
105
106     glColor3f(1.0f, 0.0f, 0.0f);
107     glVertex2f(0.0f, 1.0f);
108     glColor3f(0.0f, 1.0f, 0.0f);
109     glVertex2f(-1.0f, -1.0f);
110     glColor3f(0.0f, 0.0f, 1.0f);
111     glVertex2f(1.0f, -1.0f);
112
113     glEnd();
114
115     gwmSwapBuffers();
116 }
117
118
```