

```
1  #include <GL/glew.h>
2  #include <GL/glut.h>
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  static char* vsSource = "#version 120 \n\
7  in vec4 aPosition; \n\
8  in vec4 aColor; \n\
9  out vec4 vColor; \n\
10 uniform vec4 uvec; \n\
11 void main(void) { \n\
12     gl_Position = aPosition + uvec; \n\
13     vColor = aColor; \n\
14 }";
15
16 static char* fsSource = "#version 120 \n\
17 in vec4 vColor; \n\
18 void main(void) { \n\
19     gl_FragColor = vColor; \n\
20 }";
21
22 GLuint vs = 0;
23 GLuint fs = 0;
24 GLuint prog = 0;
25
26 char buf[1024];
27 float dist = 0.0f;
28
29 GLfloat vertices[] = {
30     -0.2, -0.2, -0.2, 1.0, // 0
31     -0.2, -0.2, +0.2, 1.0, // 1
32     -0.2, +0.2, -0.2, 1.0, // 2
33     -0.2, +0.2, +0.2, 1.0, // 3
34     +0.2, -0.2, -0.2, 1.0, // 4
35     +0.2, -0.2, +0.2, 1.0, // 5
36     +0.2, +0.2, -0.2, 1.0, // 6
37     +0.2, +0.2, +0.2, 1.0, // 7
38 };
39
40
41 GLfloat colors[] = {
42     1.0, 0.0, 0.0,
43     0.0, 1.0, 0.0,
44     0.0, 0.0, 1.0,
45     1.0, 0.0, 0.0,
46     0.0, 1.0, 0.0,
47     0.0, 0.0, 1.0,
48     1.0, 0.0, 0.0,
49     0.0, 1.0, 0.0
50 };
51
52 GLushort indices[] = { // 36 points, 12 triangles
53     0, 4, 6,
54     6, 2, 0,
55     4, 5, 7,
56     7, 6, 4,
```

```
57     1, 3, 7,
58     7, 5, 1,
59     0, 2, 3,
60     3, 1, 0,
61     2, 6, 7,
62     7, 3, 2,
63     0, 1, 5,
64     5, 4, 0,
65 };
66
67 void myinit(void) {
68     GLuint status;
69
70     printf("***** Your student number and name *****\n");
71     vs = glCreateShader(GL_VERTEX_SHADER);
72     glShaderSource(vs, 1, &vsSource, NULL);
73     glCompileShader(vs);
74     glGetShaderiv(vs, GL_COMPILE_STATUS, &status);
75     printf("vs compile status = %s\n", (status == GL_TRUE) ? "true" :
76           "false");
77     glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
78     printf("vs log = [%s]\n", buf);
79
80     fs = glCreateShader(GL_FRAGMENT_SHADER);
81     glShaderSource(fs, 1, &fsSource, NULL);
82     glCompileShader(fs);
83     glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
84     printf("fs compile status = %s\n", (status == GL_TRUE) ? "true" :
85           "false");
86     glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
87     printf("fs log = [%s]\n", buf);
88
89     prog = glCreateProgram();
90     glAttachShader(prog, vs);
91     glAttachShader(prog, fs);
92     glLinkProgram(prog);
93     glGetProgramiv(prog, GL_LINK_STATUS, &status);
94     printf("program link status = %s\n", (status == GL_TRUE) ? "true" :
95           "false");
96     glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
97     printf("link log = [%s]\n", buf);
98     glValidateProgram(prog);
99     glGetProgramiv(prog, GL_VALIDATE_STATUS, &status);
100    printf("program validate status = %s\n", (status == GL_TRUE) ? "true" :
101           "false");
102    glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
103    printf("validate log = [%s]\n", buf);
104    glUseProgram(prog);
105
106    GLuint loc;
107    GLuint vbo[1];
108    // using vertex buffer object
109    glGenBuffers(1, vbo);
110    glBindBuffer(GL_ARRAY_BUFFER, vbo[0]);
111    glBufferData(GL_ARRAY_BUFFER, 2 * 8 * 4 * sizeof(GLfloat), NULL,
112                GL_STATIC_DRAW);
```

```
108     glBufferSubData(GL_ARRAY_BUFFER, 0, 8 * 4 * sizeof(GLfloat), vertices);
109     glBufferSubData(GL_ARRAY_BUFFER, 8 * 4 * sizeof(GLfloat), 8 * 4 * sizeof
    (GLfloat),
110         colors);
111
112     loc = glGetAttribLocation(prog, "aPosition");
113     glEnableVertexAttribArray(loc);
114     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *)0);
115
116     loc = glGetAttribLocation(prog, "aColor");
117     glEnableVertexAttribArray(loc);
118     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *) (3 * 4 *
    sizeof(GLfloat)));
119
120 }
121
122 void mykeyboard(unsigned char key, int x, int y) {
123     switch (key) {
124     case 27: // ESCAPE
125         exit(0);
126         break;
127     }
128 }
129
130
131
132 void myidle(void) {
133     dist += 0.0001f;
134     if (dist > 1.5)
135         dist = 0.0f;
136
137     // redisplay
138     glutPostRedisplay();
139 }
140
141
142 void mydisplay(void) {
143     GLuint loc;
144     GLfloat d[4] = { 0.2, 0.5, 0.0, 0.0 };
145
146
147     glClearColor(0.7f, 0.7f, 0.7f, 1.0f); // gray
148     glClear(GL_COLOR_BUFFER_BIT);
149
150     loc = glGetUniformLocation(prog, "udvec");
151     glUniform4fv(loc, 1, d);
152
153
154
155     glDrawElements(GL_TRIANGLES, 12 * 3, GL_UNSIGNED_SHORT, indices);
156
157     glFlush();
158     glutSwapBuffers();
159 }
160
161 int main(int argc, char* argv[]) {
```

```
162     glutInit(&argc, argv);
163     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
164     glutInitWindowSize(500, 500);
165     glutInitWindowPosition(0, 0);
166     glutCreateWindow("*** Your Student Number and Name ***");
167     glutDisplayFunc(mydisplay);
168     glutIdleFunc(myidle);
169     glutKeyboardFunc(mykeyboard);
170     glewInit();
171     myinit();
172     glutMainLoop();
173     return 0;
174 }
175
```