

```
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 WnW
7  in vec4 aPosition; WnW
8  in vec4 aColor; WnW
9  out vec4 vColor; WnW
10 uniform float udist; WnW
11 void main(void) { WnW
12     gl_Position.x = aPosition.x + udist; WnW
13     gl_Position.yzw = aPosition.yzw; WnW
14     vColor = aColor; WnW
15 }";
16
17 static char* fsSource = "#version 120 WnW
18 in vec4 vColor; WnW
19 void main(void) { WnW
20     gl_FragColor = vColor; WnW
21 }";
22
23 GLuint vs = 0;
24 GLuint fs = 0;
25 GLuint prog = 0;
26
27 char buf[1024];
28 float dist = 0.0f;
29
30 GLfloat vertices[] = {
31     -0.5, -0.5, 0.0, 1.0,
32     +0.5, -0.5, 0.0, 1.0,
33     -0.5, +0.5, 0.0, 1.0,
34 };
35
36 GLfloat colors[] = {
37     1.0, 0.0, 0.0, 1.0, // red
38     0.0, 1.0, 0.0, 1.0, // green
39     0.0, 0.0, 1.0, 1.0, // blue
40 };
41
42 void myinit(void) {
43     GLuint status;
44
45     printf("***** Your student number and name *****Wn");
46     vs = glCreateShader(GL_VERTEX_SHADER);
47     glShaderSource(vs, 1, &vsSource, NULL);
48     glCompileShader(vs);
49     glGetShaderiv(vs, GL_COMPILE_STATUS, &status);
50     printf("vs compile status = %sWn", (status == GL_TRUE) ? "true" :
51         "false");
52     glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
53     printf("vs log = [%s]Wn", buf);
54
55     fs = glCreateShader(GL_FRAGMENT_SHADER);
56     glShaderSource(fs, 1, &fsSource, NULL);
```

```

56     glCompileShader(fs);
57     glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
58     printf("fs compile status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
59     glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
60     printf("fs log = [%s]\n", buf);
61
62     prog = glCreateProgram();
63     glAttachShader(prog, vs);
64     glAttachShader(prog, fs);
65     glLinkProgram(prog);
66     glGetProgramiv(prog, GL_LINK_STATUS, &status);
67     printf("program link status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
68     glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
69     printf("link log = [%s]\n", buf);
70     glValidateProgram(prog);
71     glGetProgramiv(prog, GL_VALIDATE_STATUS, &status);
72     printf("program validate status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
73     glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
74     printf("validate log = [%s]\n", buf);
75     glUseProgram(prog);
76
77     GLuint vbo[1];
78     // using vertex buffer object
79     glGenBuffers(1, vbo);
80     glBindBuffer(GL_ARRAY_BUFFER, vbo[0]);
81     glBufferData(GL_ARRAY_BUFFER, 2 * 3 * 4 * sizeof(GLfloat), NULL,
           GL_STATIC_DRAW);
82     glBufferSubData(GL_ARRAY_BUFFER, 0, 3 * 4 * sizeof(GLfloat), vertices);
83     glBufferSubData(GL_ARRAY_BUFFER, 3 * 4 * sizeof(GLfloat), 3 * 4 * sizeof
           (GLfloat),
84         colors);
85 }
86
87 void mykeyboard(unsigned char key, int x, int y) {
88     switch (key) {
89         case 27: // ESCAPE
90             exit(0);
91             break;
92     }
93 }
94
95
96
97 void myidle(void) {
98     dist += 0.0001f;
99     if (dist > 1.5)
100         dist = 0.0f;
101
102     // redisplay
103     glutPostRedisplay();
104 }
105
106

```

```
107 void mydisplay(void) {
108     GLuint loc;
109
110     glClearColor(0.7f, 0.7f, 0.7f, 1.0f); // gray
111     glClear(GL_COLOR_BUFFER_BIT);
112
113     loc = glGetAttribLocation(prog, "aPosition");
114     glEnableVertexAttribArray(loc);
115     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *)0);
116
117     loc = glGetAttribLocation(prog, "aColor");
118     glEnableVertexAttribArray(loc);
119     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *) (3 * 4 *
120         sizeof(GLfloat)));
121
122     loc = glGetUniformLocation(prog, "udist");
123     glUniform1f(loc, dist);
124
125     glDrawArrays(GL_TRIANGLES, 0, 3);
126
127     glFlush();
128     glutSwapBuffers();
129 }
130 int main(int argc, char* argv[]) {
131     glutInit(&argc, argv);
132     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
133     glutInitWindowSize(500, 500);
134     glutInitWindowPosition(0, 0);
135     glutCreateWindow("*** Your Student Number and Name ***");
136     glutDisplayFunc(mydisplay);
137     glutIdleFunc(myidle);
138     glutKeyboardFunc(mykeyboard);
139     glewInit();
140     myinit();
141     glutMainLoop();
142     return 0;
143 }
144
```