

```
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 float udist; \n\
11 void main(void) { \n\
12     gl_Position.x = aPosition.x + udist; \n\
13     gl_Position.yzw = aPosition.yzw; \n\
14     vColor = aColor; \n\
15 }";
16
17 static char* fsSource = "#version 120 \n\
18 in vec4 vColor; \n\
19 void main(void) { \n\
20     gl_FragColor = vColor; \n\
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 void myinit(void) {
31     GLuint status;
32
33     printf("***** Your student number and name *****\n");
34     vs = glCreateShader(GL_VERTEX_SHADER);
35     glShaderSource(vs, 1, &vsSource, NULL);
36     glCompileShader(vs);
37     glGetShaderiv(vs, GL_COMPILE_STATUS, &status);
38     printf("vs compile status = %s\n", (status == GL_TRUE) ? "true" :
39         "false");
40     glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
41     printf("vs log = [%s]\n", buf);
42
43     fs = glCreateShader(GL_FRAGMENT_SHADER);
44     glShaderSource(fs, 1, &fsSource, NULL);
45     glCompileShader(fs);
46     glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
47     printf("fs compile status = %s\n", (status == GL_TRUE) ? "true" :
48         "false");
49     glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
50     printf("fs log = [%s]\n", buf);
51
52     prog = glCreateProgram();
53     glAttachShader(prog, vs);
54     glAttachShader(prog, fs);
55     glLinkProgram(prog);
56     glGetProgramiv(prog, GL_LINK_STATUS, &status);
```

```
55     printf("program link status = %s\n", (status == GL_TRUE) ? "true" :  
        "false");  
56     glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);  
57     printf("link log = [%s]\n", buf);  
58     glValidateProgram(prog);  
59     glGetProgramiv(prog, GL_VALIDATE_STATUS, &status);  
60     printf("program validate status = %s\n", (status == GL_TRUE) ? "true" :  
        "false");  
61     glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);  
62     printf("validate log = [%s]\n", buf);  
63     glUseProgram(prog);  
64 }  
65  
66 void mykeyboard(unsigned char key, int x, int y) {  
67     switch (key) {  
68     case 27: // ESCAPE  
69         exit(0);  
70         break;  
71     }  
72 }  
73  
74 GLfloat vertices[] = {  
75     -0.5, -0.5, 0.0, 1.0,  
76     +0.5, -0.5, 0.0, 1.0,  
77     -0.5, +0.5, 0.0, 1.0,  
78 };  
79  
80 GLfloat colors[] = {  
81     1.0, 0.0, 0.0, 1.0, // red  
82     0.0, 1.0, 0.0, 1.0, // green  
83     0.0, 0.0, 1.0, 1.0, // blue  
84 };  
85  
86 void myidle(void) {  
87     dist += 0.0001f;  
88     if (dist > 1.5)  
89         dist = 0.0f;  
90  
91     // redisplay  
92     glutPostRedisplay();  
93 }  
94  
95  
96 void mydisplay(void) {  
97     GLuint loc;  
98  
99     glClearColor(0.7f, 0.7f, 0.7f, 1.0f); // gray  
100    glClear(GL_COLOR_BUFFER_BIT);  
101  
102    loc = glGetAttribLocation(prog, "aPosition");  
103    glEnableVertexAttribArray(loc);  
104    glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, vertices);  
105  
106    loc = glGetAttribLocation(prog, "aColor");  
107    glEnableVertexAttribArray(loc);  
108    glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, colors);
```

```
109
110     loc = glGetUniformLocation(prog, "udist");
111     glUniform1f(loc, dist);
112
113     glDrawArrays(GL_TRIANGLES, 0, 3);
114
115     glFlush();
116     glutSwapBuffers();
117 }
118
119 int main(int argc, char* argv[]) {
120     glutInit(&argc, argv);
121     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
122     glutInitWindowSize(500, 500);
123     glutInitWindowPosition(0, 0);
124     glutCreateWindow("*** Your Student Number and Name");
125     glutDisplayFunc(mydisplay);
126     glutIdleFunc(myidle);
127     glutKeyboardFunc(mykeyboard);
128     glewInit();
129     myinit();
130     glutMainLoop();
131     return 0;
132 }
133
```