```
1 #include <GL/glew.h>
2 #include <GL/glut.h>
3 #include <stdio.h>
4 #include <stdlib.h>
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₩
   gl_Position.x = aPosition.x + udist; ₩n₩
     gl_Position.yzw = aPosition.yzw; ₩n₩
13
     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 GLfloat vertices[] = {
       -0.5, -0.5, 0.0, 1.0,
31
32
       +0.5, -0.5, 0.0, 1.0,
33
       -0.5, +0.5, 0.0, 1.0,
34 };
35
36 GLfloat colors[] = {
       1.0, 0.0, 0.0, 1.0, // red
37
       0.0, 1.0, 0.0, 1.0, // green
38
       0.0, 0.0, 1.0, 1.0, // blue
39
40 };
41
42 void myinit(void) {
43
       GLuint status:
44
45
       printf("***** Your student number and name *****\text{\text{\text{W}}}n");
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 = %s\n", (status == GL_TRUE) ? "true" :
          "false");
51
       glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
52
       printf("vs log = [%s] Wn", buf);
53
54
       fs = glCreateShader(GL_FRAGMENT_SHADER);
55
       glShaderSource(fs, 1, &fsSource, NULL);
```

```
...ne_lecture\U0422_LEC12\U00acLEC12_pgm\U00acLEC12.1_translate_one_vbo.c
56
        glCompileShader(fs);
57
        glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
        printf("fs compile status = %s₩n", (status == GL_TRUE) ? "true" :
58
           "false"):
59
        glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
        printf("fs log = [%s] Wn", buf);
60
61
62
        prog = glCreateProgram();
        glAttachShader(prog, vs);
63
64
        glAttachShader(prog, fs);
65
        glLinkProgram(prog);
66
        glGetProgramiv(prog, GL_LINK_STATUS, &status);
        printf("program link status = %s₩n", (status == GL_TRUE) ? "true" :
67
          "false");
        glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
68
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
        glGenBuffers(1, vbo);
79
        glBindBuffer(GL_ARRAY_BUFFER, vbo[0]);
80
        glBufferData(GL_ARRAY_BUFFER, 2 * 3 * 4 * sizeof(GLfloat), NULL,
81
          GL_STATIC_DRAW);
        glBufferSubData(GL_ARRAY_BUFFER, 0, 3 * 4 * sizeof(GLfloat), vertices);
82
        glBufferSubData(GL_ARRAY_BUFFER, 3 * 4 * sizeof(GLfloat), 3 * 4 * sizeof
83
          (GLfloat).
            colors);
84
85 }
86
   void mykeyboard(unsigned char key, int x, int y) {
87
        switch (key) {
88
        case 27: // ESCAPE
89
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

103

// redisplay

glutPostRedisplay();

```
...ne_lecture\d422_LEC12\LEC12_pgm\LEC12.1_translate_one_vbo.c
```

```
:
```

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