```
1 #include <GL/glew.h>
 2 #include <GL/glut.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <math.h>
7 #define my_PI 3.141592
8
9 static char* vsSource = "#version 130 \n\
10 in vec4 aPosition; \n\
11 in vec4 aColor; \n\
12 out vec4 vColor; \n\
uniform mat4 u rotate; \n\
14 uniform float u_scale_factor; \n\
uniform vec2 u_trans_vec; \n\
16 void main(void) { \n\
17
     mat4 scalemat = mat4(u_scale_factor); \n\
18
     scalemat[3][3] = 1.0; \n\
19
     mat4 transmat = mat4(1.0); \n\
     transmat[3][0] = u_trans_vec[0]; \n\
     transmat[3][1] = u_trans_vec[1]; \n\
21
     gl_Position = transmat*u_rotate*aPosition; \n\
22
23 // gl_Position = u_rotate*transmat*aPosition; \n\
24 // gl_Position = scalemat*transmat*u_rotate*aPosition; \n\
25 // gl_Position = u_rotate*scalemat*transmat*aPosition; \n\
26 // gl_Position = transmat*u_rotate*scalemat*aPosition; \n\
   vColor = aColor; \n\
28 }";
29
30 static char* fsSource = "#version 130 \n\
31 in vec4 vColor; \n\
32 void main(void) { \n\
   gl_FragColor = vColor; \n\
34 }";
35
36 GLuint vs = 0;
37 GLuint fs = 0;
38 GLuint prog = 0;
40 char buf[1024];
41 int DRAW MODE = 0;
42 float t = 0.0f;
43
44 GLfloat vertices[] = {
45
       0.0, 0.15, 0.0, 1.0, // 0
46
       -0.1, -0.1, +0.1, 1.0, // 1
47
       0.1, -0.1, +0.1, 1.0, // 2
48
       0.1, -0.1, -0.1, 1.0, // 3
49
       -0.1, -0.1, -0.1, 1.0, // 4
50 };
51
52 GLfloat colors[] = {
       1.0, 0.0, 0.0, 1.0, //0
53
54
       0.0, 1.0, 0.0, 1.0, //1
55
       0.0, 0.0, 1.0, 1.0, //2
       1.0, 0.0, 1.0, 1.0, //3
```

```
...LEC15_online\LEC15_program\org\LEC15.1_concatenation.c
```

```
-
```

```
57
         1.0, 1.0, 0.0, 1.0
 58 };
 59
 60
    GLushort indices[] = {
         0, 1, 2,
 62
         2, 3, 0,
         4, 0, 3,
 63
         1, 0, 4,
 64
 65
         2, 3, 1,
         3, 4, 1
 66
 67 };
 68 void myinit(void) {
 69
         GLuint status;
 70
         printf("***** Your student number and name *****\n");
 71
 72
         vs = glCreateShader(GL_VERTEX_SHADER);
 73
         glShaderSource(vs, 1, &vsSource, NULL);
 74
         glCompileShader(vs);
 75
         glGetShaderiv(vs, GL_COMPILE_STATUS, &status);
 76
         printf("vs compile status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
 77
         glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
         printf("vs log = [%s]\n", buf);
 78
 79
         fs = glCreateShader(GL_FRAGMENT_SHADER);
 80
         glShaderSource(fs, 1, &fsSource, NULL);
 81
 82
         glCompileShader(fs);
 83
         glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
 84
         printf("fs compile status = %s\n", (status == GL_TRUE) ? "true" :
 85
         glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
 86
         printf("fs log = [%s]\n", buf);
 87
 88
         prog = glCreateProgram();
 89
         glAttachShader(prog, vs);
 90
         glAttachShader(prog, fs);
 91
         glLinkProgram(prog);
 92
         glGetProgramiv(prog, GL_LINK_STATUS, &status);
 93
         printf("program link status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
         glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
 94
 95
         printf("link log = [%s]\n", buf);
 96
         glValidateProgram(prog);
         glGetProgramiv(prog, GL_VALIDATE_STATUS, &status);
 97
 98
         printf("program validate status = %s\n", (status == GL_TRUE) ? "true" :
           "false");
         glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
 99
100
         printf("validate log = [%s]\n", buf);
101
         glUseProgram(prog);
102
103
         GLuint loc;
104
         GLuint vbo[1];
105
         // using vertex buffer object
106
         glGenBuffers(1, vbo);
         glBindBuffer(GL_ARRAY_BUFFER, vbo[0]);
107
         glBufferData(GL ARRAY BUFFER, 2 * 5 * 4 * sizeof(GLfloat), NULL,
108
```

```
\verb|...LEC15_online| LEC15_program| org| LEC15.1_concatenation.c| \\
```

```
GL STATIC DRAW);
109
         glBufferSubData(GL ARRAY BUFFER, 0, 5 * 4 * sizeof(GLfloat), vertices);
110
         glBufferSubData(GL_ARRAY_BUFFER, 5 * 4 * sizeof(GLfloat), 5 * 4 * sizeof
           (GLfloat),
111
             colors);
112
113
         loc = glGetAttribLocation(prog, "aPosition");
114
         glEnableVertexAttribArray(loc);
115
         glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *)0);
116
         loc = glGetAttribLocation(prog, "aColor");
117
118
         glEnableVertexAttribArray(loc);
         glVertexAttribPointer(loc, 4, GL FLOAT, GL FALSE, 0, (GLvoid *)(5 * 4 *
119
           sizeof(GLfloat)));
120
121
         glEnable(GL DEPTH TEST);
122 //
         glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
123
         glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
124 }
125
126 void mykeyboard(unsigned char key, int x, int y) {
127
         switch (key) {
         case 27: // ESCAPE
128
129
             exit(0);
130
             break;
131
         }
132 }
133
134
135 void myidle(void) {
136
         t += 0.001f;
         glutPostRedisplay();
137
138 }
139
140 GLfloat m[16];
141
    void mydisplay(void) {
142
143
         GLuint loc;
144
         glClearColor(0.7f, 0.7f, 0.7f, 1.0f); // gray
145
         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
146
         t = 0.0;
147
148 //
          t = 60.0 * my_PI/180.0;
149
150
         // rotation about x-axis
                                                       m[12] = 0.0;
151
         m[0] = 1.0; m[4] = 0.0;
                                     m[8] = 0.0;
152
         m[1] = 0.0; m[5] = cos(t);
                                     m[9] = -\sin(t);
                                                       m[13] = 0.0;
153
         m[2] = 0.0; m[6] = sin(t);
                                     m[10] = \cos(t);
                                                       m[14] = 0.0;
154
         m[3] = 0.0; m[7] = 0.0;
                                     m[11] = 0.0;
                                                       m[15] = 1.0;
155
         loc = glGetUniformLocation(prog, "u_rotate");
156
157
         glUniformMatrix4fv(loc, 1, GL FALSE, m);
158
159
         float scale_factor = 1.0;
160 //
        float scale factor = 1.5;
161
         loc = glGetUniformLocation(prog, "u_scale_factor");
```

```
...LEC15_online\LEC15_program\org\LEC15.1_concatenation.c
```

```
4
```

```
162
         glUniform1f(loc, scale factor);
163
164
        float trans_vec[] = {0.0, 0.0};
165 // float trans_vec[] = { 0.5, 0.5 };
        loc = glGetUniformLocation(prog, "u_trans_vec");
166
167
        glUniform2fv(loc, 1, trans_vec);
168
         glDrawElements(GL_TRIANGLES, 6 * 3, GL_UNSIGNED_SHORT, indices);
169
170
         glFlush();
171
         glutSwapBuffers();
172 }
173
174
175 int main(int argc, char* argv[]) {
         glutInit(&argc, argv);
176
         glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
177
178
         glutInitWindowSize(500, 500);
179
         glutInitWindowPosition(0, 0);
         glutCreateWindow("*** Your Student Number and Name ***");
180
181
        glutDisplayFunc(mydisplay);
        glutIdleFunc(myidle);
182
183
        glutKeyboardFunc(mykeyboard);
184
        glewInit();
185
        myinit();
186
         glutMainLoop();
187
         return 0;
188 }
189
```