

May 7 2012 19:10

alpha3d.c

Page 1

```
1  /*
2   * OpenGL Blending Exercise, a simple/sample OpenGL program.
3   *
4   * Usage:
5   *   g++ alpha3d.c -framework GLUT -framework OpenGL -framework Foundation
6   *
7   * This code places the following three rectangles in 3-D space;
8   *   B: Opaque black rectangle that is parallel to the XY plane.
9   *   R: Translucent red rectangle that is also parallel to XY.
10  *   G: Translucent green rectangle that is parallel to YZ
11  * The rectangle G crosses through B and R.
12  * The points here (or in OpenGL blending in general) are:
13  *   (1) Place opaque objects first.
14  *   (2) Freeze the depth buffer by glDepthMask(GL_FALSE).
15  *   (3) But keep applying the depth test for hidden surface removal.
16  *
17  *   coded by Akira Kageyama, Kobe Univ.
18  *   on 2012.05.07
19  *   at Kobe
20  */
21
22 #include <GLUT/glut.h>
23 #include <stdlib.h>
24
25 static int opaqueFirst = GL_TRUE;
26
27 /* initialize alpha blending function */
28 static void init(void)
29 {
30     glEnable(GL_BLEND);
31     glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
32     glShadeModel(GL_FLAT);
33     glClearColor(1.0, 1.0, 1.0, 0.0);
34 }
35
36 static void drawRectangleXY(float r, float g, float b, float a, float z)
37 {
38     glPushMatrix();
39     glTranslatef(0.0, 0.0, z);
40     glBegin(GL_QUADS);
41         glColor4f(r, g, b, a);
42         glVertex3f(0.0, 0.0, 0.0);
43         glVertex3f(0.7, 0.0, 0.0);
44         glVertex3f(0.7, 0.3, 0.0);
45         glVertex3f(0.0, 0.3, 0.0);
46     glEnd();
47     glPopMatrix();
48 }
49
50 static void drawRectangleYZ(float r, float g, float b, float a, float x)
51 {
52     glPushMatrix();
53     glTranslatef(x, 0.0, 0.0);
54     glBegin(GL_QUADS);
55         glColor4f(r, g, b, a);
56         glVertex3f(0.0, 0.0, 0.0);
57         glVertex3f(0.0, 0.3, 0.0);
58         glVertex3f(0.0, 0.3, -5.0);
59         glVertex3f(0.0, 0.0, -5.0);
60     glEnd();
61     glPopMatrix();
62 }
63
64 void display(void)
65 {
66     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

alpha3d.c

May 7 2012 19:10

alpha3d.c

Page 2

```
67     glEnable(GL_DEPTH_TEST);
68
69     if (opaqueFirst) {
70         drawRectangleXY(0.0, 0.0, 0.0, 1.0, -2.0);
71         glDepthMask(GL_FALSE);
72         drawRectangleYZ(1.0, 0.0, 0.0, 0.3, -1.0);
73         drawRectangleYZ(0.0, 1.0, 0.0, 0.3, 0.35);
74         glDepthMask(GL_TRUE);
75     }
76     else {
77         glDepthMask(GL_FALSE);
78         drawRectangleXY(1.0, 0.0, 0.0, 0.3, -1.0);
79         drawRectangleYZ(0.0, 1.0, 0.0, 0.3, 0.35);
80         glDepthMask(GL_TRUE);
81         drawRectangleXY(0.0, 0.0, 0.0, 1.0, -2.0);
82     }
83     glFlush();
84 }
85
86 void reshape(int w, int h)
87 {
88     glViewport(0, 0, w, h);
89
90     glLoadIdentity();
91     gluPerspective(15.0, (double)w / (double)h, 1.0, 100.0);
92     gluLookAt(-1.0, -0.6, 5.0, 0.0, 0.0, 0.0, 1.0, 0.0);
93 }
94
95 void keyboard(unsigned char key, int x, int y)
96 {
97     switch (key) {
98     case 't':
99     case 'T':
100        opaqueFirst = !opaqueFirst;
101        glutPostRedisplay();
102        break;
103    case 27: /* Escape key */
104        exit(0);
105        break;
106    default:
107        break;
108    }
109 }
110
111 int main(int argc, char** argv)
112 {
113     glutInit(&argc, argv);
114     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
115     glutInitWindowSize(400, 400);
116     glutCreateWindow(argv[0]);
117     init();
118     glutReshapeFunc(reshape);
119     glutKeyboardFunc(keyboard);
120     glutDisplayFunc(display);
121     glutMainLoop();
122     return 0;
123 }
```

1