

```
1: /* checker.c
2:  * This program texture maps a checkerboard image onto
3:  * several rectangles, demonstrating mappings between texture
4:  * coordinates and object coordinates.
5:  *
6:  * If running this program on OpenGL 1.0, texture objects are
7:  * not used.
8:  */
9:
10:
11: #include <GL/glut.h>
12: #include <stdlib.h>
13: #include <stdio.h>
14:
15: /*      Create checkerboard texture      */
16: #define checkImageWidth 64
17: #define checkImageHeight 64
18: static GLubyte checkImage[checkImageHeight][checkImageWidth][4];
19:
20: #ifdef GL_VERSION_1_1
21: static GLuint texName;
22: #endif
23:
24: void makeCheckImage(void)
25: {
26:     int i, j, c;
27:
28:     for (i = 0; i < checkImageHeight; i++) {
29:         for (j = 0; j < checkImageWidth; j++) {
30:             c = (((i&0x8)==0)^((j&0x8)==0))*255;
31:             checkImage[i][j][0] = (GLubyte) c;
32:             checkImage[i][j][1] = (GLubyte) c;
33:             checkImage[i][j][2] = (GLubyte) c;
34:             checkImage[i][j][3] = (GLubyte) 255;
35:         }
36:     }
37: }
38:
39: void init(void)
40: {
41:     glClearColor (0.0, 0.0, 0.0, 0.0);
42:     glShadeModel(GL_FLAT);
43:     glEnable(GL_DEPTH_TEST);
44:
45:     makeCheckImage();
46:     glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
47:
48: #ifdef GL_VERSION_1_1
49:     printf("OpenGL 1.1\n");
50:     glGenTextures(1, &texName);
51:     glBindTexture(GL_TEXTURE_2D, texName);
52: #else
53:     printf("OpenGL 1.0\n");
54: #endif
55:
56:     glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
57:     glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
58:     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
59:         GL_NEAREST);
60:     glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
61:         GL_NEAREST);
62: #ifdef GL_VERSION_1_1
63:     glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth,
```

```
64:         checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,
65:         checkImage);
66: #else
67:     glTexImage2D(GL_TEXTURE_2D, 0, 4, checkImageWidth,
68:         checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,
69:         checkImage);
70: #endif
71: }
72:
73: void display(void)
74: {
75:     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
76:     glEnable(GL_TEXTURE_2D);
77:     glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
78: #ifdef GL_VERSION_1_1
79:     glBindTexture(GL_TEXTURE_2D, texName);
80: #endif
81:
82:     glBegin(GL_QUADS);
83:     // Lower left
84:     glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, -1.0, 0.0);
85:     glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 1.0, 0.0);
86:     glTexCoord2f(1.0, 1.0); glVertex3f(0.0, 1.0, 0.0);
87:     glTexCoord2f(1.0, 0.0); glVertex3f(0.0, -1.0, 0.0);
88:
89:     // Upper left.
90:     glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, 1.5f, 0.0);
91:     glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 2.5f, 0.0);
92:     glTexCoord2f(1.0, 1.0); glVertex3f(-1.0, 2.5f, 0.0);
93:     glTexCoord2f(1.0, 0.0); glVertex3f(-1.0, 1.5f, 0.0);
94:
95:     // Upper middle.
96:     glTexCoord2f(0.0, 0.0); glVertex3f(0.0, 1.5f, 0.0);
97:     glTexCoord2f(0.0, 2.0); glVertex3f(0.0, 2.5f, 0.0);
98:     glTexCoord2f(2.0, 2.0); glVertex3f(1.0, 2.5f, 0.0);
99:     glTexCoord2f(2.0, 0.0); glVertex3f(1.0, 1.5f, 0.0);
100:
101:     // Upper right
102:     glTexCoord2f(0.0, 0.0); glVertex3f(1.5f, 1.5f, 0.0);
103:     glTexCoord2f(0.0, 0.5); glVertex3f(1.5f, 2.5f, 0.0);
104:     glTexCoord2f(0.5, 0.5); glVertex3f(2.5f, 2.5f, 0.0);
105:     glTexCoord2f(0.5, 0.0); glVertex3f(2.5f, 1.5f, 0.0);
106:
107:     // Lower right.
108:     glTexCoord2f(0.0, 0.0); glVertex3f(1.0, -1.0, 0.0);
109:     glTexCoord2f(0.0, 1.0); glVertex3f(1.0, 1.0, 0.0);
110:     glTexCoord2f(1.0, 1.0); glVertex3f(2.41421f, 1.0, -1.41421f);
111:     glTexCoord2f(1.0, 0.0); glVertex3f(2.41421f, -1.0, -1.41421f);
112:     glEnd();
113:     glFlush();
114:     glDisable(GL_TEXTURE_2D);
115: }
116:
117: void reshape(int w, int h)
118: {
119:     glViewport(0, 0, (GLsizei) w, (GLsizei) h);
120:     glMatrixMode(GL_PROJECTION);
121:     glLoadIdentity();
122:     gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 30.0);
123:     glMatrixMode(GL_MODELVIEW);
124:     glLoadIdentity();
125:     //glTranslatef(0.0, 0.0, -3.6f);
126:     glTranslatef(0.0, 0.0, -5.0f);
```

```
127: }
128:
129: void keyboard (unsigned char key, int x, int y)
130: {
131:     switch (key) {
132:         case 27:
133:             exit(0);
134:             break;
135:         default:
136:             break;
137:     }
138: }
139:
140: int main(int argc, char** argv)
141: {
142:     setvbuf(stdout, (char *)NULL, _IONBF, 0);
143:     setvbuf(stderr, (char *)NULL, _IONBF, 0);
144:
145:     glutInit(&argc, argv);
146:     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
147:     glutInitWindowSize(400, 400);
148:     glutInitWindowPosition(100, 100);
149:     glutCreateWindow(argv[0]);
150:     init();
151:     glutDisplayFunc(display);
152:     glutReshapeFunc(reshape);
153:     glutKeyboardFunc(keyboard);
154:     glutMainLoop();
155:     return 0;
156: }
```