



الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

Arab Academy for Science, Technology & Maritime Transport

COLLEGE OF ENGINEERING AND TECHNOLOGY  
DEPARTMENT OF COMPUTER ENGINEERING

Lecturer: Prof. Dr. Mohamed Taher El-Sonni  
Teaching Assistant(s): Eng. Mohamed A. Aslan

## Lab #7

### Mixing 2D & 3D

## C++ Template

Listing 1: image.cpp

```
1 #include <cstdlib>
2 #include <cstdio>
3 #include <cmath>
4 #include <string>
5
6 #include "image.h"
7
8 Image::Image(std::string filename)
9 {
10     FILE* image = fopen(filename.c_str(), "rb");
11     unsigned char c;
12     for(int i=0; i<18; i++)
13         fscanf(image, "%c", &c);
14     unsigned char b[4];
15     fscanf(image, "%c%c%c%c", &b[0], &b[1], &b[2], &b[3]);
16     width = (b[3] << 24) | (b[2] << 16) | (b[1] << 8) | b[0];
17     fscanf(image, "%c%c%c%c", &b[0], &b[1], &b[2], &b[3]);
18     height = (b[3] << 24) | (b[2] << 16) | (b[1] << 8) | b[0];
19     for(int i=0; i<28; i++)
20         fscanf(image, "%c", &c);
21     for(int i=0; i<height; i++)
22     {
23         for(int j=0; j<width; j++)
24         {
25             char r, g, b;
26             fscanf(image, "%c%c%c", &b, &g, &r);
27             pixels[i][j].red = r;
28             pixels[i][j].green = g;
29             pixels[i][j].blue = b;
30         }
31         if((width * 3) % 4 == 0)
32             continue;
33         for(int j = 0; j< 4-((width*3)%4); j++)
34             fscanf(image, "%c", &c);
```

```

35     }
36     fclose(image);
37 }
38
39 void Image::display(int win_x, int win_y)
40 {
41     glBegin(GL_POINTS);
42     for(int i = 0, h = win_y ; i < height ; i++, h++)
43     {
44         for(int j = 0, w = win_x ; j < width ; j++, w++)
45         {
46             float r = ((float)pixels[i][j].red) / 255.0;
47             float g = ((float)pixels[i][j].green) / 255.0;
48             float b = ((float)pixels[i][j].blue) / 255.0;
49             glColor3f(r, g, b);
50             glVertex2i(w, h);
51         }
52     }
53     glEnd();
54 }

```

Listing 2: image.h

```

1  #include <cstdlib>
2  #include <cstdio>
3  #include <cmath>
4  #include <string>
5  #include <GL/gl.h>
6  #include <GL/glu.h>
7  #include <GL/glut.h>
8
9  #ifndef IMAGE_H
10 #define IMAGE_H
11
12 struct RGB
13 {
14     unsigned char red;
15     unsigned char green;
16     unsigned char blue;
17 };
18
19 class Image
20 {
21 private:
22     RGB pixels[2048][2048];
23     int width;
24     int height;
25 public:
26     Image(std::string);
27     void display(int, int);
28 };
29
30 #endif

```

Listing 3: test.cpp

```

1  #include <GL/gl.h>
2  #include <GL/glu.h>
3  #include <GL/glut.h>
4  #include <stdio.h>
5  #include <math.h>
6
7  #include "image.h"
8
9

```

```

10 Image *img;
11
12 void loadImages()
13 {
14     img = new Image("background.bmp");
15 }
16
17 void DrawCube()
18 {
19     glPushMatrix();
20     glBegin(GL_QUADS);
21     /* 1st face */
22     glColor3f(0, 0, 0); glVertex3f(-1, -1, -1);
23     glColor3f(0, 0, 1); glVertex3f(-1, -1, 1);
24     glColor3f(0, 1, 1); glVertex3f(-1, 1, 1);
25     glColor3f(0, 1, 0); glVertex3f(-1, 1, -1);
26
27     /* 2nd face */
28     glColor3f(1, 0, 0); glVertex3f( 1, -1, -1);
29     glColor3f(1, 0, 1); glVertex3f( 1, -1, 1);
30     glColor3f(1, 1, 1); glVertex3f( 1, 1, 1);
31     glColor3f(1, 1, 0); glVertex3f( 1, 1, -1);
32
33     /* 3rd face */
34     glColor3f(0, 0, 0); glVertex3f(-1, -1, -1);
35     glColor3f(0, 0, 1); glVertex3f(-1, -1, 1);
36     glColor3f(1, 0, 1); glVertex3f( 1, -1, 1);
37     glColor3f(1, 0, 0); glVertex3f( 1, -1, -1);
38
39     /* 4th face */
40     glColor3f(0, 1, 0); glVertex3f(-1, 1, -1);
41     glColor3f(0, 1, 1); glVertex3f(-1, 1, 1);
42     glColor3f(1, 1, 1); glVertex3f( 1, 1, 1);
43     glColor3f(1, 1, 0); glVertex3f( 1, 1, -1);
44
45     /* 5th face */
46     glColor3f(0, 0, 0); glVertex3f(-1, -1, -1);
47     glColor3f(0, 1, 0); glVertex3f(-1, 1, -1);
48     glColor3f(1, 1, 0); glVertex3f( 1, 1, -1);
49     glColor3f(1, 0, 0); glVertex3f( 1, -1, -1);
50
51     /* 6th face*/
52     glColor3f(0, 0, 1); glVertex3f(-1, -1, 1);
53     glColor3f(0, 1, 1); glVertex3f(-1, 1, 1);
54     glColor3f(1, 1, 1); glVertex3f( 1, 1, 1);
55     glColor3f(1, 0, 1); glVertex3f( 1, -1, 1);
56
57     glEnd();
58     glPopMatrix();
59 }
60
61 // Mixing 2D & 3D rendering
62 void myDisplay()
63 {
64     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
65
66     // Draw 2D
67     glMatrixMode(GL_PROJECTION);
68     glPushMatrix();
69     glLoadIdentity();
70     glOrtho(0, 1280, 0, 1024, -1000, 1);
71     glMatrixMode(GL_MODELVIEW);
72     img->display(0, 0);
73     glMatrixMode(GL_PROJECTION);
74     glPopMatrix();
75
76     // Draw 3D
77     glMatrixMode(GL_MODELVIEW);
78     glPushMatrix();
79     glLoadIdentity();

```

```

80     glTranslatef(0, 0, -10);
81     glRotatef(30, 1, 0, 0);
82     glRotatef(10, 0, 1, 0);
83     DrawCube();
84     glPopMatrix();
85
86     glutSwapBuffers();
87 }
88
89 void reshape(int width, int height)
90 {
91     glMatrixMode(GL_PROJECTION);
92     glLoadIdentity();
93     gluPerspective(20, width / (float) height, 5, 15);
94     glViewport(0, 0, width, height);
95     glMatrixMode(GL_MODELVIEW);
96     glutPostRedisplay();
97 }
98
99 void keyboard(unsigned char ch, int x, int y)
100 {
101     if(ch == 'q')
102         exit(0);
103     glutPostRedisplay();
104 }
105
106 int main(int argc, char **argv)
107 {
108     glutInit(&argc, argv);
109     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
110     glutInitWindowSize(600, 600);
111     glutInitWindowPosition(50, 50);
112     glutCreateWindow("OpenGL Image Template");
113     glClearColor(1.0, 1.0, 1.0, 0.0);
114     glEnable(GL_DEPTH_TEST);
115     glutDisplayFunc(myDisplay);
116     glutReshapeFunc(reshape);
117     glutKeyboardFunc(keyboard);
118     loadImages();
119     glutMainLoop();
120     return 0;
121 }

```

## Compile:

- `c++ -c image.cpp -o image.o`
- `c++ -c test.cpp -o test.o`
- `c++ test.o image.o -o test -lglut -lGLU`