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 #6

Loading Bitmap Images with OpenGL

C++ Template

Listing 1: image.cpp

```cpp
#include <cstdlib>
#include <cstdio>
#include <cmath>
#include <string>

#include "image.h"

Image::Image(std::string filename)
{
    FILE* image = fopen(filename.c_str(), "rb");
    unsigned char c;
    for(int i=0 ; i<18 ; i++)
        fscanf(image, "%c", &c);
    unsigned char b[4];
    fscanf(image, "%c%c%c%c", &b[0], &b[1], &b[2], &b[3]);
    width = (b[3] << 24) | (b[2] << 16) | (b[1] << 8) | b[0];
    fscanf(image, "%c%c%c%c", &b[0], &b[1], &b[2], &b[3]);
    height = (b[3] << 24) | (b[2] << 16) | (b[1] << 8) | b[0];
    for(int i=0 ; i<28 ; i++)
        fscanf(image, "%c", &c);
    for(int i=0 ; i<height ; i++)
    {
        for(int j=0; j<width ; j++)
        {
            char r, g, b;
            fscanf(image, "%c%c%c", &b, &g, &r);
            pixels[i][j].red = r;
            pixels[i][j].green = g;
            pixels[i][j].blue = b;
        }
        if((width * 3) % 4 == 0)
            continue;
        for(int j = 0 ; j< 4-((width*3)%4) ; j++)
            fscanf(image, "%c", &c);
```

```
    }
    fclose(image);
}

void Image::display(int win_x, int win_y)
{
    glBegin(GL_POINTS);
    for(int i = 0, h = win_y ; i < height ; i++, h++)
    {
        for(int j = 0, w = win_x ; j < width ; j++, w++)
        {
            float r = ((float)pixels[i][j].red) / 255.0;
            float g = ((float)pixels[i][j].green) / 255.0;
            float b = ((float)pixels[i][j].blue) / 255.0;
            glColor3f(r, g, b);
            glVertex2i(w, h);
        }
    }
    glEnd();
}
```

Listing 2: image.h

```
#include <cstdlib>
#include <cstdio>
#include <cmath>
#include <string>
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>

#ifndef IMAGE_H
#define IMAGE_H

struct RGB
{
    unsigned char red;
    unsigned char green;
    unsigned char blue;
};

class Image
{
private:
    RGB pixels[2048][2048];
    int width;
    int height;
public:
    Image(std::string);
    void display(int, int);
};

#endif
```

Listing 3: test.cpp

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>

#include "image.h"


```

```
10 Image *img1;
11 Image *img2;
12
13 void loadImages()
14 {
15     img1 = new Image("car.bmp");
16     img2 = new Image("computer.bmp");
17 }
18
19 void myDisplay()
20 {
21     glClear(GL_COLOR_BUFFER_BIT);
22     glMatrixMode(GL_MODELVIEW);
23     glLoadIdentity();
24     glPointSize(4.0);
25
26     glPushMatrix();
27         img1->display(0, 0);
28     glPopMatrix();
29     glPushMatrix();
30         img2->display(300, 400);
31     glPopMatrix();
32
33     glutSwapBuffers();
34 }
35
36 int main(int argc, char **argv)
37 {
38     glutInit(&argc, argv);
39     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
40     glutInitWindowSize(600, 600);
41     glutInitWindowPosition(50, 50);
42     glutCreateWindow("OpenGL Image Template");
43     glClearColor(1.0, 1.0, 1.0, 0.0);
44     glMatrixMode(GL_PROJECTION);
45     glLoadIdentity();
46     glOrtho(0, 600, 0, 600, 0, -1);
47     glutDisplayFunc(myDisplay);
48     loadImages();
49     glutMainLoop();
50     return 0;
51 }
```

Compile:

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