Lab Report

Week 2

Ritobroto Maitra (1301CS50) 02/02/2016

Title

- Draw basic primitives (a line, a circle and a polygon) in OpenGL.

Procedure

We create three files for the given code and then compile and run them with the openGL linker to get the desired output. The following steps are to be followed :

1. Write the attached code in three files, namely *line.c*, *circle.c* and *polygon.c*.
2. Compile the C source files as follows :

   g++ line.c -framework OpenGL -framework GLUT -o line
   g++ circle.c -framework OpenGL -framework GLUT -o circle
   g++ polygon.c -framework OpenGL -framework GLUT -o polygon

3. Run the program by using the command

   ./line
   ./circle
   ./polygon

```c
#include <math.h>
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include <stdio.h>

//Initialize OpenGL
void init(void) {
  glClearColor(1, 1, 1, 1);
  glViewport(0, 0, 500, 500);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(0, 500, 0, 500, 1, -1);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
}

void drawLines(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0,0,1);
    glPointSize(3.0);

    glBegin(GL_LINES);
    glVertex2d(200, 150);glVertex2d(400, 350);
    glEnd();
    glFlush();


}
```

```c

int main(int argc, char**argv) {
    glutInit(&argc, argv);
    glutInitWindowPosition(10,10);
    glutInitWindowSize(500,500);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutCreateWindow("Example");
    init();
    glutDisplayFunc(drawLines);
    glutMainLoop();
}
```

line.c

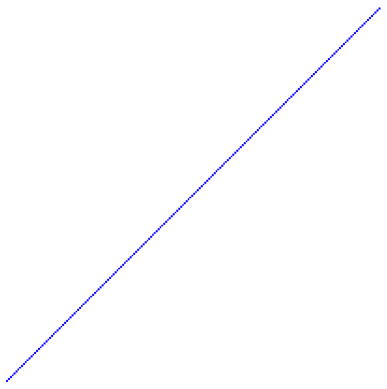


Fig : Output of the above code

```c
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include <math.h>
#include <stdio.h>
void Draw() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);

    glBegin(GL_QUADS);
      glColor3f (0.0, 0.0, 0.0);

      glVertex3f (0.1, 0.1, 0.0);
      glVertex3f (0.9, 0.1, 0.0);
      glVertex3f (0.9, 0.9, 0.0);
      glVertex3f (0.1, 0.9, 0.0);

    glEnd();

    glFlush();
    // DrawCircle(0.5, 0.5, 0.2, 5);
}

void DrawCircle()
{
    float cx = 0.5, cy = 0.5, r = 0.2, num_segments = 50;
    glBegin(GL_LINE_LOOP);
    for(int ii = 0; ii < num_segments; ii++)
    {
        float theta = 2.0f * 3.1415926f * float(ii) / float(num_segments);//get the current angle

        float x = r * cosf(theta);//calculate the x component
        float y = r * sinf(theta);//calculate the y component

        glVertex2f(x + cx, y + cy);//output vertex

    }
    glEnd();
    glFlush();
}


void Initialize() {
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}

int main(int iArgc, char** cppArgv) {
    glutInit(&iArgc, cppArgv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(950, 950);
    glutInitWindowPosition(200, 200);
    glutCreateWindow("Universum");
    Initialize();
    Draw();
    glutDisplayFunc(DrawCircle);
    glClear(GL_COLOR_BUFFER_BIT);
    // glColor3f(1.0, 1.0, 1.0);
    // DrawCircle(0.5, 0.5, 0.2, 5);

    glutMainLoop();
    return 0;

}
```

circle.c

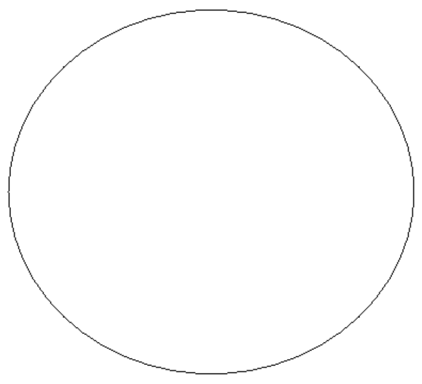


Fig : Output of the above code

```c
#include <math.h>
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include <stdio.h>

//Initialize OpenGL
```

```c
void init(void) {
  glClearColor(1, 1, 1, 1);
  glViewport(0, 0, 500, 500);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(0, 500, 0, 500, 1, -1);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
}

void drawLines(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0,0,1);
    glPointSize(3.0);

    glBegin(GL_LINES);
    glVertex2d(200, 150);glVertex2d(400, 350);
    glVertex2d(400, 350);glVertex2d(500, 200);
    glVertex2d(500, 200);glVertex2d(200, 150);
    glEnd();
    glFlush();


}

int main(int argc, char**argv) {
    glutInit(&argc, argv);
    glutInitWindowPosition(10,10);
    glutInitWindowSize(500,500);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutCreateWindow("Example");
    init();
    glutDisplayFunc(drawLines);
    glutMainLoop();
}
```

polygon.c

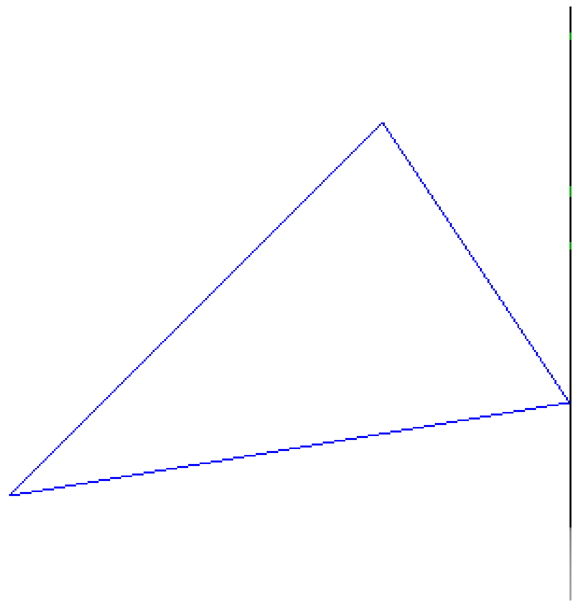


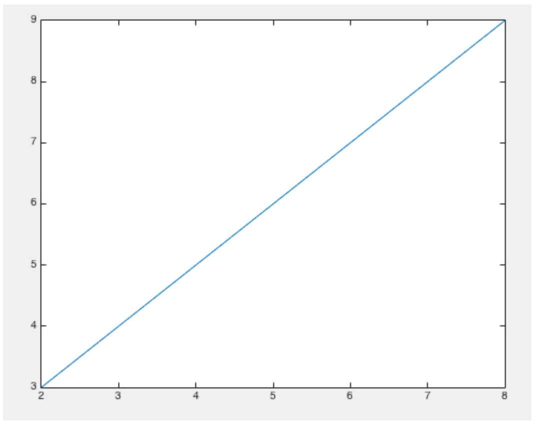
Fig : Output of the above code

Plotting in MATLAB

We plot the same figures in *MATLAB* as well.

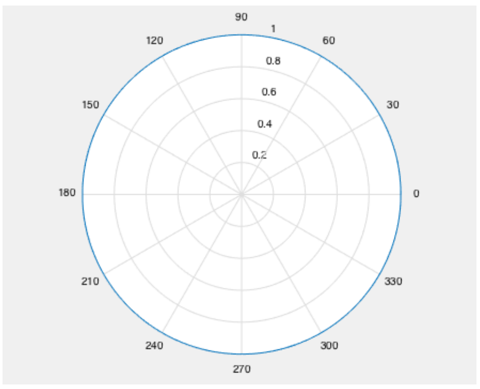
Drawing a line

```
function line = drawLine(x1, y1, x2, y2)
%This function draws line between two points
X = [x1, x2];
Y = [y1, y2];
plot(X,Y);
end
```



Drawing a circle

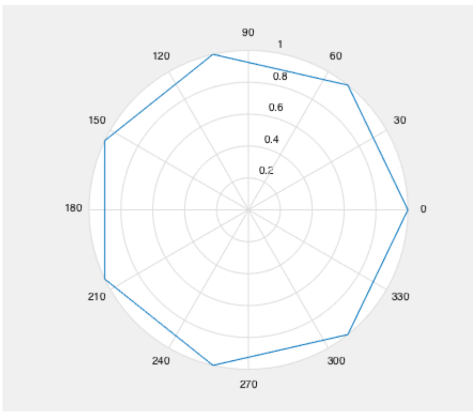
```
function circle = DrawCircle()
% This function draws a circle
sides = 11000;
angle = 2*pi/sides;
angle_in_degree = 0:angle:360-angle;
radius=ones(1,numel(angle_in_degree));
polar(angle_in_degree,radius);
end
```



Drawing a polygon

```
function polygon = DrawPolygon(sides)
% This function draws any n-sided polygon
angle = 2*pi/sides;
angle_in_degree = 0:angle:360-angle;
radius=ones(1,numel(angle_in_degree));
```

```
polar(angle_in_degree,radius);
end
```



90
1
120
60
0.8
0.6
150
30
0.4
0.2
180
0
210
330
240
300
270