

Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

وزارة التعليم

Umm AlQura University

جامعة أم القرى

Adham University College

الكلية الجامعية بأضم

Computer Science Department

قسم الحاسب الآلي



CS
Department

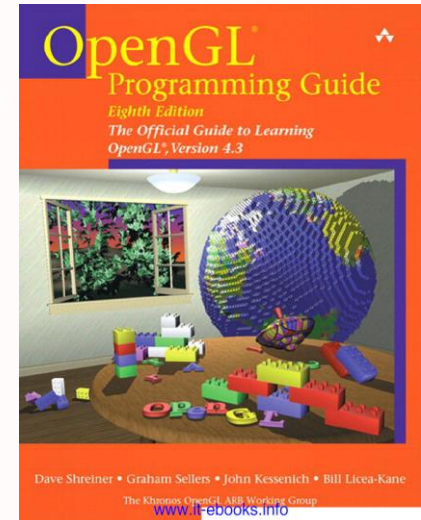
Computer Graphics Course, 3-6803430



T.Mariah Khayat

References

- Lab Lectures, Computer Graphics, Taif University, Faculty Of Computers And Information Technology, TA. Maha Thafar &TA. Haifa Alshehri, TA.Sohair Soliman & L.Shakila Bano.
- OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.3, 8th edition, Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane & The Khronos OpenGL ARB Working Group, Addison-Wesley.



T.Mariah Khayat

Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

جامعة أم القرى

Umm AlQura University

جامعة أم القرى

Adham University College

الكلية الجامعية بأضم

Computer Science Department

قسم الحاسب الآلي

Lecture Five

Drawing Circles and Ellipse using OpenGL

Computer Graphics
Course, 3-6803430

LAB

T.Mariah Khayat

content

1. Circles in OpenGL
2. Draw Circle using OpenGL
3. Draw Ellipse using OpenGL
4. Draw Traffics Signal using OpenGL
5. Important Notes

T.Mariah Khayat

Circles in OpenGL

- A **circle** is a simple **shape** of **Euclidean geometry** consisting of those **points** which are the same **distance** from a given point called the **center**.
- The common distance of the points of a circle from its center is called its **radius**.

There are several way to draw a circle:

1) Draw a circle by algorithms:

- Simple circle drawing algorithm
- Midpoint Circle Algorithm.
- Bresenham's Circle Algorithm.
- DDA Algorithm.

2) Draw a circle by arithmetic equations:

- Trigonometric functions.

Circles in OpenGL

- Draw a circle by algorithms:

- 1. Simple circle drawing algorithm

- In an x-y **Coordinate System**, the circle with center (a, b) and radius r is the set of all points (x, y) such as the equation:

$$(x - a)^2 + (y - b)^2 = r^2$$

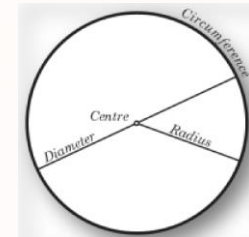
- If the circle is centered at the origin $(0, 0)$, then the equation simplifies to:

$$x^2 + y^2 = r^2$$

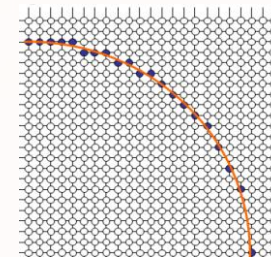
- where: **x = x coordinate**, **y = y coordinate**, **r = radius**

- **Note:** This method : **not very GOOD** because:

- 1. the resulting circle has large gaps.
 - 2. the calculations are not very efficient with **square root** operations.

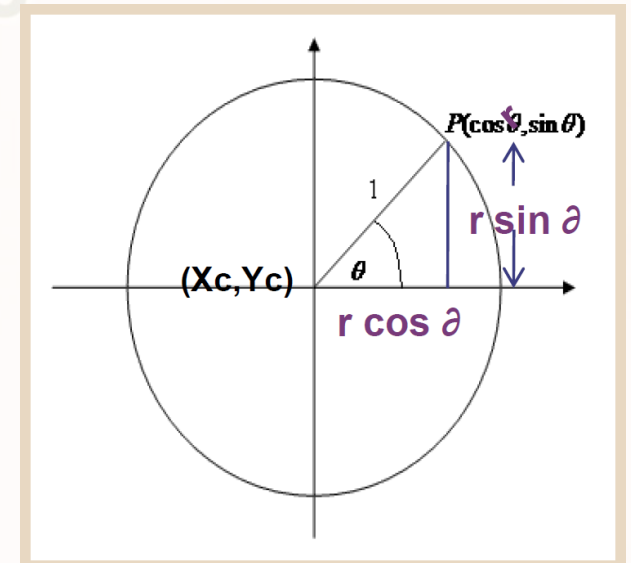


$$y = \mp \sqrt{r^2 - x^2}$$



Circles in OpenGL

- **Draw a circle by arithmetic equations:**
 - **Using the trigonometric functions:**
 - If the circle is centered at the origin (X_c, Y_c):
 - $\sin \theta = (Y - Y_c) / r \rightarrow Y = Y_c + r * \sin \theta$
 - $\cos \theta = (X - X_c) / r \rightarrow X = X_c + r * \cos \theta$
 - If the circle is centered at the origin (0, 0), then the equation simplifies to
 - $Y = r \sin \theta$
 - $X = r \cos \theta$



Draw circle using OpenGL

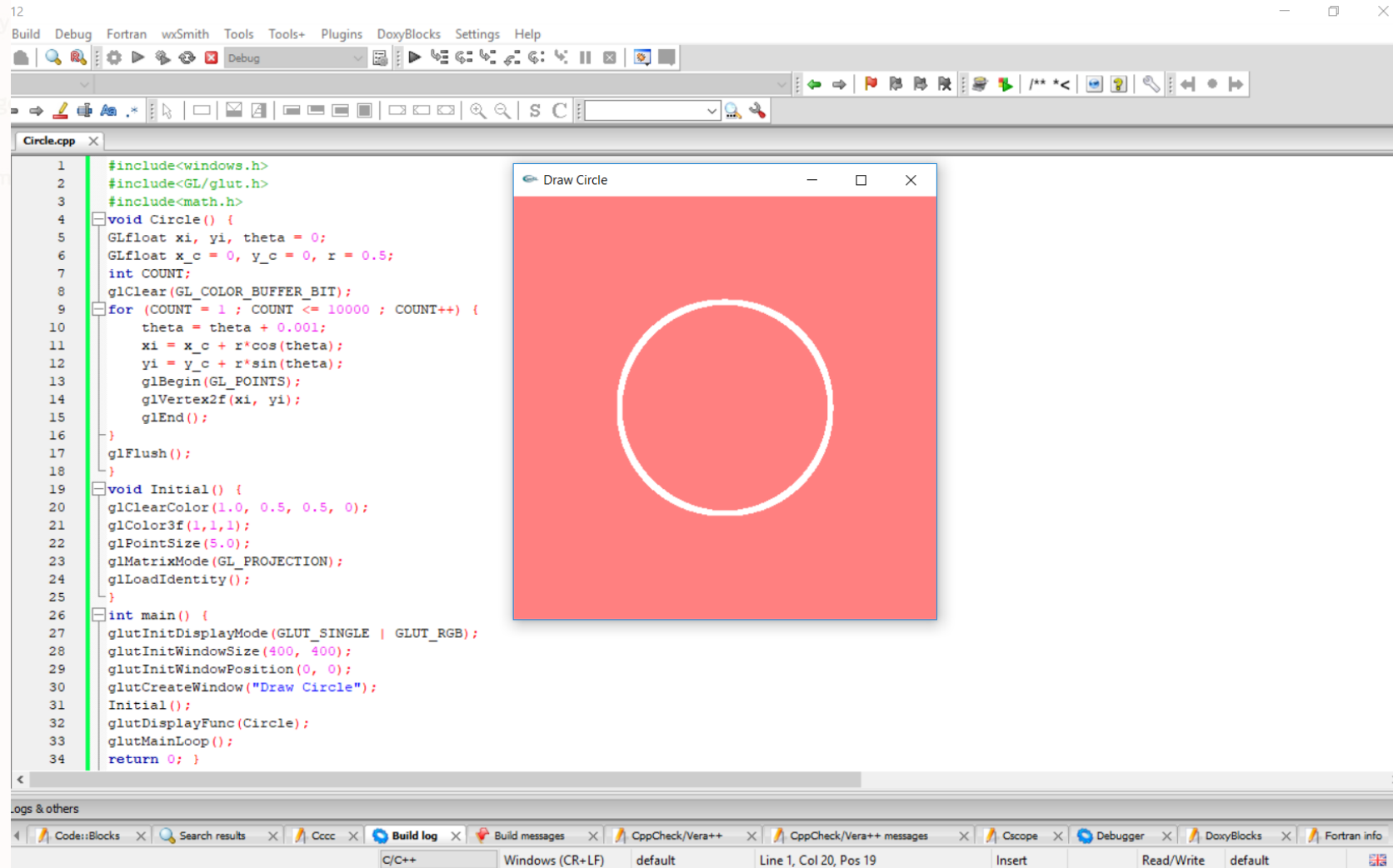
```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Circle() {
    GLfloat xi, yi, theta = 0;
    GLfloat x_c = 0, y_c = 0, r = 0.5;
    int COUNT;
    glClear(GL_COLOR_BUFFER_BIT);
    for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
        theta = theta + 0.001;
        xi = x_c + r*cos(theta);
        yi = y_c + r*sin(theta);
        glBegin(GL_POINTS);
        glVertex2f(xi, yi);
        glEnd();
    }
    glFlush();
}
void Initial() {
    glClearColor(1.0, 0.5, 0.5, 0);
    glColor3f(1,1,1);
    glPointSize(5.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
}
int main() {
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Draw Circle");
    Initial();
    glutDisplayFunc(Circle);
    glutMainLoop();
    return 0; }
```

Computer Graphics
Course, 3-6803430

LAB

T.Mariah Khayat

Draw circle using OpenGL



The screenshot shows a C++ IDE with a file named `Circle.cpp` open. The code implements a circle-drawing function using OpenGL and GLUT. The `Circle` function calculates points on a circle using trigonometric functions and draws them as points. The `Initial` function sets up the OpenGL environment, including clearing the color buffer, setting the color to red, and initializing the window. The `main` function initializes GLUT, creates a window titled "Draw Circle", and calls the `Initial` and `Circle` functions.

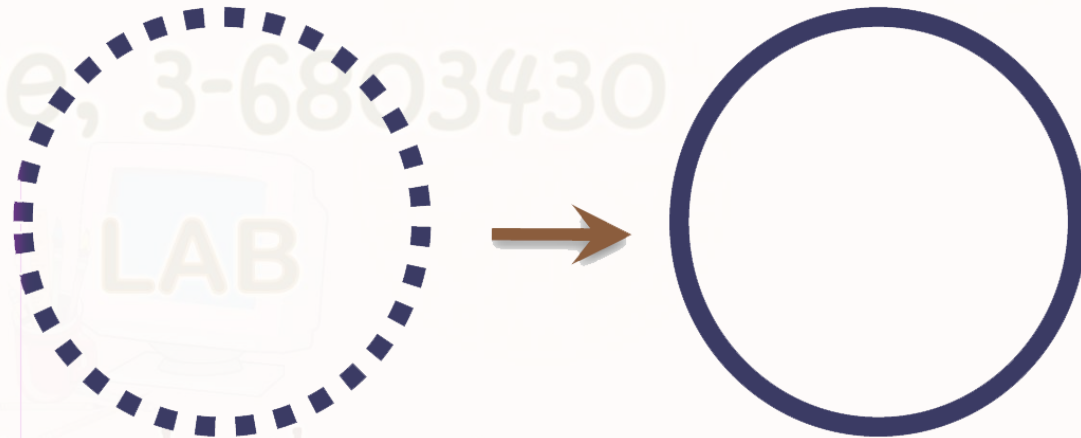
```
1  #include<windows.h>
2  #include<GL/glut.h>
3  #include<math.h>
4  void Circle() {
5      GLfloat xi, yi, theta = 0;
6      GLfloat x_c = 0, y_c = 0, r = 0.5;
7      int COUNT;
8      glClear(GL_COLOR_BUFFER_BIT);
9      for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
10         theta = theta + 0.001;
11         xi = x_c + r*cos(theta);
12         yi = y_c + r*sin(theta);
13         glBegin(GL_POINTS);
14         glVertex2f(xi, yi);
15         glEnd();
16     }
17     glFlush();
18 }
19 void Initial() {
20     glClearColor(1.0, 0.5, 0.5, 0);
21     glColor3f(1,1,1);
22     glPointSize(5.0);
23     glMatrixMode(GL_PROJECTION);
24     glLoadIdentity();
25 }
26 int main() {
27     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
28     glutInitWindowSize(400, 400);
29     glutInitWindowPosition(0, 0);
30     glutCreateWindow("Draw Circle");
31     Initial();
32     glutDisplayFunc(Circle);
33     glutMainLoop();
34     return 0; }
```

The window titled "Draw Circle" displays a red square with a white circle centered in the middle, representing the output of the program.

Draw circle using OpenGL

- Note: This program to Draw a circle in OpenGL:
- By draw a large number of points near each other inside loop and this points draw a circle as the following figure:

Points



T.Mariah Khayat

Draw Ellipse using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Ellipse() {
    GLfloat xi, yi, theta = 0;
    GLfloat x_c = 0, y_c = 0, r_x = 0.8, r_y = 1.8;
    int COUNT;
    glClear(GL_COLOR_BUFFER_BIT);
    for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
        theta = theta + 0.001;
        xi = x_c + r_x*cos(theta);
        yi = y_c + r_y*sin(theta);
        glBegin(GL_POINTS);
        glVertex2f(xi, yi);
        glEnd();
    }
    glFlush();
}
void Initial() {
    glClearColor(1.0, 0.5, 0.1, 0);
    glColor3f(1,1,1);
    glPointSize(5.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-2, +2, -2, +2);
}
int main() {
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Draw Ellipse");
    Initial();
    glutDisplayFunc(Ellipse);
    glutMainLoop();
    return 0; }
```

Computer Graphics
Course, 3-6803430

LAB

T.Mariah Khayat

CS
Department

Draw Ellipse using OpenGL

17.12

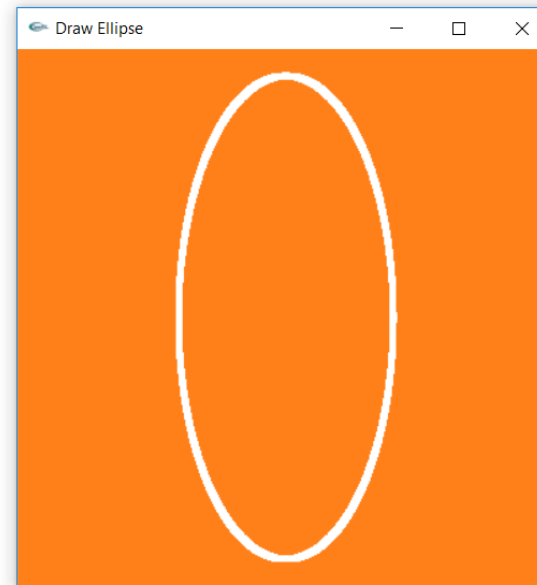
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Initial() : void



Ellipse.cpp

```
2  #include<GL/glut.h>
3  #include<math.h>
4  void Ellipse() {
5      GLfloat xi, yi, theta = 0;
6      GLfloat x_c = 0, y_c = 0, r_x = 0.8, r_y = 1.8;
7      int COUNT;
8      glClear(GL_COLOR_BUFFER_BIT);
9      for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
10         theta = theta + 0.001;
11         xi = x_c + r_x*cos(theta);
12         yi = y_c + r_y*sin(theta);
13         glBegin(GL_POINTS);
14         glVertex2f(xi, yi);
15         glEnd();
16     }
17     glFlush();
18 }
19 void Initial() {
20     glClearColor(1.0, 0.5, 0.1, 0);
21     glColor3f(1,1,1);
22     glPointSize(5.0);
23     glMatrixMode(GL_PROJECTION);
24     glLoadIdentity();
25     gluOrtho2D(-2, +2, -2, +2);
26 }
27 int main() {
28     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
29     glutInitWindowSize(400, 400);
30     glutInitWindowPosition(0, 0);
31     glutCreateWindow("Draw Ellipse");
32     Initial();
33     glutDisplayFunc(Ellipse);
34     glutMainLoop();
35     return 0; }
```

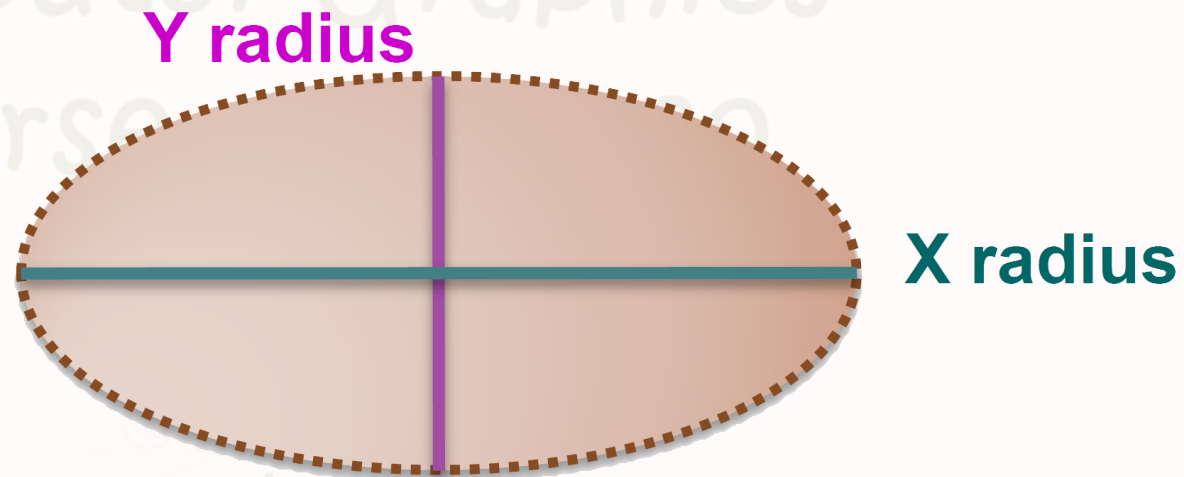


Logs & others

Draw Ellipse using OpenGL

- **NOTE:**

- To draw ellipse:
- The different between circle & ellipse only in radius, in ellipse we have 2 radius:
 1. X radius.
 2. and Y radius.



T.Mariah Khayat

Draw Graphics Signal using OpenGL

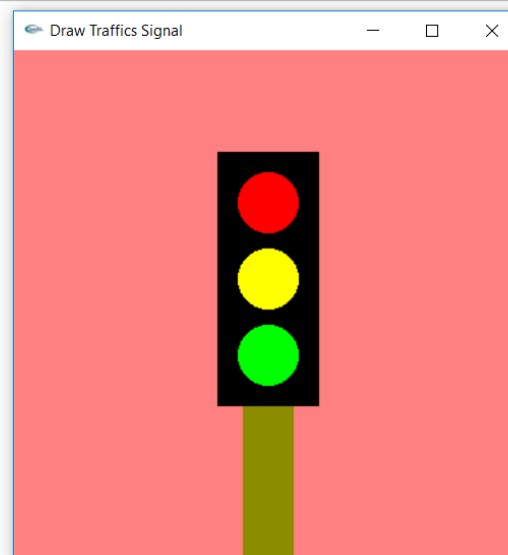
Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main() : int

TrafficsSingal.cpp

```
1  #include<windows.h>
2  #include<GL/glut.h>
3  #include<math.h>
4  void Traffics() {
5      GLfloat x_i, y_i, theta = 0;
6      GLfloat x_c, y_c, r;
7      glClear(GL_COLOR_BUFFER_BIT);
8      int i;
9      glColor3f(0,0,0); //Black Rectangle
10     glBegin(GL_QUADS);
11     glVertex2d(40, 30);
12     glVertex2d(60, 30);
13     glVertex2d(60, 80);
14     glVertex2d(40, 80);
15     glEnd();
16     glColor3f(0.55, 0.55, 0);
17     glBegin(GL_QUADS);
18     glVertex2d(45, 0);
19     glVertex2d(55, 0);
20     glVertex2d(55, 30);
21     glVertex2d(45, 30);
22     glEnd();
23     x_c = 50; y_c = 70; r = 6; //Circle1 (RED)
24     glBegin(GL_POLYGON);
25     for (i = 1; i <= 10000; i++) {
26         theta+=0.001;
27         x_i = x_c + r*cos(theta);
28         y_i = y_c + r*sin(theta);
29         glColor3f(1,0,0);
30         glVertex2d(x_i, y_i); }
31     glEnd();
32     x_c = 50; y_c = 55; r = 6; //Circle2 (YELLOW)
33     glBegin(GL_POLYGON);
34     for (i = 1; i <= 10000; i++) {
```



Logs & others

Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran info

C/C++

Windows (CR+LF)

default

Line 66, Col 12, Pos 1566

Insert

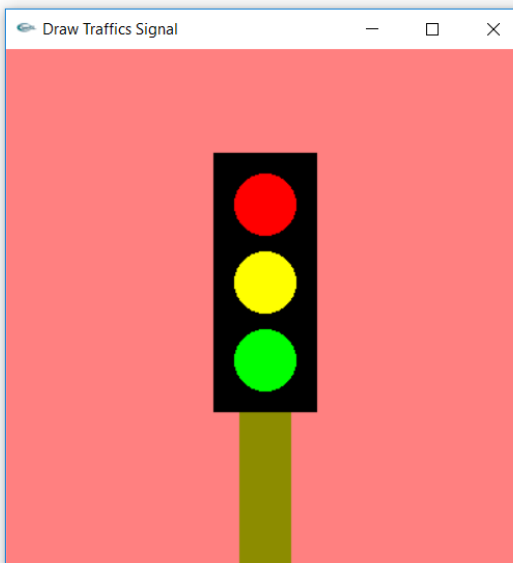
Read/Write

default

UK

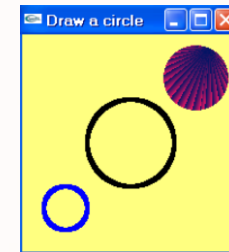
Draw Graphics Signal using OpenGL

```
Code::Blocks 17.12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main(): int
TrafficsSignal.cpp
30 glVertex2d(x_i, y_i);
31 glEnd();
32 x_c = 50; y_c = 55; r = 6; //Circle2 (YELLOW)
33 glBegin(GL_POLYGON);
34 for (i = 1; i <= 10000; i++) {
35     theta+=0.001;
36     x_i = x_c + r*cos(theta);
37     y_i = y_c + r*sin(theta);
38     glColor3f(1,1,0);
39     glVertex2d(x_i, y_i);
40 glEnd();
41 x_c = 50; y_c = 40; r = 6; //Circle3 (Blue)
42 glBegin(GL_POLYGON);
43 for (i = 1; i <= 10000; i++) {
44     theta+=0.001;
45     x_i = x_c + r*cos(theta);
46     y_i = y_c + r*sin(theta);
47     glColor3f(0,1,0);
48     glVertex2d(x_i, y_i);
49 } glEnd();
50 glFlush();
51 void Initial() {
52     glClearColor(1.0, 0.5, 0.5, 0);
53     glMatrixMode(GL_PROJECTION);
54     glLoadIdentity();
55     gluOrtho2D(0, 100, 0, 100);
56 } int main() {
57     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
58     glutInitWindowSize(400, 400);
59     glutInitWindowPosition(0, 0);
60     glutCreateWindow("Draw Traffic Signal");
61     Initial();
62     glutDisplayFunc(Traffics);
63     glutMainLoop();
64     return 0;
65 }
logs & others
```



Important Notes

- You can draw a circle by 2 shapes:
 1. **Solid (Fill) Circle** by use `glBegin(GL_POLYGON)` function and write a for loop inside it as the program(3).
 2. **Edge (empty) Circle** by use:
 - `glBegin(GL_POINTS)` inside the for loop as program(1).
 - `glBegin(GL_LINE_LOOP)` or `glBegin(GL_LINE_STRIP)` as we
 - Use `glBegin(GL_POLYGON)` with `glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)`
- You can use **Stipple pattern (lines, polygons)** with draw a circle.
- You can draw one circle by **multiple colors** using: `glColor3f(1, V, 200)` or `glColor3ub(V1, V2, V3)` functions inside for loop and use also variables in this function as the figure.
- You can draw a circle in any **place** by change the center point(X_c , Y_c).
- You can draw a circle by any **size** by change **radius** value.



Exercise

CS
Department

- **Draw a Circle that satisfies the following specifications:**
 - **Point Size= 16.**
 - **$X = x_c = 0$, $Y = y_c = 0$, Radius = 0.8, and theta = 0.**
 - **Circle Color Line = Blue.**
 - **Background Color = Yellow.**
 - **Window Title Bar = "My Circle".**

T.Mariah Khayat

Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

وزارة التعليم

Umm AlQura University

جامعة أم القرى

Adham University College

الكلية الجامعية بأضم

Computer Science Department

قسم الحاسب الآلي

وصلى الله وبارك على نبيينا محمد

The End Summary of Lecture Five

T.Mariah Khayat

الأستاذة/ مارية خياط

Adham University College

الكلية الجامعية بأضم

mskhayat@uqu.edu.sa

T.Mariah Khayat