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

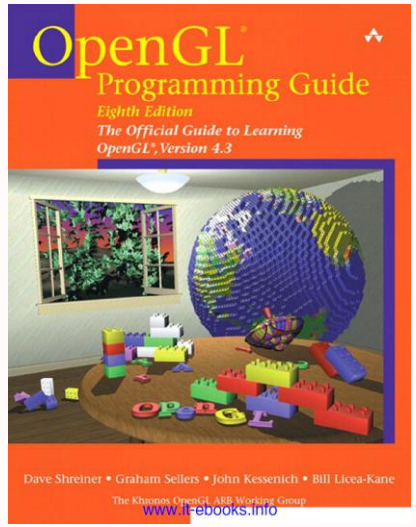
CS Department

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.*



Lecture Five

Drawing Circles and Ellipse using OpenGL

content

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:**

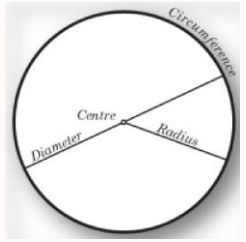
1. Simple circle drawing algorithm

- 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:

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

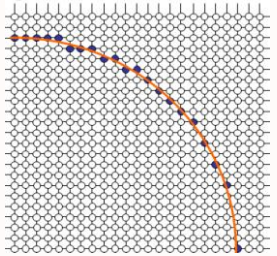
- If the circle is centered at the origin (0, 0), then the equation simplifies to:

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



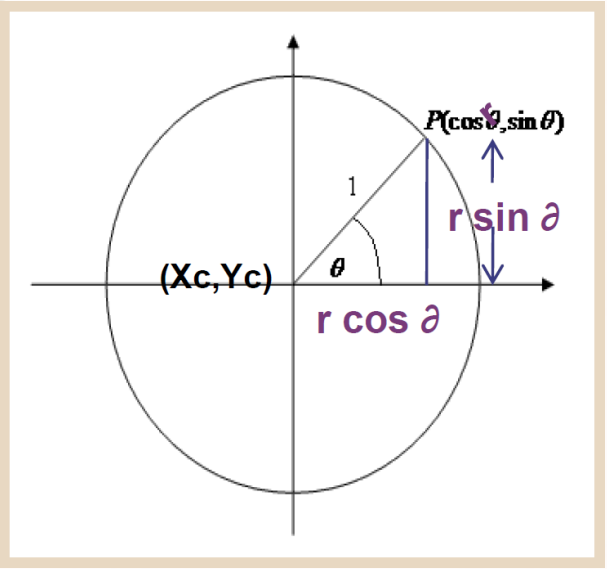
- where: **x= x coordinate ,y= y coordinate ,r = radius**
- **Note:** This method : **not very GOOD** because:
  1. the resulting circle has large gaps.
  2. the calculations are not very efficient with **square root** operations.

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



circles in OpenGL

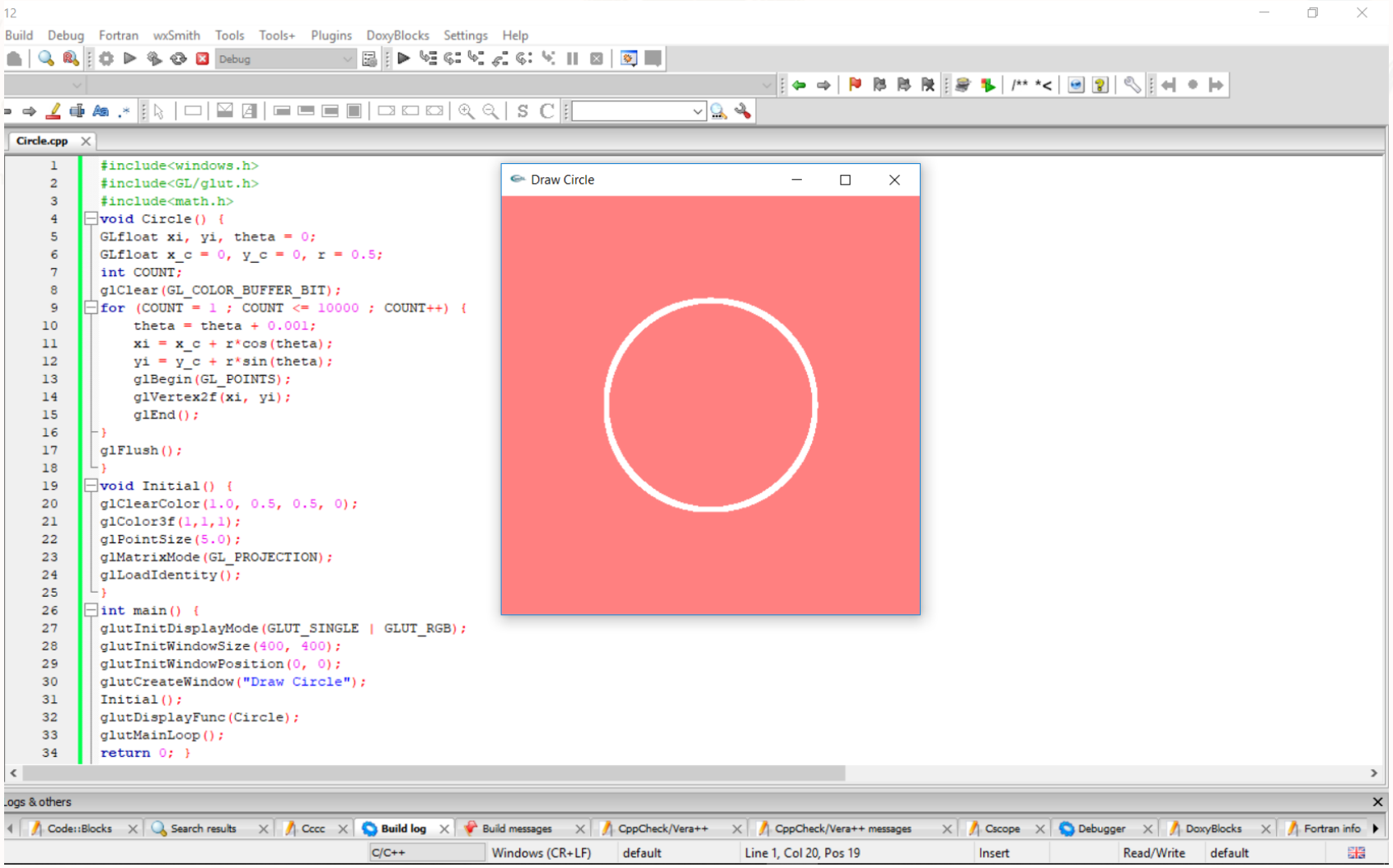
- **Draw a circle by arithmetic equations:**
  - ***Using the*** **trigonometric functions:**
  - If the circle is centered at the origin (Xc, Yc):
    - $\sin\theta = (Y - Yc)/r \quad \rightarrow \quad Y = YC + r * \sin\theta$
    - $\cos\theta = (X - Xc)/r \quad \rightarrow \quad X = XC + r * \cos\theta$
  - 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

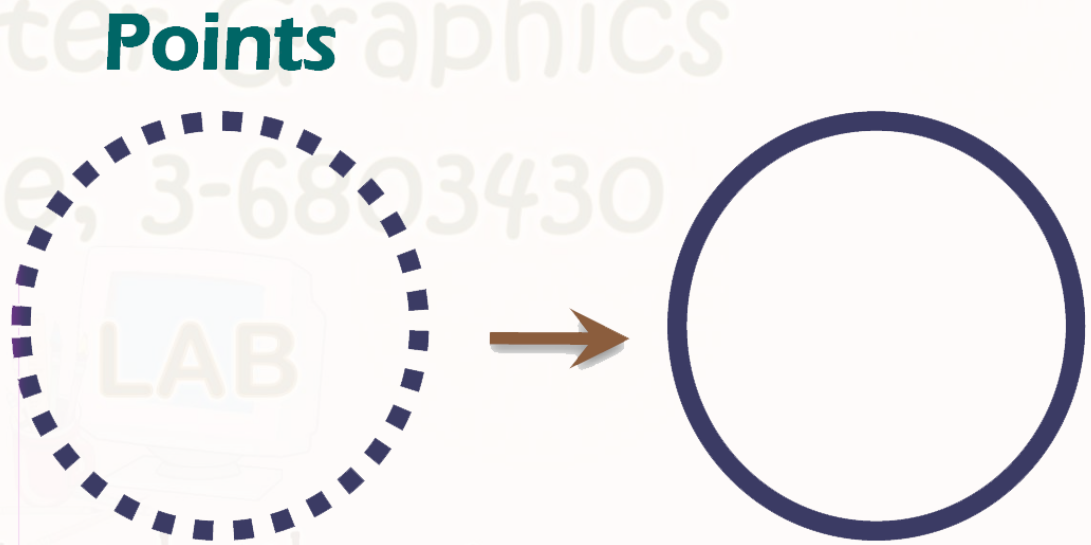
```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Circle() {
GLfloat xi, yi, theta = 0;
GLfloat x_c = 0, y_c = 0, r = 0.5;
int COUNT;
glClear(GL_COLOR_BUFFER_BIT);
for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
    theta = theta + 0.001;
    xi = x_c + r*cos(theta);
    yi = y_c + r*sin(theta);
    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; }
```

Draw circle using OpenGL



Draw circle using OpenGL

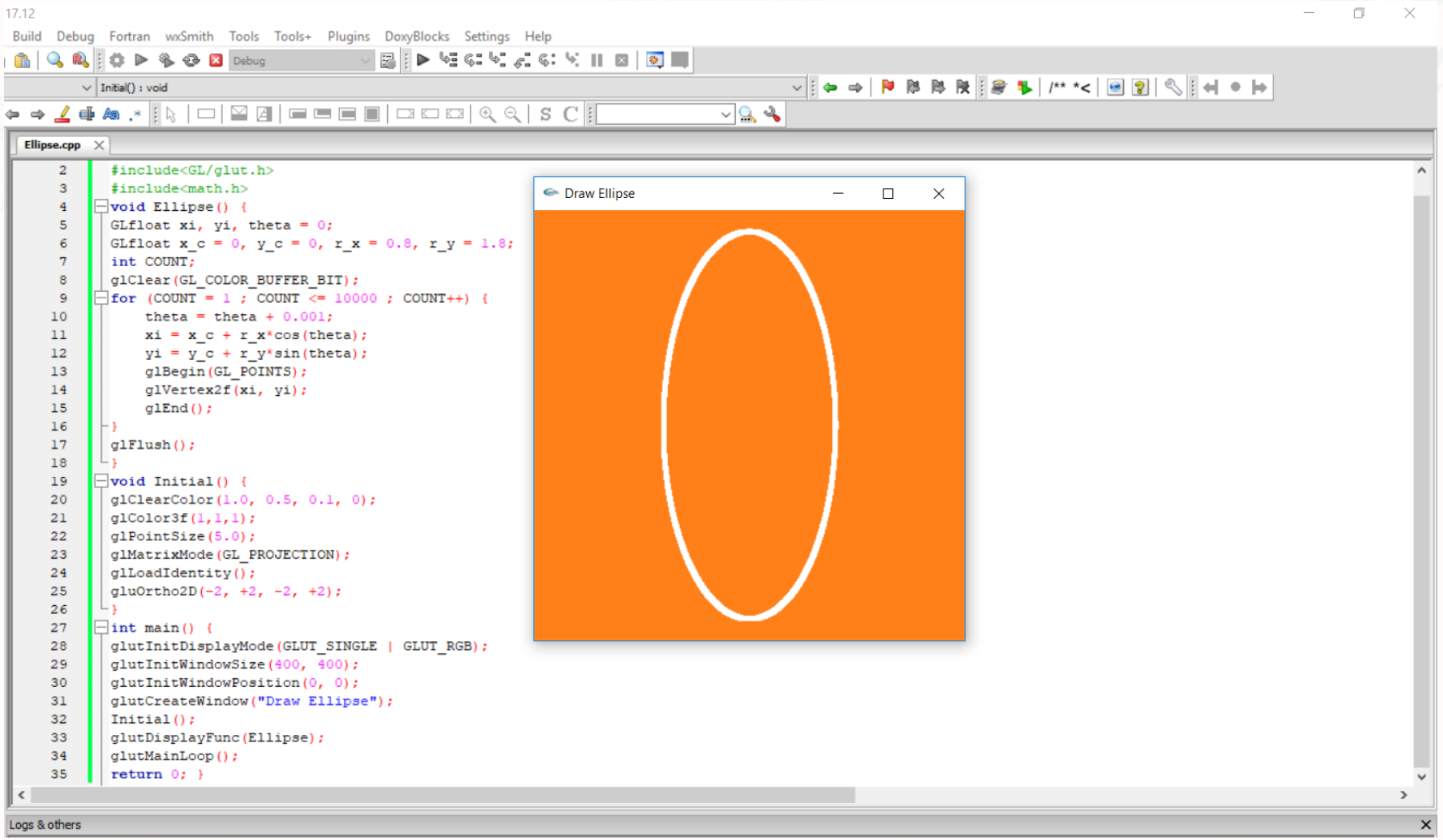
- Note: This program to Draw a circle in OpenGL:
- By draw a large number of points near each other inside loop and this points draw a circle as the following figure:



Draw Ellipse using OpenGL

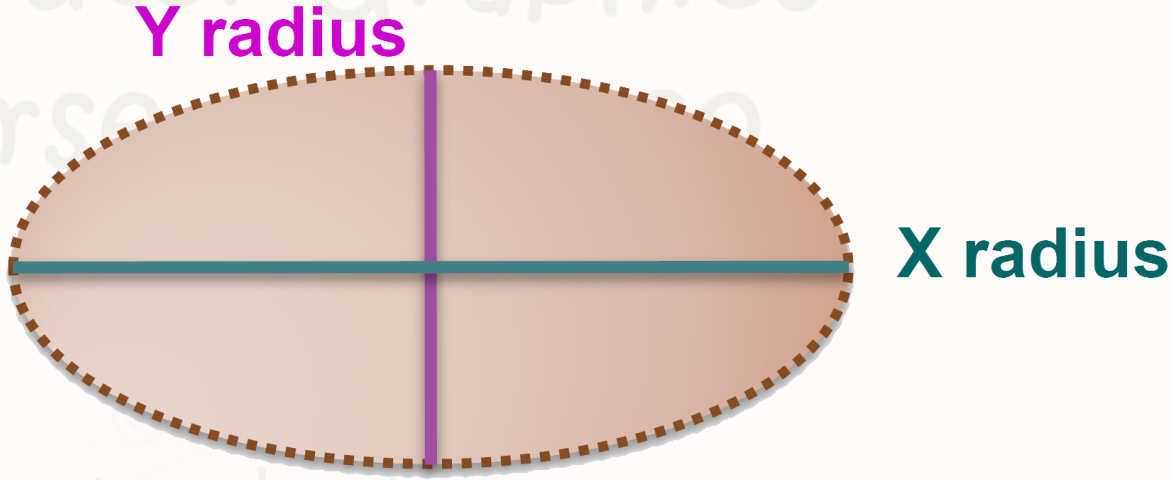
```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Ellipse() {
GLfloat xi, yi, theta = 0;
GLfloat x_c = 0, y_c = 0, r_x = 0.8, r_y = 1.8;
int COUNT;
glClear(GL_COLOR_BUFFER_BIT);
for (COUNT = 1 ; COUNT <= 10000 ; COUNT++) {
    theta = theta + 0.001;
    xi = x_c + r_x*cos(theta);
    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; }
```

Draw Ellipse using OpenGL

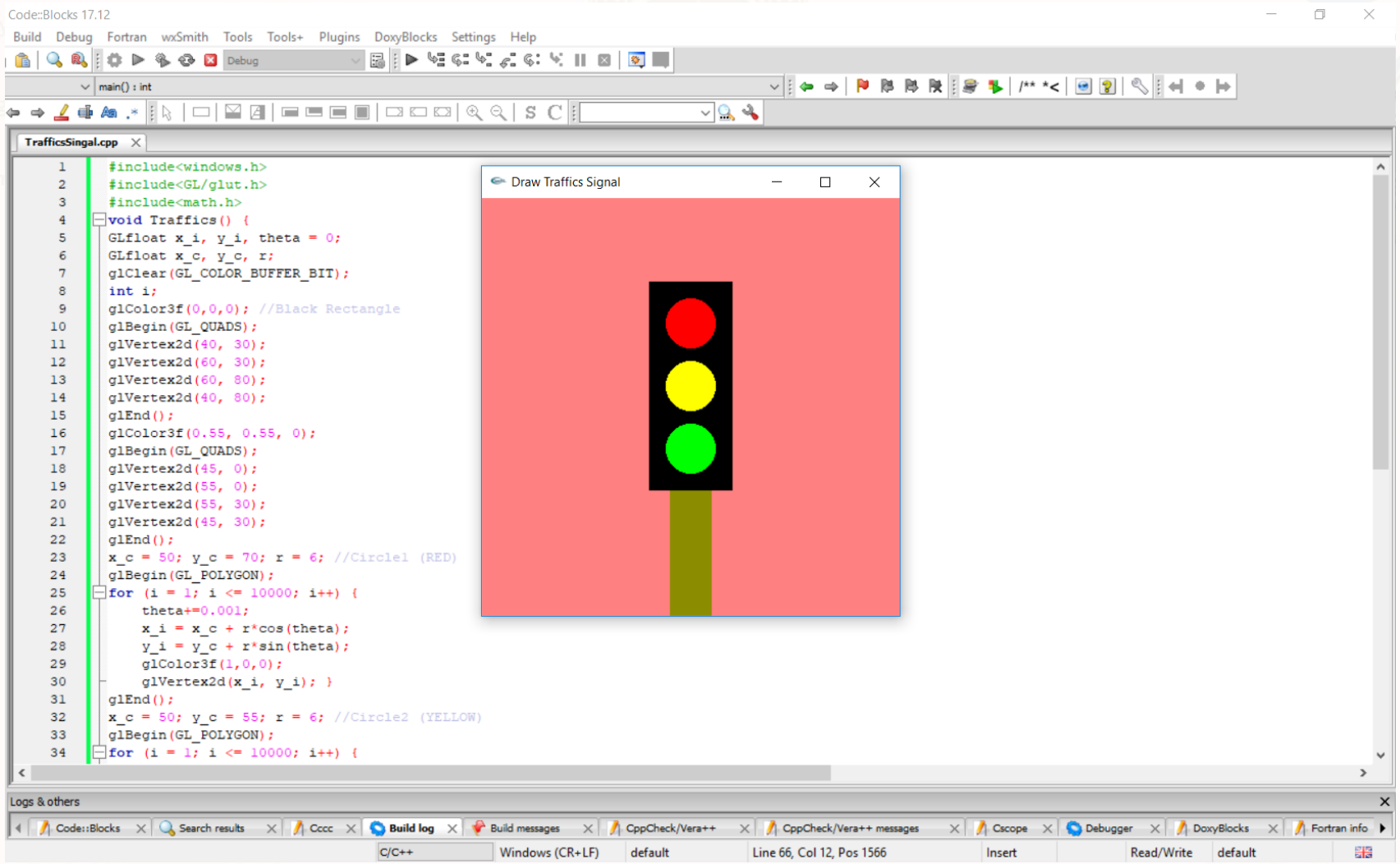


Draw Ellipse using OpenGL

- **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.



Draw Graphics Signal using OpenGL



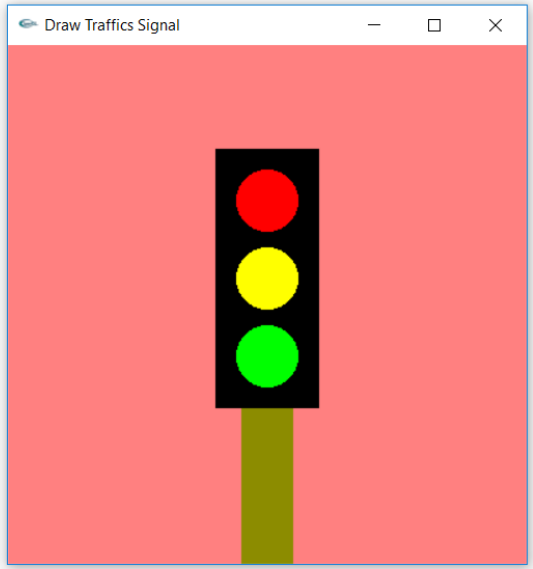
Draw Graphics Signal using OpenGL

Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

TrafficsSingal.cpp

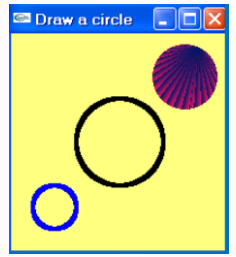
```
30      glVertex2d(x_i, y_i); }
31  glEnd();
32  x_c = 50; y_c = 55; r = 6; //Circle2 (YELLOW)
33  glBegin(GL_POLYGON);
34  for (i = 1; i <= 10000; i++) {
35      theta+=0.001;
36      x_i = x_c + r*cos(theta);
37      y_i = y_c + r*sin(theta);
38      glColor3f(1,1,0);
39      glVertex2d(x_i, y_i);}
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 Traffics Signal");
61  Initial();
62  glutDisplayFunc(Traffics);
63  glutMainLoop();
64  return 0; }
```



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(Xc , Yc).
- You can draw a circle by any **size** by change **radius** value.

Excercise

- **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”.**

وصلى الله وبارك على نبينا محمد

The End Summary of Lecture Five

T.Mariah Khayat
الأستاذة/ مارية خياط
Adham University College
الكلية الجامعية بأضم

mskhayat@uqu.edu.sa