**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



**Computer Graphics Assignment #6**

**Draw a Ellipse using Mid-point Algorithm**

**Submitted By:**

Binod Marikhu

013BSCCSIT013

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil K. Sah  Lecturer, Department of Computer Science |  |

**Submission on: August 25, 2015**

**STATEMENT**

Draw a ellipse using mid-point algorithm

**ALGORITHM**

1. Input rX, rY and ellipse center (a,b) and obtain the first point on an ellipse centered on the origin as (x,y)=(0,rY)
2. Evaluate sqX=rX\*rX and sqY=rY\*rY
3. Calculate the initial parameter in region 1 as pX=sqY-sqX\*y+sqX/4;
4. If (pX<0), plot the pixel (x+1,y) and evaluate pX+=2\*sqY\*x+sqY

else plot the pixel (x+1, y-1) and evaluate pX+=2\*sqY\*x+sqY-2\*sqX\*y;

1. Continue executing step 4 until sqY\*x<sqX\*y
2. Evaluate the initial deciding parameter for region 2 by formula

pY=sqY\*x\*x+sqX\*y\*y-sqX\*sqY;s

1. If (pY>0) plot the pixel (x,y-1) and evaluate pY-=2\*sqX\*y+sqX;

Else plot the pixel (x+1,y-1) and evaluate pY+=2\*sqY\*x-2\*sqX\*y+rX\*rX;

1. Execute the step 7 unless y!=0
2. Find the corresponding symmetry coordinates for other 3 quadrants and plot the pixels for those quadrants
3. End

**SOURCE CODE**

//---------------------------------------------------------------------------

#include <vcl\vcl.h>

#pragma hdrstop

#include "Ellipse.h"

//---------------------------------------------------------------------------

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::DrawEllipseClick(TObject \*Sender)

{

float sqX, sqY, a, b, rX, rY;

float pX, pY;

float x,y;

//(a,b)-Center of the ellipse

a=StrToInt(centerX->Text);

b=StrToInt(centerY->Text);

//Radius for the ellipse

rX=StrToInt(radiusX->Text);

rY=StrToInt(radiusY->Text);

sqX=rX\*rX;

sqY=rY\*rY;

x=0; y=rY;

//REGION 1

//INITIAL CONDITION FOR REGION 1

pX=sqY-sqX\*y+sqX/4;

//while (sqY\*x>=sqX\*y){

while (sqY\*x<sqX\*y){

x++;

if(pX<0){

//x++;

pX+=2\*sqY\*x+sqY;

} else {

y--;

pX+=2\*sqY\*x+sqY-2\*sqX\*y;

}

DisplayScreen->Canvas->Pixels[x+a][y+b]=RGB(250,000,000);

DisplayScreen->Canvas->Pixels[-x+a][y+b]=RGB(000,250,000);

DisplayScreen->Canvas->Pixels[x+a][-y+b]=RGB(000,000,250);

DisplayScreen->Canvas->Pixels[-x+a][-y+b]=RGB(100,150,200);

}

//REGION2

x+=0.5;

y--;

//INITIAL CONDITION FOR REGION 2

pY=sqY\*x\*x+sqX\*y\*y-sqX\*sqY;

while(y!=0){

y--;

if(pY>0){

//y--;

pY-=2\*sqX\*y+sqX;

} else {

x++;//y--;

pY+=2\*sqY\*x-2\*sqX\*y+rX\*rX;

}

DisplayScreen->Canvas->Pixels[x+a][y+b]=RGB(250,000,000);

DisplayScreen->Canvas->Pixels[-x+a][y+b]=RGB(000,250,000);

DisplayScreen->Canvas->Pixels[x+a][-y+b]=RGB(000,000,250);

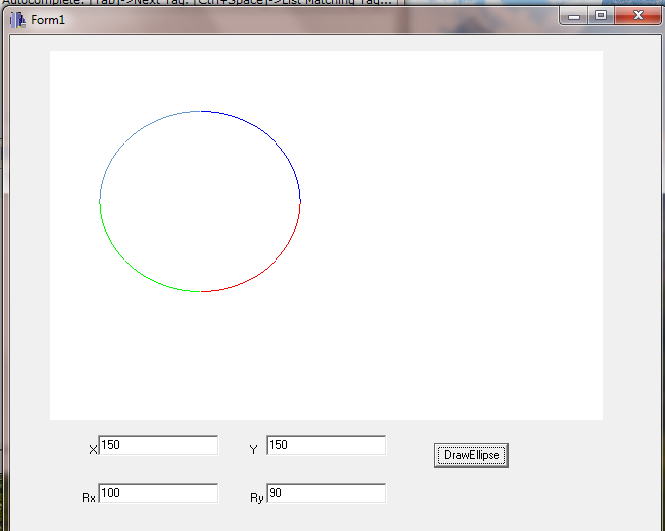
DisplayScreen->Canvas->Pixels[-x+a][-y+b]=RGB(100,150,200);

}

}

//---------------------------------------------------------------------------

**OUTPUT**

****

**CONCLUSION**

Hence, an ellipse having center (150,150), major axis 100units and minor axis 90 units was drawn using mid-point algorithm in C++ builder.