**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



**Computer Graphics Assignment #6**

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

**Submitted By:**

Bishal Pandey

013BSCCSIT016

**Submitted to:**

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

**Submission on: August 25, 2015**

**STATEMENT: Implement midpoint ellipse algorithm in C++ builder**

**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 "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int xc,yc,rx,ry,x,y,p;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::Draw\_ellipseClick(TObject \*Sender)

{

xc=StrToInt(Edit1->Text);

yc=StrToInt(Edit2->Text);

rx=StrToInt(Edit3->Text);

ry=StrToInt(Edit4->Text);

x=0;

y=ry;

p=(ry\*ry)-(rx\*rx\*ry)+((rx\*rx)/4);

while((2\*x\*ry\*ry)<(2\*y\*rx\*rx))

{

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(155,125,125);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(125,1,0);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,155,1);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(155,15,100);

if(p<0)

{

x=x+1;

p=p+(2\*ry\*ry\*x)+(ry\*ry);

}

else

{

x=x+1;

y=y-1;

p=p+(2\*ry\*ry\*x+ry\*ry)-(2\*rx\*rx\*y);

}

}

p=((float)x+0.5)\*((float)x+0.5)\*ry\*ry+(y-1)\*(y-1)\*rx\*rx-rx\*rx\*ry\*ry;

while(y>=0)

{

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(0,155,100);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(155,100,0);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,155,100);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(150,150,150);

if(p>0)

{

y=y-1;

p=p-(2\*rx\*rx\*y)+(rx\*rx);

}

else

{

y=y-1;

x=x+1;

p=p+(2\*ry\*ry\*x)-(2\*rx\*rx\*y)-(rx\*rx);

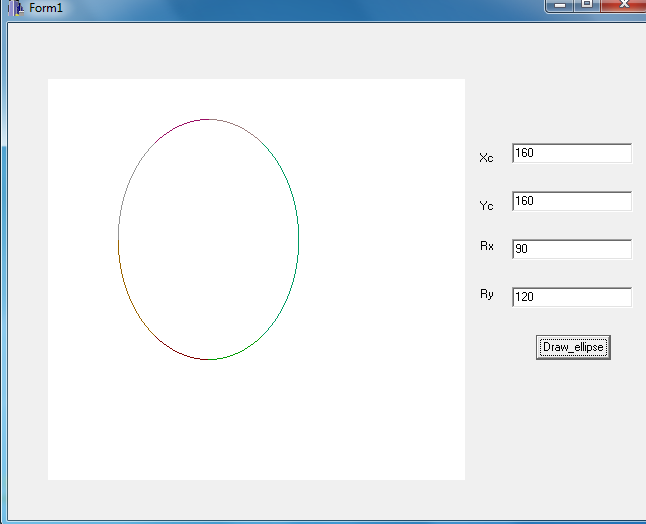
}

}

}

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

**OUTPUT**

****

**CONCLUSION**

Hence, an ellipse having center (160,160), major axis 90 units and minor axis 120 units was drawn using mid-point algorithm in C++ builder.