ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

**(Affiliated to Tribhuvan University)**



**Computer Graphics**

**Lab Assignment #6**

**DRAW AN ELLIPSE USING MIDPOINT ALGORITHM**

**Submitted By**

Alok Shrestha

013BSCIT005

B.Sc. CSIT Year II/IV Semester

**Submitted To**

Er. Anil Shah

Lecturer

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

August 2015

**STATEMENT**

DRAW AN ELLIPSE USING MIDPOINT ALGORITHM

**ALGORITHM**

**Step 1**: Input radius rx, ry and ellipse center (Xc, Yc) and obtain the first point on the circumference of a circle centered on the origin as (X0, Y0) = (0, ry)

**Step 2:** Calculate the initial values of the decision parameter in region 1 as

P10 = r2y – r2x ry + 1/4 r2x

**Step 3:** At each Xk position in region 1, starting at k = 0, perform the following test:

If P1k < 0, the next point to plot is (Xk+1, Yk) and

P1k+1 = P1k+2 r2yXk+1 + r2y

Otherwise the next point is (Xk+1, Yk-1) and

P1k+1 = P1k+2 r2yXk+1 - 2r2xYk+1 + r2y

With

2 r2yXk+1=2 r2yXk+ 2r2y

2r2xYk+1=2r2xYk- 2r2x

**Step 4:** Calculate the initial values of the decision parameter in region 2 as

P20 = r2y(X0+1/2)2+ r2x(Y0 – 1)2- r2x r2y

**Step 5:** At each position starting at Yk position in region 2, starting at k = 0,

perform the following test:

If P2k > 0, the next point to plot is (Xk, Yk-1) and

P2k+1 = P2k - 2 r2yYk+1 + r2x

Otherwise the next point is (Xk+1, Yk-1) and

P2k+1 = P2k - 2 r2yXk+1 - 2r2xYk+1 + r2x

**Step 6**: Determine symmetry points in the other three octants

**Step 7**: Move each pixel position(X, Y) onto the circular path centered on

(Xc, Yc) and plot the coordinate values as

X = X + Xc Y = Y + Yc

**Step 8:** Repeat steps for region 1 until 2 r2yX>=2 r2xY.

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int xc, yc, rx, ry;

int x, y,p;

int i;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::Button1Click(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(100,125,150);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(100,1,0);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,125,150);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(200,15,150);

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(100,125,150);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(100,1,0);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,125,150);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(200,15,150);

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);

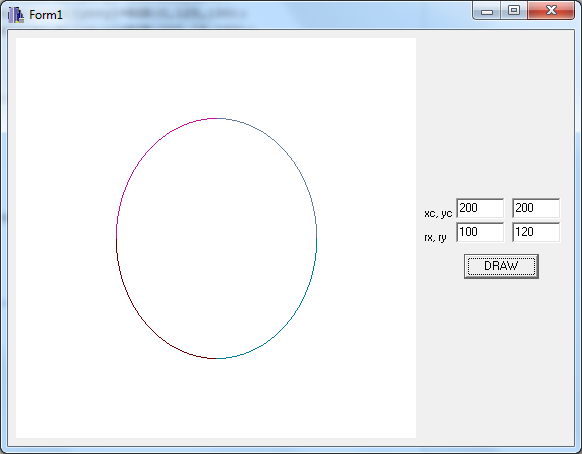
}

}

}

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

**OBSERVATION/S**



**CONCLUSION**

Hence an ellipse was designed using C++builder using mid-point algorithm.