ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Computer Graphics Lab Assignment #6**

**Submitted by:**

Arun Sanjel

013BSCCSIT010

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil Sah  Lecturer, St. Xavier’s College |  |

**STATEMENT**

“Draw a ellipse using midpoint ellipse algorithm where radius and center are provided by user.”

**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::butDrawClick(TObject \*Sender)

{

int xc,yc,rx,ry;

xc = StrToInt(inXc->Text);

yc = StrToInt(inYc->Text);

rx = StrToInt(inRx->Text);

ry = StrToInt(inRy->Text);

int x, y, p;

x=0; //Initial Coordinates

y=ry; //Initial Coordinates

p=(ry\*ry)-(rx\*rx\*ry)+((rx\*rx)/4); // Initial Decision Parameter for Region 1

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

{

//Region 1 with Symmetric Property

outPanel->Canvas->Pixels[xc+x][yc-y]=RGB(255,0,0);

outPanel->Canvas->Pixels[xc-x][yc+y]=RGB(0,255,0);

outPanel->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,255);

outPanel->Canvas->Pixels[xc-x][yc-y]=RGB(2055,155,0);

if(p<0)

{

x+=1;

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

}

else

{

x+=1;

y-=1;

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; // Initial Decision parameter for Region 2

while(y>=0)

{

//Region 2 with Symmetric Property

outPanel->Canvas->Pixels[xc+x][yc-y]=RGB(255,120,255);

outPanel->Canvas->Pixels[xc-x][yc+y]=RGB(100,255,130);

outPanel->Canvas->Pixels[xc+x][yc+y]=RGB(110,155,255);

outPanel->Canvas->Pixels[xc-x][yc-y]=RGB(123,145,185);

if(p>0)

{

y-=1;

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

}

else

{

y-=1;

x+=1;

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

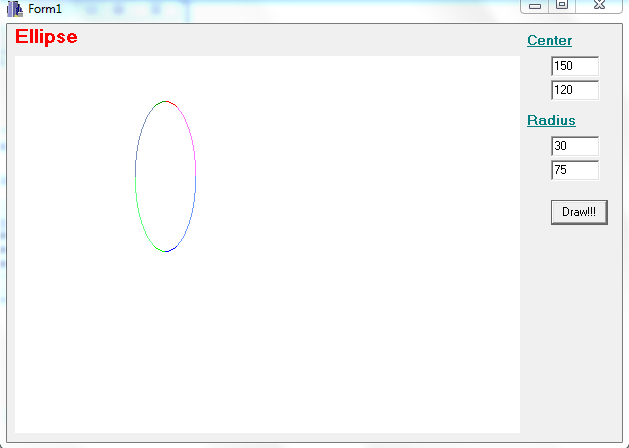
}

}

}

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

**OUTPUT**

****

**CONCLUSION**

Thus, ellipse was drawn using midpoint algorithm where user provided center and radius of the ellipse using C++ builder.