**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**COMPUTER GRAPHICS**

**LAB ASSIGNMENT #6**

**Submitted by:**

Linus Dhakal

013BSCCSIT022

**Submitted to:**

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

**Date of submission:** 28th August, 2015

**STATEMENT: WAP A PROGRAM TO IMPLEMENT MD-POINT ALGORITHM TO DRAW AN ELLIPSE.**

**ALGORITHM:**

**Step1:** Input *rx*, *ry*, and ellipse center (*xc*, *yc*), and obtain the first point on an ellipse centered on the origin as

(*x*0, *y*0) = (0, *ry*)



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

**Step 3:** At each *xi* position, starting at *i* = 0, if *p*1*i* < 0, the next point along the ellipse centered on (0, 0) is (*xi* + 1, *yi*) and



Otherwise, the next point is (*xi* + 1, *yi* – 1) and



and continue until

**Step 4:** (*x*0, *y*0) is the last position calculated in region 1. Calculate the initial parameter in region 2 as



**Step 5:** At each *yi* position, starting at *i* = 0, if *p*2*i* > 0, the next point along the ellipse centered on (0, 0) is (*xi*, *yi* – 1) and



otherwise, the next point is (*xi* + 1, *yi* – 1) and



Use the same incremental calculations as in region 1. Continue until *y* = 0.

**Step 6:** For both regions determine symmetry points in the other three quadrants.

**Step 7:** Move each calculated pixel position (x, y) onto the elliptical path centered on (*xc*, *yc*) and plot the coordinate values

***x* = *x* + *xc* , *y* = *y* + *yc***

**SOURCE CODE:**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "eclipse.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

\_\_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(10,125,10);

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

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,25,15);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(20,15,150);

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(250,0,0);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,250,0);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,250);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(100,200,250);

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);

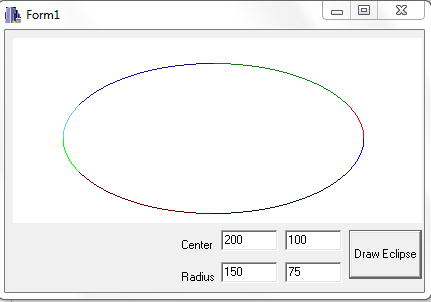
}

}

}

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

OUTPUT:



**CONCLUSION:**

Hence, we were able to draw an ellipse in C++ Builder by implementing mid-point algorithm.

**REFERENCE:**

[1] <http://www.embarcadero.com/products/cbuilder>