ST. XAVIER'S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**COMPUTER GRAPHICS**

**LAB ASSIGNMENT # 6**

**SUBMITTED BY:**

Arjun Neupane  
013BSCCSIT009

2nd Year/4th Sem

**SUBMITTED TO:**

|  |  |
| --- | --- |
| **Er. Anil K. Sah**  Supervisor,  Department of Computer Science  St. Xavier’s College |  |

**Date of Submission:** 25th August, 2015

**STATEMENT:**

Draw an ellipse using Mid-Point Algorithm.

**ALGORITHM:**

1. Get parameters *a*, *b*, *h*, *k* for center coordinate *h* and *k* and major & minor axis lengths 2*a* and 2*b.*
2. Calculate the initial decision parameter value in the first region: http://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/empty.gifhttp://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/eq0023M.gifhttp://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/empty.gif.
3. Use the formulas above to iterate *px*k+1 until *b*2*x*>*a*2*y*.
4. Rename the current (*x*k,*y*k) as (*x*0,*y*0) and calculate the initial decision parameter value in the 2nd region: http://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/empty.gifhttp://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/eq0024M.gifhttp://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/empty.gif.
5. Use the formulas above to iterate *py*k+1 until *y* <= 0.
6. For both regions plot the other three symmetry points.

1. Shift to center at *h*, *k*.

**PROGRAM 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::EllipseClick(TObject \*Sender)

{

int x,y;

int xc,yc,Rx,Ry;

int p;

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(100,0,0);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(100,200,0);

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

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(205,255,0);

if(p<0)

{

x=x+1;

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

}

else

{

x=x+1;

y=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;

while(y>=0)

{

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

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(100,200,0);

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

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(205,255,0);

if(p>0)

{

y=y-1;

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

}

else

{

y=y-1;

x=x+1;

p+=(2\*Ry\*Ry\*x)-(2\*Rx\*Rx\*y)-(Rx\*Rx);

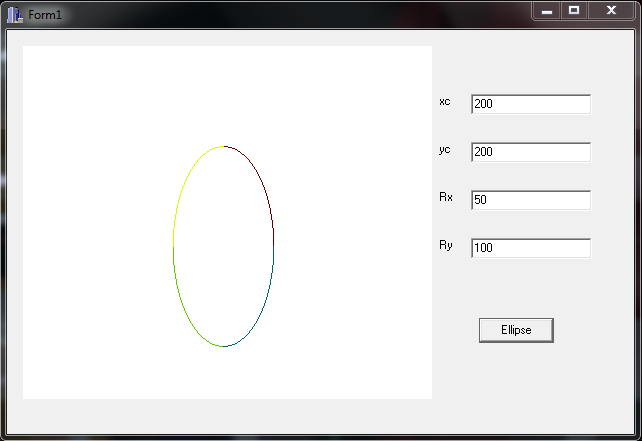
}

}

}

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

**OUTPUT SCREEN:**



**CONCLUSION:**

Hence, an ellipse was drawn using mid-point algorithm.