**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**COMPUTER GRAPHICS**

**LAB ASSIGNMENT #06**

**Submitted by:**

Rojesh Tamrakar

013BSCCSIT032

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Anil K. Sah** |  |

Lecturer

Department of Computer Science

Date of submission: 25th August, 2015

# STATEMENT: To implement the mid-point ellipse algorithm to draw a ellipse:

# ALGORITHM:

# Get parameters a, b, h, k for center coordinate h and k and major & minor axis lengths 2a and 2b.

# Calculate the initial decision parameter value in the first region: .

# Use the formulas above to iterate pxk+1 until b2x>a2y.

# Rename the current (xk,yk) as (x0,y0) and calculate the initial decision parameter value in the 2nd region: .

# Use the formulas above to iterate pyk+1 until y <= 0.

# For both regions plot the other three symmetry points.

# Shift to center at h, k.

# 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;

# //---------------------------------------------------------------------------

# \_\_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; //Initial Coordinates

# y=ry; //Initial Coordinates

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

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

# { //Region 1 with Symmertic Property

# Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,10,10);

# Image1->Canvas->Pixels[xc-x][yc+y]=RGB(10,255,10);

# Image1->Canvas->Pixels[xc+x][yc+y]=RGB(10,10,255);

# Image1->Canvas->Pixels[xc-x][yc-y]=RGB(255,255,10);

# 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; // Intial Decision parameter for Region 2

# while(y>=0)

# {

# //Region 2 with Symmertic Property

# Image1->Canvas->Pixels[xc+x][yc-y]=RGB(10,255,255);

# Image1->Canvas->Pixels[xc-x][yc+y]=RGB(255,10,255);

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

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

# 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);

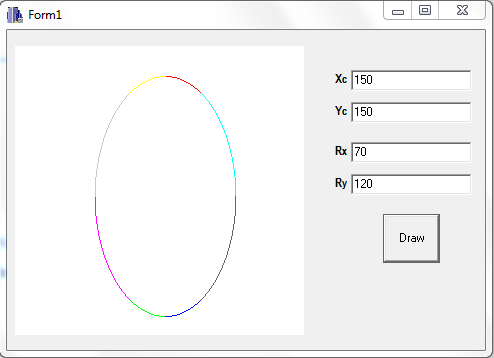
# }

# }

# }

# //---------------------------------------------------------------------------

# OUTPUT



# CONCLUSION:

Hence, the mid-point ellipsse algorithm was implemented using C++ builder. The mid-point ellipse algorithm helped to draw the ellipse by entering the radius and center.

**REFERENCE:**

<http://geofhagopian.net/sablog/Slog-october/sablog-10-24-05.htm>