**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



**Computer Graphics Assignment #6**

**Draw an ellipse using mid-point algorithm**

**Submitted By:**

Bikash Paneru

013BSCCSIT012

**Submitted to:**

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

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

**STATEMENT**

Write a program to draw an ellipse using the mid-point algorithm.

­

**ALGORITHM**

1. Get parameters (cX,cY) for center and a and b as radius for x and y axis respectively.
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.gif
3. Set x=0 and y=b
4. Use these formulas to iterate *px*k+1 until *b*2*x*>*a*2*y*.
   1. x=x+1
   2. If P<0, P = P + 2b2x+b2
   3. Else, y=y-1, P = P+2b2x+b2-2a2y
   4. Plot a pixel at (x,y) and its reflections plus (cX,cY) at other three quadrants
5. Continue from the previous values of x and y to calculate the initial decision parameter value in the 2nd region:

http://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/eq0024M.gifhttp://geofhagopian.net/sablog/Slog-october/sablog-10-24-05_files/empty.gif.

1. Use these formulas to iterate *py*k+1 until *y>=0*.
   1. y=y-1
   2. If P>0, P = P + 2a2(y-1)+a2
   3. Else, x=x+1, P = P+2b2x -2a2y+a2
   4. Plot a pixel at (x,y) and its reflections plus (cX,cY) at other three quadrants

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "ellipse.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

void plotInFourQuadrants(int x, int y, int cX, int cY, TCanvas\* Canvas,int region) {

//Draw the points

int colorDecider = (region-1)\*255;

Canvas->Pixels[x+cX][y+cY] = RGB(colorDecider,0,0);

Canvas->Pixels[x+cX][-y+cY] = RGB(0,255,colorDecider);

Canvas->Pixels[-x+cX][-y+cY] = RGB(0,colorDecider,255);

Canvas->Pixels[-x+cX][y+cY] = RGB(colorDecider,255,100);

}

void drawRegion1(int cX, int cY, int rX, int rY, TCanvas\* Canvas, int&x, int&y) {

int P; //The Decision Parameter

P = rY\*rY - rX\*rX\*rY + ((rX\*rX)/4); //Initial Decision Parameter

while((2\*rY\*rY\*x)<(2\*rX\*rX\*y)) {

x+=1; //Increment x in each step

//Check decision parameter for value of y

if(P<0) {

P += 2\*rY\*rY\*x + (rY\*rY);

//There is no change in y

} else {

y-=1; //Decrement y

P += 2\*rY\*rY\*x + rY\*rY - 2\*rX\*rX\*y;

}

plotInFourQuadrants(x,y,cX,cY,Canvas,1);

}

}

void drawRegion2(int cX, int cY, int rX, int rY, TCanvas\* Canvas, int& x, int& y) {

int P; //The Decision Parameter

P = rY\*rY\*(x+0.5)\*(x+0.5) + rX\*rX\*(y-1)\*(y-1) - rX\*rX\*rY\*rY; //Initial Decision Parameter

while(y>0) {

y-=1; //Decrement y in each step

//Check decision parameter for value of x

if(P>0) {

P -= 2\*rX\*rX\*(y-1) + rX\*rX;

//There is no change in x

} else {

x+=1; //Increment x

P += 2\*rY\*rY\*x - 2\*rX\*rX\*y + rX\*rX;

}

plotInFourQuadrants(x,y,cX,cY,Canvas,2);

}

}

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

void \_\_fastcall TForm1::drawButtonClick(TObject \*Sender)

{

//Get the center and two radius from the input form

int cX = StrToInt(centerX->Text);

int cY = StrToInt(centerY->Text);

int rX = StrToInt(radiusX->Text);

int rY = StrToInt(radiusY->Text);

//Points to draw

int x = 0;

int y = rY;

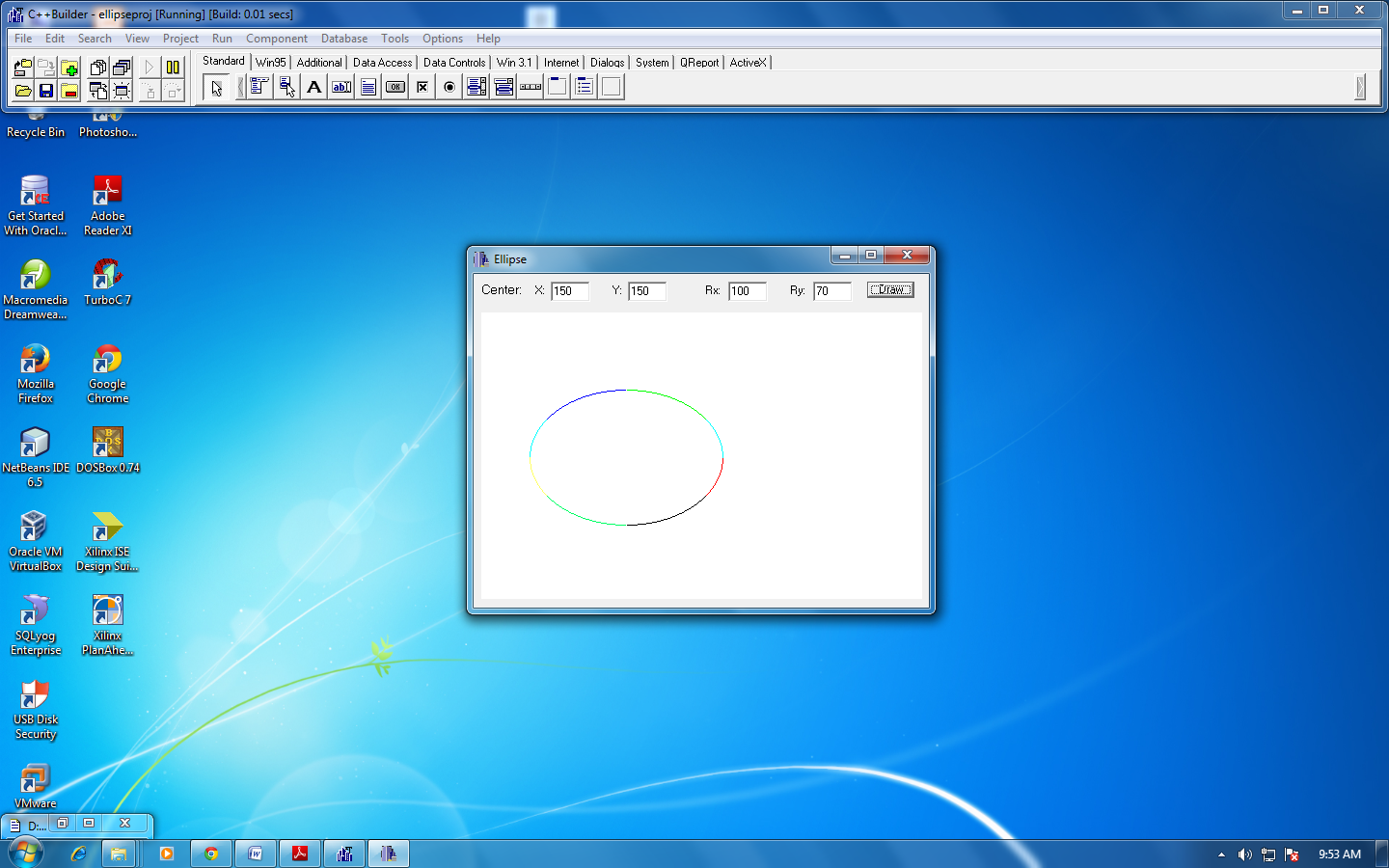
drawRegion1(cX,cY,rX,rY,drawArea->Canvas,x,y);

drawRegion2(cX,cY,rX,rY,drawArea->Canvas,x,y);

}

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

**OUTPUT:**

****

**CONCLUSION**

Hence, a program to draw an ellipse using the mid-point algorithm was implemented by using C++ with C++Builder.