**ST.XAVIER’S COLLEGE**

Maitighar, Kathmandu



**Computer Graphics Assignment #6**

Draw a Ellipse using Mid-point Algorithm

**Submitted By:**

Dwarika Shiwakoti

013BSCCSIT020

**Submitted to:**

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

Submitted on 28th August 2015

**STATEMENT:**

**WRITE A PROGRAM TO IMPLEMENT MID POINT ELLIPSE ALGORITHM IN C++ BUILDER.**

**ALGORITHM:**

**Step 1:** input center (xc,yc) and rx and ry for the ellipse and obtain the first point as (x0,y0)=(0, ry)

**Step 2:** calculate initial decision parameter value in region 1 as

P10=

**Step 3:** at each xk­ position, in region1, starting at k=0, compute

xk+1=xk+1

if p1k=p1k+2xk+1+

yk+1=yk+1

otherwise next point to plot is

yk+1=yk-1

P1k+1=P1k+xk+1+-2yk+1 with xk+1=xk+1 and yk+1=yk-1

**Step 4:** calculate the initial value of decision parameter at region 2 using last calculated point say(x0,y0) in region 1 as

P20=

**Step 5:** at each yk position in region 2 starting at k=0, perform computation

yk+1=y-1

if P2k>0, then

xk+1=xk

P2k+1=P2k-

Otherwise

xk+1=xk+1

P2k+1=P2k+2 where xk+1=xk+1 and yk+1=yk+1

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

**Step 7:** move each calculated point (xk,yk) on to the centered (xc,yc) ellipse path as

xk=xk+xc

yk=yk+yc

**Step 8:** repeat the process for region 1 until and region 2 until (xk,yk)=(rx,0).

**SOURCE CODE:**

#include <vcl\vcl.h>

#pragma hdrstop

#include "elip.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

int xc,yc,rx,ry;

xc = StrToInt(Edit1->Text);

yc = StrToInt(Edit2->Text);

rx = StrToInt(Edit3->Text);

ry = StrToInt(Edit4->Text);

int x, y, p;

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(255,0,0);

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

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

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(2055,155,0);

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; // Intial Decision parameter for Region 2

while(y>=0)

{

//Region 2 with Symmertic Property

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,120,255);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(100,255,130);

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(110,155,255);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(123,145,185);

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);

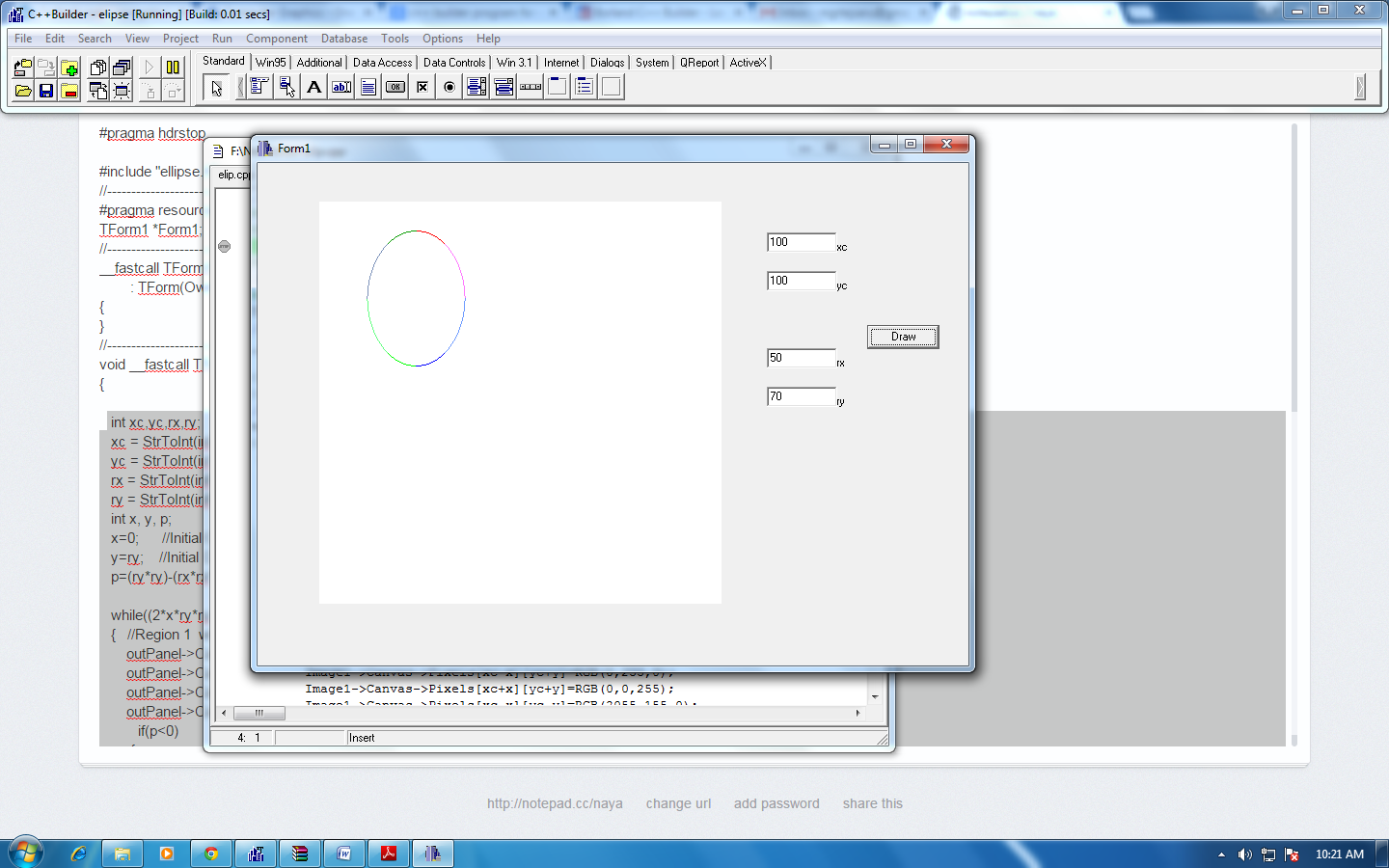
}

}

}

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

**OUTPUT SCREEN**



**CONCLUSION**

Hence, the midpoint ellipse algorithm was implemented in C++ Builder according to the given centre and radius.