ST. XAVIER’S COLLEGE

**Maitighar,Kathmandu**

**(Affiliated to Tribhuvan University)**

****

**Computer graphics**

**Lab Assignment #5**

**Submitted By**

Ajita Khatiwada

B.Sc. CSIT

Year II/III Semester

013BSCIT004

**Submitted To**

Er. Anil Kumar Shah

Lecturer

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

August 21, 2015

Statement

To Draw a circle using midpoint circle algorithm.

ALGORITHM

Step 1: Input the radius and circle centre(xc,yc) and obtain the first point on circle centers at origin as

(X0, y0)= (0, r).

Step 2: Calculate initial decision parameter

P0=5/4-r

Step 3: At each x, position starting at k=0 perform the tests

If p0<0 next point along the circle centre at (0, 0) is (xk+1,yk)

Pk+1=pk+2xk+1+1)

Otherwise the next point along circle is (xk+1,yk-1)

Pk+1=pk+2xk+1+1-2yk-1

Step 4: Determine symmetry point on the other seven octants

Step 5: Move each calculated pixels positions (x,y) in to circle path centered at (xc,yc) as

x=x+xc

y=y+yc

Step 6: Repeat 3 through 5 until x>=y

SOURCE CODES :

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "CIRCLE.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int xc,yc,radius,p,y,x;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::Edit1Change(TObject \*Sender)

{

xc=StrToInt(Edit1->Text);

}

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

void \_\_fastcall TForm1::Edit2Change(TObject \*Sender)

{

yc=StrToInt(Edit2->Text);

}

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

void \_\_fastcall TForm1::Edit3Change(TObject \*Sender)

{

radius=StrToInt(Edit1->Text);

}

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

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

{

int x=0,y=radius;

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

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

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

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

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

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(127,127,127);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(127,255,0);

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,255,127);

int p=1-radius;

while(x<y)

{

if(p<0)

{

x=x+1;

p=p+2\*x+1;

}

else

{

x=x+1;

y=y-1;

p=p+2\*(x-y)+1;

}

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

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(20500,255,255);

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

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(100,255,0);

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

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(127,100,127);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(127,255,0);

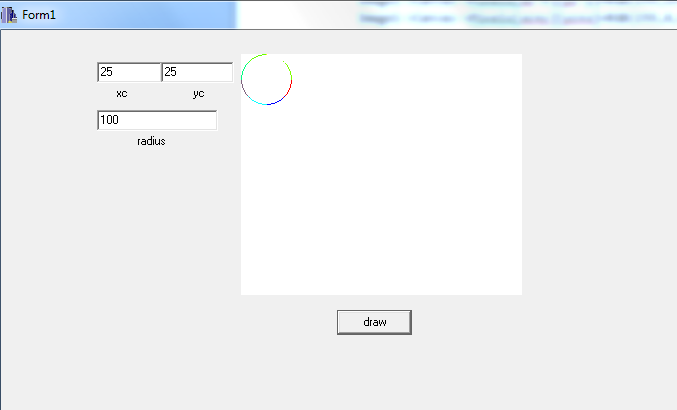
Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,255,127);

}

}

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

OUTPUT :



Conclusion

hence a circle was drawn in C++ builder using midpoint circle algorithm.