ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Computer Graphics Lab Assignment #5**

**Submitted by:**

Prashraya Hada

013BSCCSIT027

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil Sah  Lecturer, St. Xavier’s College |  |

Statement

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 Code

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "circ.h"

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

#pragma resource "\*.dfm"

int x,y,pk,r,xc,yc;

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

xc=StrToInt(Edit1->Text);

yc=StrToInt(Edit2->Text);

r=StrToInt(Edit3->Text);

x=0;

y=r;

pk=1-r;

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

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(50,255,0);

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

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

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(100,50,255);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(50,100,100);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(225,0,255);

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

while (x<y)

{

if(pk<0)

{

x=x+1;

pk=pk+(2\*x)+1;

}

else

{

x=x+1;

y=y-1;

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

}

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

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(50,255,0);

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

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

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(100,50,255);

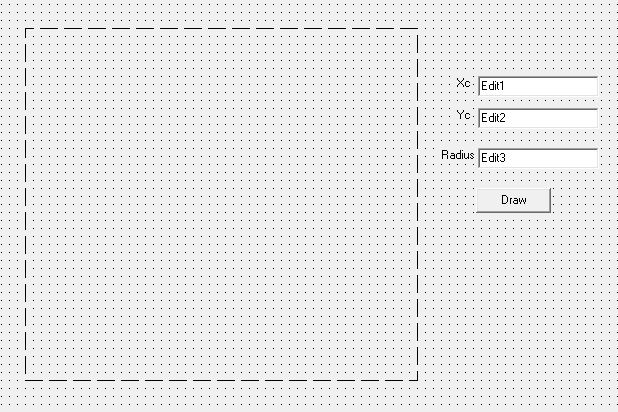
Image1->Canvas->Pixels[xc-x][yc+y]=RGB(50,100,100);

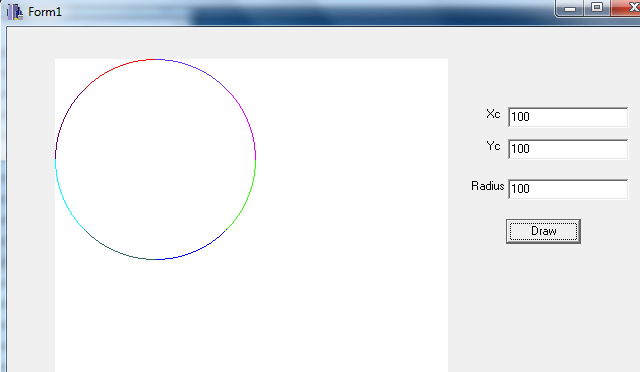
Image1->Canvas->Pixels[xc+y][yc-x]=RGB(225,0,255);

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

}

}

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

Output

Conclusion

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