ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Computer Graphics Lab Assignment #5**

**To draw a circle using midpoint algorithm**

**SUBMITTED BY:**

Hemanchal Joshi

013BSCCSIT021

**SUBMITTED TO:**

**Er. Anil K. Sah**

Lecturer, Dept. of Computer Science

St.Xavier’s college, Maitighar

**Submission Date:** 21’st August 2015

**ALGORITHM**

**Step 1:** start of the process

**Step 2:** Input the radius and circle centre(xc,yc) and obtain the first point on circle centers at origin as

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

**Step 3:** Calculate initial decision parameter

P0=5/4-r

**Step 4:** 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 5:** Determine symmetry point on the other seven octants

**Step 6:** Move each calculated pixels positions (x,y) in to circle path centered at (xc,yc) as

x=x+xc

y=y+yc

**Step 7:** Repeat 3 through 5 until x>=y

**Step 8:**  End of the process

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

int xc,yc,r;

xc=StrToInt(Edit2->Text);

yc=StrToInt(Edit3->Text);

r=StrToInt(Edit1->Text);

int p,x,y;

x=0;

y=r;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(255,0255,0);

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

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

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

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

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

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

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,0,255); p=1-r;

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(120,255,47);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(45,179,0);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(50,210,20);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(195,34,57);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(32,25,210);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(110,130,120);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(75,210,10);

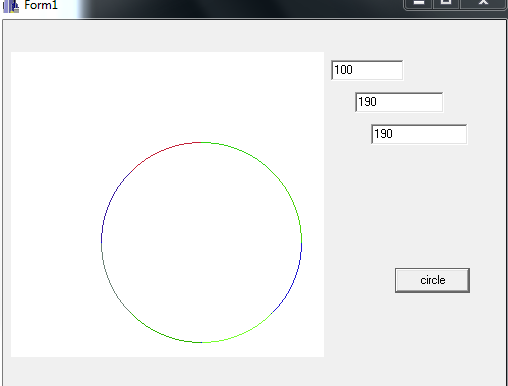
Image1->Canvas->Pixels[xc-y][yc-x]=RGB(59,29,160);

}

}

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

**OUTPUT SCREEN:**



**CONCLUSION:**

Hence, a circle was drawn using the Circle mid-point algorithm.