**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**COMPUTER GRAPHICS**

**LAB ASSIGNMENT #5**

**Submitted by:**

Lokendra puri

013BSCCSIT023

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Anil K. Sah** |  |

Lecturer

Department of Computer Science

Date of submission: 21st August, 2015

**Statement: Implement midpoint circle algorithm.**

**Algorithm:**

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

(x0, y0) = (0, r).

Step 2: calculate initial decision parameter

Po=

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

If Pk<0 next point along the center at (0, 0) is (xk+1, yk)

Pk+1= Pk+2 xk+1+1

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

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

Where 2 xk+1=2 xk+2 and 2 yk+1=2 yk+2

Step 4: determine symmetry point on the other seven octants.

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

x=x+xc and y=y+yc

Step 6: repeat 3 through 5 until x≥y.

Step 7: terminate**.**

**Source code:**

**//---------------------------------------------------------------------------**

**#include <vcl\vcl.h>**

**#pragma hdrstop**

**#include "Unit1.h"**

**//---------------------------------------------------------------------------**

**#pragma resource "\*.dfm"**

**TForm1 \*Form1;**

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

**//---------------------------------------------------------------------------**

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

**: TForm(Owner)**

**{**

**}**

**//---------------------------------------------------------------------------**

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

**{**

**r=StrToInt(Edit1->Text);**

**xc=StrToInt(Edit2->Text);**

**yc=StrToInt(Edit3->Text);**

**x=0;**

**y=r;**

**pk=1-r;**

**Image1->Canvas->Pixels[x][y]=RGB(100,200,200);**

**while(x<y)**

**{**

**if(pk<0)**

**{**

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

**x++;**

**Image1->Canvas->Pixels[xc+x][yc+y]=RGB(25,28,255);**

**Image1->Canvas->Pixels[xc-x][yc+y]=RGB(210,200,10);**

**Image1->Canvas->Pixels[xc+x][yc-y]=RGB(210,210,255);**

**Image1->Canvas->Pixels[xc-x][yc-y]=RGB(222,200,200);**

**Image1->Canvas->Pixels[xc+y][yc+x]=RGB(198,00,85);**

**Image1->Canvas->Pixels[xc-y][yc+x]=RGB(250,50,100);**

**Image1->Canvas->Pixels[xc+y][yc-x]=RGB(210,205,255);**

**Image1->Canvas->Pixels[xc-y][yc-x]=RGB(120,25,189);**

**}**

**else**

**{**

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

**x++;**

**y--;**

**Image1->Canvas->Pixels[xc+x][yc+y]=RGB(120,150,189);**

**Image1->Canvas->Pixels[xc-x][yc+y]=RGB(190,0,180);**

**Image1->Canvas->Pixels[xc+x][yc-y]=RGB(150,210,250);**

**Image1->Canvas->Pixels[xc-x][yc-y]=RGB(1,255,135);**

**Image1->Canvas->Pixels[xc+y][yc+x]=RGB(160,255,135);**

**Image1->Canvas->Pixels[xc-y][yc+x]=RGB(2,200,68);**

**Image1->Canvas->Pixels[xc+y][yc-x]=RGB(29,255,288);**

**Image1->Canvas->Pixels[xc-y][yc-x]=RGB(215,2,205);**

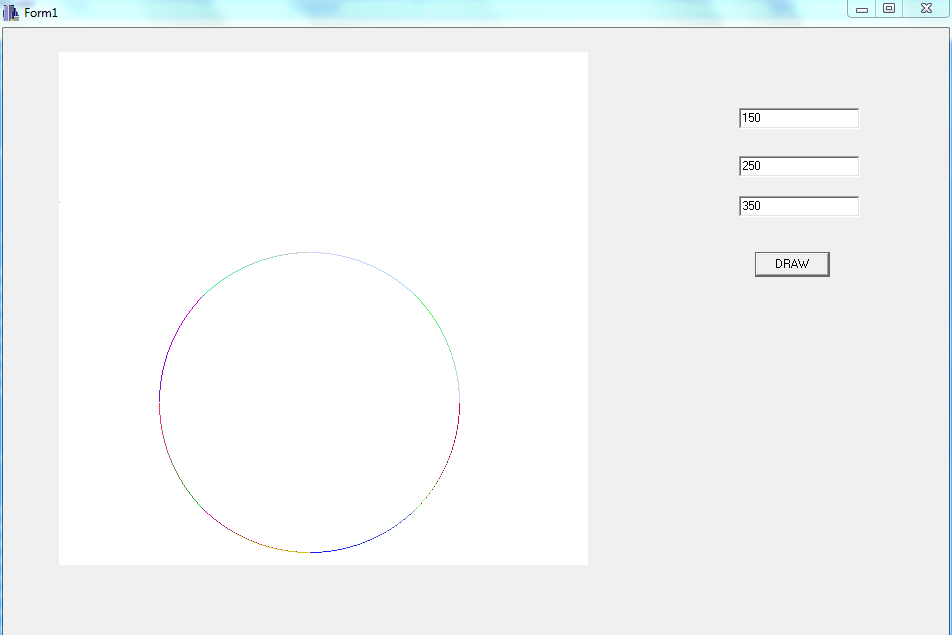
**}**

**}**

**}**

**//-----------------------------------------**

**Output:**

****

**Conclusion:**

Hence, midpoint circle algorithm was implemented.