**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**COMPUTER GRAPHICS**

**LAB ASSIGNMENT#5**

**Submitted by:**

Tej Bahadur Saru

013BSCCSIT044

**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 "cir.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

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

r=StrToInt(Edit1->Text);

xc=StrToInt(Edit2->Text);

yc=StrToInt(Edit3->Text);

x=0;

y=r;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,500);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,500,0);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(500,0,0);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(250,250,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(250,0,250);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(0,250,250);

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

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

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(0,0,500);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,500,0);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(500,0,0);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(250,250,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(250,0,250);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(0,250,250);

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

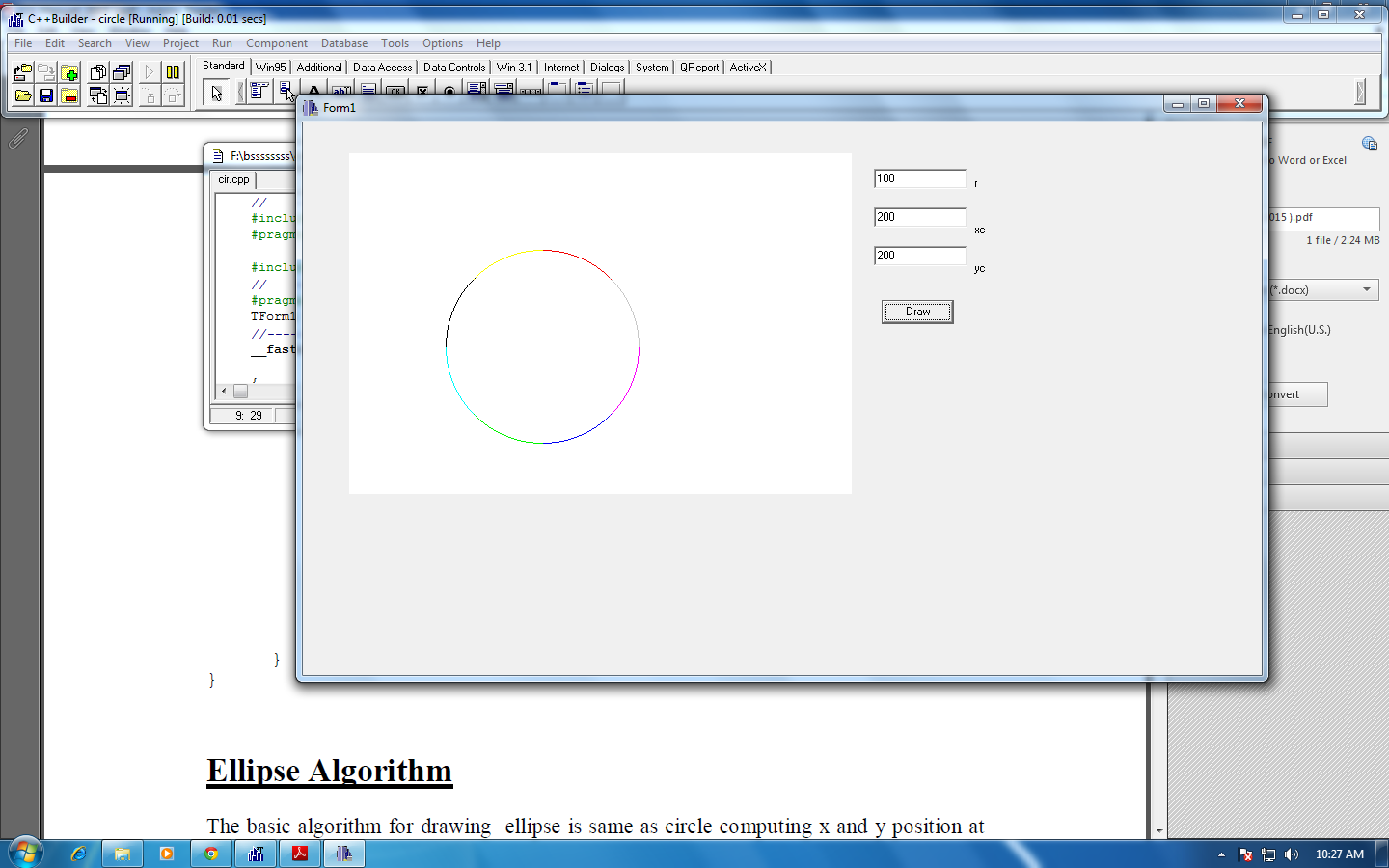
Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,0,0);

}

}

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

**OUTPUT:**

****

**CONCLUSION:**

Hence, midpoint circle algorithm was implemented.