ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

****

**Computer Graphics**

**Lab Assignment #5**

**SUBMITTED BY:**

**Siddhant Rimal**

**013BSCCSIT039**

**SUBMITTED TO**

|  |  |
| --- | --- |
| **Er. Anil Sah**  **( Lecturer )** |  |
| **Department of Computer Science** | |

Submission Date: August 21st 2015

**OBJECTIVE 4.1: TO DRAW A CIRCLE USING MID POINT CIRCLE ALGOTHM**

|  |  |
| --- | --- |
| **Algorithm:** | |
| Step 1 | Start |
| Step 2 | Input radius r and circle centre (xc , yc ), and obtain the first point on circle  centered at origin as.  (xo, yo ) = (0,r). |
| Step 3 | Calculate initial decision parameter po=5/4-r |
| Step 4 | At each xk position, starting at k=0, perform the tests:  If pk<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 the circle is (xk+1,yk-1)  pk+1=pk+2xk+1+1-2yk+1  where, 2xk+1=2xk+2 and 2yk+1=2yk-2 |
| Step 5 | Determine symmetry point on the other seven octants |
| Step 6 | Move each calculated pixels positions (x,y) in to circle path centred at (xc,yc) as  x=x+xc, y=y+yc |
| Step 7 | Repeat Steps 4 through 6 until x>=y |
| Step 8 | Stop |

|  |
| --- |
| //**SOURCE CODE**--------------------------------------------------------------------- |
| #include <vcl\vcl.h> |
| #pragma hdrstop |
| #include "circle.h" |
| //--------------------------------------------------------------------------- |
| #pragma resource "\*.dfm" |
| TForm2 \*Form2; |
| int r=1,xc=1,yc=1,x=1,y=1,k=0; |
| float pk=0; |
| //--------------------------------------------------------------------------- |
| \_\_fastcall TForm2::TForm2(TComponent\* Owner) |
| : TForm(Owner) |
| { |
| } |
| //--------------------------------------------------------------------------- |
| void \_\_fastcall TForm2::makeCircleClick(TObject \*Sender) |
| { |
| r=StrToInt(R->Text); |
| xc=StrToInt(Xc->Text); |
| yc=StrToInt(Yc->Text); |
| x=0; //STEP1 |
| y=r; |
| pk=5/4-r; //STEP2 |
| do{ //STEP4 and STEP5 |
| Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc-x][yc-y]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc+x][yc-y]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc+y][yc+x]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc+y][yc-x]=RGB(0,0,255); |
| Image1->Canvas->Pixels[xc-y][yc+x]=RGB(0,0,255); |
| if(pk<0){ |
| x=x+1; |
| pk=pk+2\*x+1; |
| } |
| else { |
| x=x+1; |
| y=y-1; |
| pk=pk+2\*x+1-2\*y+1; |
| } |
| }while(x<=y); //STEP6 |
| } |
| //--------------------------------------------------------------------------- |

**OUTPUT:**

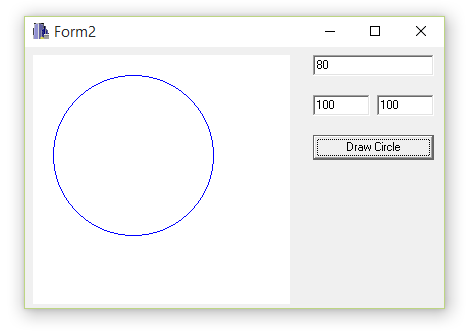


Fig: Drawing a Circle with Mid Point circle algorithm

**CONCLUSION:**

The program could plot a simple circle with the help of mid point circle algorithm. The program needs Radius(R), Xc and Yc co-ordinate values to operate.