ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Computer Graphics**

**Lab Assignment #5**

**Submitted by:**

Pankaj Niroula  
013BSCCSIT025

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil K. Sah  Lecturer Department of Computer Science |  |

Date: 21 August, 2015

**Algorithm for Midpoint Circle Algorithm**

Step 1: Input radius r and circle center (xc, yc)

Step 2: Obtain the first point on circle centered at origin as (x0, y0) = (0, r)

Step 3: Calculate initial decision parameter

P0 = 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 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 + 1

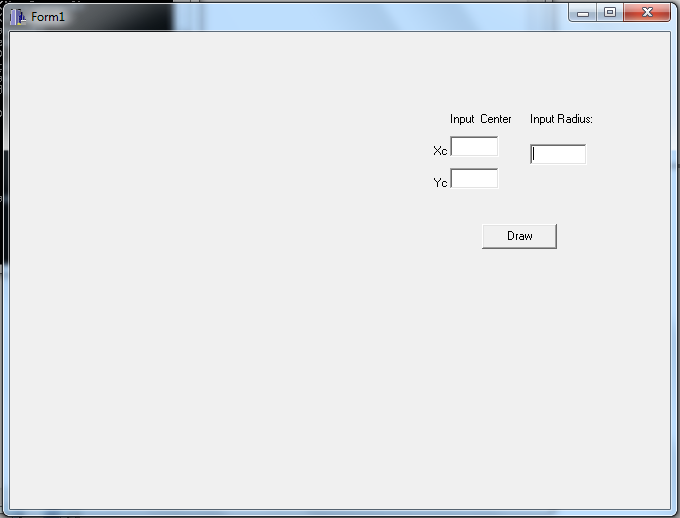
y = y + 1

Step 7: Repeat 4 through 6 until x >= y.

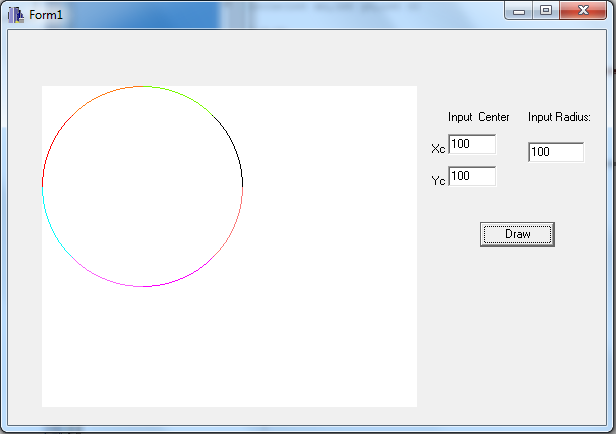
**Source Code**

|  |
| --- |
| //---------------------------------------------------------------------------  #include <vcl\vcl.h>  #pragma hdrstop  #include "ProjCir.h"  //---------------------------------------------------------------------------  #pragma resource "\*.dfm"  TForm1 \*Form1;  int r, x, y, xc, yc, p;  void draw(int x, int y, int xc, int yc);  //---------------------------------------------------------------------------  \_\_fastcall TForm1::TForm1(TComponent\* Owner)  : TForm(Owner)  {  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Edit1Change(TObject \*Sender)  {  r = StrToInt(Edit1->Text);  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Edit2Change(TObject \*Sender)  {  xc = StrToInt(Edit2->Text);  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Edit3Change(TObject \*Sender)  {  yc = StrToInt(Edit3->Text);  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::draw(int x, int y, int xc, int yc)  {  Image1->Canvas->Pixels[xc+x][yc+y] = RGB(250,0,250);  Image1->Canvas->Pixels[xc-x][yc+y] = RGB(250,100,250);  Image1->Canvas->Pixels[xc+x][yc-y] = RGB(120,250,0);  Image1->Canvas->Pixels[xc-x][yc-y] = RGB(255,122,21);  Image1->Canvas->Pixels[xc+y][yc+x] = RGB(250,122,122);  Image1->Canvas->Pixels[xc-y][yc+x] = RGB(10,250,250);  Image1->Canvas->Pixels[xc+y][yc-x] = RGB(0,0,0);  Image1->Canvas->Pixels[xc-y][yc-x] = RGB(255,0,0);  }  void \_\_fastcall TForm1::Button1Click(TObject \*Sender)  {  x = 0;  y = r;  p = 1-r;  draw(x, y, xc, yc);  while(x<y)  {  if(p<0)  {  x++;  p = p+2\*x+1;  }  else  {  x++;  y--;  p = p+2\*(x-y)+1;  }  draw(x, y, xc, yc);  }  }  //- |

**Input Screen**

****

**Output Screen**

****

**Conclusion**

Thus, the mid point circle algorithm can be used to draw circles as shown above.