**ST. XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

****

**COMPUTER GRAPHICS**

**LAB ASSIGNMENT #5**

**Submitted by:**

NARAYAN PANTHI

013BSCCSIT024

**Submitted to:**

Er. Anil Sah

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

# STATEMENT: MID-POINT CIRCLE ALGORITHM

* Input radius *r* and circle centre *(xc, yc)*, then set the coordinates for the first point on the circumference of a circle centred on the origin as:



* Calculate the initial value of the decision parameter as:



* Starting with *k = 0* at each position *xk*, perform the following test. If *pk < 0*, the next point along the circle centred on *(0, 0)* is *(xk+1, yk)* and:



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



* Determine symmetry points in the other seven octants
* Move each calculated pixel position *(x, y)* onto the circular path centred at *(xc, yc)* to plot the coordinate values:



* Repeat steps 3 to 5 until *x >= y*

# SOURCE CODE:

|  |
| --- |
| #include <vcl\vcl.h>  #pragma hdrstop  #include "Unit1.h"  //---------------------------------------------------------------------------  #pragma resource "\*.dfm"  TForm1 \*Form1;  int a,b,r;  //---------------------------------------------------------------------------  \_\_fastcall TForm1::TForm1(TComponent\* Owner)  : TForm(Owner)  {  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button1Click(TObject \*Sender)  {  a=StrToInt(Edit2->Text);  b=StrToInt(Edit3->Text);  r=StrToInt(Edit1->Text);  int p,x,y;  x=0;  y=r;  Image1->Canvas->Pixels[a+x][b+y]=RGB(255,0,0);  Image1->Canvas->Pixels[a-x][b+y]=RGB(255,0,255);  Image1->Canvas->Pixels[a+x][b-y]=RGB(0,255,255);  Image1->Canvas->Pixels[a-x][b-y]=RGB(255,255,255);  Image1->Canvas->Pixels[a+y][b+x]=RGB(255,0,0);  Image1->Canvas->Pixels[a-y][b+x]=RGB(130,130,130);  Image1->Canvas->Pixels[a+y][b-x]=RGB(103,200,0);  Image1->Canvas->Pixels[a-y][b-x]=RGB(0,200,130);  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[a+x][b+y]=RGB(0,0,255);  Image1->Canvas->Pixels[a-x][b+y]=RGB(0,255,255);  Image1->Canvas->Pixels[a+x][b-y]=RGB(255,255,255);  Image1->Canvas->Pixels[a-x][b-y]=RGB(255,255,0);  Image1->Canvas->Pixels[a+y][b+x]=RGB(255,0,0);  Image1->Canvas->Pixels[a-y][b+x]=RGB(127,127,127);  Image1->Canvas->Pixels[a+y][b-x]=RGB(127,255,0);  Image1->Canvas->Pixels[a-y][b-x]=RGB(0,255,127);  }  } |

# OUTPUT:

