**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



**Computer Graphics Assignment #5**

**Draw a circle using midpoint algorithm**

**Submitted By:**

Bikash Paneru

013BSCCSIT012

**Submitted to:**

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

**Date of Submission: 21st August, 2015**

**STATEMENT**

Write a program to draw a circle using the midpoint algorithm.

­

**ALGORITHM**

1. Take center (x0,y0) and radius r as input.
2. Set x=0 and y=radius
3. Calculate the initial decision parameter, P = 1-y
4. Do while (x<=y)
   1. Increment the value of x by 1
   2. If P<0
      * Calculate the next decision parameter as, P = P+2\*x+1
   3. else
      * Calculate the next decision parameter as, P = P+2\*(x-y)+1;
      * Decrement the value of y by 1
   4. Plot pixels at
      * (x0+x,y0+y)
      * (x0+x,y0-y)
      * (x0-x,y0+y)
      * (x0-x,y0-y)
      * (x0+y,y0+x)
      * (x0-y,y0+x)
      * (x0+y,y0-x)
      * (x0-y,y0-x)

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "circle.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

void drawPixels(int x, int y, int x0, int y0, TCanvas \* Canvas) {

//First Octane

Canvas->Pixels[x0+x][y0+y] = RGB(0,0,0);

//Second Octane

Canvas->Pixels[x0+x][y0-y] = RGB(255,30,0);

//Third Octane

Canvas->Pixels[x0-x][y0+y] = RGB(0,30,255);

//Fourth Octane

Canvas->Pixels[x0-x][y0-y] = RGB(200,200,0);

//Fifth Octane

Canvas->Pixels[x0+y][y0+x] = RGB(0,0,0);

//Sixth Octane

Canvas->Pixels[x0-y][y0+x] = RGB(100,0,100);

//Seventh Octane

Canvas->Pixels[x0+y][y0-x] = RGB(0,200,200);

//Eighth Octane

Canvas->Pixels[x0-y][y0-x] = RGB(200,0,200);

}

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

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

{

//Get the center and radius values

int x0=StrToInt(centerX->Text);

int y0=StrToInt(centerY->Text);

int radius = StrToInt(radiusInput->Text);

//x and y are the points that will be drawn

//The drawing point will start from 0,radius

int x = 0;

int y = radius;

int P=1-y; //The initial decision parameter

while(x<=y) {

++x; //X is incrememted in every step

if(P<0) {

P+=2\*x+1; //The next decision parameter

} else {

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

--y; //Plot y one coordinate time the next time

}

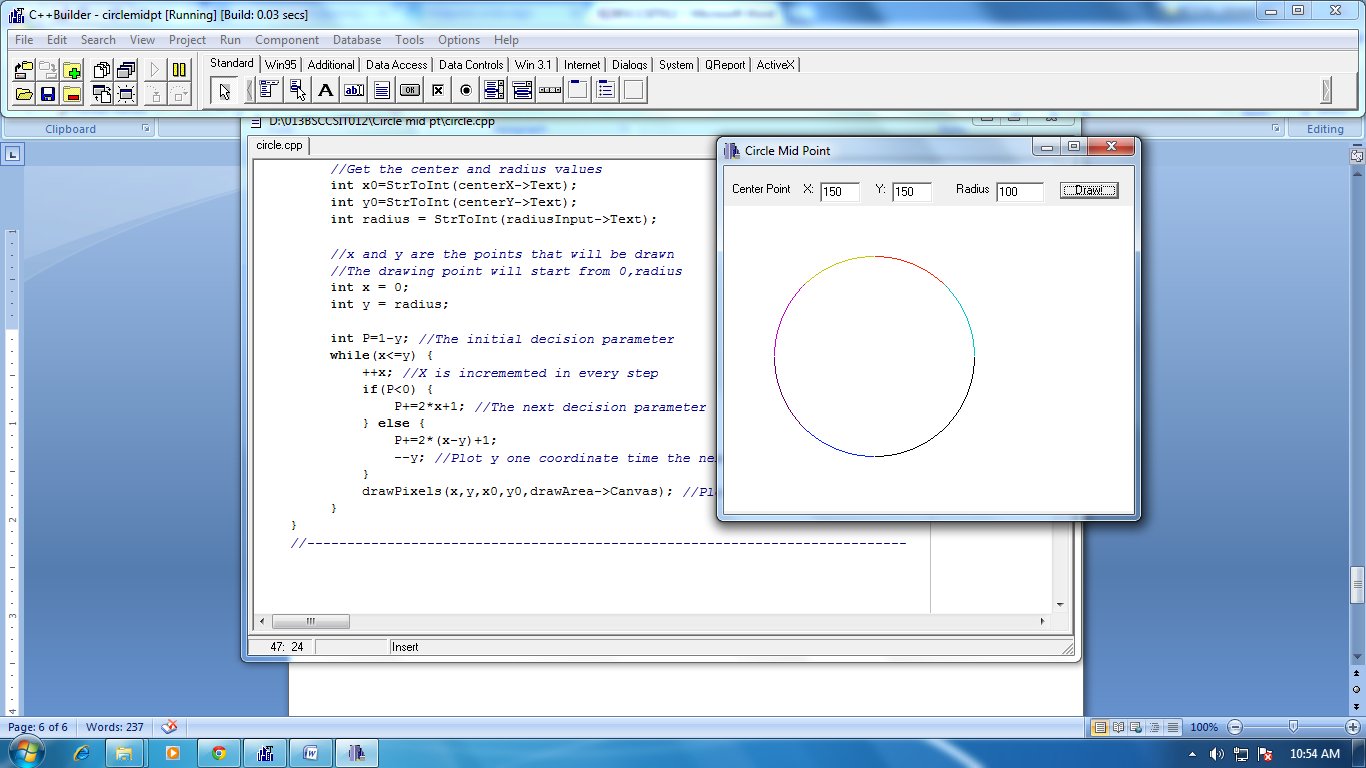
drawPixels(x,y,x0,y0,drawArea->Canvas); //Plot the Pixel

}

}

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

**OUTPUT:**

****

**CONCLUSION**

Hence, a program to draw a circle using the midpoint algorithm was implemented by using C++ with C++Builder.