**ST.XAVIER,S COLLEGE**

**Maitighar, Kathmandu**



Digital Logic Lab Assignment #3

**To implement DDA line drawing algorithm.**

Yub Raj Basnet

013BScCSIT048 (4th Semester)

**Submitted to**

|  |  |
| --- | --- |
| Er. Anil Sah  (Lecturer, St.Xavier’s College ) |  |

**Submitted date: Aug 11, 2015**

# STATEMENT

To implement the DDA line drawing algorithm:

# ALGORITHM:

* The DDA starts by calculating the smaller of dy or dx for a unit increment of the other. A line is then sampled at unit intervals in one coordinate and corresponding integer values nearest the line path are determined for the other coordinate.
* Considering a line with positive slope, if the slope is less than or equal to 1, we sample at unit x intervals (dx=1) and compute successive y values as

y_{k+1} = y_k + m

* Subscript k takes integer values starting from 0, for the 1st point and increases by 1 until endpoint is reached. y value is rounded off to nearest integer to correspond to a screen pixel.
* For lines with slope greater than 1, we reverse the role of x and y i.e. we sample at dy=1 and calculate consecutive x values as

x_{k+1} = x_k + \frac{1}{m}

* Similar calculations are carried out to determine pixel positions along a line with negative slope. Thus, if the absolute value of the slope is less than 1, we set dx=1 if  x_{start}<x_{end} i.e. the starting extreme point is at the left.

**Source Code**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "assm3.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int x1,y1,x2,y2;

void draw();

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

x1=StrToInt(Edit1->Text);

y1=StrToInt(Edit2->Text);

x2=StrToInt(Edit3->Text);

y2=StrToInt(Edit4->Text);

draw();

}

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

void draw()

{

int dx, dy, steps, m;

float incrx, incry,x,y;

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy))

steps=abs(dx);

else

steps=abs(dy);

incrx=dx/steps;

incry=dy/steps;

x=x1;

y=y1;

Image1->Canvas->Pixels[x][y]=RGB(255,0,0);

for(m=1;m<=steps;m++)

{

x=x + incrx;

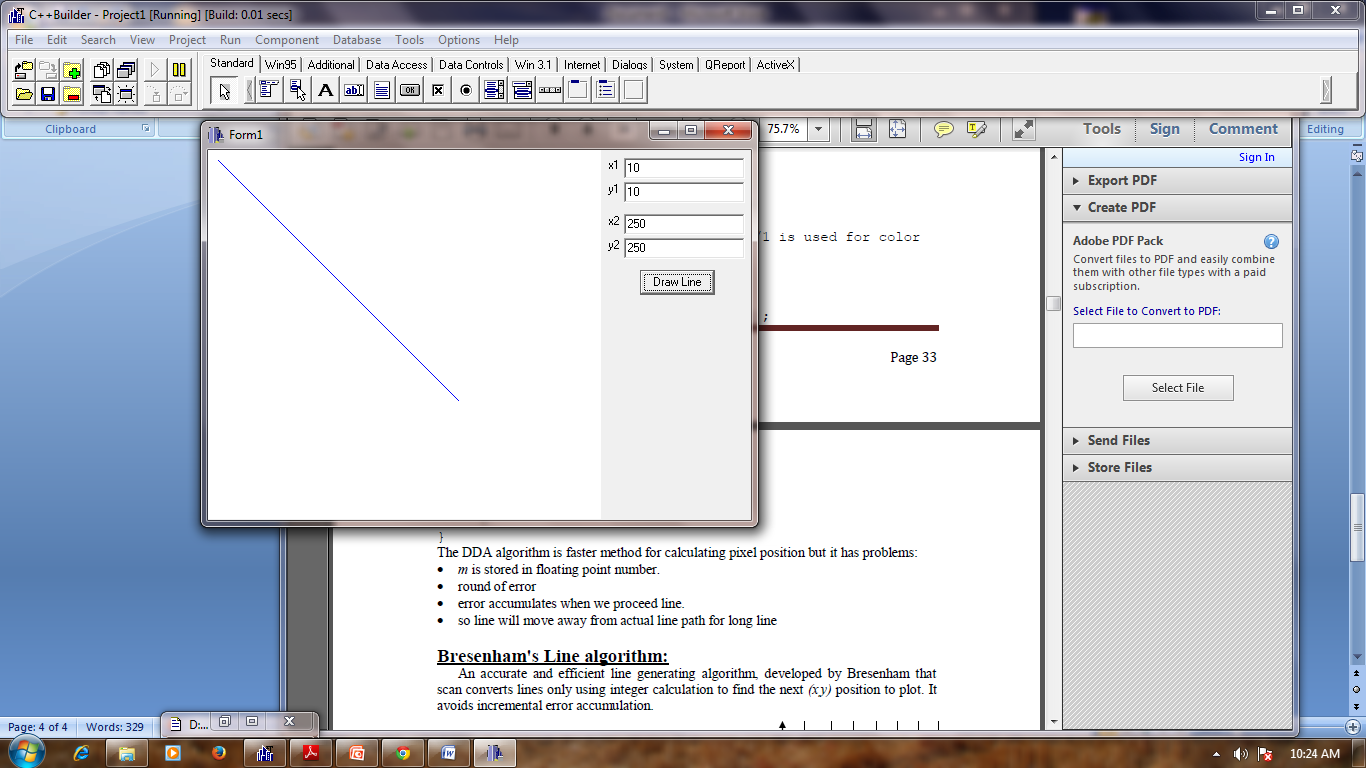
y=y + incry;

Image1->Canvas->Pixels[x][y]=RGB(255,0,0);

}

}

# Output



# Conclusion:

Hence, the DDA algorithm was implemented using C++ builder.

# Reference

https://users.soe.ucsc.edu/~pang/160/f12/slides/dda2.pdf