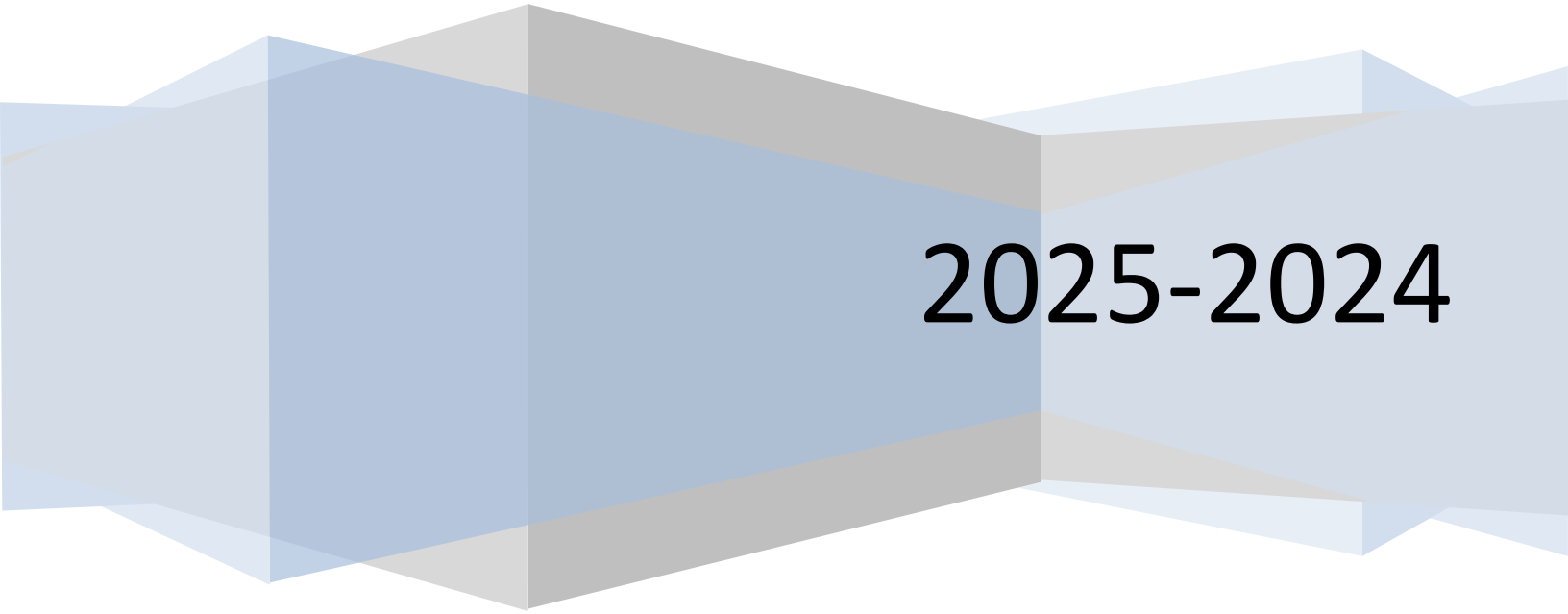
4th lvl Network management branch

Multimedia 2

Lab course 2

Assistant Lecture Zainab Ali Yakoob



2025-2024

Simple Line Drawing

Input (P1(x1,y1), P2(x2,y2))

output (Line)

Apply Simple Line Drawing Alg.

```
dx := ( x2 – x1 );
dy := ( y2 – y1 );
if dx <> 0 then
   b := dy / dx;
else
   b:= 0;
a:= y1 – x1 * b;
Procedure Switch( var  x,y : integer )
   begin
     t := x ;
     x := y ;
     y := t  ;
    end;

Begin
If  abs ( x1 – x2 ) > abs ( y1 – y2)  then
     begin
     ( * gaps between x's  is greater than y's .
          Trace horizontal  *)
     If  x1 > x2  then
        begin
           Switch ( x1 , x2 );
           Switch ( y1 , y2 );
        end;

     dx := ( x2 – x1 );
     dy := ( y2 – y1 );
     if dx <> 0 then
        b := dy / dx;
```

```
    else
       b:= 0;
    a:= y1 – x1 * b;

    for x := x1 to x2 do
       begin
          y:= round ( a + x * b);
          PutPixel (x, y, color);
       end;
    end
else
    begin
    ( * gaps between y's  is greater than x's .
       Trace vertically *)
    If  y1 > y2  then
    begin
      Switch ( y1 , y2 );
      Switch ( x1 , x2 );
    end;

    dx := ( x2 – x1 );
    dy := ( y2 – y1 );
    if dx <> 0 then
       b := dy / dx;
    else
       b:= 0;
    a:= y1 – x1 * b;
     for y := y1 to y2 do
     begin
        if  b <> 0 then
           x := round ( ( y – a ) / b);
        else
           x := 0;
           PutPixel (x, y, color);
     end;
  end;
end;
```

Digital Differential Analyzer Drawing

Input (P1(x1,y1), P2(x2,y2))

output (Line)

Apply DDA Alg.

```
Start

        If ABS( X2-X1) > ABS (Y2-Y1) Then
                                        Length=ABS (X2-X1)
                                Else
                                        Length=ABS (Y2-Y1)

dX = (X2-X1) / Length
dY = (Y2-Y1) / Length

X=X1+ 0.5 * Sign(ΔX)     ;   ΔX = X2 – X1
Y=Y1+ 0.5 * Sign(ΔY)     ;   ΔY = Y2 – Y1

For I=1 to Length
  Begin
        PutPixel ( Int(X) , Int (Y) )
        X=X+dX
        Y=Y+dY
          End

Finish

Sign function returns :  -1  if its argument is  < 0
                         :   0  if its arguments is = 0
                         :  +1 if its arguments is  > 0
```

Translation

Input (P1(x1,y1), P2(x2,y2), …….)

output (shape)

Apply Translation using DDA Alg.

```
Start
‘draw a rectangle

Call DDA(x1,y1,x2,y2)

Call DDA(x2,y2,x3,y3)

Call DDA(x3,y3,x4,y4)

Call DDA(x4,y4,x1,y1)

   ‘draw the rectangle after translation by 3 units in the X
                direction and   2  units in the Y direction


Call DDA(x1+3,y1+2,x2+3,y2+2)

Call DDA(x2+3,y2+2,x3+3,y3+2)

Call DDA(x3+3,y3+2,x4+3,y4+2)

Call DDA(x4+3,y4+2,x1+3,y1+2)

end
```

Scaling

Input (P1(x1,y1), P2(x2,y2), …….)

output (shape)

Apply Scaling using DDA Alg.

Start

‘draw a rectangle

Call DDA(x1,y1,x2,y2)

Call DDA(x2,y2,x3,y3)

Call DDA(x3,y3,x4,y4)

Call DDA(x4,y4,x1,y1)

‘draw the rectangle after Scaling with SX=2,SY=2

Call DDA(x1*2,y1*2,x2*2,y2*2)

Call DDA(x2*2,y2*2,x3*2,y3*2)

Call DDA(x3*2,y3*2,x4*2,y4*2)

Call DDA(x4*2,y4*2,x1*2,y1*2)

end

Scaling by a fixed point

Input ((12,4),(20,4),(12,8),(20,8))

output (shape before and after scaling)

Apply Scaling using DDA Alg.

```
Start
‘draw a rectangle
Call DDA(x1,y1,x2,y2)
Call DDA(x2,y2,x3,y3)
Call DDA(x3,y3,x4,y4)
Call DDA(x4,y4,x1,y1)
‘Translate the shape at the point (12,4) as the fixed point
X1=x1-12 : y1=y1-4
X2=x2-12 : y2=y2-4
X3=x3-12 : y3=y3-4
X4=x4-12 : y4=y4-4
‘Scale the rectangle after translation with SX=2,SY=2
X1=x1*2 : y1=y1*2
X2=x2*2 : y2=y2*2
X3=x3*2 : y3=y3*2
X4=x4*2 : y4=y4*2
```

‘Translate the shape back to the point (12,4)

X1=x1+12 : y1=y1+4

X2=x2+12 : y2=y2+4

X3=x3+12 : y3=y3+4

X4=x4+12 : y4=y4+4

‘draw the rectangle after translation and Scaling at the fixed point

Call DDA(x1,y1,x2,y2)

Call DDA(x2,y2,x3,y3)

Call DDA(x3,y3,x4,y4)

Call DDA(x4,y4,x1,y1)

end

Load a video

Input (Execute code to load a video)

output (display the video and it's path in a text)

Load a video program

```
Public load_vid As New OpenFileDialog
Public Video_frames As string=""

Public Sub Button1_Click(sender As System.Object, e As
System.EventArgs) Handles Button1.Click

    If load_vid.ShowDialog()=Windows.Forms.DialogResult.OK then
   Video_frames = f.FileName
   AxWindowsMediaPlayer1.URL = Video_frames
   End If
  Textbox1.text= Video_frames
 End Sub
```

Get the frames from video

Input (load a video)

output (display the frames)

get the frames program

```
Public ffmpegPath As string= "C:\v\ffmpeg.exe" ’ FFmpeg is a free and open-source software with libraires work with video and audio you must install it from the internet
Public WithEvents proc As New process
Public load_vid As New OpenFileDialog
Public Video_frames As string=””

‘this button open the vedio and start the ffmpegPath


Public Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
    proc.StartInfo.FileName = ffmpegPath

     If load_vid.ShowDialog()=Windows.Forms.DialogResult.OK then
    Video_frames = f.FileName
    AxWindowsMediaPlayer1.URL = Video_frames
    End If
   Textbox1.text= Video_frames
 End Sub
```

```
‘This button get the frames and save it in a specific path

Public Sub Button2_Click(sender As System.Object, e As System.EventArgs)
Handles Button1.Click

Dim Frame_Time As Double =
AxWindowsMediaPlayer1.Ctlcontrols.currentPosition
Dim SaveAs As String = "C:\output\F" & Frame_Time.ToString.Replace(".", "-
") &  ".jpg"
proc.StartInfo.Arguments = " -ss " & "00:05:20 -t 00:00:30" & " -i " & Chr(34) &
Video_frames & Chr(34) & " -vframes 1 -f image2 " & Chr(34) & SaveAs &
Chr(34)
TextBox2.Text = proc.StartInfo.Arguments
proc.Start()

End Sub
```