

اذ Wave no Colour
 Light يتحدد wave
 Range دا لا العين يتوقف
 Red \leftrightarrow violet
 infrared أقل أعلى ultraviolet
 و دل من ينثر ضوء

أى object يُؤدي اللumen بناه

الضوء لما يتحدد على دا لا ينثر اللumen بناه
 لو الضوء ينثر اللumen هبيقل دا لا الضوء ضعيف اللumen يخف

object reflection coefficient = object color

$\frac{\text{داللumen}}{\text{داللumen}}$ دا المترافق عليه

ضئيل ألوان Specular Color ميتاليك
 دا لا يعتمد على تردد الضوء
 defused Color

ambient Color يعرفني ألا بالمنقار الالام بالليل

الـ input يتأثر بـ object Material ، KeyPoint احذر الـ object يتأثر
 وصف الـ Action - Light - وصف الـ Light - وصف الـ Light - وصف الـ Light

كون منظم

المسوحة صوب بـ اسنان

Computer Graphics تطبيقات الـ

Games - Generated in ~~Films~~ Movies -

(ed System {computer editing design}) -

Simulation محاكيات حاسوبية for entertainment

وكان عايزه اجري حاسوب على زى الطياره من محاكمات
و هم بمحاسنها ادخل على الكمبيوتر

Metaverse - Virtual

لأن استوفى حاجات حاسوبية مع حاجات
(غير حاسوبية)

$$R \cos \theta \quad R \sin \theta$$

message

الـ New Windows

interrupt من Windows API بـ Windows

{ WinMain - WindowProcedure } كل برنامج عليه

Dispatcher

Window Procedure يتابع على

action اللى بيحصل على الشاشة

screen

dll dynamic

طريقه بداخل DLL هي هناك ديناميكي Function واحده والمسماة GetProcAddress والمعيار لها اسمها GetProcAddress

لما ادخلت اتحاد DLL function

الى المتصفح او المتصفح المتصفح Compiles

Call GetProcAddress لومتصفح وLoadLibrary

or start يليك مع DLL Linker

H INSTANCE

H → handle

INSTANCE → instance مثيل كل

كمان H instance كمان P instance

instance ينتمي الى Linked List

طبعاً له قيمة لا P دردوب Null يغير دردوب

اول صوره تعمل run لكن لو Not Null بصير في واحد يترن قبل بي

كل doc بيعمل Application يفتح

وخصوص كل فайл doc مفتوح بال Word ويبيله الفيل

طريقنا كبر و صعب و يتبع الورود عاليته لعرف من الفاعل الاسمي
Parsing also "LPSTRCmd" cmd
Long Pointer to string \leftarrow LPSTR
Long Pointer \rightarrow Short Pointer \rightarrow Variable

شكل الورقة و كثيرو صيغته اداه \Rightarrow int Csh \Rightarrow ~~محضه أي حابده كل حاجاته لها~~ قسم متغير Variable

~~APIENTRY~~ \Rightarrow Call Convention عالي الاستدعاء

هذه بسحاقياً حفظ الـ fun \rightarrow Sun \rightarrow Caller الملي يسترجع fun

الدستور يعتمد على البرنامجه وعمولاته
 Pascal يعيّن

Push x , Push y

new Push
Stack ۱۰
Data

C

Push y, Push x

"Callee" يرجع إلى return address في Pop مامات،
حيث يكتب "X, Y, Z" بين مسافر Pop،
ويمثل "Caller" الذي كتب الفائز "Callee" عليه push
Standard Call frame، دارج كل طلب، وكل Pop يرجع

Windows with one window
one button (Text) or  button وال

one window one window
one class one window

own for Procedure لـ
once Per message

unfixed int

UINT M+ المسالة
الآن

Long Parameters \Rightarrow x, y and word
word Parameter \Rightarrow keyboard or mouse
word Parameter و mouse

الآن نرسم خط لرسم معين
نعطي البداية و النهاية
و ينبع خط من نقطتين
endPoint
(x₁, y₁) and (x₂, y₂)

حيث نقطتين كيصر كـ
Approximate و ينبعوا

Keyboard على انه لو حال آخر من wordParam
mouse وال

الدول المعدلة Algo بـايل

Parameters

$x_1, y_1, x_2, y_2, \text{Color}$

Direct Method

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

دعاً يُحيى لِلّٰهِ عَزَّوَجَلَّ اعْوَادُ
وَكُوٰنٰدُ الْمُحْمَدٰ اعْوَادُ
وَكُوٰنٰدُ الْمُحْمَدٰ

قربي من الدفعات
أول من ٤٥ درهم أقل
نعطيه تعاطف الـ X مع الـ Y
لومحاته حلme

bij 5 crows

$(x_2, y_2) \in (x_1, y_1)$ ⇒ $y_2 > y_1$

الـ slope أكبر من 45 درجة



كذلك ، سنت نقطتين

والميل خطى كده هبيان

الفروق بينهم

طبعاً سنت نقطتين اختيارات
dependent x, independent y

طبعاً حاصل على 5 نقطتين سويف على اللمس
لبعضها البعض

أنا بزود لك صاروصل لـ x_2
طبعاً افترضنا x_2 أبعد فضل
و بالفعل مع y

$$y = y_1 + (x - x_1) * \left(\frac{y_2 - y_1}{x_2 - x_1} \right)$$

```
void drawLine1(HDC hdc, int x1, y1, x2, y2, color)
{
    int dx = x2 - x1, dy = y2 - y1;
    if (abs(dx) > abs(dy))
    {
        if (x1 > x2)
            swap(x1, y1, x2, y2);
```

int x = x1;

SetPixel(hdc, x, y1, C);

swap صرديع

while ($x < x_2$) {

$$\text{double } y = y_1 + (x - x_1) * \frac{y_2 - y_1}{x_2 - x_1}$$

typeCasting double

منطق

Zero (يُمْكِن) int من \rightarrow double * int في

SetPixel(hdc, x, Round(y)),);

$x++ ;$

for

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

slope < 1

$$y = y_1 + (x - x_1) * (y_2 - y_1) / (x_2 - x_1)$$

$$x = x_1 + (y - y_1) * (x_2 - x_1) / (y_2 - y_1)$$

↑

slope > 1

We need floating Point operation inside the loop

Standard Addition, multiplication, rounding

1.34215 ... Multiplication by 2 \rightarrow دلبيات وعمليات ضرب

1.34215 ... \rightarrow Mantissa علامة الملايين من المجموعة

Mantissa

Exponent \rightarrow floating Point Num

$$3524.2 = \underbrace{0.3524}_\text{Mantissa} \times 10^4 \rightarrow \text{Exponential}$$

$$1.1101 \times 2^{e_1} * 1.01101 \times 2^{e_2}$$

Shift and Add, ~~Addition~~, Normalization
 fraction \rightarrow significand \rightarrow exponent \rightarrow standard \rightarrow significand \rightarrow
 1.

Multiplication \rightarrow significand \rightarrow exponent \rightarrow significand \rightarrow
 fraction \rightarrow significand \rightarrow

Simple Digital differential analyzer (DDA)

→ remove ~~the~~ floating Point Mult. Plication

Equation of Line

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x} = m$$

$$y = y_1 + (x - x_1) \times m \quad \text{when } |\Delta x| > |\Delta y|$$

$$x = x_1 + (y - y_1) \times \frac{1}{m} \quad \text{when } |\Delta x| < |\Delta y|$$

$$\frac{1}{m} = m_i$$

از ای احباب و جدیده بدلله یا قدیمه

mul واقيمهما iterative function

Case $|\Delta x| > |\Delta y|$

- initial value of $y = y_1$
 - change of $y = y_{\text{new}} - y_{\text{old}}$

$$x+1 \in \mathbb{N}^{\text{close}}$$

~~X ice of time~~

$$\text{Change of } y = (y \text{ at } x+1) - (y \text{ at } x)$$

$$\text{Change} = (y_1 + (x+1 - x_1) * m) - (y_1 + (x - x_1) * m)$$

y_{new} y_{old}

$$\text{Change} = m$$

Before the loop: $y = y_1$, $x = x_1$

Inside the loop : $x = x + 1$
float $\rightarrow y = y + m$

When $|\Delta y| > |\Delta x|$

Before the Loop

$$x = x_1 \quad y = y_1$$

In the loop : $y = y + 1$ $x = x + m;$

float

امسواحه صوبه ب ائمه

Void LineDDA(HDC hdc, int x₁, int y₁,
int x₂, int y₂, COLORREF C)

{

$$\text{int } dx = x_2 - x_1, \quad dy = y_2 - y_1;$$

if (abs(dx) > abs(dy)) { if (x₁ > x₂) swap(x₁, y₁, x₂, y₂); }

int x = x₁;

double y = y₁;

double m = (double) dy / dx; ~~مكتوب خطأ~~

setPixel(hdc, x₁, y₁, C); ~~line 1~~

while (x < x₂)

{ x++;

y += m;

setPixel(hdc, x, Round(y), C);

} if (y₁ > y₂) swap(x₁, y₁, x₂, y₂);

} else ~~if~~ " |dx| < |dy| "

{ double x = x₁;

int y = y₁;

setPixel(hdc, x, y, C);

double m; = (double) dx / dy;

while (y < y₂)

{ y++;

x += mi;

setPixel(hdc, x, y, C);

} }

الدالة (مع مفهوم الاقات والضرب المزدوج)
Floating Point Multiplication

نحو الـ user يعطيه مركبة left button لـ user أو اخليه يد و من الممكن

Lbutton down

يعرف نقطة البداية

Lbutton up

يعرف نقطة النهاية ويرسم

switch static int x,y
جعرف سيناريو كل static int x,y
x,y
جخليها static int x,y
جخليها static int x,y

WM_LBUTTONDOWN:

x = LOWORD (lp)

y = HIWORD (lp)

break

WM_LBUTTONUP

hdc = GetDC(hWnd);

LineDDA(hdc, x, y, LOWORD(lp), HIWORD(lp),
RGB(255, 0, 0));

ReleaseDC(hWnd, hdc);

break;

Bresenhan's Algorithm

مبنیو سی ای
floating Point Operation

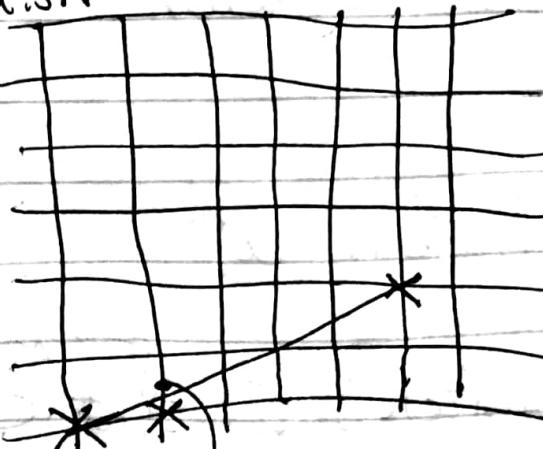
1- Start with the First Point

x_1, y_1

2- Mid Point

X direction من خطوط من

y direction من خطوط من



MidPoint
 $(x+1, y + \frac{1}{2})$

3- If midPoint لا يكفي لخط
والا فوق

يُنصح برسمي MidPoint

حسنو ال Line اقرب لل نقطه اللي فوق

والا كفت ملين دعوه ب midPoint

للو العين بوبنت كفت الخط هما هذ نقطه اللي فوق

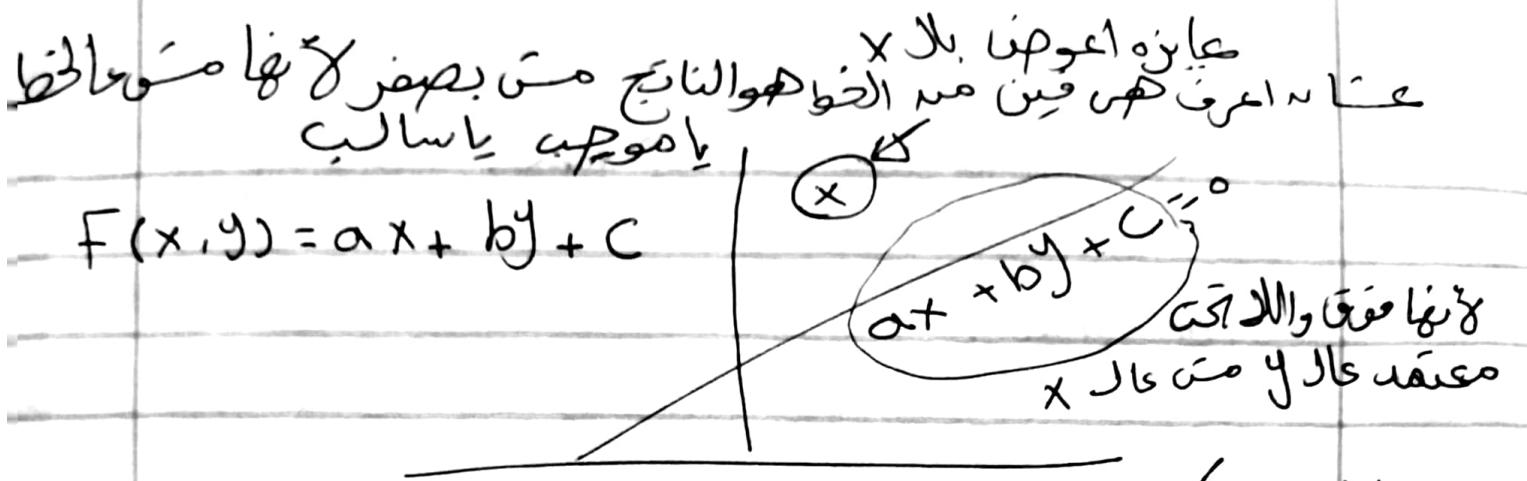
يعن ازد $x+1$ و $y+1$

للو فوق الخط هما هذ نقطه اللي تحت

$x+1$

حلب دنا هعم رف ازاي ال MidPoint دى عرق رال

تحت الخط



لو قاربنا b كان $f(x, y)$ للناتج بناءً لخوبص x \rightarrow variable Positive

Assume $b > 0$

$$f(x, y) = \begin{cases} > 0 & \text{when } (x, y) \text{ is above the line} \\ = 0 & \text{when } (x, y) \text{ on the line} \\ < 0 & \text{when } (x, y) \text{ is under the line} \end{cases}$$

يعن كل وبنا على سير

في الالجوريتم الاوامر DDA كانت لازم x او y تكون integer هنا الا دلتين

Line eqn

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$

$$\Delta x(y - y_1) - \Delta y(x - x_1) = 0$$

Line test $f(x, y) = \Delta x(y - y_1) - \Delta y(x - x_1)$
we can assume that $\Delta x > 0$

Line eq Δ

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$

$$\Delta x(y - y_1) - \Delta y(x - x_1) = 0$$

Line test f_n $f(x, y) = \Delta x(y - y_1) - \Delta y(x - x_1)$

We can assume that $\Delta x > 0$

$$d = f(x+1, y + \frac{1}{2}) = \Delta x(y + \frac{1}{2} - y_1) - \Delta y(x+1 - x_1)$$

float
 $\frac{1}{2}$ اليمين

يمكن اضرب x عاشر اضيع ال $\frac{1}{2}$ لأن أنا أعاين اعربي أو $d + \Delta x(y + 1 - y_1) - 2\Delta y(x+1 - x_1)$ فالقيمة نفسها متغيرات على المعادلة

$$d = f(x+1, y + \frac{1}{2}) = \Delta x(2y + 1 - 2y_1) - 2\Delta y(x+1 - x_1)$$

Pre Conditions

لذلك قبل ما أدخل الدخوه رقم أنا أكده إن

Slope of the line $< 1 \equiv \Delta x \geq |\Delta y|$

$\Delta x > 0$ Positive

$\Delta y > 0$ Positive

y independent x independent

نكتة المعاين

y dependent

المسوحة صوب بـ اعلى

Algorithm draft

$$x = x_1 \quad y = y_1$$

setPixel(hdc, x, y, c);

while ($x < x_2$) {

$$d = \Delta x * (2 * y + 1 - 2 * y_1) - 2 * \Delta y * (x + 1 - x_1);$$

if ($d > 0$) $x++;$

else { $x++;$ $y++;$ }

Set Pixel (hdc, x, y, c);

}

Swap if $d < 0$ لو

negative x, y تبادل لو

$y--$, $x-- \Leftarrow y++, x++$ يبقى بدل

Swap if $x_1 < x_2$ لو

(x_1, y_1) (x_2, y_2)

$\Delta x < \Delta y$ بفرض
y تغل على x

Improved algorithm

$$d = \Delta x * (2*y + 1 - 2*x_1) - 2*\Delta y * (x + 1 - x_1)$$

الآن نصل إلى خط آخر غير الخط الأصلي

use DDA approach to Compute d iteratively

Initial Value of d

$$\rightarrow x = x_1, \quad y = y_1$$

ج

$$d_{\text{initial}} = \Delta x * (2*y_1 + 1 - 2*x_1) - 2*\Delta y * (x_1 + 1 - x_1)$$

$$d_{\text{initial}} = \Delta x - 2*\Delta y$$

Change in d

if ($d > 0$) then

$$\text{change} = (d \text{ at } (x+1, y)) - (d \text{ at } (x, y))$$

$$= [\Delta x * (2*y + 1 - y_1) - 2*\Delta y * (x + 2 - x_1)] -$$

$$[\Delta x * (2*y + 1 - y_1) - 2*\Delta y * (x + 1 - x_1)]$$

$$= -2 * \Delta y$$

يمكننا تبسيط выражة المعا达 (II)Expression II

حيث يزيد بمعامل x

else

$$\text{change}_2 = [\text{dat}(x+1, y+1)] - [\text{dat}(x, y)]$$

$$= \Delta x * (2y + 3 - 2*y_1) - 2 * \Delta y * (x + 2 - x_1)$$

$$- [\Delta x * (2*y + 1 - 2*y_1) - 2 * \Delta y * (x + 1 - x_1)]$$

$$= 2 * \Delta x - 2 * \Delta y = 2 * (\Delta x - \Delta y)$$

change 1,2

y, x اثبات $\Delta x, \Delta y$ اثبات
loop \rightarrow y, x اثبات $\Delta x, \Delta y$

Pre Conditions

$$\Delta x > 0, \Delta y > 0, \Delta x \geq \Delta y$$

$$x = x_1$$

$$\underline{y} = y_+$$

$$d = \Delta x - 2 * \Delta y ;$$

~~change1 = -2 * Δy;~~

$$\text{change}_2 = 2 * (\Delta x - \Delta y);$$

SetPixel(hdc, x, y, c);

White ($x < x_2$)

~~if (d > 0) { x++; d += change; }~~

```
else { x++, y++; d+=change2; }
```

~~SetPixel(hdc,x,y,c);~~

3

DDA \Rightarrow Multiplication (ستخرج من الـ DDA بـ)

integer operation, addition.

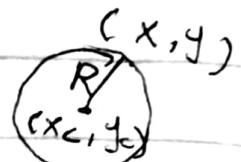
جے سی ال پرنسپنچ نکس ای ای
جي، گلہر ، پاراپلا ، کوئہ جی
جانشین

Circle drawing algorithm.

Cartesian form :

$$\sqrt{(x - x_c)^2 + (y - y_c)^2} = R$$

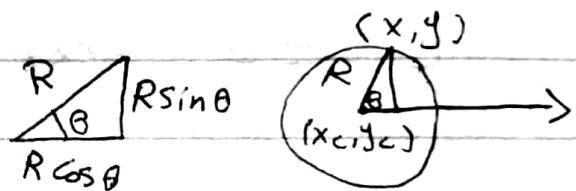
$$(x - x_c)^2 + (y - y_c)^2 = R^2$$



when $(x_c, y_c) = (0, 0)$:

$$x^2 + y^2 = R^2 \Leftarrow \text{Computation}$$

Polar form



$$x = x_c + R \cos \theta$$

$$y = y_c + R \sin \theta \quad ? \quad \theta \in [0, 2\pi]$$

\downarrow if $(x_c, y_c) = (0, 0)$

$$x = R \cos \theta$$

$$y = R \sin \theta$$

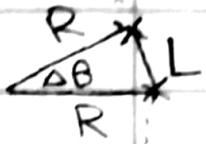
$y_c, x_c - R$ لاتصل بالدایره ایسا

لے ای دلائی x, y ایسی دایره ایسا

$360^\circ \rightarrow 0^\circ \theta$ ایسی دایره ایسا

$2\pi \rightarrow 0$

عندما اقرب التقاطعين دول سريعاً



$L \leq 1$ مدخل

الزاوية من الممتد بمسافة طول القوس والزاوية من الممتد بمسافة طول القوس

$$R * \Delta\theta < 1$$

$$\Delta\theta = \frac{L}{R}$$

$$\Delta\theta \leq \frac{1}{R}$$

$$\text{choose } \Delta\theta = \frac{1}{R}$$

أقل حاجة ممكن أحدهما هو $\frac{1}{R}$ وهو أقصى احتمالاته كون $L \leq 1$ أقل حاجة ممكن $L \leq 1$ لحل المسألة

Void CirclePolar(HDC hdc, int xc, int yc,

int R, ColorREF C)

{ double dtheta = 1.0 / R;

for (double theta=0; theta < 6.28; theta += dtheta)

Cos, Sin θ $360^\circ = 2\pi$ عرفنا 2π

Radium لكتور C++ في

double x = xc + R * Cos(theta);

double y = yc + R * Sin(theta);

SetPixel(hdc, Round(x), Round(y), C);

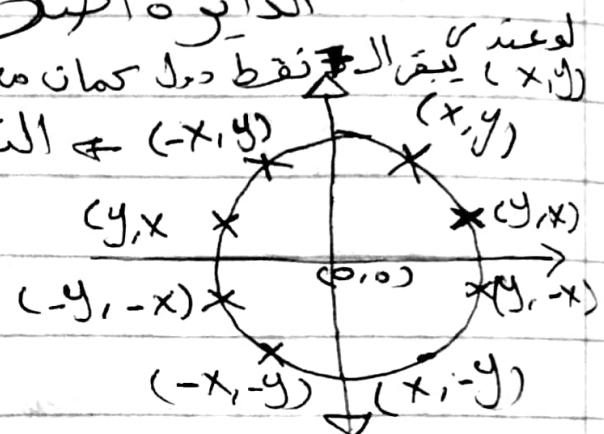
$$\cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots$$

$$\sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots$$

دي كده ممكنة كبيرة في الخوارزم
 \cos, \sin, \tan
 لأن فيه وقت طويل

طبعاً ناله ارسم الدائرة كلها
 Symmetry فيها

$$x^2 + y^2 = R^2$$



Exploiting symmetry

If we Compute (x, y) on the circle

$$x^2 + y^2 = R^2$$

$$\text{we get 8 point of } (x - x_c)^2 + (y - y_c)^2 = R^2$$

$$(x_c + x, y_c + y)$$

$$(x_c - x, y_c + y)$$

$$(x_c - x, y_c - y)$$

$$(x_c + x, y_c - y)$$

$$(x_c + y, y_c + x)$$

$$(x_c - y, y_c + x)$$

$$(x_c - y, y_c - x)$$

$$(x_c + y, y_c - x)$$

دول كده نقط

لو جب نقطه اخوه

جيب عصاها

7 نقط من غير حاجة

Void Draw8Points (HDC hdc, int xc, int yc,
int x, int y, COLORREF c)

{
setPixel(hdc, xc+x, yc+y, c);
setPixel(hdc, xc-x, yc+y, c);
setPixel

لـ ٦٢ جـ ٤ جـ ٣ جـ ٢ جـ ١

دـ ٥ جـ ٤ جـ ٣ جـ ٢ جـ ١

Void CirclePolar(HDC hdc, int xc, int yc,
{ double int R, COLORREF c)

double theta = 1.0/R; theta = 0

double x = R, y = 0;

Draw8Points(hdc, xc, yc, R, 0, c);

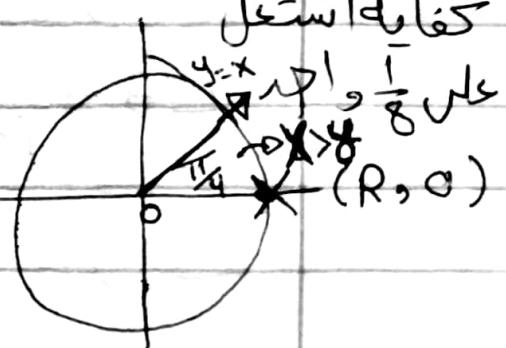
while (x > y) {

theta += dtheta;

x = R * Cos(theta);

y = R * Sin(theta);

Draw8Points(hdc, x, y, Round(x), Round(y), c), } }



Loop لـ ٦٢ جـ ٤ جـ ٣ جـ ٢ جـ ١
iteration فـ ٣ جـ ٢ جـ ١ جـ ٠ جـ ١
الـ ٣ جـ ٢ جـ ١ جـ ٠ جـ ١ جـ ٠ جـ ١

Iterative Polar algorithm

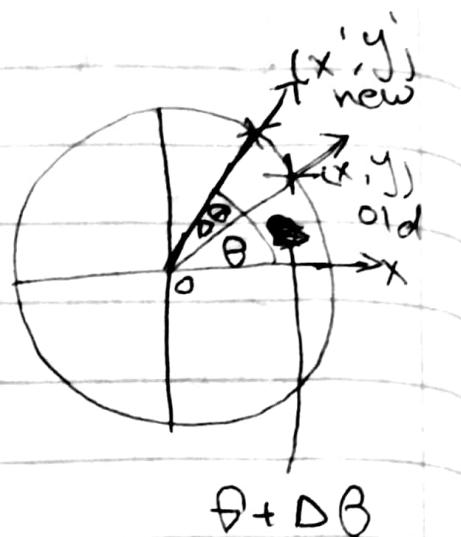
$$x = R \cos \theta$$

$$y = R \sin \theta$$

$$x' = R \cos(\theta + \Delta\theta)$$

$\cos \rightarrow$ جمع زاويتين

$$\cos \cos - \sin \sin$$



$$x' = R \cos \theta \cos \Delta\theta - R \sin \theta \sin \Delta\theta$$

$$x' = x \cos \theta - y \sin \Delta\theta$$

loop لـ loop و ممكن كـ $\Delta\theta$

$$\Delta\theta = \frac{1}{R} \quad \text{const}$$

لكن الـ $\Delta\theta$ من θ إلى $\theta + \Delta\theta$

$$y' = R \sin(\theta + \Delta\theta)$$

$\sin \rightarrow$ جمع زاويتين Sin

$$\cos \sin + \sin \cos$$

$$= R \cos \theta \sin \Delta\theta + R \sin \theta \cos \Delta\theta$$

$$y' = x \sin \Delta\theta + y \cos \Delta\theta$$

Void CircleIt(HDC hdc, int xc, int yc,
int R, COLORREF C)

{ double dtheta = 1.0/R;
double ct = cos(dtheta), st = sin(dtheta);
double x = R, y = 0; } = y ~ Curve 4 Point

Draw8Points(hdc, xc, yc, R, 0, C)

while (x > y)

{ double x1 = x * ct - y * st;

y = x * st + y * ct;

x = x1;

Draw8Points(hdc, xc, yc, Round(x), Round(y)) } } } }

Cartesian algorithm

$$x^2 + y^2 = R^2$$

X independent variable

Y dependent,

well,

$$y = \sqrt{R^2 - x^2}$$

$$x = \pm \sqrt{R^2 - y^2}$$

(X is independent) (Y is independent)

x++;

y++

$$y = \sqrt{R^2 - x^2}$$

$$x = \sqrt{R^2 - y^2}$$

Cartesian eqn

Circle

$$x^2 + y^2 = R^2$$

$$y = \pm \sqrt{R^2 - x^2}$$

$$x = \pm \sqrt{R^2 - y^2}$$

x is independent

y is independent

if $|Slope| \leq 1$

if $|Slope| > 1$

$$\text{Slope} = \frac{dy}{dx}$$

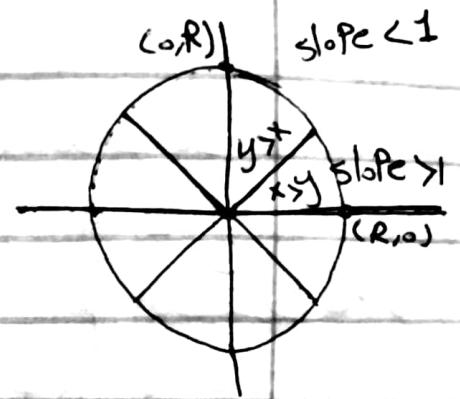
$$x^2 + y^2 = R^2$$

Take derivative

$$2x + 2y \frac{dy}{dx} = 0 \quad \therefore \frac{dy}{dx} = -\frac{x}{y}$$

$$\left| \frac{dy}{dx} \right| < 1 \text{ when } \frac{|x|}{|y|} < 1$$

if $|x| < |y|$,
then $\text{Slope} < 1$



void Circle_Cartesian(HDC hdc,
int xc, int yc, int R,
COLORREF c) {

int x=0;

double y=R;

Draw8Points(hdc, xc, yc, 0, R, c);

while (x < y)

{ x++; (double)

 y = sqrt(R * R - x * x);

 Draw8Points(hdc, xc, yc, x, Round(y), c);

و، int كلية، float على الاجزاء، مثل

Bresenhan's Circle algorithm.

Circle eqn: $x^2 + y^2 - R^2 = 0$

Circle test function:

$$f(x,y) = x^2 + y^2 - R^2$$

$$f(x,y) = \begin{cases} < 0 & (x,y) \text{ is inside the circle} \\ = 0 & (x,y) \text{ is on the circle} \\ > 0 & (x,y) \text{ is outside the circle} \end{cases}$$

first step $x=0$, $y=\mathbb{R}$

Second

Draft Version

Void Draft Circle (HDC hdc, int xc, int yc,
int R, ColorREF C)

$$\{ \text{int } x = 0, y = R; \}$$

Draw8Points(hdc, xc, yc, x, y, c);

while ($x < y$) {

$$\text{midPoint} \leftarrow d = (x+1)^2 + (y - \frac{1}{2})^2 - R^2;$$

else if ($d \leq 0$) $x++;$

فوق الخط & else { x++; y-- } } و

Draw8Points(hdc, xc, yc, x, y, c) } }

$$y^2 - y + \frac{1}{4}$$

الخطوة الثانية (second step): معايرة الدرجة الثانية

Improve Version (1) Based first order difference

$$d = (x+1)^2 + (y - \frac{1}{2})^2 - R^2$$

Initial value of d (at $x=0, y=R$)

$$\begin{aligned} d_{\text{initial}} &= 1 + (R - \frac{1}{2})^2 - R^2 \\ &= 1 + R^2 - R + \frac{1}{4} - R^2 \\ &= \frac{5}{4} - R \approx 1 - R \end{aligned}$$

الربع من الميزة
معيار (السارة)
من الممكن
السارة من الميزة

Change in d :

if ($d < 0$) then

$$\text{change} = (d \text{ at } x+1, y) - (d \text{ at } x)$$

\leftarrow الجديدة من التغير

$$= [(x+2)^2 + (y - \frac{1}{2})^2 - R^2] - [(x+1)^2 + (y - \frac{1}{2})^2 - R^2]$$

\leftarrow d_{new} \leftarrow d_{old}

اتيق فرق بين مربعي a و b كأن يقول
 $(a-b)(a+b)$ والنتائج

$$= (1)(2x+3) = 2x+3$$

else

$$\text{change} = (d \text{ at } (x+1, y-1)) - (d \text{ at } (x, y))$$

$$\begin{aligned}
 &= \left[(x+2)^2 + \left(y - \frac{3}{2}\right)^2 - R^2 \right] - \left[(x+1)^2 + \left(y - \frac{1}{2}\right)^2 - R^2 \right] \\
 &= 2x + 3 + (-1)(2y - 2) \\
 &= 2x - 2y - 5 = 2(x - y) + 5
 \end{aligned}$$

loop ال حبل هو

Void Circle Bresl(HDC hdc, int xc, int R,
ColorREF c){

int x=0, y=R, d=1-R;

Draw8Points(hdc, xc, yc, x, y, c);

while (x < y) {

if (d < 0)

{ ~~d = d + 2x + 3;~~
~~x++;~~

d_1

else {
~~d = 2*(x - y) + 5;~~
~~x++; y--;~~

d_2

Draw8Points(hdc, xc, yc, x, y, c);

ای جیسے ای مولپل سینڈ

DDA پر جو دو d_2 , d_1 اپنے

iterative d_2, d_1 comes

Improvement 2

Compute $d_1 = 2x + 3$ and

$d_2 = 2(x-y) + 5$ iterative

initial Values ($x=0, y=R$)

$$d_1 \text{ initial} = 3$$

$$d_2 \text{ initial} = 5 - 2R$$

Changes :

if ($d < 0$ then)

$$\text{Change in } d_1 = [2(x+1) + 3] - [2x + 3] = 2$$

$$\text{Change in } d_2 = [2(x+1-y) + 5] - [2(x-y) + 5] \\ = 2$$

else

$$\text{Change in } d_1 = [2(x+1) + 3] - [2x + 3] = 2$$

$$\text{Change in } d_2 = [2(x+1-(y-1)) + 5] - [2(x-y) + 5] = 4$$

end if

Void CircleBres2(HDC hdc,int xc,int yc,int R,
ColorREF c){

int x=0,y=R,d3=1-R,d1=3,d2=5-2R,-

Draw8Points(hdc,xc,yc,x,y,c);

while(x < y)

{ if (d < 0) {

d+=d1;

d1+=2;

x++; } }

else { d+=d2;

d2+=4; x++; y--;

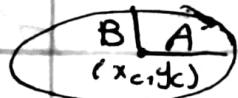
Draw8Points(hdc,xc,yc,x,y,c); }}

Ellipse Algorithm

نحوی بخراصی

equation of Ellipse in Cartesian Space

$$\frac{(x - x_c)^2}{A^2} + \frac{(y - y_c)^2}{B^2} = 1$$



Polar :

$$x = x_c + A \cos \theta$$

$$y = y_c + B \sin \theta$$

$$\theta \in [0, 2\pi]$$

Exploit (4-Similarity)

utility:

Draw 4 Points

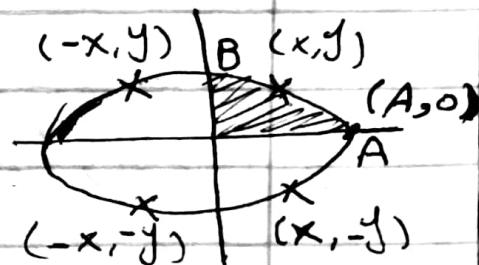
in first Quarter

Polar algorithm:

$$x = A \cos \theta$$

$$y = B \sin \theta$$

$$\theta \Rightarrow \text{range from } 0 \text{ to } \frac{\pi}{2}$$



$$\text{increment} \Rightarrow \frac{1}{\max(A, B)}$$

لـ Δx

على مسافة المقدمة الى
النقطة التي

تحسبها هي تقربي

عن يعمد لها فاصح القيمة على مسافة المقدمة تقربي

والشكل يبقى أدق

المسموحة صواب بـ اعتماد

while $\theta < 90$, $y < B$, $x > 0$

$$x = A \quad y = 0 \quad \theta = 0$$

Draw 4 Points

$$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1$$

$$x = \pm$$

Polar \Rightarrow محمد على المحadle الصغرية

الربح هي معلم على مستوى مساحة
يتأثر على الـ Modules أو المدار ؟ كبير

\Rightarrow Parametric Curves and Splines

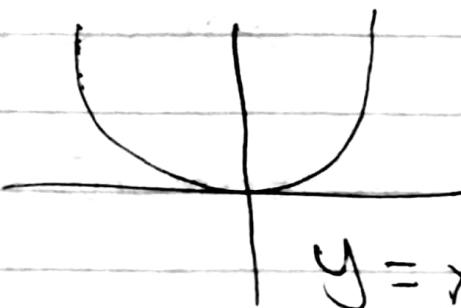
يمكن انشئ الـ Curve لخطوط بـ داخل
مساحة لكن لوالد Lines صيغة كعوب

محمد على الـ Line

$$x = a_2 t^2 + b_1 t + a_0$$



$$y = b_2 t^2 + b_1 t + b_0$$



$$y = x^2$$

$$t \in [0, 1]$$

الى Curve من

المعادلة التالية ممكن اقصى حاجة يحمل



Curve Control Points

Req: to draw
the Curve Passing
through

These Points

عندما عارفه ببعض الـ Control Points
فيمكننا رسم

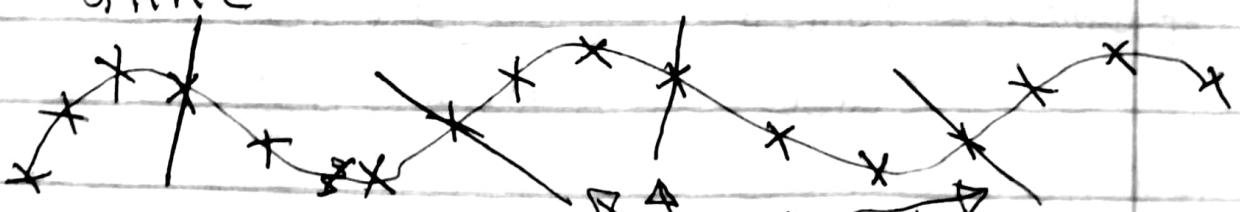
عدد الـ Points = درجة الـ Curve

$1 - \text{Points}$ = درجة الـ Curve

كل ما الا درجة المقادير يمكن
التحكم فيها من خلال user على

можем ان نقسم الـ Curve الى اجزاء
واحدها درجة من درجة 3
وهذا يعني كل 4 نقاط

Spline



درازيم الوصلات من تبقى

المسوحة صوب بـ B-Splines

Parametric Cubic Curves

$$x(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$y(t) = b_3 t^3 + b_2 t^2 + b_1 t + b_0$$

لـ t ممـن 3d لـ x و y

لـ curve الـ slope = slope of curve الـ second derivative

~~smooth~~ يـ \Rightarrow smooth يـ \Rightarrow second derivative يـ \Rightarrow slope يـ \Rightarrow second derivative يـ \Rightarrow smooth

$$x' = 3a_3 t^2 + 2a_2 t + a_1$$

$$y' = 3b_3 t^2 + 2b_2 t + b_1$$

$$P(t) = (x(t), y(t)) \quad t \in [0, 1]$$

~~$S(t) = (x'(t), y'(t))$~~

~~slope~~ \Rightarrow vector يـ \Rightarrow slope يـ \Rightarrow x و y

4 Control Points \Rightarrow user يـ \Rightarrow بين \Rightarrow curve على \Rightarrow ارساله الى

Possible control information:

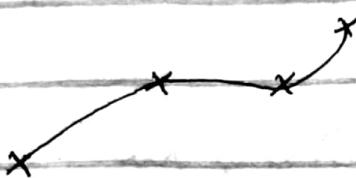
$$\textcircled{1} \quad x(0), x\left(\frac{1}{3}\right), x\left(\frac{2}{3}\right), x(1) .$$

$$y(0), y\left(\frac{1}{3}\right), y\left(\frac{2}{3}\right), y(1) .$$

Lagrange

يسخدم الـ 4 Control Point

يمكننا ~~الرسم~~ Curve



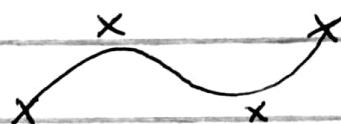
الـ Lagrange 2 Curves ~~لـ 4 Points~~ في الـ 4 Points ينبع من بينهم

(Bezier Method)

~~الـ 4 Points~~ بين عالي التأثير ~~ومنخفض~~ Curve

Curve ~~ومنخفض~~ Control

بعضهم ملحوظ Point



وأيضاً ضمن الـ 4 Points

(Hermite Method)

سيغير واحد من Control Points والباقي بـ

\textcircled{2}

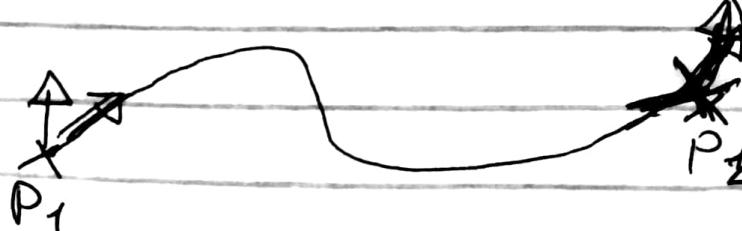
$$\begin{aligned} x(0), x'(0), x(1), x'(1) &\rightarrow P(0), S(0) \\ y(0), y'(0), y(1), y'(1) &\rightarrow P(1), S(1) \end{aligned}$$

أول نقطه

وآخر نقطه

وأدين لـ Congent

بتاعها



المسوحة صوب بـ انتقال

for ($t = 0$ $t \leq 1$ $t + 0, \dots$)

{

$x =$

$y =$

Set Pixel with rounding

}

B و A ال loop مجبى من بوج

Coefficient احسب ال Control Point احسب

Manual دود a , b

و جر ين ال loop للغ فرق دى

interpolation طرق

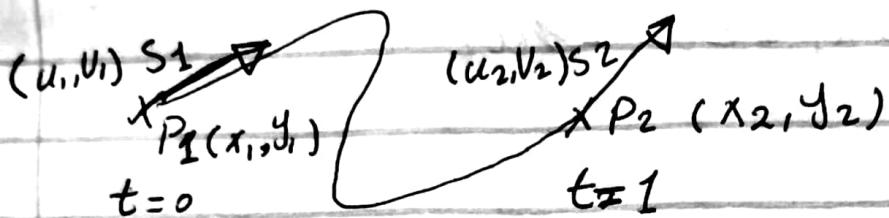
Hermite Curve (Cubic)

$$x(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$y(t) = b_3 t^3 + b_2 t^2 + b_1 t + b_0$$

$$x'(t) = 3a_3 t^2 + 2a_2 t + a_1$$

$$y'(t) = 3b_3 t^2 + 2b_2 t + b_1$$



Constraints

$$x(0) = x_1 \implies a_0 = x_1 \quad (1)$$

$$\text{slope } x'(0) = u_1 \implies a_1 = u_1 \quad (2)$$

$$x(1) = x_2 \implies a_3 + a_2 + a_1 + a_0 = x_2 \quad (3)$$

$$x'(1) = u_2 \implies 3a_3 + 2a_2 + a_1 = u_2 \quad (4)$$

given us $u_1, u_2, x_1, x_2, a_1, a_2, a_3, a_4$ \rightarrow المعرفة بالتحريك في

(3) becomes $a_3 + a_2 = x_2 - x_1 - u_1$ \rightarrow جملة (3)

(4) becomes $3a_3 + 2a_2 = u_2 - u_1$ \rightarrow جملة (4)

$$\begin{aligned} \alpha_3 + \alpha_2 &= x_2 - x_1 - u_1 \quad (3) \quad * -3 \\ 3\alpha_3 + 2\alpha_2 &= u_2 - u_1 \quad (4) \quad \leftarrow + 8 \end{aligned}$$

$$(3)' * -3 + (4)': -\alpha_2 = -3x_2 + 3x_1 + 2u_1 + u_2$$

$$\alpha_2 = 3x_2 - 3x_1 - 2u_1 - 2u_2$$

Subs in (3')

$$\begin{aligned} \alpha_3 &= x_2 - x_1 - u_1 - \underbrace{3x_2 + 3x_1 + 2u_1 + u_2}_{\alpha_2} \\ &= -2x_2 + 2x_1 + u_1 + u_2 \end{aligned}$$

$$\begin{bmatrix} \alpha_3 \\ \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ u_1 \\ x_2 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ v_1 \\ y_2 \\ v_2 \end{bmatrix}$$

$$\begin{bmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 & Y_1 \\ U_1 & V_1 \\ X_2 & Y_2 \\ U_2 & V_2 \end{bmatrix}$$

↓ ↓ ↓

C H G

Geometry Matrix

الكل G للuser
وال H لجهاز

محتاج ارسم ال Curve

اذا اخلي dt متحركة بين تطبيق
one curve متحركة

Matrix \Rightarrow arrays , Pointer \Rightarrow Pointer

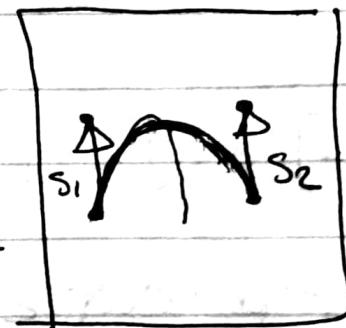
ممكن سكتها

double H[4][4] = { { 2, 1, -2, 1 },
{ -3, -2, 3, -1 },
{ 0, 1, 0, 0 },
{ 1, 0, 0, 0 } }

البطء او for loop تتابع اول t الى t من 0 لـ 1

ال Line DDA على اساع ال DDA
حسب القيمة الفعلية (الحالية) بخلاف (القيمة المطلوبة)
قبلها

تحصل Change ل معادله من الدرجة الرابعة
Change من الدرجة الثالثة كمان
تحصل من الدرجة الثانية كمان
و لكنها



عندما user يضغط على
ع نفسه مطروح الاولي منه سبعين
يدخل الى Slope

Animation

حول job والآن الى Computer
ويجدون امسح المثلث وغیر ال Position
و لكنها صيغة animation

امسح ال Job اذ اي ياتي لما اللون يلون
الClear او اعمل back ground
كلها

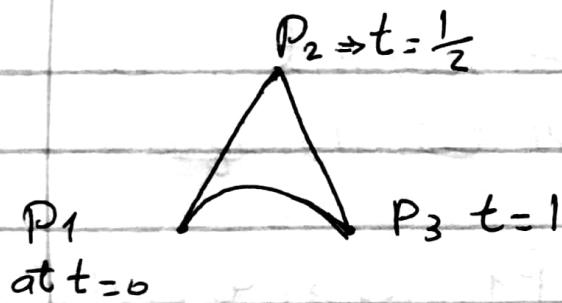
Thread

الغايات من بذاتها في
create Thread() function

x', y' هو الميل slope الـ
و دا هو ميل المترابع لـ curve
Graph rule

Cubic Bézier Curves :

Constraints

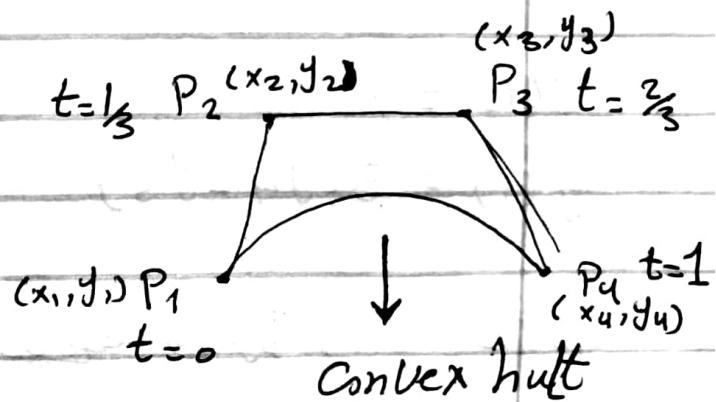


Bézier

صيغة التاريـة

لـ 3 points

Quadratic



هي الرجـة الـ 4
على 4 points

Cubic

Constraints:

$$x(0) = x_1$$

$$x'(0) = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{\frac{1}{3} - 0} = 3(x_2 - x_1)$$

$$X(0) = X_4$$

$$X'(0) = \frac{\Delta x}{\Delta t} = \frac{X_4 - X_3}{1 - 3} = 3(X_4 - X_3)$$

$$y(0) = y_1$$

$$y'(0) = 3(y_2 - y_1)$$

$$y(1) = y_4$$

$$y'(1) = 3(y_4 - y_3)$$

slope على النقاط الأولى والثانية والثالثة والرابعة
بتبعها والنقطة الثالثة والرابعة

P_4 الثالثة، P_1 الرابعة

slope 1 $\Rightarrow X'(0)$ slope 2 $\Rightarrow X'(1)$

To get the Bresler Coefficient

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$

$$X(0) = X_1 \Rightarrow a_0 = X_1$$

$$X'(0) = 3(X_2 - X_1) \Rightarrow a_1 = 3(X_2 - X_1)$$

$$X'(t) = 3a_3t^2 + 2a_2t + a_1$$

$$X(1) = X_4 \Rightarrow a_3 + a_2 + a_1 + a_0 = X_4$$

$$X'(1) = 3(X_4 - X_3) \Rightarrow 3a_3 + 2a_2 + a_1 = 3(X_4 - X_3)$$

$$\xrightarrow{* - 2} a_3 + a_2 = X_4 - X_1 - 3X_2 + 3X_1$$

$$= X_4 + 2X_1 - 3X_2$$

$$3a_3 + 2a_2 = 3x_4 - 3x_3 - 3x_2 + 3x_1 \approx$$

-2 و ۱۹۲۷ از پل می خواهد

$$a_3 = x_4 - x_1 + 3x_2 - 3x_3$$

$$a_2 = 3x_1 - 6x_2 + 3x_3$$

$$\begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

shift \div
3C₂

$$\begin{aligned} 1 &= ((1-t) + t) \\ &= (1-t)^3 + \binom{3}{1} (1-t)^2 t + \binom{3}{2} (1-t) t^2 \\ &\quad + t^3 \\ &= (1-3t^2 + 3t - t^3) + (3t-6t^2 + 3t^3) \\ &\quad + (3t^2 - 3t^3) + (t^3) \end{aligned}$$