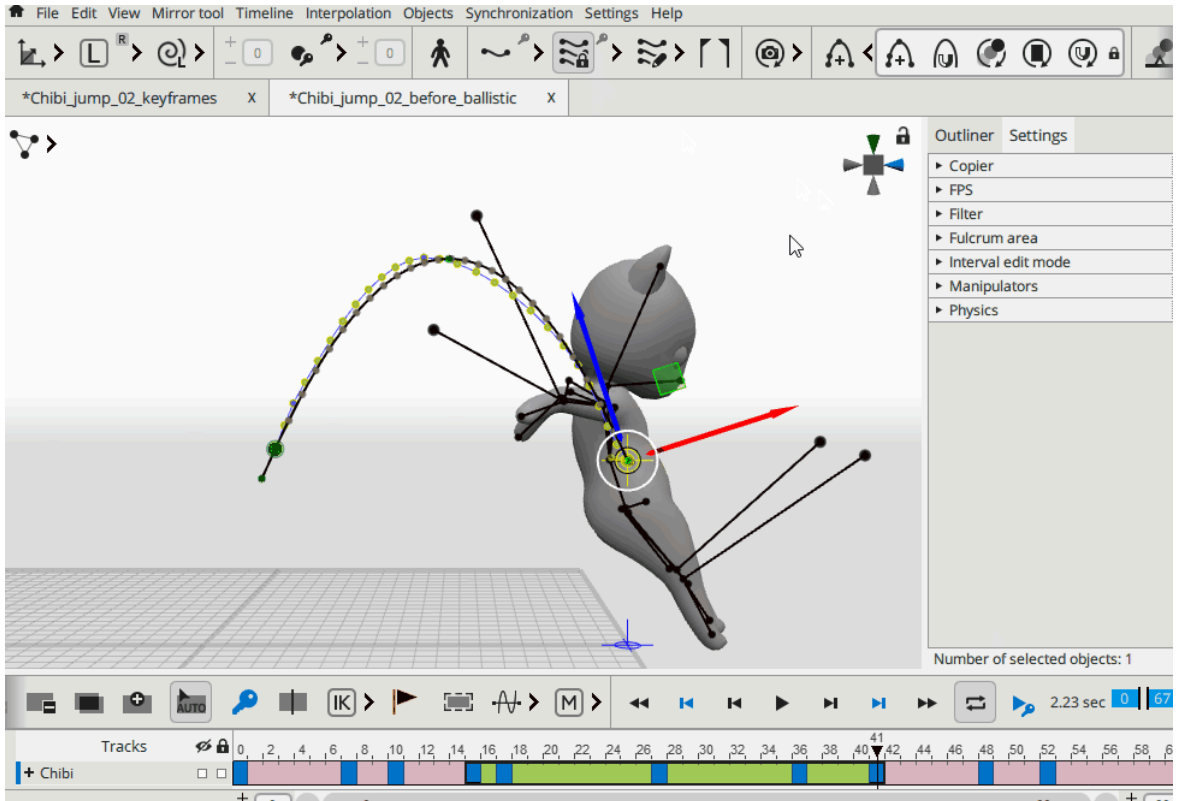
CC3501: Computación Gráfica

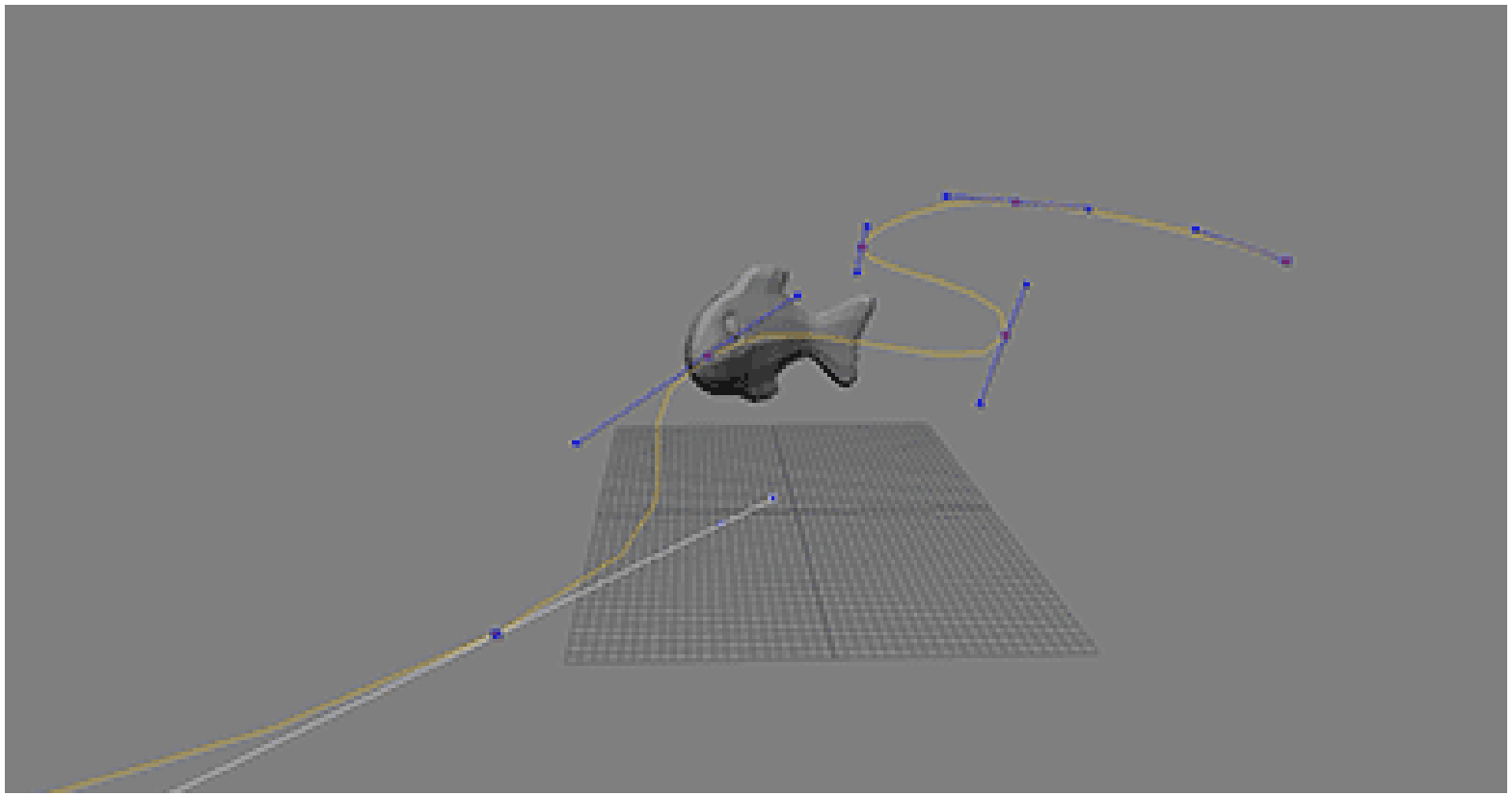
Auxiliar 13: Curvas

Auxiliares: Julieta Coloma y Catalina Gajardo

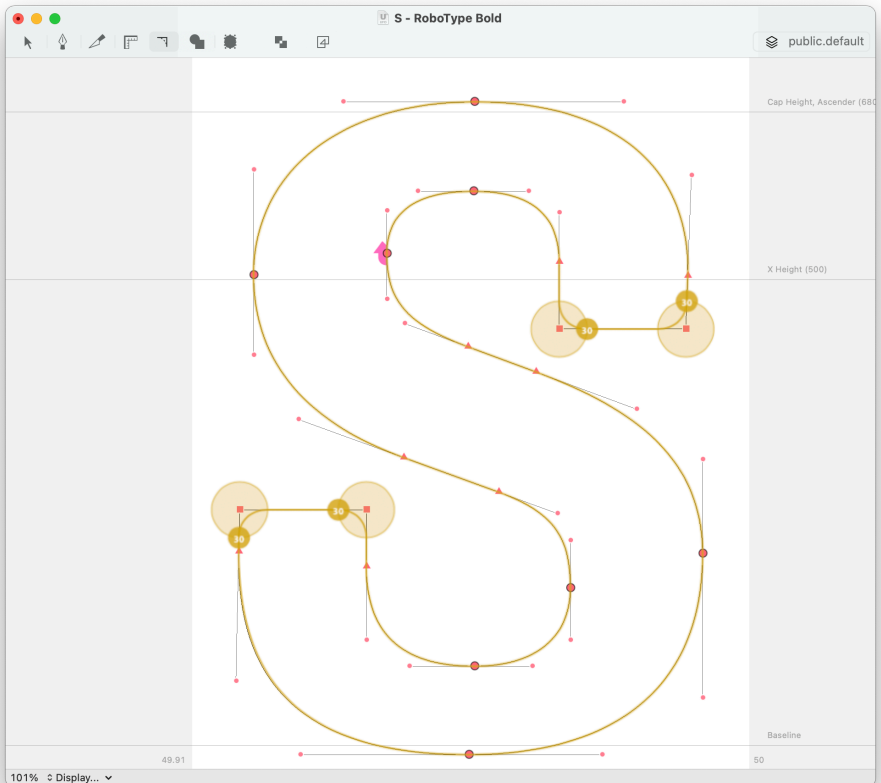

Editing Harbinger Hybrid - Steel Track
Done
TEST RATINGS
TOOLS
Live data
Results
Heatmaps
Excitement
Fear
Nausea
Speed
Lateral g-force
Vertical g-force
Forward g-force
Station
Utilities
Loops
Rolls
Non-inverted
TOOLS
Autocomplete
Angle Snap
Auto Tunnel
Height Markers
Track Supports
Camera
Park Management
My Blueprints
Toolkit Items
Coasters
Track Rides
Rides
Shops and Facilities
Scenery
Building
Paths
Terrain
Review Park


File Edit View Mirror tool Timeline Interpolation Objects Synchronization Settings Help
*Chibi_jump_02_keyframes
*Chibi_jump_02_before_ballistic
Outliner
Settings
Copier
FPS
Filter
Fulcrum area
Interval edit mode
Manipulators
Physics
Number of selected objects: 1
2.23 sec
Tracks
Chibi

[Gif]



Gif



In mathematics, a **spline** is a function defined piecewise by polynomials. In interpolating problems, spline interpolation is often preferred to polynomial interpolation because it yields similar results, even when using low degree polynomials, while avoiding Runge's phenomenon for higher degrees.

In the computer science subfields of computer-aided design and computer graphics, the term *spline* more frequently refers to a piecewise polynomial (parametric) curve. Splines are popular curves in these subfields because of the simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design.

The term spline comes from the flexible spline devices used by shipbuilders and draftsmen to draw smooth shapes.

Introduction [edit source]

The term "spline" is used to refer to a wide class of functions that are used in a and/or smoothing. The data may be either one-dimensional or multi-dimensio are normally determined as the minimizers of suitable measures of roughnes curvature) subject to the interpolation constraints. Smoothing splines may be interpolation splines where the functions are determined to minimize a weigh approximation error over observed data and the roughness measure. For a nu roughness measure, the spline functions are found to be finite dimensional in their utility in computations and representation. For the rest of this section, w polynomial splines and use the term "spline" in this restricted sense.

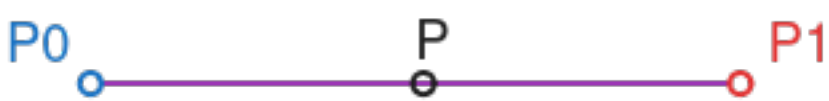
Curvas de Bézier

Bézier Curves

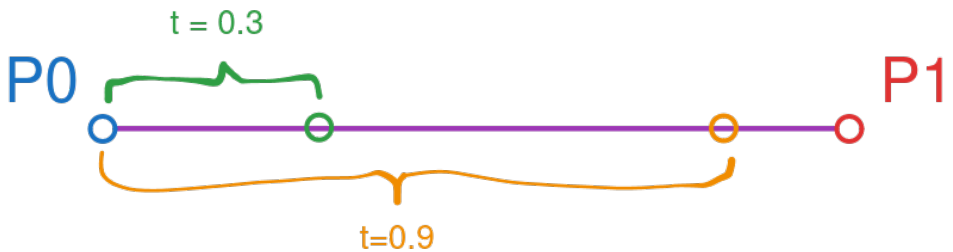
Lerp

Digamos que queremos conectar dos puntos P0 y P1

Podemos usar **lerp** (Linear Interpolation) para obtener todos los puntos de en medio y conectar P_0 con P_1



Donde el punto **P** está dado por $P = (1 - t)P_0 + P_1 t$

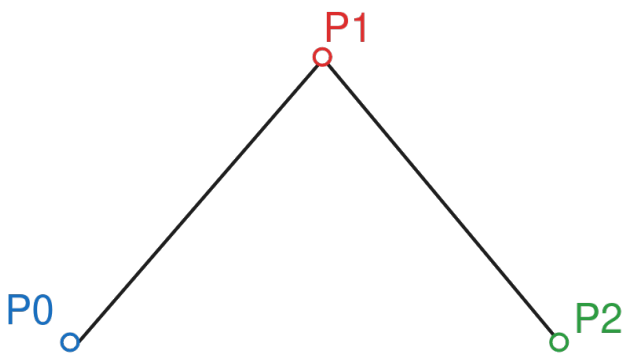


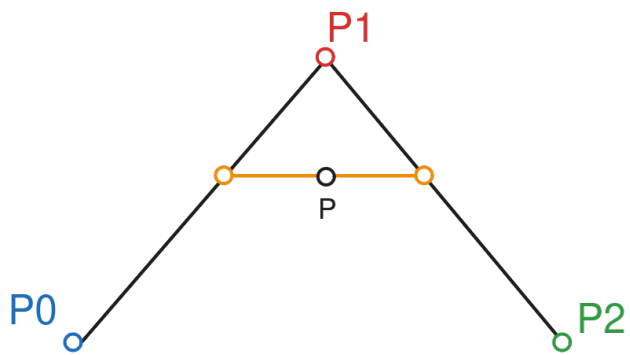
De esta forma obtenemos que $P(0) = P_0$ y $P(1) = P_1$

Ya pero queremos curvas no rectas >:(

Agreguemos otro punto, y usamos lerp de nuevo.

Luego tomamos 2 puntos de cada recta y hacemos **lerp** entre ellos





Si $P_0(t) = (1-t)P_0 + tP_1$ y $P_1(t) = (1-t)P_1 + tP_2$ son los **lerp** entre P_0, P_1 y P_1, P_2 respectivamente. Entonces

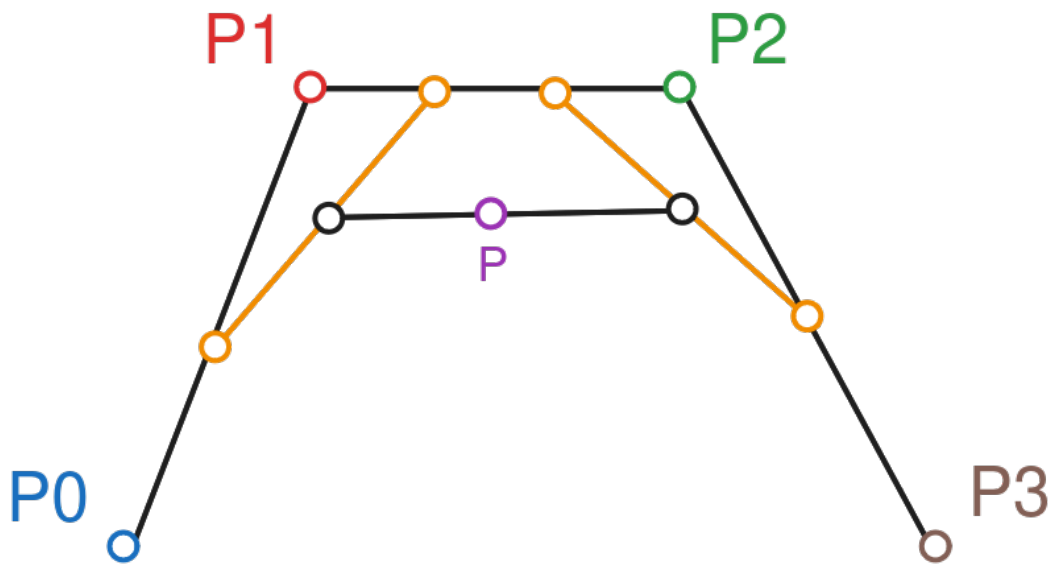
$$P = (1-t)P_0(t) + tP_1(t)$$

$$P = (1-t)((1-t)P_0 + tP_1) + t((1-t)P_1 + tP_2)$$

Esto sería una curva de Bézier Cuadrática

Desmos: Quadratic Bézier Curve

Agreguemos otro punto



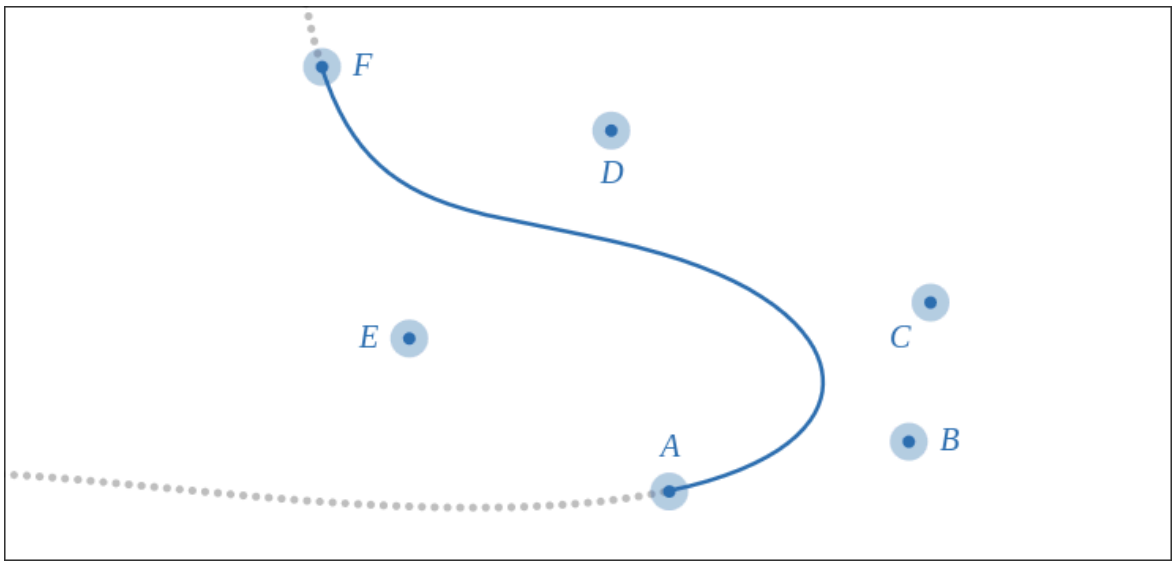
Siguiendo el mismo procedimiento, de hacer lerp sobre el lerp…

Este algoritmo se llama De CastleJau's Algorithm

Y eso sería una curva de Bézier cúbica

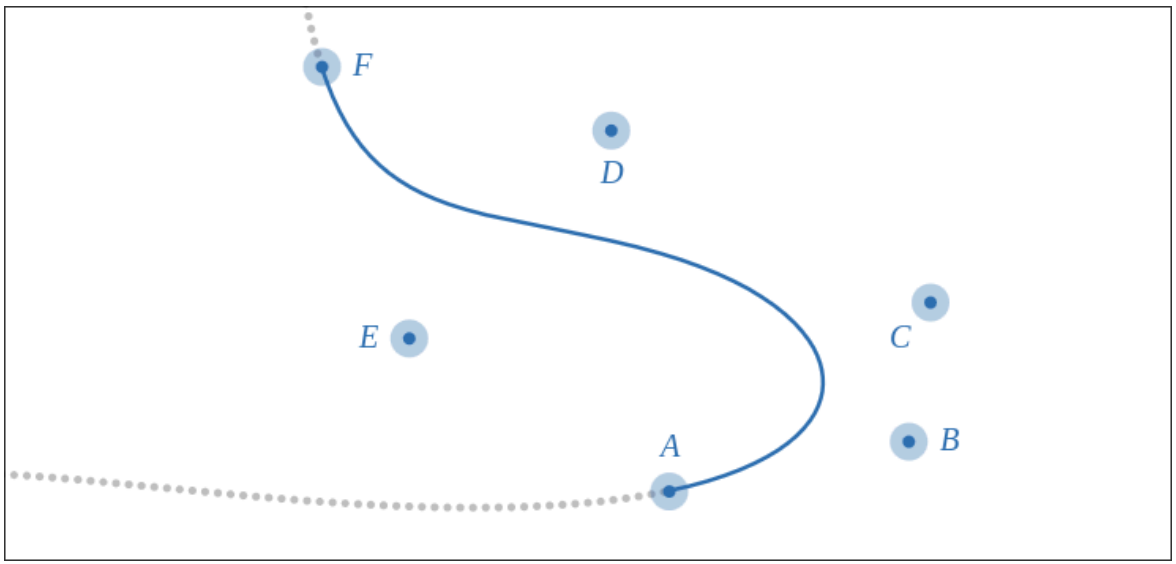
Desmos: Cubic Bézier Curve

¿Qué pasa si seguimos agregando puntos?



Desmos: Con más puntos

¿Qué pasa si seguimos agregando puntos?



Desmos: Con más puntos

No es muy útil :(

- Tenemos poco control sobre la forma de la curva.
- Es caro de calcular: `lerp(lerp(lerp(lerp())))`

Bézier Curve + Bézier Curve + Bézier Curve +Bézi-

Bézier Splines

La idea es unir varias curvas de Bézier → Desmos: Multiple Bézier

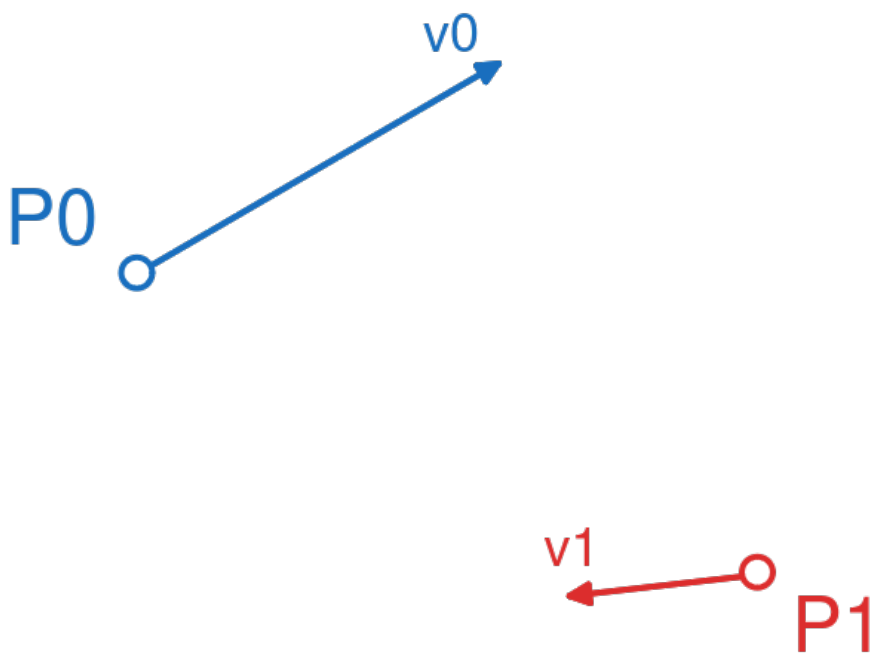
Ahora sí

- Tenemos más control sobre la curva
- El más barato de calcular

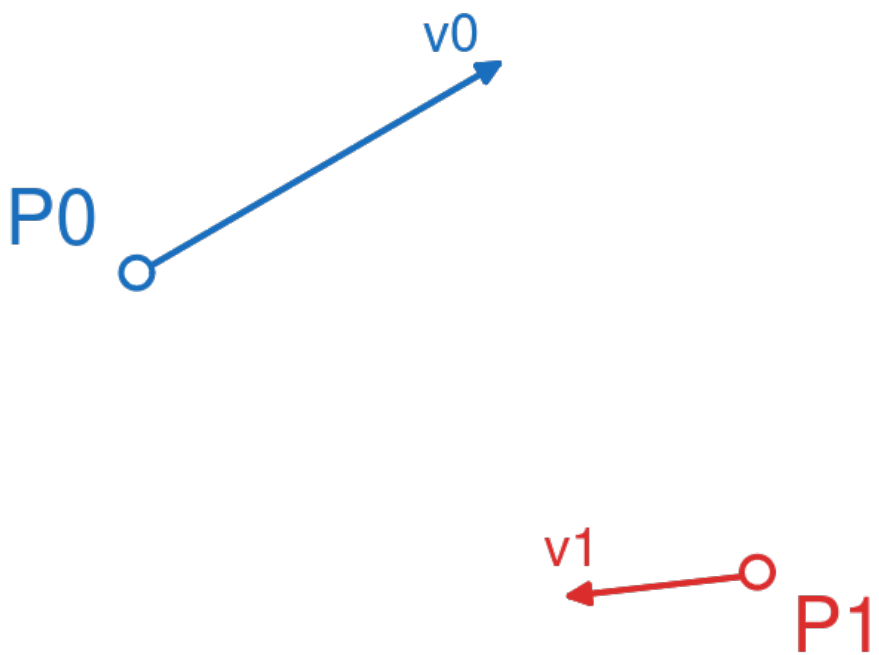
Este tipo de spline es muy usada (Photoshop, Fonts)

Curvas de Hermite

Ahora vamos a partir con dos puntos, y las velocidades que deben tener en la curva.



Ahora vamos a partir con dos puntos, y las velocidades que deben tener en la curva.



Conocemos lo siguiente:

$$P(0) = P_0$$
$$P(1) = P_1$$
$$P'(0) = v_0$$
$$P'(1) = v_1$$

Planteamos una solución:

$$P(t) = at^3 + bt^2 + ct + d$$

Se ve que $d = P_0$ y $c = v_0$. Para sacar a y b obtenemos las ecuaciones

$$P'(1) = v_1 = 3a + 2b + v_0$$
$$P(1) = P_1 = a + b + v_0 + P_0$$

Y luego toca despejarlas...

Deberían obtener que en forma matricial la ecuación debería ser:

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ v_0 \\ v_1 \end{bmatrix}$$

Desmos: Hermite Curve

Pregunta

Se pide utilizar una curva de Bezier para controlar el movimiento de un tiburón :0

Debe usar la función a continuación:

Spline y función oscilante

Propuesto

Modifique el código anterior para generar el movimiento entre dos puntos dados con una Curva de Hermite.

Considere que Hermite está dado por lo siguiente:

$$P(t) = \begin{bmatrix} H_0(t) & H_1(t) & H_2(t) & H_3(t) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}, \text{ donde}$$

$$H_0(t) = 2t^3 - 3t^2 + 1$$

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

Recomendaciones personales

Goddess Freya Holmér

The Continuity of Splines

The Beauty of Bézier Curves