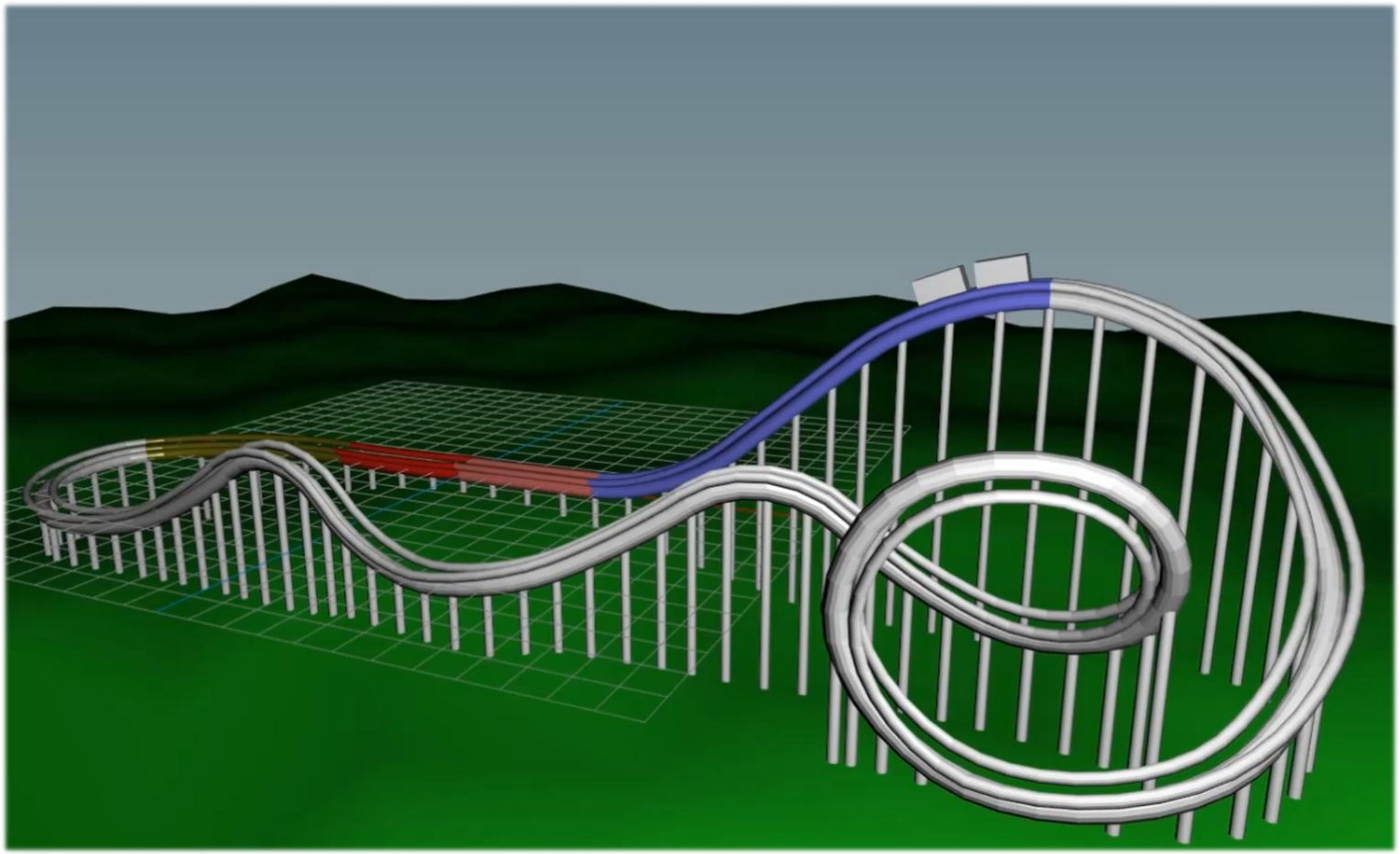
Construção e Simulação Virtual de Montanhas-Russas Voltadas para Jogos Digitais



César Gasparini Fernandes

Introdução

Acredita-se que as primeiras montanhas-russas foram construídas no século XVII e tinham de 20 a 24 metros

Introdução

Objetivos

- Criar um construtor de montanhas-russas fácil de usar;
- Simular o carro no trilho.

Motivações

- Jogos de infância;
- Lidar com Curvas de Bézier;
- Aplicar Álgebra Linear e Cálculo;
- Aprender a técnica de extrusão;
- Manipular modelos por seus vértices.

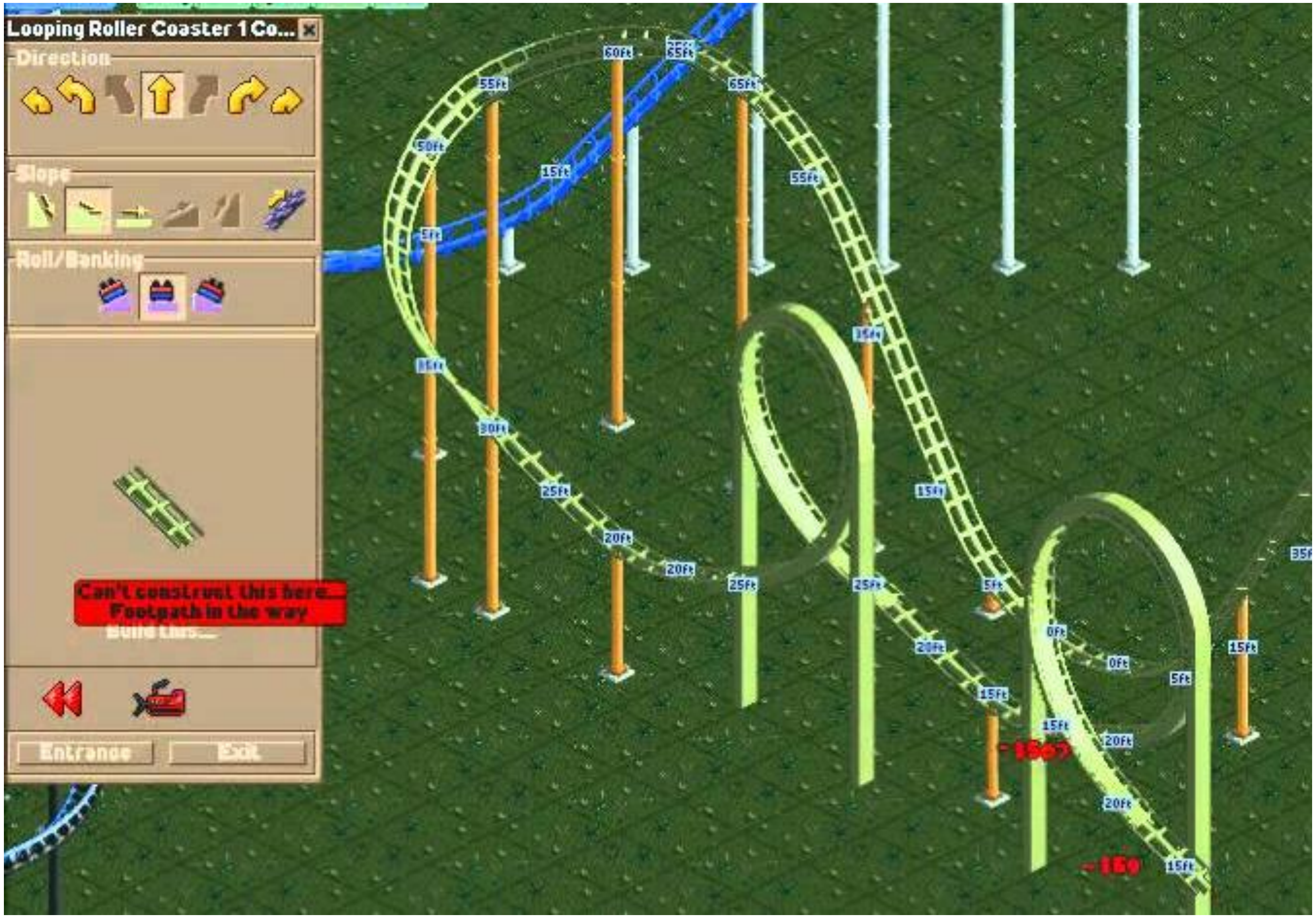
Montanhas-russas e o mundo virtual

As montanhas-russas foram e ainda continuam sendo exploradas em jogos e simuladores.

Elas estão fortemente presentes em jogos de simulação de parques e em simuladores presentes em óculos de realidade virtual.

Simuladores de montanha-russa são mais comuns do que construtores.

Evolução das montanhas-russas virtuais



Roller Coaster Tycoon

Fonte: google.com

Evolução das montanhas-russas virtuais

Roller Coaster Tycoon 3

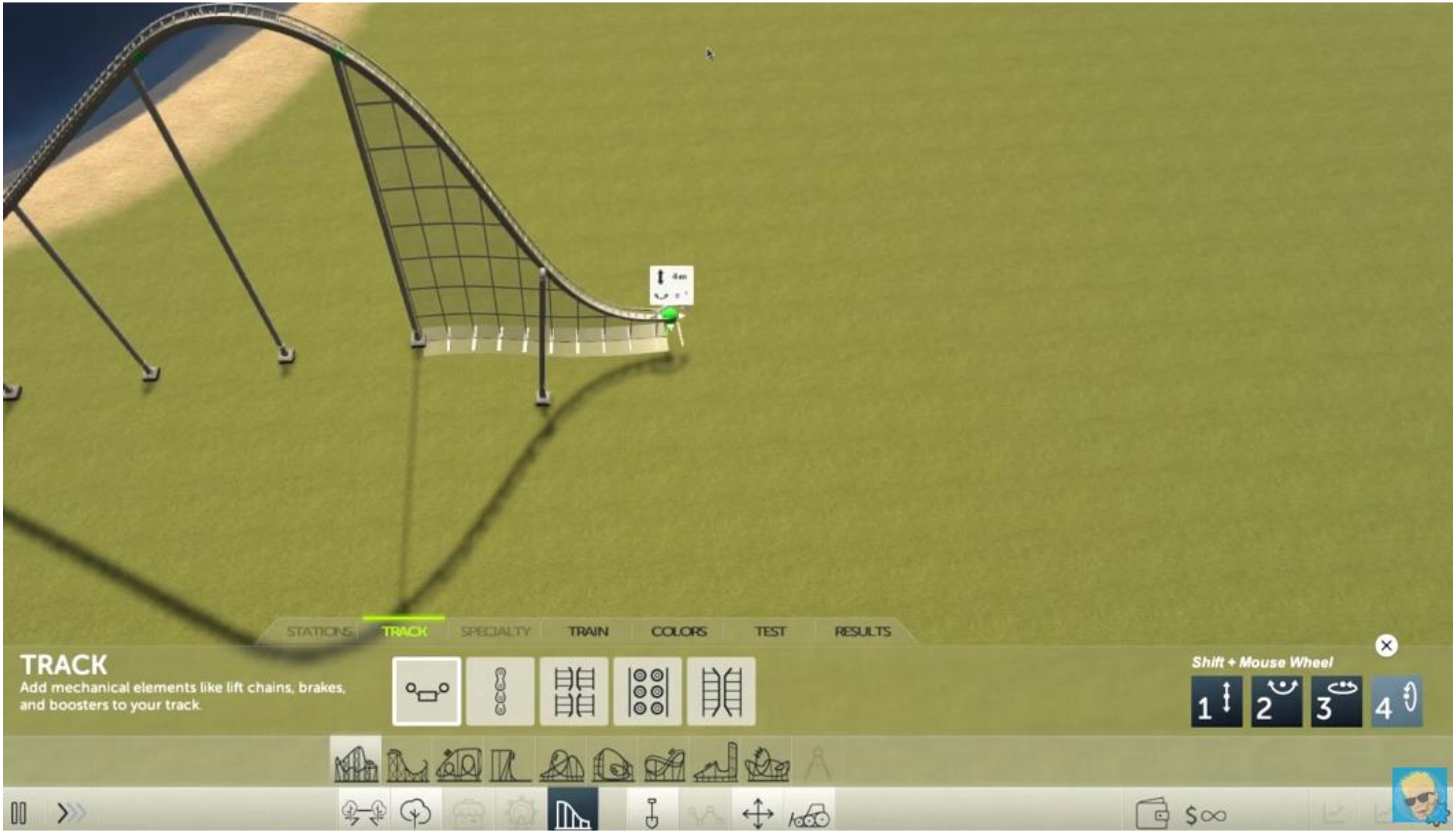
Fonte: youtube.com

Evolução das montanhas-russas virtuais

Parkitect

Fonte: youtube.com

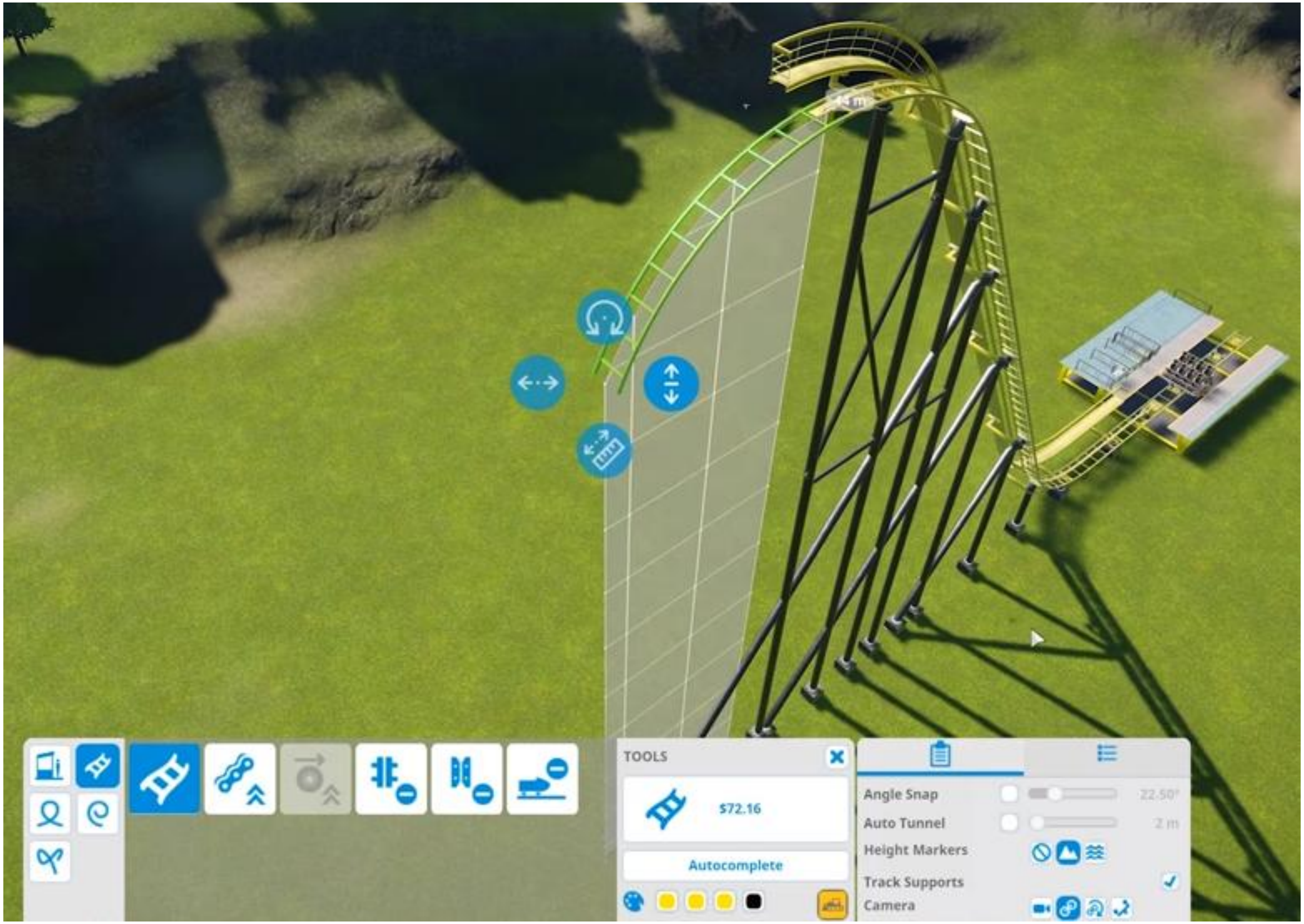
Evolução das montanhas-russas virtuais



Roller Coaster Tycoon World

Fonte: youtube.com

Evolução das montanhas-russas virtuais



Planet Coaster

Fonte: youtube.com

Metodologia

Como construir e simular uma montanha-russa

Apenas um aviso

As ideias a seguir foram totalmente tiradas da minha cabeça.

Não sei como elas estão nos livro (provavelmente estão parecidas).

Portanto, caso você queira reproduzir a montanha-russa da maneira que fiz, um bom material para um estudo inicial são esses slides.

Por isso, me dediquei o máximo que pude neles. Todas as imagens a seguir foram feitas por mim.

Caso você encontre algum erro, por favor me contate: cesar.gaspfer@gmail.com

Construindo

Como podemos “construir” a montanha-russa?

“Alongando” um cilindro ao longo de uma curva descrita pelo jogador.

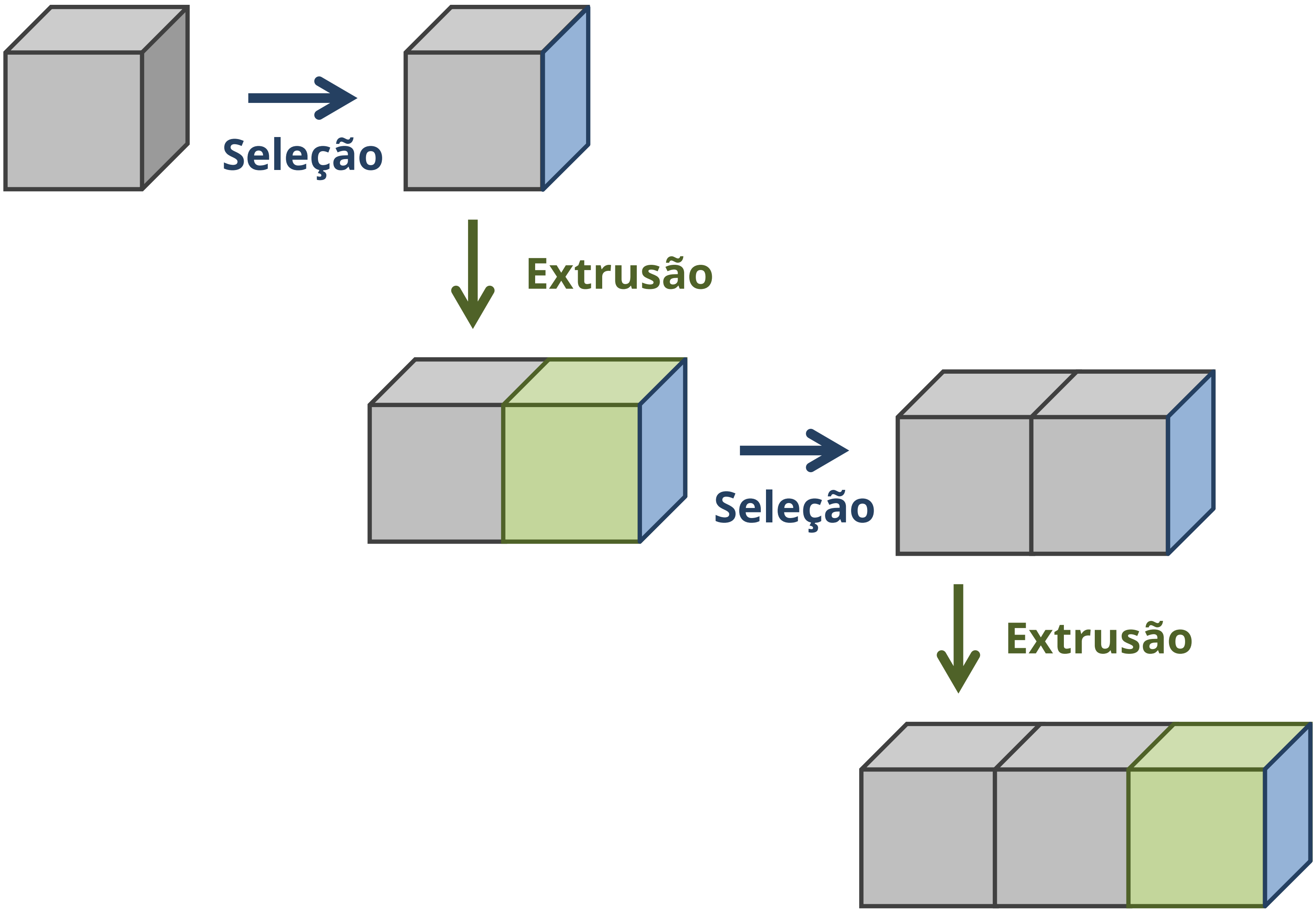
Como?

Extrusão!

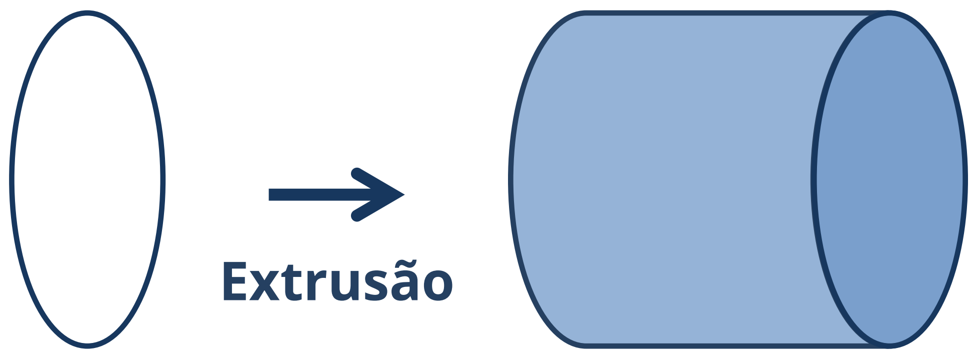
Extrusão

Gera um sólido ou uma superfície 3D
ao estender uma curva ou uma superfície 2D
ao longo de uma curva

Técnica de extrusão a partir de superfícies 2D



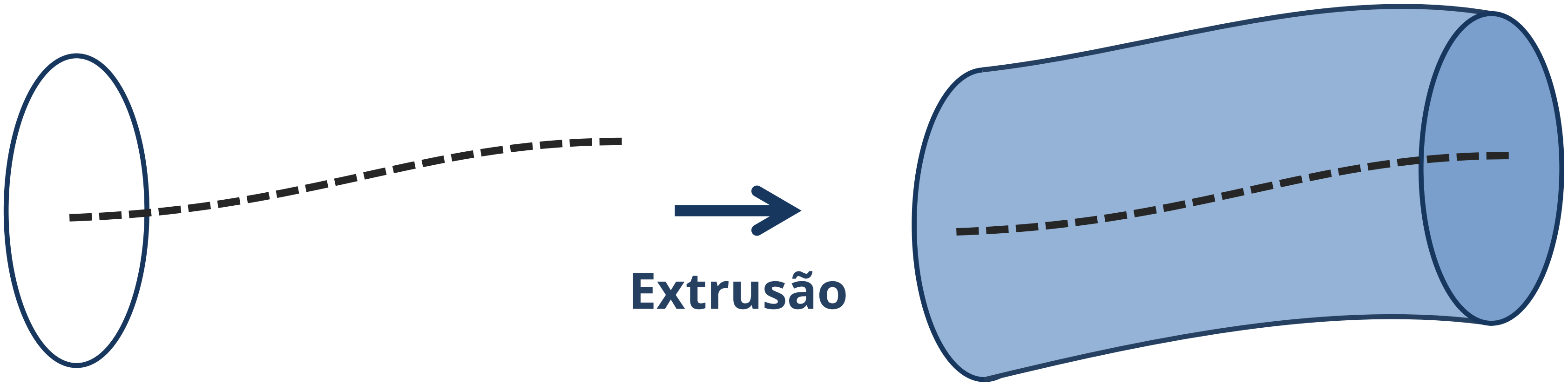
Técnica de extrusão a partir de curvas



$$\begin{cases} x^2 + y^2 = 1 \\ z = 0 \end{cases}$$

$$\begin{cases} x^2 + y^2 = 1 \\ 0 \le z \le 1 \end{cases}$$

Técnica de extrusão ao longo de uma curva



$$\begin{cases} x^2 + y^2 = 1 \\ z = 0 \end{cases}$$

Técnica de Extrusão

A técnica de extrusão é uma das técnicas mais elementares para modelagem.

Ela permite que o artista transforme um cubo em uma espaçonave.

Presente nos softwares de modelagem como Autodesk Maya, Blender, Cinema 4D.

Possibilita que a nova superfície 3D siga uma curva de Bézier.

Curva de Bézier

Uma curva descrita por um polinômio de grau n definido por n+1 pontos

Curva de Bézier

Seja B(t) uma curva de Bézier e P_i um ponto no espaço com $0 \leq i \leq n$, sendo n pertencente aos naturais.

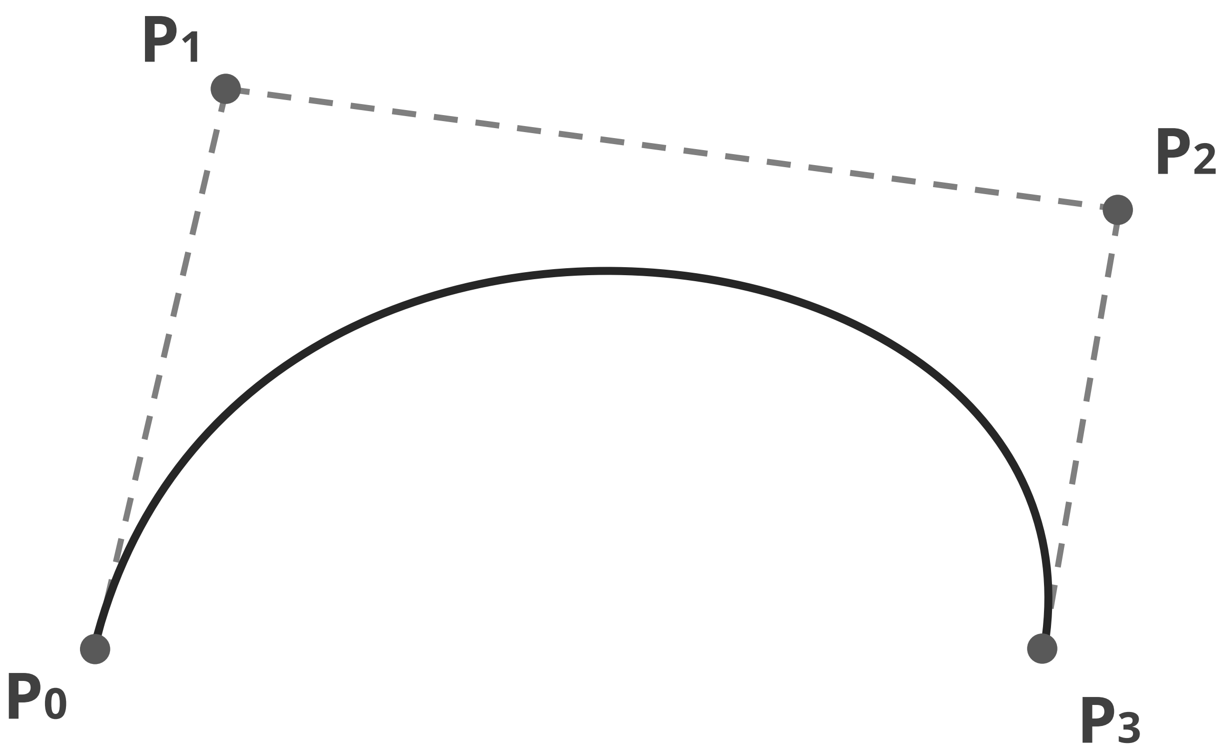
B(t) é definido como:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1 - t)^{n-i} t^i P_i$$

Curva de Bézier Cúbica

A curva de Bézier cúbica é definida por

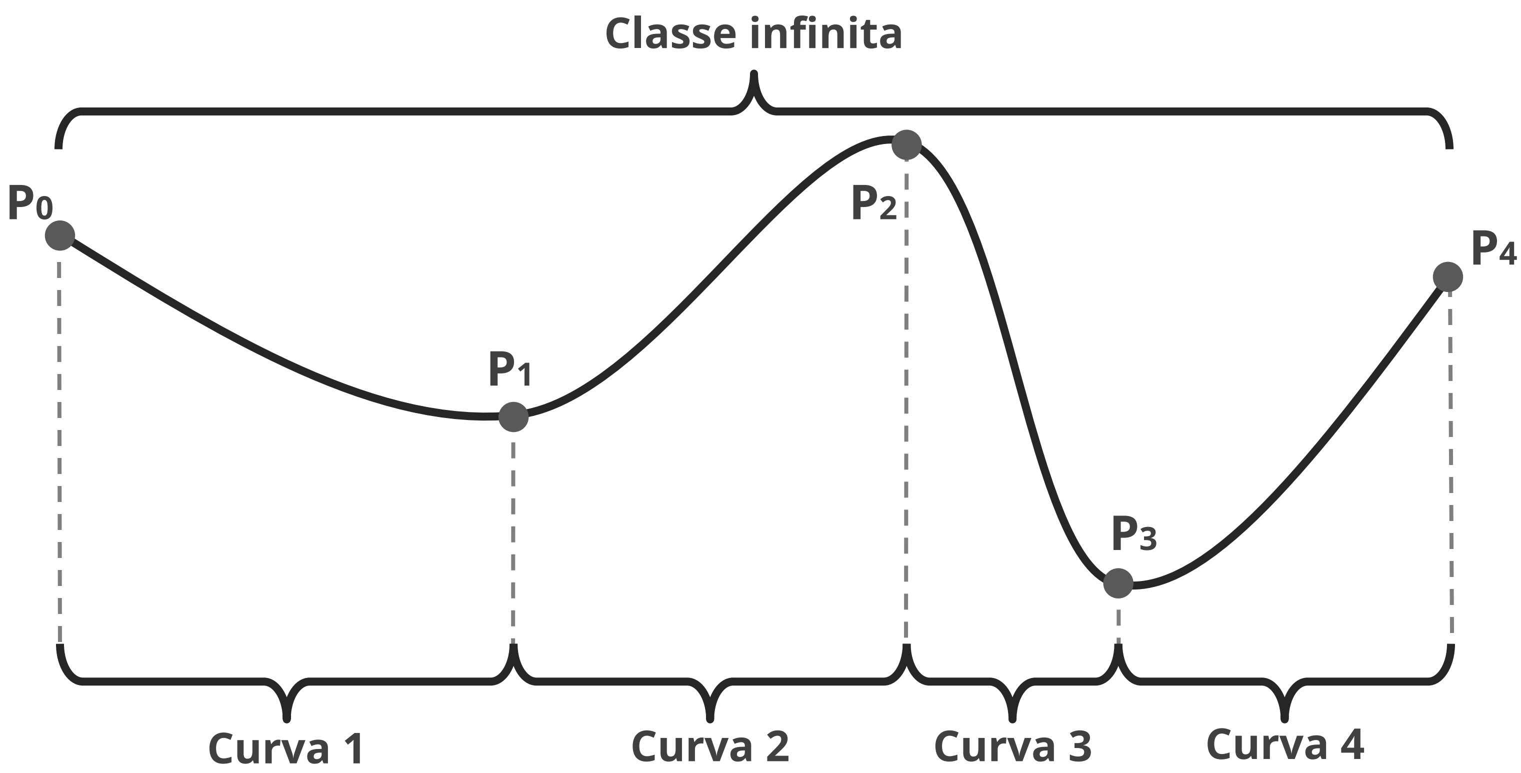
$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, t \in [0,1]$$



Curva de Bézier Cúbica

Uma propriedade interessante é a possibilidade de controlar as derivadas em B(0) e B(1) utilizando P_1 e P_2.

Ela torna possível com que composições de Bézier sejam de classe infinita (infinitamente derivável).



Rotacionando

Orientando os pontos ao longo da curva

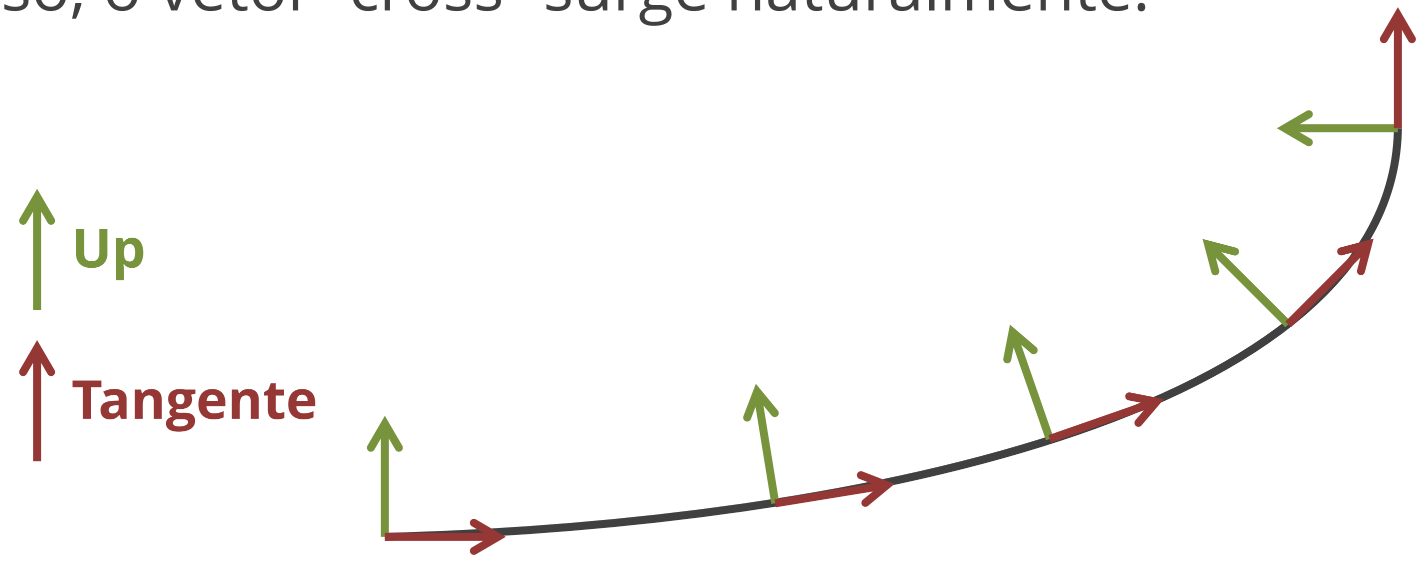
Rotacionando

Temos um conjunto de pontos que descrevem uma curva para ser extrusada.

Devemos orientá-los ao longo da curva de Bézier para dar a ideia de que o modelo é rotacionado ao longo dela.

Para isso podemos utilizar a derivada da curva e atribuí-la um vetor "up" inicial.

Com isso, o vetor "cross" surge naturalmente.

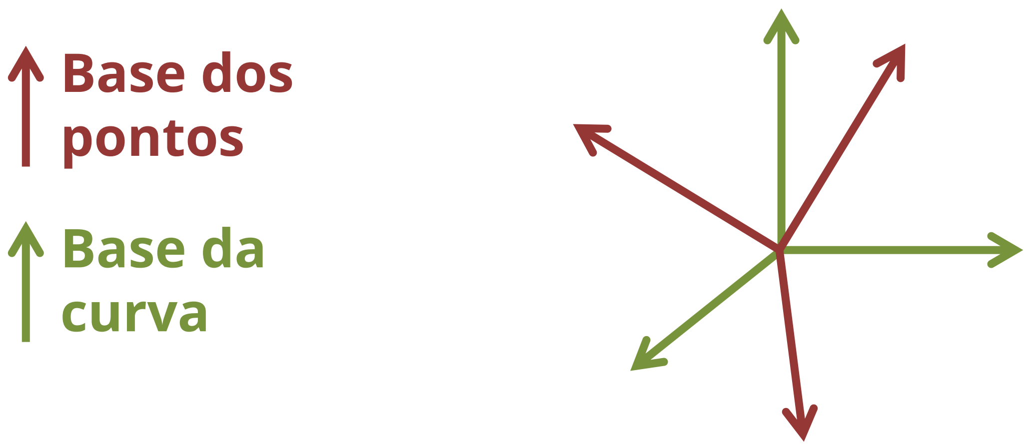


Rotacionando

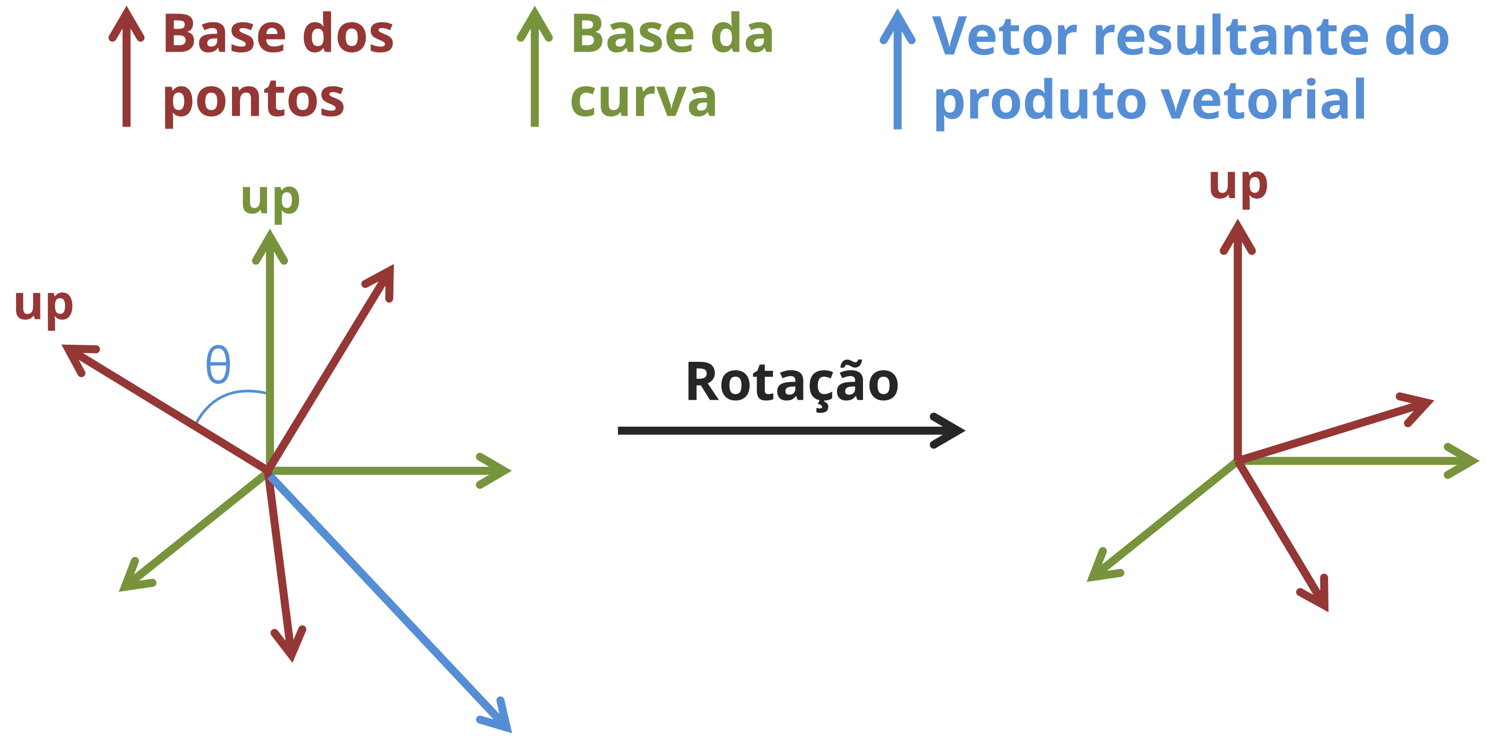
Temos a base ortonormal do conjunto de pontos.

Portanto, para orientá-la, basta fazer uma transformação de base com a orientação do ponto da curva, que é uma base ortonormal.

Para transformar as bases, basta fazer duas rotações.



Primeiro rotacione ao redor do vetor resultante do produto vetorial entre os vetores “up” das duas bases para que eles tenham mesma direção e sentido.



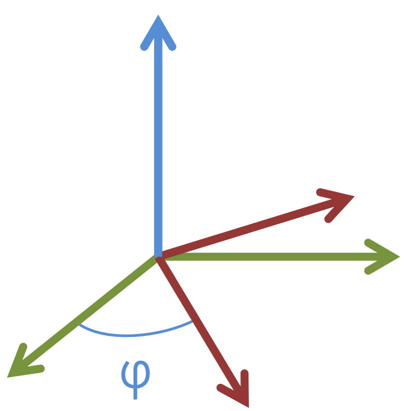
Rotacionando

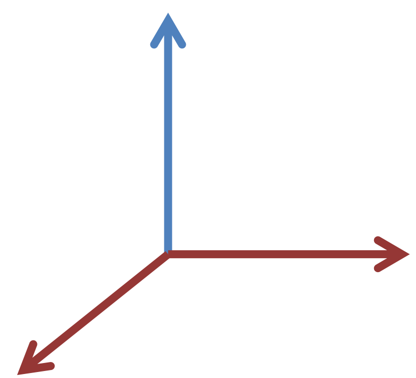
Em seguida rotacione ao redor do vetor "up" para que as bases fiquem iguais.

Base dos pontos

Base da curva

Vetor "up"





Interação do jogador

O jogador possui 4 graus de liberdade:

- Comprimento do segmento;
- Elevação;
- Rotação;
- Inclinação.

A ordem de avaliação é:

Elevação –> **Rotação** -> **Inclinação** –> **Comprimento**

Para seguir os parâmetros dados pelo jogador, o ultimo segmento construído “entra” perpendicularmente em uma esfera imaginária para calcular a forma do próximo segmento seguindo os parâmetros definidos pelo usuário.

Interação do jogador

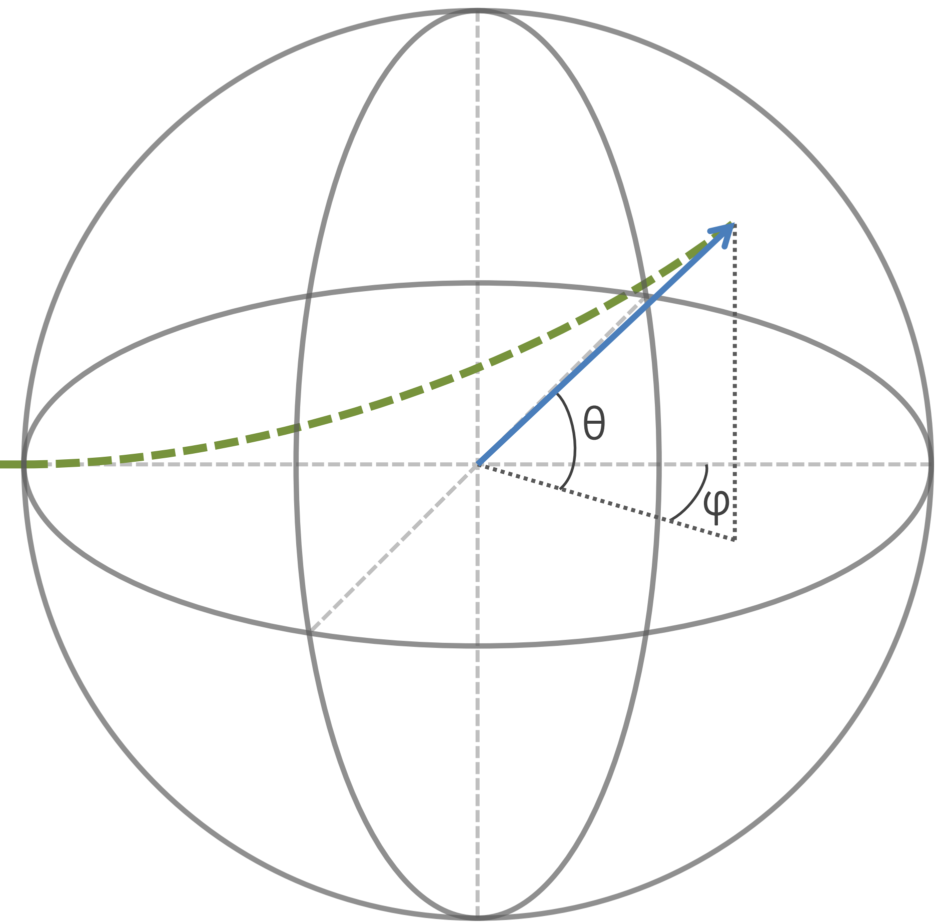
é o segmento anterior

é o segmento a ser construído

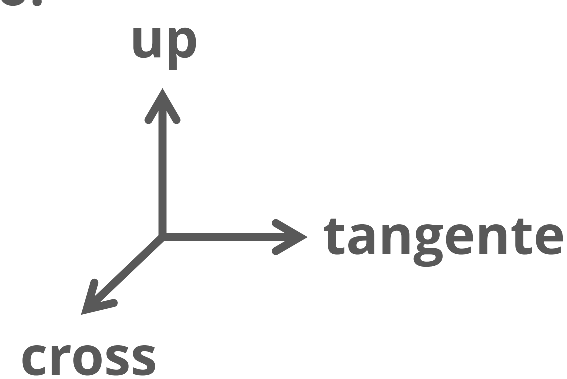
θ é o ângulo de elevação

φ é o ângulo de rotação

é o raio direcional



Orientação:



Elevação

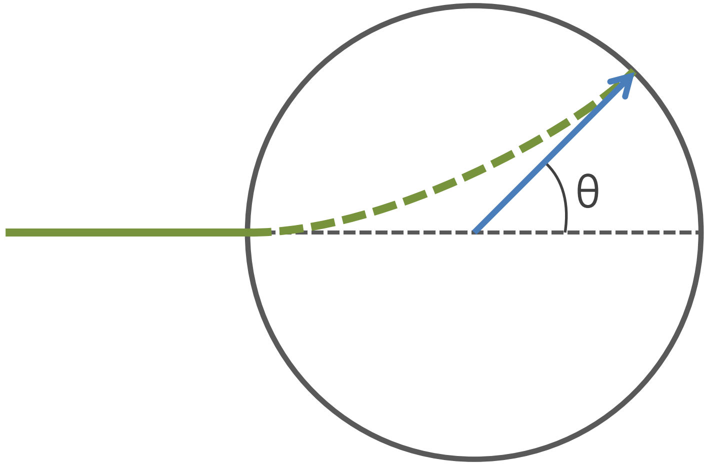
A elevação é a primeira rotação feita. Apenas é rotacionado θ em relação ao vetor resultante do produto vetorial entre os vetores “up” e “tangente” (ou seja, o vetor “cross”).

é o raio direcional

θ é o ângulo de elevação

é o segmento anterior

é o segmento a ser construído



Rotação

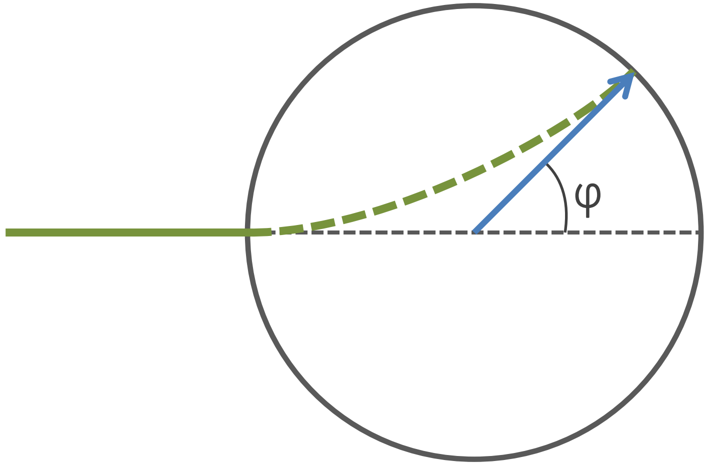
Depois de elevado, o raio direcional é rotacionado φ ao redor do vetor “up”.

é o raio direcional

φ é o ângulo de rotação

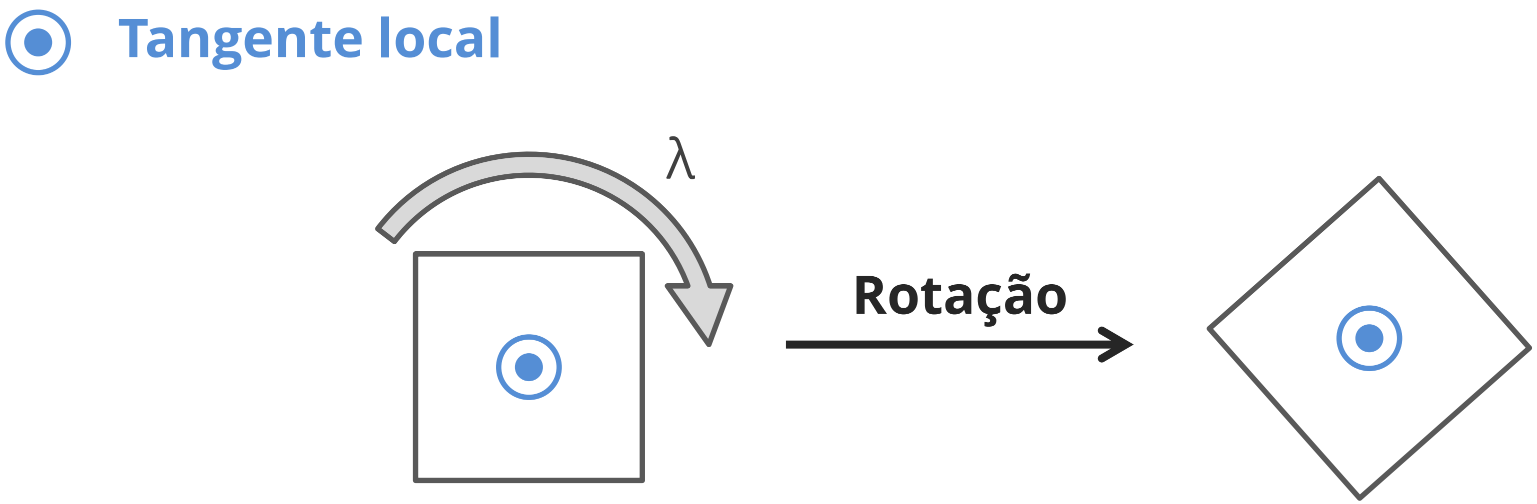
é o segmento anterior

é o segmento a ser construído



Inclinação

Depois de todos esses cálculos e rotações, basta apenas rotacionar o conjunto de pontos ao redor da tangente local.

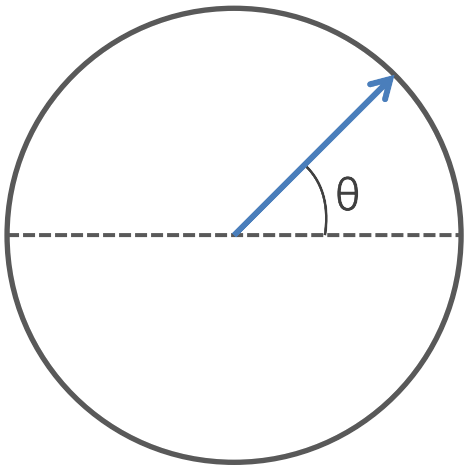


Comprimento

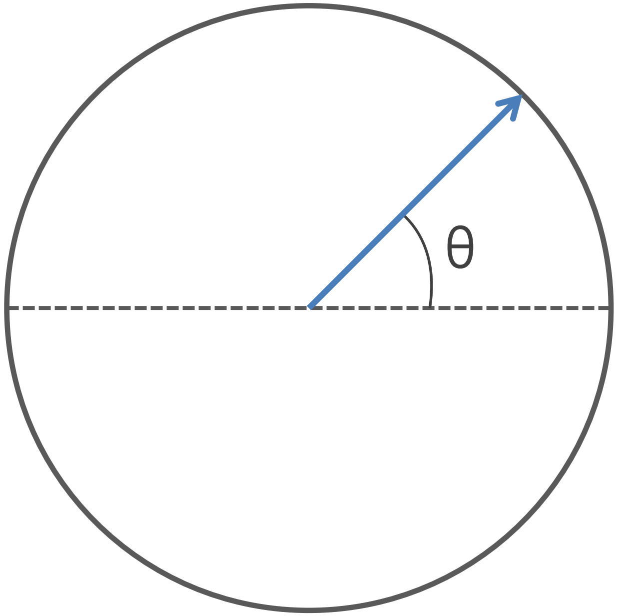
O comprimento altera o tamanho do raio direcional, multiplicando escalarmente ele.

Ele é apenas um escalar do vetor, a elevação e a rotação também influenciam em seu tamanho.

Comprimento = 6



Comprimento = 8



Comprimento

Para se manter o comprimento proposto pelo jogador, é necessário fazer o cálculo da norma do raio direcional.

Para isso devemos seguir as seguintes fórmulas:

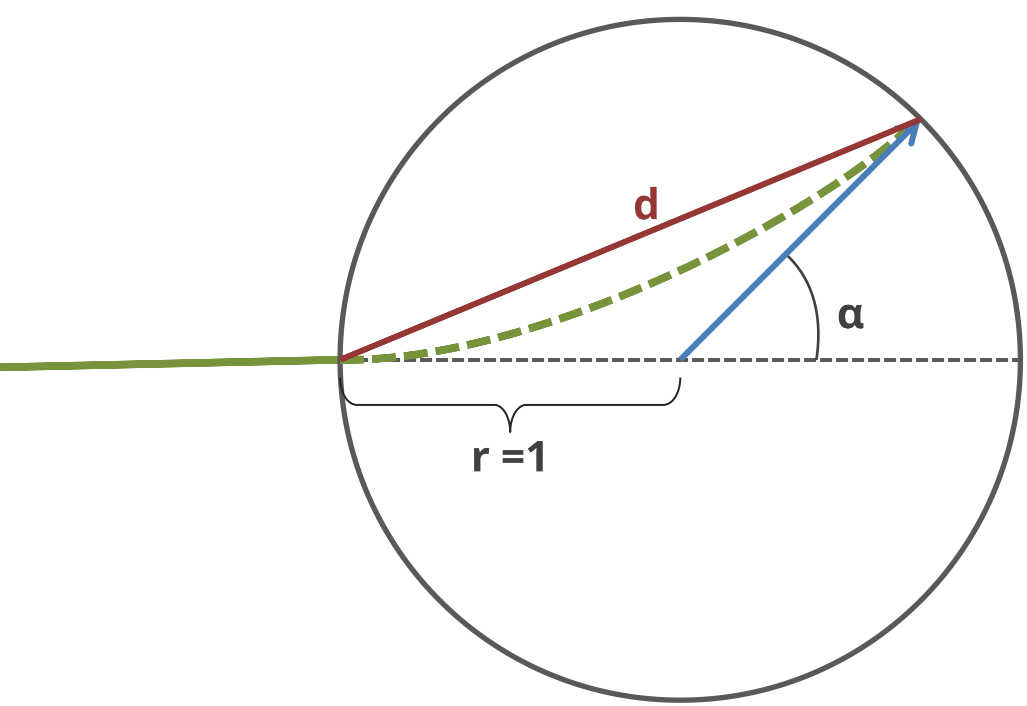
$$\|raio\ direcional\| = \frac{comprimento}{2L} \qquad L = \frac{\alpha d}{2\sin\left(\frac{\alpha}{2}\right)}$$

$$L = \frac{\alpha d}{2\sin\left(\frac{\alpha}{2}\right)}$$

$$d = \|B'(0) + B'(1)\|$$

$$\alpha = \arccos\left(B'(0).B'(1)\right)$$

Comprimento



é o raio direcional

é o segmento anterior

é o segmento a ser construído

$$\|raio\ direcional\| = \frac{comprimento}{2L} \qquad L = \frac{\alpha d}{2\sin\left(\frac{\alpha}{2}\right)}$$

$$L = \frac{\alpha d}{2\sin\left(\frac{\alpha}{2}\right)}$$

$$d = \|B'(0) + B'(1)\|$$

$$\alpha = \arccos\left(B'(0).B'(1)\right)$$

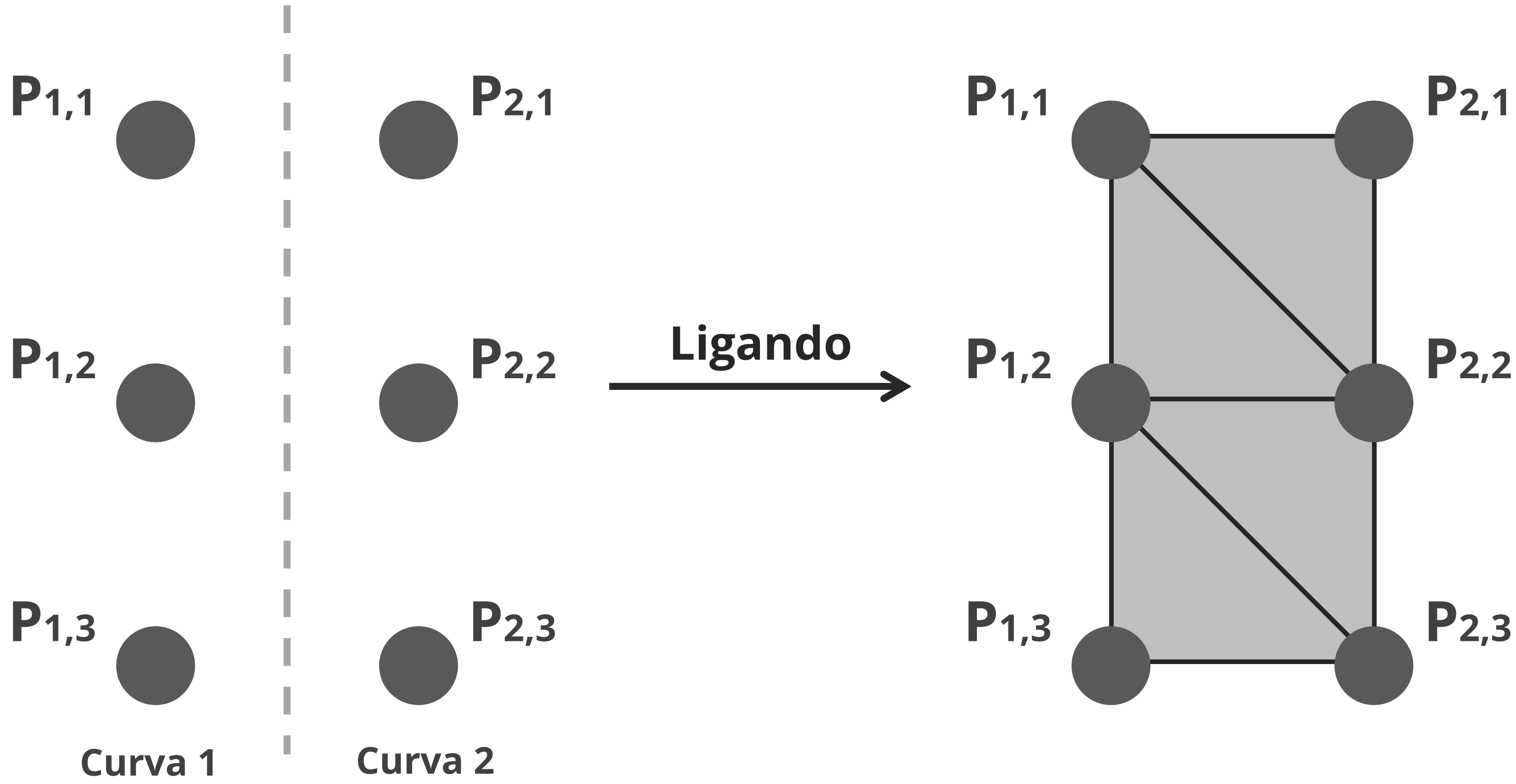
Lembrando que: $\lim_{\alpha \to 0} L = 1$

Ligando os Pontos

Como vamos criar as faces do modelo

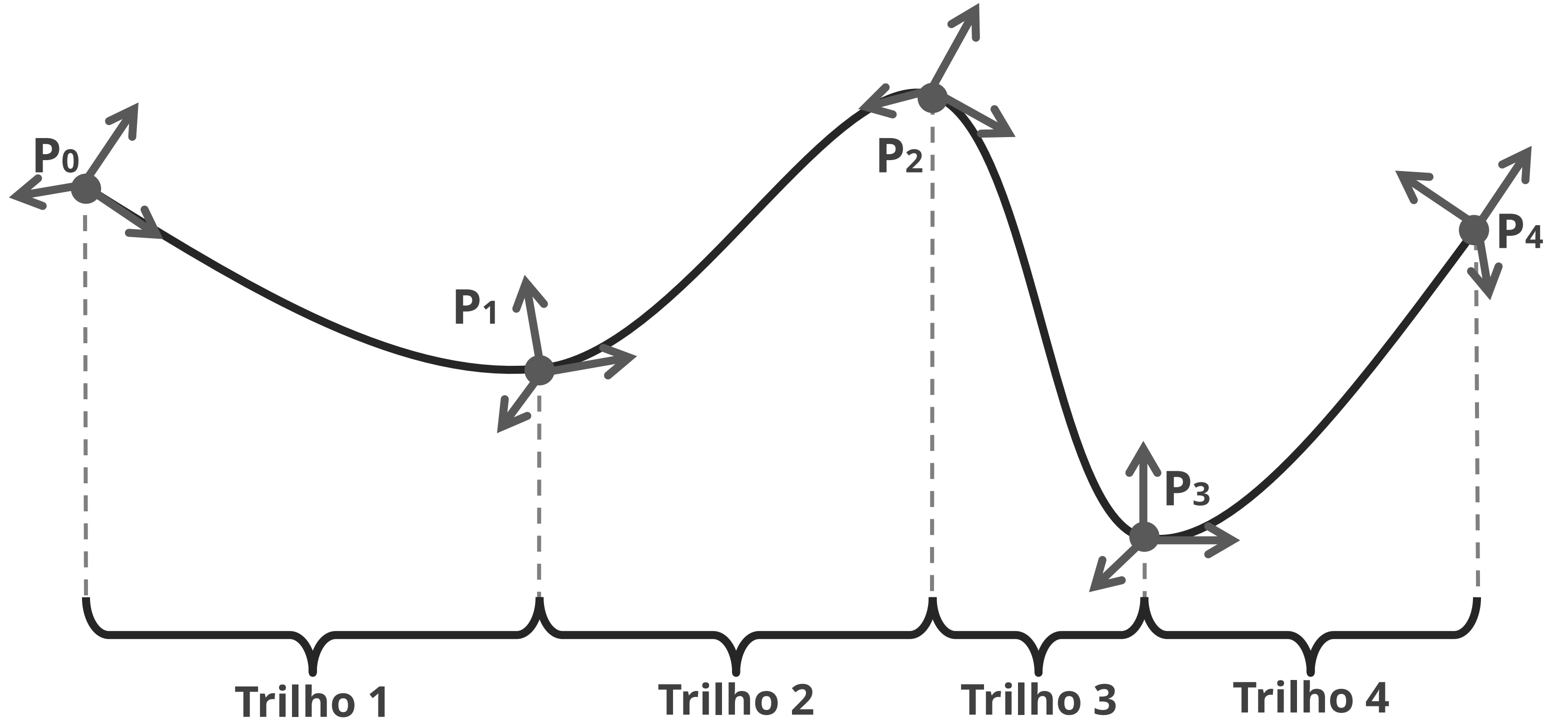
Criando faces

Para criar as faces da montanha-russa enquanto se extrusa, basta ligar os pontos de duas curvas da seguinte forma:



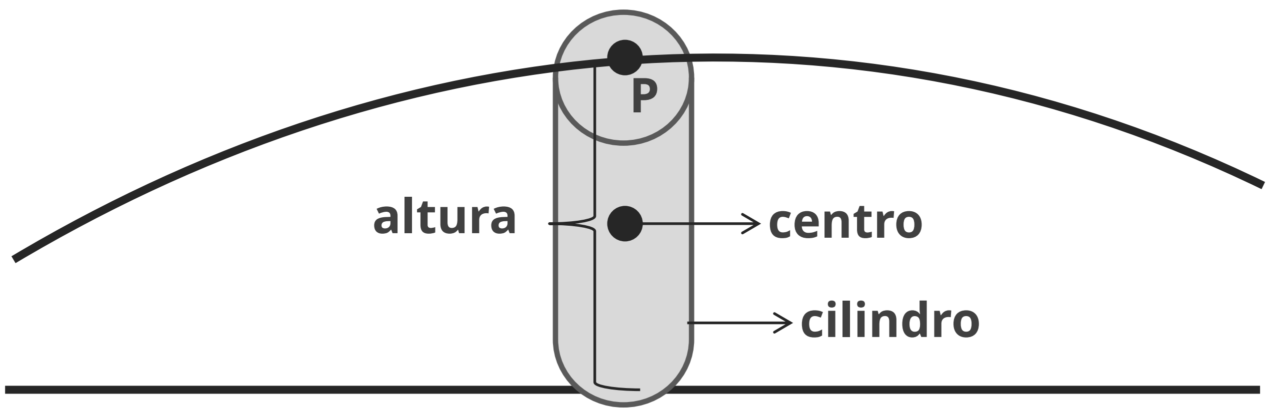
Composição

Para compor curvas de Bézier, basta colocar o começo de uma no final de outra. No caso da montanha-russa, mantendo as orientações.

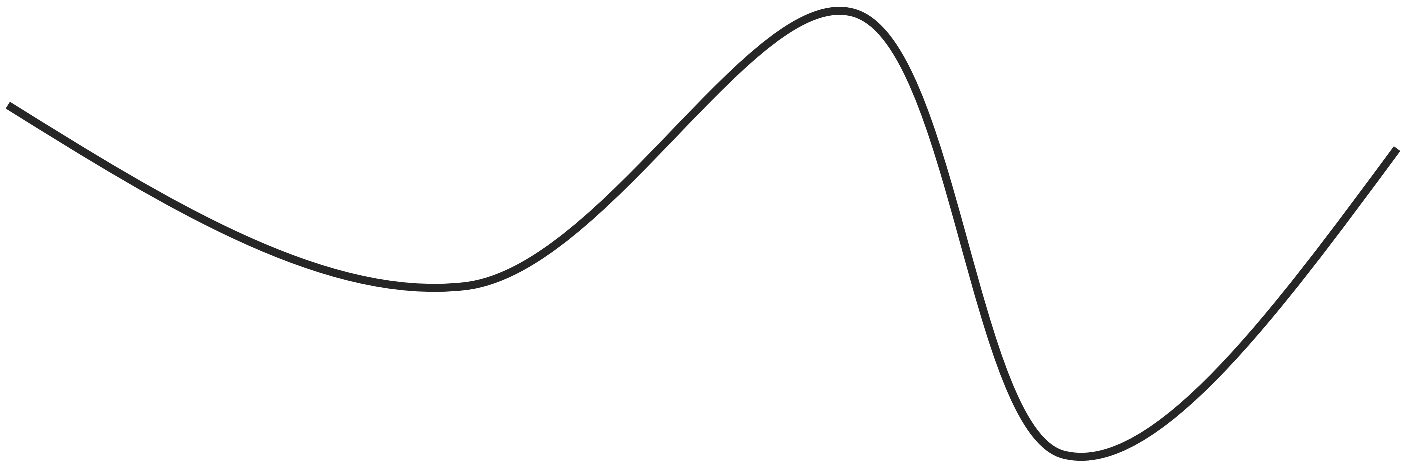


Suportes

Para adicionar os suportes, devemos selecionar um ponto da curva, calcular sua altura em relação ao chão e instanciar um cilindro com essa altura e centralizado entre o ponto e o chão.



Agora temos nossa montanha-russa construída.



Simulando

Como fazer o carro se movimentar

Tipos de trilho

A montanha-russa possui 5 tipos de trilhos:

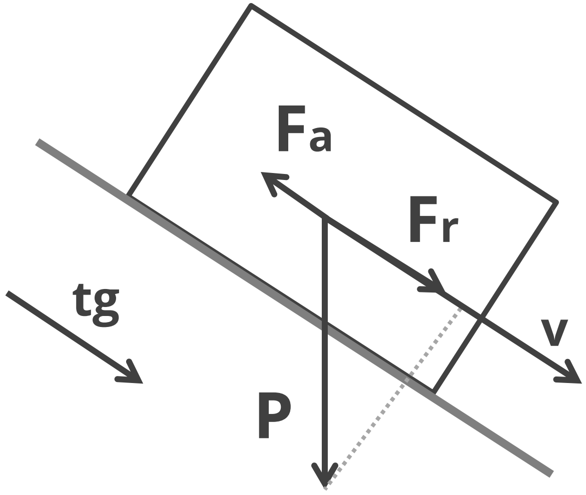
- Plataforma;
- Normal;
- Alavanca;
- Freio;
- Final.

Cada tipo de trilho aplica um tipo de força sobre o carro.

Observação: a unidade de medida para o espaço é unidade do mundo virtual e a chamaremos de “u”.

Normal

Um trilho normal aplica a força da gravidade (**P**) e a força atrito (**F_a**). A força resultante (**F_r**) é definida como:



$$F_r = (P.tg)tg + F_a$$

$$P = 9.8 * m$$

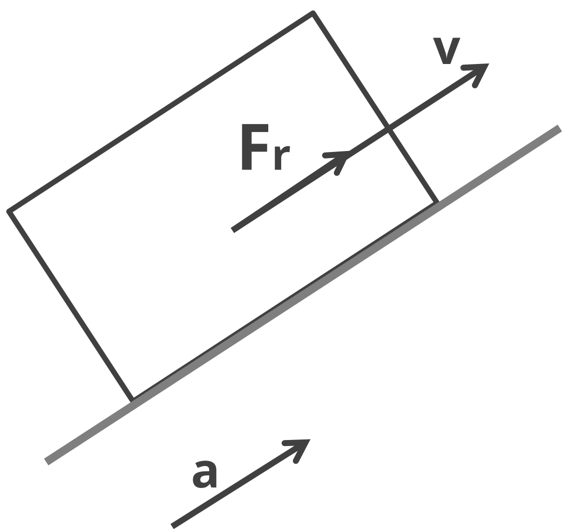
$$F_a = -0.08 * v$$

tg = tangente do carro
v = velocidade do carro
m = massa do carro

(Esquema meramente ilustrativo, dimensões fora de escala)

Alavanca

Caso a velocidade escalar esteja maior do que 4 u/s, é aplicada a força do trilho normal. Caso contrário, é aplicada a seguinte força:



$$\boxed{F_r = a * m}$$

v = velocidade do carro

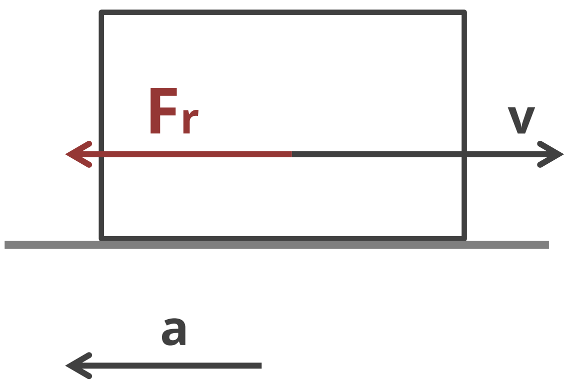
m = massa do carro

a = aceleração = 5 u/s²

(Esquema meramente ilustrativo, dimensões fora de escala)

Freio

Caso a velocidade escalar esteja menor do que 5 u/s², é aplicada a força do trilho normal. Caso contrário, é aplicada a seguinte força:



$$\boxed{F_r = a * m}$$

tg = tangente do carro
v = velocidade do carro
m = massa do carro
a = aceleração = 5 u/s²

(Esquema meramente ilustrativo, dimensões fora de escala)

Plataforma

Caso o carro esteja com velocidade escalar menor do que 4u/s, é aplicado a força da alavanca com aceleração igual a 10 u/s², sendo m a massa do carro.

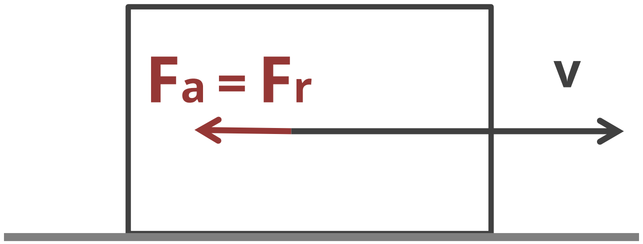
Ou seja, se o carro estiver devagar, ele é empurrado

Caso contrário, é aplicado a força do freio com aceleração igual a 10 u/s², sendo m a massa do carro.

Ou seja, se o carro estiver rápido, ele é freado

Final

O carro sofre uma desaceleração com magnitude igual a 18 u/s².



(Esquema meramente ilustrativo, dimensões fora de escala)

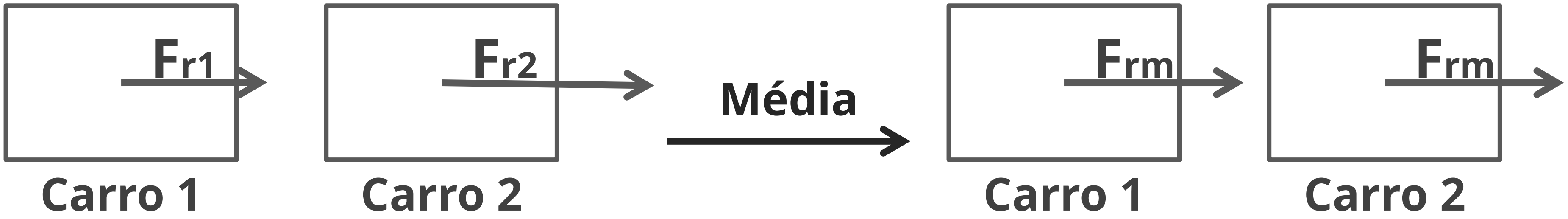
Observação: como é o trilho final, o carro pode apenas ter velocidade maior ou igual a 0 u/s².

Aplicação da força em mais de um carro

Para aplicar as forças nos carros, primeiro se calcula as forças resultantes de cada carro individualmente.

Em seguida, se calcula a média das forças resultantes e a aplica a cada carro.

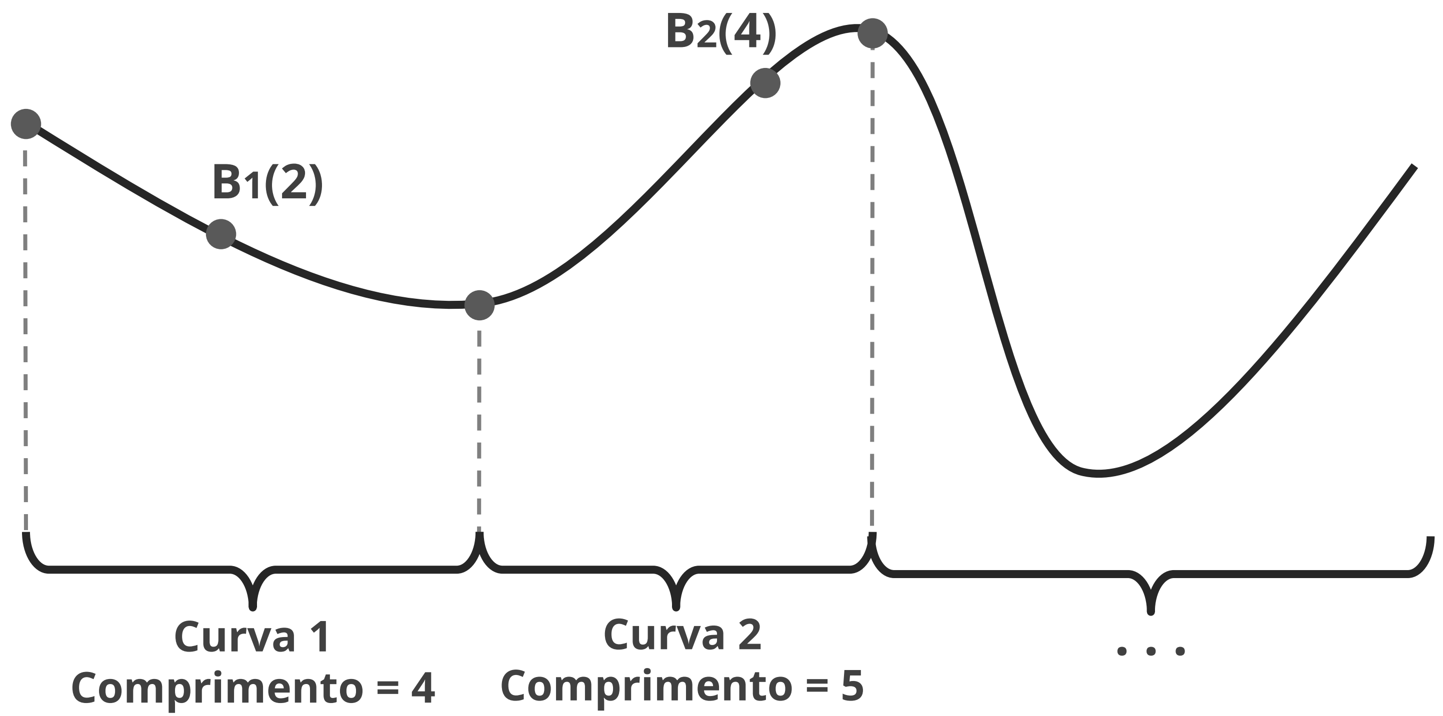
O conjunto de carros são considerados como apenas um corpo por estarem rigidamente ligados.



Posição

A posição é em função da composição das curvas de Bézier.

É feito algo do tipo $B_i(j)$, onde i representa qual curva estamos posicionando, Bi a função da curva e j a posição local da curva, com j pertencendo a [0, comprimento da curva].

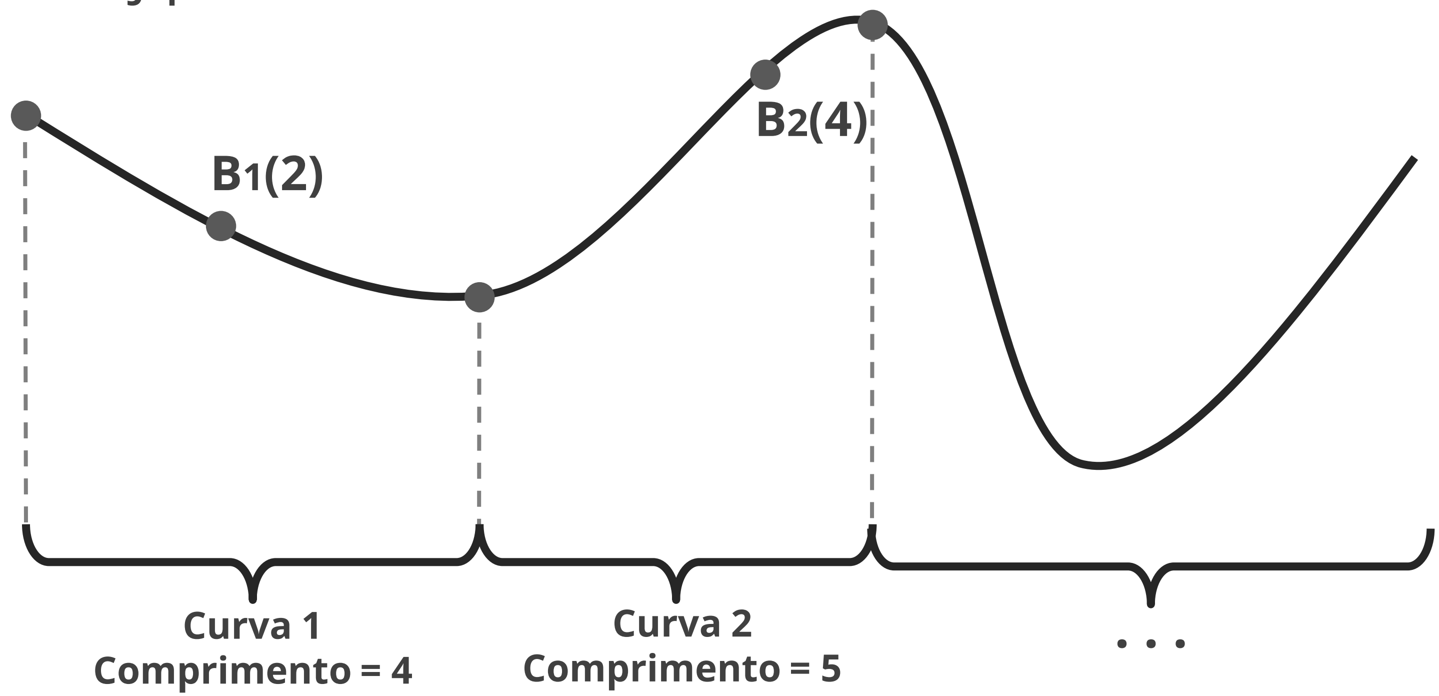


Posição

Para calcularmos a posição do carro, basta então alterar o "j" e o "i" de $B_i(j)$.

Para isso, como projetamos a força anteriormente, incrementar j com a velocidade escalar do carro.

Caso o j atualizado não pertença a [0, comprimento da curva], basta calcular a curva i' que ele pertence e atualizar o j para ser relativo a curva i'.



Posição

Note que $B_i(j)$ é uma função do R para o R^3.

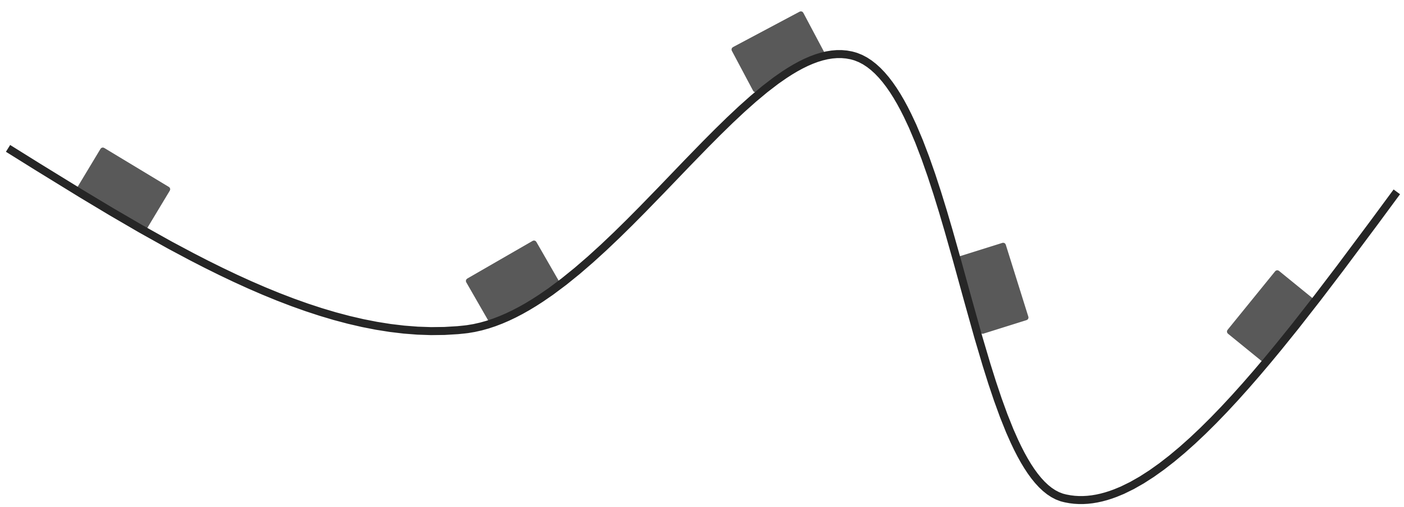
Portanto a posição pertence a um espaço unidirecional (1D), ou seja, tecnicamente o carro pode apenas andar "em linha reta".

Logo, é impossível o carrinho sair do trilho, não importa o "delta t" do step.

Temos então um sistema de partículas altamente especializado.

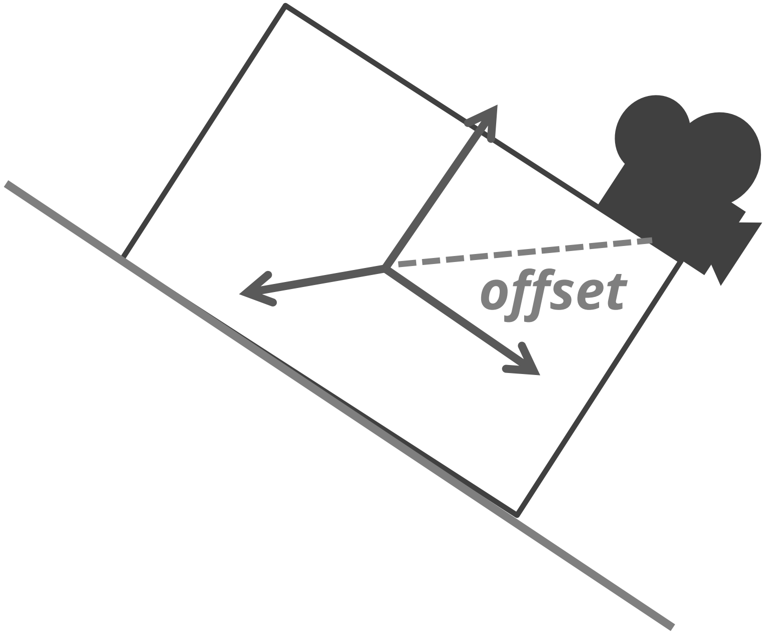
Simulando

Agora temos nossa montanha-russa construída e simulada.



Primeira Pessoa

Para colocar a câmera em “primeira pessoa”, bastou apenas mudar a orientação e posição da câmera para a do carro e deslocar localmente um *offset*.



Resultados

Hora da demonstração

Conclusões

Hora da demonstração

Críticas

Melhorar o design pode torná-lo mais atratível

Paralelizar pode ser uma boa ideia

Reestruturação do código – orientá-lo melhor a objetos e utilizar apenas numpy ao invés de vetores

Término da montanha-russa e salvar dados

Resultados alcançados

Modelar a montanha-russa de maneira genérica

Simular o carro

Criar uma interação fácil entre usuário e a montanha-russa

Completar todo o cronograma

Cronograma

Conteúdo	Estimativa	Feito
Modelo do segmento simples de trilho	5%	5%
Splines corretamente implementadas	15%	15%
Replicação dos segmentos simples de trilho	20%	20%
Ligação correta dos segmentos simples de trilho	10%	10%
Implementação dos suportes	10%	10%
Simulação do carro	25%	25%
Interface de usuário	5%	5%
Implementação da superfície do terreno	5%	5%
Acoplagem da câmera ao carro	5%	5%
Total	100%	100%

100% concluído

O que aprendi

Técnica de extrusão

Manipular geometria

Álgebra Linear e Cálculo

Sistema de partículas

Agradecimentos

Marcel Parolin Jackowski
Aulas e código base

Matheus Tavares Bernardino
Códigos dos EPs de 420

Perguntas

Alguém tem alguma?

Obrigado!

Boas férias e boas festas a todos!