

Entrega 02

Cálculo 01

Profª Renata

404

Not Found

Breno Groba de Azevedo, Igor Almeida, João Victor
Feria, Kaio Inglez e Gustavo Diniz;

. Cálculos de derivadas no jogo

Troca do dia pela noite com a utilização de
blend e derivadas

Derivada aqui significa o quanto blend muda por unidade de tempo (t).

Nos períodos do entardecer e amanhecer, a derivada é diferente de zero — o valor blend muda com o tempo.

A derivada é:

+10 no entardecer: valor crescente (céu escurecendo)

-10 no amanhecer: valor decrescente (céu clareando)

Quando a derivada é zero (durante o dia e a noite), significa que a iluminação é constante.

Intervalo de t	Fase do Ciclo	Fórmula de blend(t)	Valor Inicial → Final	Energia (kWh/dia)	Tipo de Variação
0.0 - 0.4	Dia	blend = 0	0 → 0	0	Constante (sem variação)
0.4 - 0.5	Entardecer	blend = Lerp(0, 1, (t - 0.4) / 0.1)	0 → 1	+10	Crescimento linear
0.5 - 0.9	Noite	blend = 1	1 → 1	0	Constante
0.9 - 1.0	Amanhecer	blend = Lerp(1, 0, (t - 0.9) / 0.1)	1 → 0	-10	Decrescimento linear

Exemplo prático com luz: `sunLight.intensity = Mathf.Lerp(1f, 0f, blend);`

- Durante o entardecer, blend vai de 0 a 1 → luz vai de 1 a 0.,
- Derivada da intensidade da luz seria negativa nesse trecho — representa escurecimento constante.

Derivada da movimentação da câmera do jogador:

Tempo (s)	Posição X	Posição Z	Tecla Pressionada	Δ posição (X, Z)	Velocidade (X, Z)
0	0	0	--	--	--
1	1	0	D	(+1, 0)	(1, 0) m/s

2	1	1	W	(0, +1)	(0, 1) m/s
3	0	1	A	(-1, 0)	(-1, 0) m/s
4	0	0	S	(0, -1)	(0, -1) m/s

Interpretação

- A **variação de posição** (Δx ou Δz) é o quanto o jogador se moveu ao pressionar uma tecla.
- Dividindo pela variação de tempo (Δt), temos a **velocidade**, que é a **derivada da posição**.
- Cada movimento do jogador altera sua posição ao longo dos eixos, e isso pode ser tratado como uma **função posição(t)**.
- O jogo calcula essa variação em pequenos intervalos de tempo para simular a física do movimento.

Derivadas dos perfis de consumo energético

A cidade é composta por construções comuns, como casas, prédios e comércios, que consomem recursos continuamente. Cada morador terá um comportamento diferente em relação ao consumo: a quantidade de habitantes será gerada de forma aleatória, assim como o nível de consumo de cada um, que pode variar entre três níveis:

- **Nível 1:** consome pouco;
- **Nível 2:** consome moderadamente;
- **Nível 3:** consome muito.

O jogador enfrentará desafios como falta de energia, escassez de água, poluição e

outros problemas urbanos típicos. Com o passar dos dias dentro do jogo, o consumo da cidade se acumula, exigindo que o jogador tome decisões conscientes e eficazes para garantir o bom funcionamento e a evolução da cidade.

Para tornar a cidade mais inteligente, o jogador poderá investir em tecnologias sustentáveis e realizar **upgrades nas construções**, que reduzem o consumo de recursos com o tempo. Essas melhorias representam práticas reais de eficiência energética e hídrica adotadas em cidades inteligentes ao redor do mundo.

Explicação em termos de Derivadas:

Considere o consumo total de energia como uma função do nível de upgrade

aplicado, ou seja: $C(n) = C_0 \cdot (1 - r)^n$

Onde:

- $C(n)$: consumo total após o nível de upgrade n
- $C_0 = 535 \text{ kWh/dia}$: consumo inicial sem upgrade
- $r = 0,10$: taxa de redução (10%)
- N : nível de upgrade

Nível de Upgrade (n)	Consumo (kWh/dia)	Redução Absoluta (kWh/dia)	Redução Percentual (%)
0 (inicial)	535,00	00,00	0%
1	481,50	53,50	10%
2	433,35	48,15	19%
3	390,02	43,34	27%
4	351,02	39,00	34%
5	315,92	35,10	41%

A fórmula usada na tabela:

$$C(n) = 535 \cdot (0,90)^n$$