

Entrega 1 — Modelagem Matemática e Funções Aplicadas ao Jogo

Disciplina: Cálculo I (PI)

1. Objetivo

Aplicar conceitos de álgebra e funções para modelar matematicamente as principais mecânicas do jogo de memória desenvolvido na Unity, relacionando os comportamentos de rotação, tempo e lógica de acerto com expressões e funções matemáticas.

2. Definição Matemática do Jogo

O jogo é um jogo de memória digital em 3D, no qual o jogador precisa virar cartas e encontrar pares iguais antes que o tempo acabe.

Cada carta realiza uma rotação suave (flip) em torno de um eixo quando o jogador clica com o mouse, e o cronômetro controla o tempo total da partida (por exemplo, 120 segundos).

3. Identificação dos Conjuntos Numéricos e Operações Algébricas

O jogo utiliza principalmente:

- **Números reais (\mathbb{R}):** representam o tempo contínuo (em segundos) e os ângulos de rotação (em graus).
- **Números inteiros (\mathbb{Z}):** usados para contar pares encontrados e rodadas.
- **Operações algébricas:** soma, subtração e interpolação (transformação gradual de um valor em outro).

4. Modelagem Matemática da Rotação da Carta

A rotação de uma carta é feita por mudança gradual suave entre duas rotações (inicial e final).

No código CardFlip.cs:

```
card.transform.rotation = Quaternion.Slerp(  
    startRotation,  
    endRotation,  
    elapsedTime / flipDuration  
);
```

Matematicamente, essa interpolação é representada por uma função contínua de tempo:

$$R(t) = R_0 + (R_f - R_0) \cdot \frac{t}{T}, \quad 0 \leq t \leq T$$

Onde:

- R_0 = rotação inicial (em graus)
- R_f = rotação final (180°)
- T = duração da animação (0,35 s)
- t = tempo decorrido desde o início da rotação

Essa é uma **função linear**, pois o movimento é proporcional ao tempo, sem aceleração. Graficamente, seria uma reta que cresce de até .

5. Modelagem Matemática do Cronômetro

O cronômetro (CronometroJogo.cs) decresce com o tempo até chegar a zero:

```
tempoRestante -= Time.deltaTime;
```

Matematicamente:

$$T(t) = T_0 - t$$

Onde:

- T_0 = tempo inicial em segundos (por exemplo, 120 s)
- $T(t)$ = tempo restante no instante t

É uma função **decrecente linear**, cujo gráfico é uma reta que parte de 120 e chega a 0. Quando $T(t)=0$, o jogo termina.

6. Aplicação de Equações e Funções para Definir Regras do Jogo

1. Regra de rotação (flip):

A rotação total ocorre quando $t = T$. Isso define o momento exato em que a carta está totalmente virada.

2. Regra de tempo:

O jogador vence antes que $T(t) = 0$. A perda é determinada por:

Fim de jogo se $T(t) \leq 0$

3. Regra de comparação de cartas:

Quando duas cartas são abertas, o sistema verifica:

Par encontrado se $C_1 = C_2$

7. Relação com Conceitos de Função

- **Funções lineares:** O tempo decresce de forma linear.
- **Funções contínuas:** A rotação é contínua (sem saltos), definida no intervalo $[0, 0.35]$.
- **Condições lógicas (funções por partes):** A lógica do jogo pode ser vista como função por partes:

$$f(t) = \begin{cases} \text{Virar carta,} & \text{se clicar e não estiver animando} \\ \text{Esperar,} & \text{se já houver duas viradas} \\ \text{Desvirar,} & \text{se não for par} \end{cases}$$

8. Conclusão

A modelagem matemática do jogo combina funções lineares e contínuas para representar a rotação das cartas e a contagem regressiva do tempo. Essas relações permitem controlar a lógica do jogo de forma previsível e suave, demonstrando a aplicação prática da álgebra e das funções no desenvolvimento de jogos digitais.

Scripts Utilizados

- CardFlip.cs
- GameManager.cs
- CronometroJogo.cs