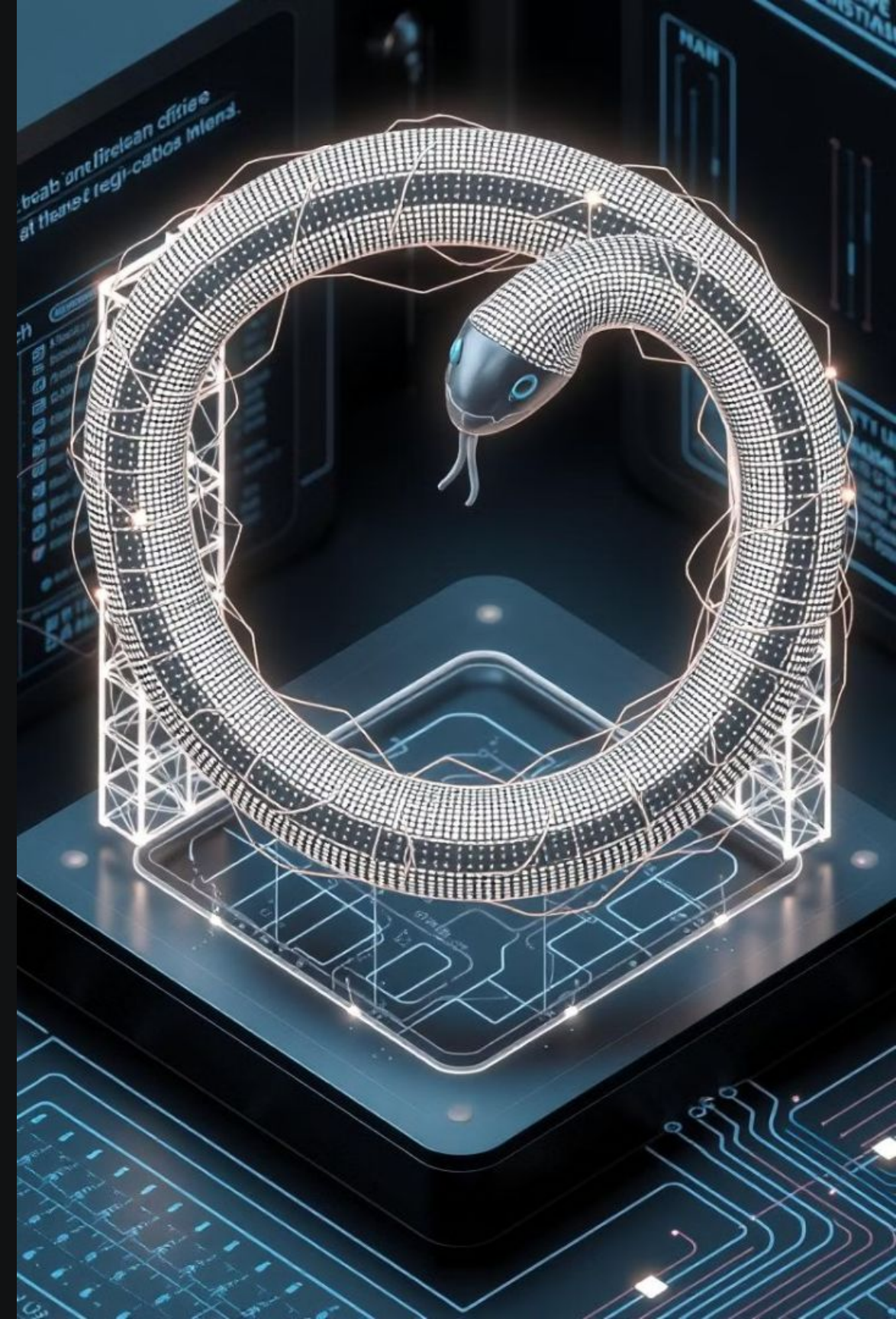


# Projeto Snake com Deep Q-Learning

Aprendizado por Reforço com Engenharia de Software

Caue Yanagihara, Eduardo Cavazin, Vinicius Margonar, Renato Platz



# Introdução

1

## Deep Q-Learning

Aplicação no jogo Snake.

2

## Aprendizado

Tentativa e erro com reforços.

3

## Engenharia de Software

Integração de boas práticas.



# Fundamentos Técnicos

**$Q(s, a)$**

Função de valor de ação  
estimada.

**Entrada**

11 variáveis de estado.

**Saídas**

3 ações possíveis.



# Estratégia de Exploração



**Positive Reinforcement**  
**Negative Reinforcement**  
**Shaped Reward Function**



**Recompensas**

Comer: +10, Colisão: -10.



**Ajuste Fino**

Comportamento do agente.



# Experimentos e Testes

## FPS Testados:

10;

50;

100;

## Gamma Testado:

Entre 0.5 e 0.95;

## Epsilon Testados:

80;

100;

200;

# Resultados Observados

10

FPS

Agente cauteloso.

100

FPS

Decisões rápidas.

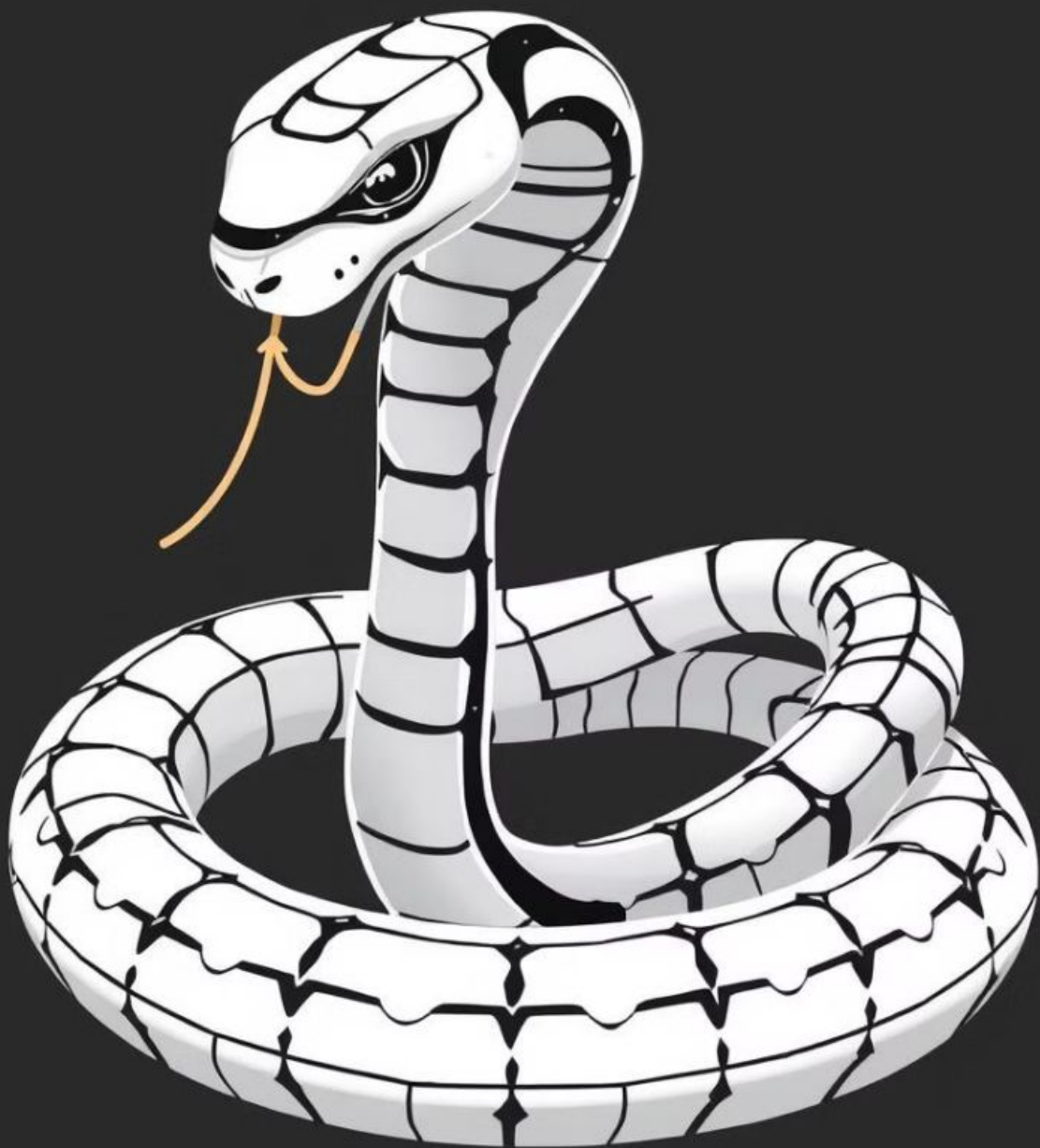
84

Score

FPS 50, recompensa +15.



# Comportamentos da IA



1

**Evita Colisões**

2

**Busca Comida**

3

**Estratégias**

Zig-zag, loops.

# Estrutura de Código

1

**agent.py**

Lógica da IA.

2

**game.py**

Ambiente e regras.

3

**helper.py**

Visualização.



# Conclusão

1

## Deep Q-Learning

Eficaz, mas exige ajustes finos e ajustes contínuos conforme a evolução do projeto

2

## Aprendizado

Parâmetros como *gamma*, *epsilon* e recompensas influenciam fortemente o comportamento

3

## Engenharia de Software

Refatoração e design modular foram essenciais para clareza e flexibilidade do projeto

4

## Correlação

O projeto conecta a teoria da IA com a prática de Engenharia de Software, aplicando o ciclo de PSA

