UNIVERSIDADE REGIONAL INTEGRADA
DO ALTO URUGUAI E DAS MISSÕES

TDE - Implementação de Reinforcement Learning

Aluno: João Vitor Paska Rossetto
Professor: Lucas Bucior

Erechim RS
2025

1. Introdução

Este relatório apresenta o desenvolvimento e avaliação de um agente inteligente baseado na técnica de Reinforcement Learning (Aprendizado por Reforço), utilizando o método Q-Learning, conforme introduzido na disciplina de Inteligência Artificial da URI Erechim.

O projeto consiste na criação de um ambiente simulado em forma de grid 2D, onde um agente deve aprender a:

- Navegar pelo ambiente;
- Coletar todos os suprimentos disponíveis;
- Evitar zumbis, que atuam como estados de punição;
- Ultrapassar obstáculos (rochas);
- Minimizar o número total de passos;
- Alcançar a área segura (porta) com a maior recompensa acumulada possível.

Este ambiente foi desenvolvido com interface gráfica utilizando Pygame, possibilitando visualizar tanto o comportamento aleatório inicial quanto a política ótima aprendida após o treinamento.

2. Fundamentação Teórica

2.1. Aprendizado por Reforço

O Aprendizado por Reforço (Reinforcement Learning – RL) é uma abordagem em que um agente aprende por meio de interação direta com o ambiente. Ele realiza ações, observa seus efeitos e recebe recompensas, ajustando seu comportamento para maximizar o ganho futuro acumulado.

Este paradigma é baseado em tentativa e erro, sendo especialmente eficaz em problemas de tomada de decisão sequencial.

2.2. Q-Learning

O Q-Learning é um algoritmo off-policy que visa estimar a função de valores Q(s, a), representando o valor esperado de tomar uma ação a em um estado s.

Sua atualização segue a equação:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

Onde:

- α é a taxa de aprendizado
- γ é o fator de desconto
- r é a recompensa recebida
- s’ é o próximo estado

Essa técnica permite que o agente aprenda a política ótima mesmo sem conhecer o modelo do ambiente.

3. Estrutura do Ambiente Construído

O ambiente foi implementado como um grid de tamanho fixo, carregado pelo arquivo simulator.py.

Cada célula do grid pode conter:

- Agente
- Zumbi
- Rocha
- Suprimentos
- Área segura (porta)
- Espaço vazio

O agente pode se mover em quatro direções:

- Cima
- Baixo
- Esquerda
- Direita

4. Modos de Ambiente

4.1. Modo A – Mapa Fixo 1

Mapa estático, utilizado para testes iniciais do comportamento do agente. Permite validar a aprendizagem em um layout controlado. (Figura na pág. 4)

4.1. Modo B – Mapa Fixo 2

Versão alternativa, com distribuição diferente de zumbis, rochas e suprimentos. Utilizado para avaliar se a política aprendida se adapta a estruturas diferentes. (Figura na pág. 5)

4.3. Modo CUSTOM

Permite que o próprio aluno defina manualmente um mapa customizado, possibilitando:

- Criação de cenários específicos
- Ajuste de desafios
- Avaliação do comportamento do agente em ambientes mais complexos

É o modo utilizado para registros e capturas de tela durante a simulação (Figura na pág. 6)

4.4. Modo RANDOM

Gera mapas totalmente aleatórios, servindo para avaliar se o agente aprende a generalizar. Implementado como demonstração adicional. (Figura na pág. 7)

5. Simulação Rodando

Print da Simulação. (Figura na Pág 8)

Administrador: Windows Pow

```
PS C:\Users\joaovpr\Documents\Faculdade\7_Semestre\IA\ReinforcementLearning> python main.py
pygame 2.6.1 (SDL 2.28.4, Python 3.12.7)
Hello from the pygame community. https://www.pygame.org/contribute.html
----------------------------------
TREINANDO O AGENTE...........
Episódio:     0 | Recompensa média (últ. 1000): -15.00 | Epsilon: 0.999
Episódio:  1000 | Recompensa média (últ. 1000): -13.13 | Epsilon: 0.367
Episódio:  2000 | Recompensa média (últ. 1000):   7.16 | Epsilon: 0.135
Episódio:  3000 | Recompensa média (últ. 1000):  40.36 | Epsilon: 0.050
Episódio:  4000 | Recompensa média (últ. 1000):  59.43 | Epsilon: 0.018
Episódio:  5000 | Recompensa média (últ. 1000):  64.87 | Epsilon: 0.010
Episódio:  6000 | Recompensa média (últ. 1000):  66.69 | Epsilon: 0.010
Episódio:  7000 | Recompensa média (últ. 1000):  66.17 | Epsilon: 0.010
Episódio:  8000 | Recompensa média (últ. 1000):  65.52 | Epsilon: 0.010
Episódio:  9000 | Recompensa média (últ. 1000):  66.17 | Epsilon: 0.010
Episódio: 10000 | Recompensa média (últ. 1000):  66.60 | Epsilon: 0.010
Episódio: 11000 | Recompensa média (últ. 1000):  65.92 | Epsilon: 0.010
TESTANDO O AGENTE (passo a passo):
------------------------------------------------------
Passo 01: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total:  -1.0
Passo 02: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total:  -2.0
Passo 03: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total:  -3.0
Passo 04: DIREITA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total:  +7.0
Passo 05: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total:  +6.0
Passo 06: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total:  +5.0
Passo 07: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total:  +4.0
Passo 08: DIREITA → ANDANDO                    Recompensa:  -1.0 | Total:  +3.0
Passo 09: DIREITA → ANDANDO                    Recompensa:  -1.0 | Total:  +2.0
Passo 10: DIREITA → ANDANDO                    Recompensa:  -1.0 | Total:  +1.0
Passo 11: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total:  +0.0
Passo 12: DIREITA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +10.0
Passo 13: DIREITA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +20.0
Passo 14: BAIXO   → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +30.0
Passo 15: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +29.0
Passo 16: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +28.0
Passo 17: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +27.0
Passo 18: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +26.0
Passo 19: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +25.0
Passo 20: ESQUERDA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +35.0
Passo 21: DIREITA → ANDANDO                    Recompensa:  -1.0 | Total: +34.0
Passo 22: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +33.0
Passo 23: DIREITA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +43.0
Passo 24: DIREITA → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +53.0
Passo 25: CIMA    → COLETOU SUPRIMENTO         Recompensa: +10.0 | Total: +63.0
Passo 26: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +62.0
Passo 27: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +61.0
Passo 28: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +60.0
Passo 29: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total: +59.0
Passo 30: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total: +58.0
Passo 31: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total: +57.0
Passo 32: CIMA    → ANDANDO                    Recompensa:  -1.0 | Total: +56.0
Passo 33: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +55.0
Passo 34: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +54.0
Passo 35: ESQUERDA → ANDANDO                    Recompensa:  -1.0 | Total: +53.0
Passo 36: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +52.0
Passo 37: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +51.0
Passo 38: BAIXO   → ANDANDO                    Recompensa:  -1.0 | Total: +50.0
Passo 39: BAIXO   → ALCANÇOU ÁREA SEGURA       Recompensa: +20.0 | Total: +70.0
------------------------------------------------------
Status final: ALCANÇOU ÁREA SEGURA
Presentes coletados: 8 de 8
Passos executados: 39
Recompensa total acumulada: 70
------------------------------------------------------
----------------------
Modo de ambiente: CUSTOM
Status: ALCANÇOU ÁREA SEGURA
Presentes coletados: 8 de 8
Quantidade de Passos: 39
Recompensa total acumulada: 70
```

6. Sistema de Recompensas

A definição das recompensas foi essencial para induzir o comportamento desejado:

Evento	Recompensa
Andar	-1
Coletar suprimento	+10
Chegar à saída	+20
Ser atacado por zumbi	-10
Ir à porta sem coletar tudo	-1

Esses valores foram estabelecidos para:

- Premiar a coleta de suprimentos
- Penalizar caminhos longos (efeito do -1 por passo)
- Evitar colisões com zumbis
- Incentivar alcançar a porta apenas após completar os objetivos

7. Treinamento e Parâmetros do Agente

Durante o treinamento, foram utilizados:

- Episódios: 11 mil
- Epsilon decay: converge até 0.01
- α (taxa de aprendizado): 0.1
- γ (desconto): 0.9

O treinamento segue um ciclo:

- Resetar o ambiente
- Executar ações (ε-greedy)
- Receber recompensas
- Atualizar tabela Q
- Armazenar desempenho médio

Durante o processo, como evidenciado na imagem da simulação (pág. 6), as recompensas médias estabilizam entre 60 e 70, demonstrando convergência satisfatória da política aprendida.

8. Resultados Obtidos

Após o treinamento, o agente demonstrou:

- ❖ Coleta completa dos suprimentos

Em todos os testes, o agente aprendeu que coletar todos os presentes gera maior recompensa final.

- ❖ Evita zumbis de forma eficiente

Ao longo do treinamento, colisões diminuem até praticamente desaparecer.

- ❖ Caminho eficiente

O agente aprendeu a priorizar caminhos que minimizem penalidades por movimento.

- ❖ Política estável

Os resultados indicam que o agente aprendeu uma política consistente, mesmo em mapas alternativos.

8. Discussão

O comportamento aprendido demonstra que:

- A tabela Q se estabiliza de forma sólida
- O agente explora bem no início e explora economicamente no final
- O sistema de recompensas foi bem calibrado
- A abordagem Q-Learning é adequada ao problema

O único modo que apresenta variação significativa é o RANDOM (como esperado), pois a geração aleatória às vezes cria mapas impossíveis ou pouco favoráveis.

9. Conclusão

Com base nos experimentos, conclui-se que o agente foi capaz de aprender políticas ótimas, maximizando a recompensa total e cumprindo o objetivo proposto: coletar suprimentos e chegar à área segura de forma eficiente.

O projeto permitiu aplicar conceitos fundamentais de IA, como:

- Formulação de ambiente
- Política ε-greedy
- Atualização da função Q
- Análise de convergência
- Representação de estados em grid

Além disso, proporcionou experiência prática com simulação, visualização gráfica com Pygame e estratégias de RL.