IART Trabalho 2

Aprendizagem por Reforço em Jogo do Tipo Solitário

Jogo: BoxWorld 2

Realizado por:
Diogo Mendes, up201605360
Diogo Sousa, up201706409

Introdução

- Este projeto foi desenvolvido no âmbito da cadeira de Inteligência Artificial e tem como objetivo a aplicação de algoritmos de aprendizagem por reforço num jogo do tipo solitário, neste caso o BoxWorld 2. De forma a concretizá-lo, treinamos um agente que resolve os níveis do jogo no menor número de movimentos possíveis.

Ferramentas e Algoritmos utilizados

- Implementação dos algoritmos de aprendizagem por reforço Q-Learning e SARSA.
- Github , Eclipse e Windows.

Regras do Boxworld 2

- O BoxWorld 2 consiste num jogo de puzzle em que o jogador controla uma personagem num mundo 2D visto de cima com o objetivo de chegar ao fim de cada nível. No entanto para fazer isso tem de movimentar certas caixas para poder ter um caminho a percorrer. A personagem pode empurrar as caixas para a posição oposta da sua posição desde que o espaço esteja vazio. O objetivo é completar cada nível com o mínimo de movimentos e 'empurrões' possível.
- No entanto, devido a problemas de desenvolvimento do trabalho, os níveis do nosso ambiente de treino não possuem caixas, sendo só necessário o agente deslocar-se para a saída.

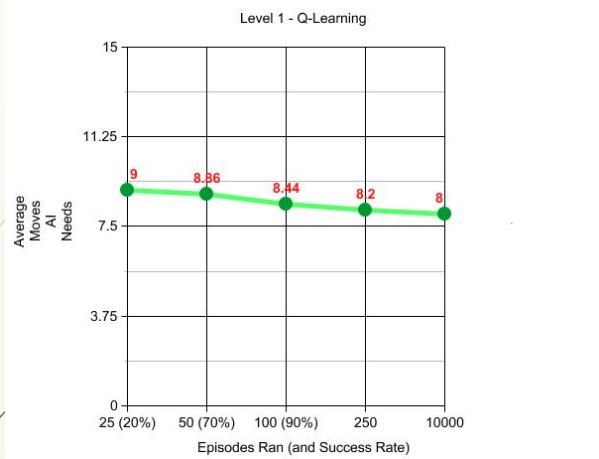
Funcionamento da Aplicação

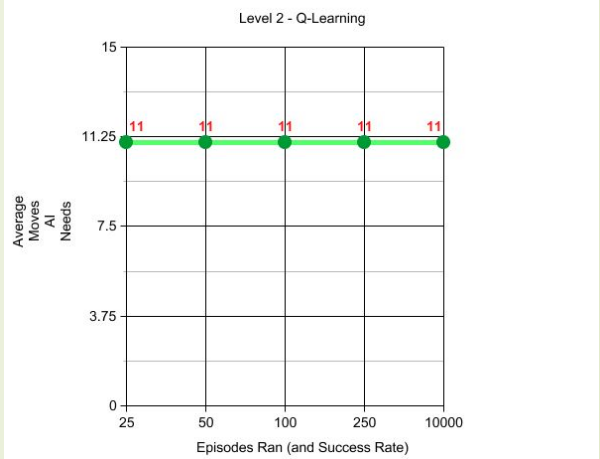
- Após compilar e executar o programa, é exibido um menu onde o utilizador tem três opções: jogar individualmente, utilizar algoritmos de aprendizagem por reforço ou sair.
- Caso escolha a segunda opção, poderá optar pelo Q-Learning ou pelo SARSA, definindo quantas iterações por nível o algoritmo terá para completar o mesmo (entre 25 e 10000 iterações).
- Os resultados do agente após o treino são exibidos.
- Mais detalhes podem ser encontrados no relatório.

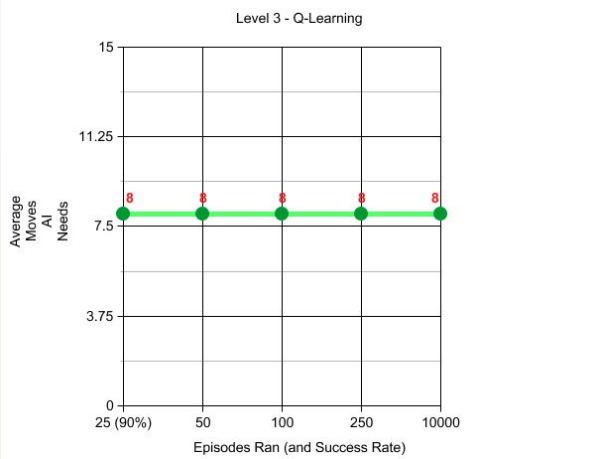
Resultados/Gráficos

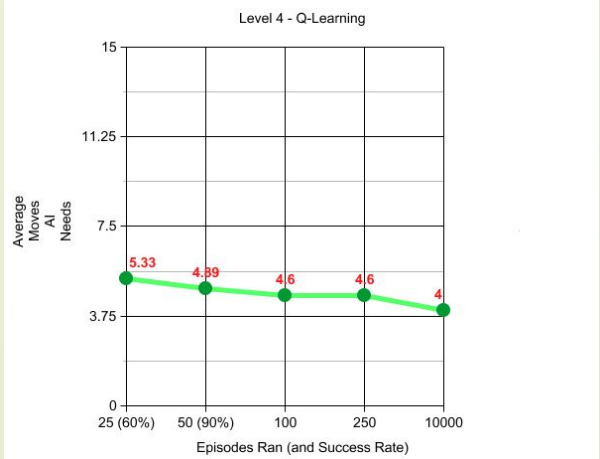
- Os seguintes gráficos mostram a obtenção do número médio de turnos necessários do agente para passar cada nível, após ter treinado durante um determinado número de episódios.
- Para obter a média, cada teste é corrido dez vezes.
- O eixo do X representa o número de episódios que o agente treinou, e a taxa de sucesso, isto é, as vezes que o agente consegue chegar ao final do nível com sucesso (se esta taxa é omitida, é 100%).
- O eixo do Y mostra a média de turnos necessários para passar o nível após o treino.

Resultados/Gráficos

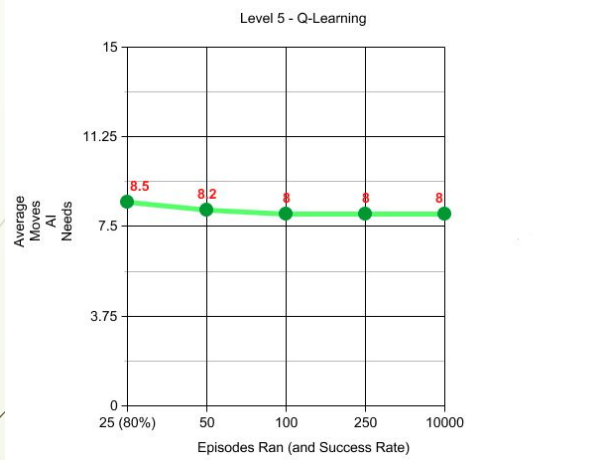


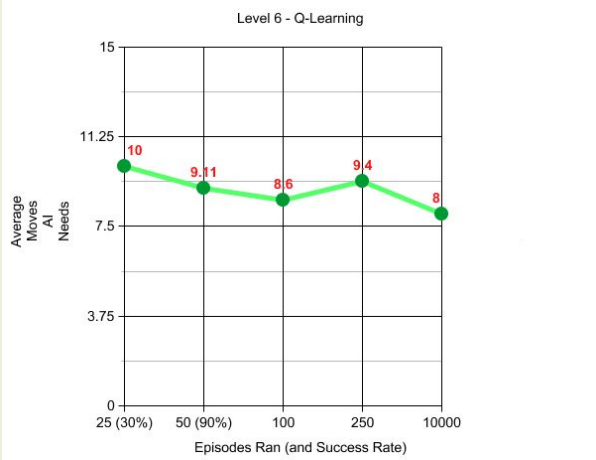


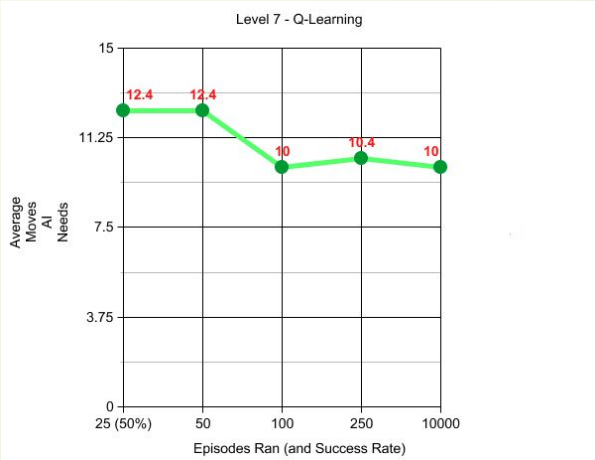




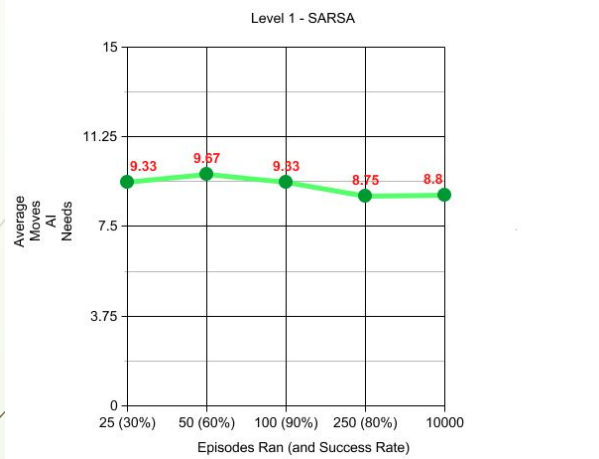
Resultados/Gráficos

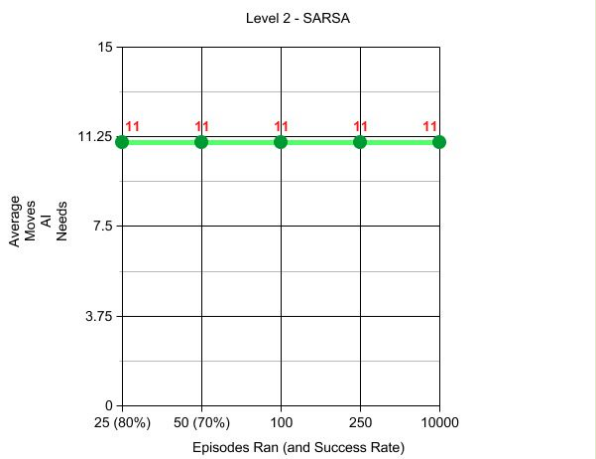


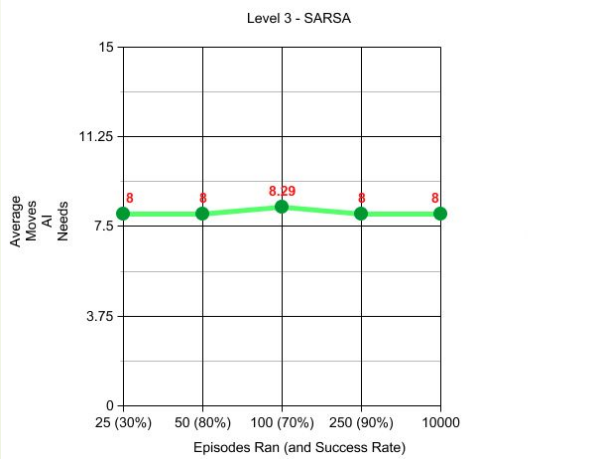


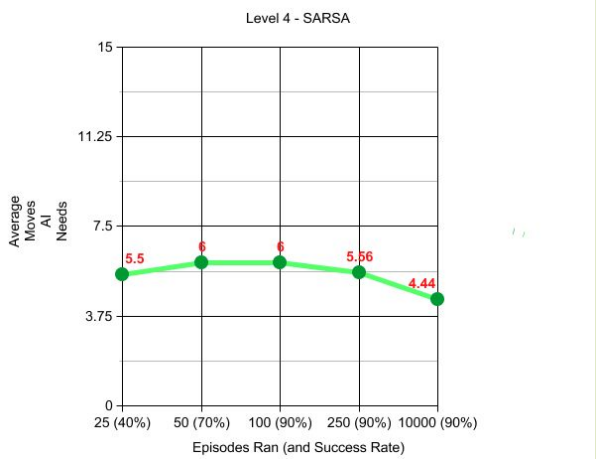


Resultados/Gráficos

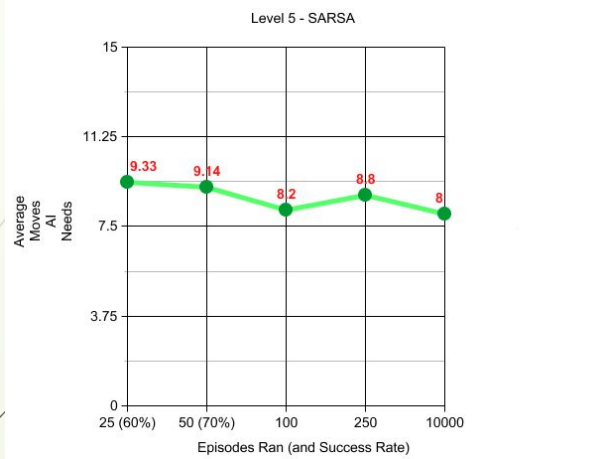


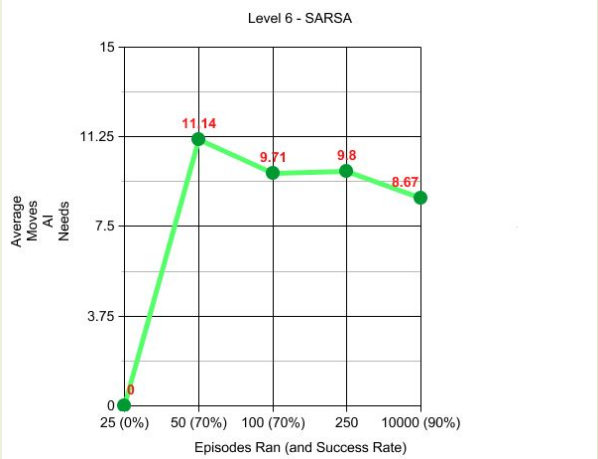


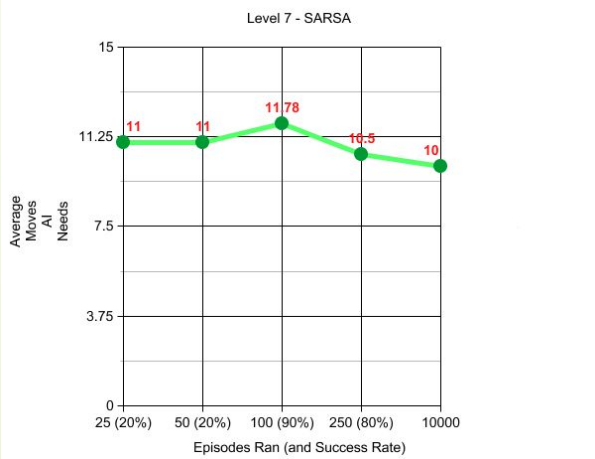




Resultados/Gráficos







Análise dos Resultados

- O algoritmo Q-Learning foi implementado com sucesso, produz bons resultados sem precisar de treinar durante muitos episódios, e quando-lhe é dado um treino 'longo', é capaz de resolver os níveis perfeitamente (no menor número de turnos possível).
- No entanto, tivemos dificuldades em implementar o algoritmo SARSA, e julgo que isto se mostra nos resultados, o número mínimo de turnos para passar o nível desce muito lentamente, não consegue passar os níveis de forma confiável, e muitas vezes não consegue chegar a resolver níveis perfeitamente mesmo quando ao agente é dado muito tempo para treinar.

Referências de Trabalho

- Alguns exemplos dos algoritmos Q-Learning: https://www.cse.unsw.edu.au/~cs9417ml/RL1/algorithms.html
- Machine Learning Notebook: https://github.com/rhiever/Data-Analysis-and-Machine-Learning-Projects/blob/master/example-data-science-notebook/Example%20Machine%20Learning%20Notebook.ipynb

Conclusões

- A utilização de algoritmos de aprendizagem por reforço fornece um ganho significativo no que diz respeito ao tempo gasto para se solucionar o problema, pois é possível, em apenas algumas iterações, chegar rapidamente a soluções válidas, formando rapidamente uma solução ideal. Sentimos dificuldades em implementar mais algoritmos por falta de tempo e menos um colega de grupo.