

DIGDUG

A I A G E N T

Licenciatura em Engenharia Informática

Inteligência Artificial
2023/2024

Rodrigo Graça, 107634
Pedro Rei, 107463
Tiago Cruz, 108615



Classes Utilizadas:

No student.py não foram utilizadas classes mas sim 5 funções muito importantes:

- `key_maker` - função que dá os estados e que define a key como o retorno da função `make_decision`
- `make_decision` - função que chama as 3 funções de estratégia.
- `enemy_stuck_strategy` - função que define a estratégia quando o inimigo está preso
- `camping_strategy` - função que dá a estratégia quando é vantajoso o Digdug acampar
- `default_strategy` - função que retorna a estratégia default de o Digdug ir atrás dos inimigos para os atacar

No classes.py foram criadas varias classes cada uma com o seu propósito, e com o objetivo de tomar a decisão do Digdug.

Algoritmo de Pesquisa:

Como método de pesquisa decidimos optar por uma abordagem mais simples, utilizando ifs para o Digdug tomar as suas decisões.

Pode-se dizer que separamos os ifs em 3 condições maiores que são diferentes estratégias de ataque aos inimigos.

Com isto quero dizer que temos estas condições principais:

- EnemyStuck - caso um inimigo esteja à muito tempo em menos de 3 posições, pode ser por estar preso ou por estar a fugir do Digdug, o Digdug percebe que se tem de aproximar mais dele para o poder atacar com sucesso
- Camping - caso o inimigo esteja numa sequencia de posições favoravel, o Digdug vai tentar acampar numa posição proxima de forma a facilitar o ataque
- Default - caso nenhum dos casos anteriores se verifique, o Digdug segue uma estratégia default de apenas ir atrás dos inimigos e ataca-los

Dentro de cada estratégia, podemos encontrar mais ifs de pesquisa, no entanto, usamos os ifs talvez não da forma mais convencional, em que se coloca a condição que será usada com mais frequência a ser verificada primeiro, mas sim por uma ordem de importância, priorizando pequenas decisões mas importantes para manter o Digdug vivo.

Ou seja, sabendo que por exemplo, enquanto o Digdug está se a aproximar dum inimigo, que com certeza é das condições que mais vezes ocorre, podem muitas vezes aparecer obstaculos como pedras, e essa seria uma condição com maior importância de verificar antes do que o facto de o Digdug já estar perto do inimigo ou não.

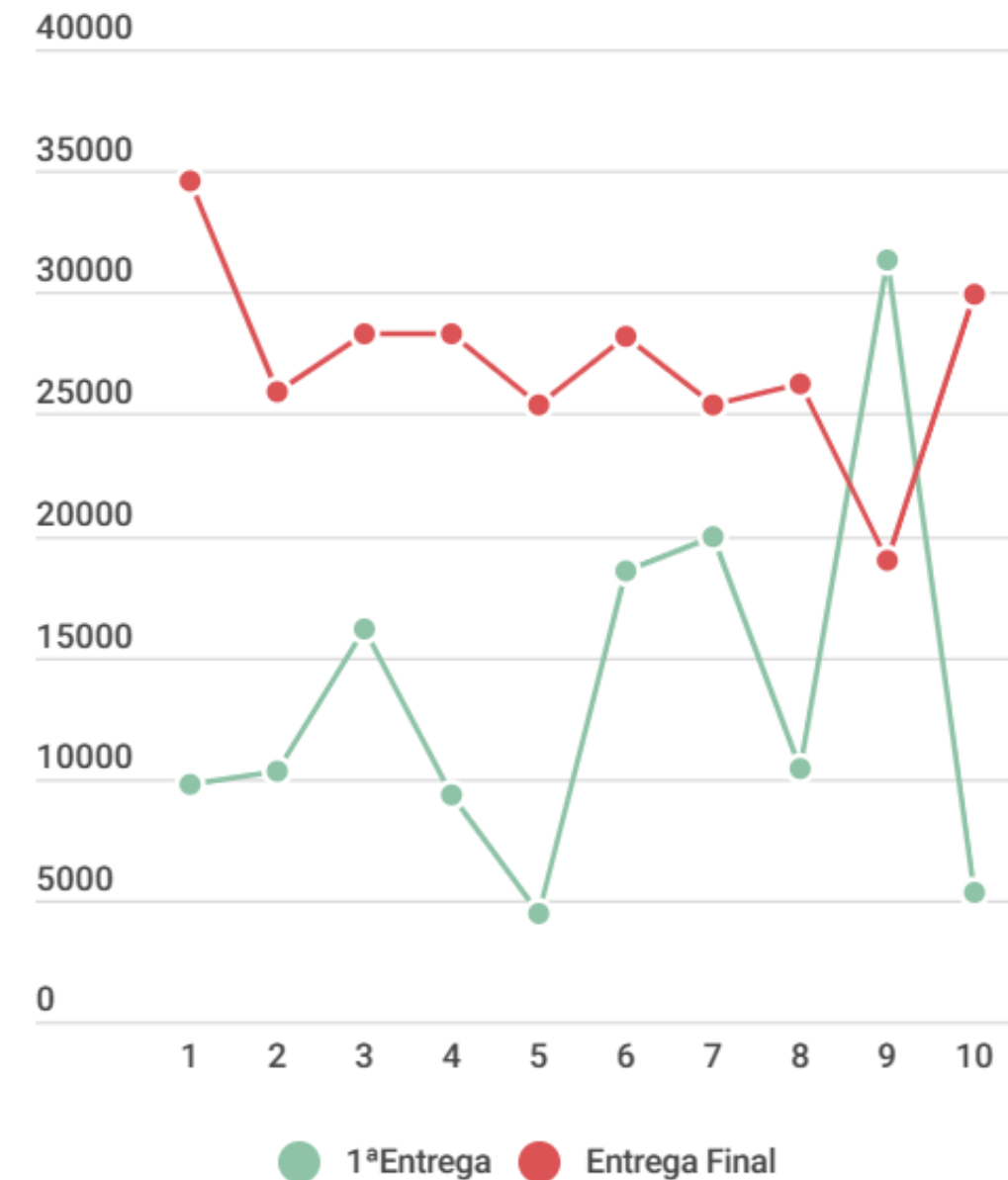
No nosso caso, verificar tanto as pedras, como inimigos wallpassing, ou com fogo, ou se estamos nas bordas, é uma prioridade em relação ao facto de nos termos de aproximar do inimigo.

Comparação:

Comparando o nosso AI Agent da primeira entrega com esta, pode-se dizer que houve uma notória evolução.

Antes a média dos scores era cerca de 13k, com um highscore de 31k, mas sendo um caso unico por conta de os resultados serão muito instaveis. Atualmente estamos com uma média calculado por nós de 25k, com resultados mais estaveis e superiores, e já obtivemos scores a mais de 40k.

Em relação ao que foi melhorado, na primeira entrega apenas tínhamos a estratégia default implementada e, nem todas as suas classes estavam tão aprimoradas como atualmente, por exemplos a Rock, que ainda era um problema para o Digdug. Enquanto para a entrega final implementamos novas estratégias, em especial a do Camping, que foi o que fez a nossa pontuação subir maioritariamente.



Teste de Entrega Final feito por nós, como forma de comparação



Conclusão:

Para o metodo de pesquisa implementado, os resultados estão proximos do esperado, apesar de ainda ser possivel melhorar algumas classes, com as estratégias utilizadas pensamos que não seria possivel obter um resultado muito melhor.

No entanto, sabemos que talvez com estratégias diferentes mas com o mesmo metodo conseguia-se um resultado bastante melhor. tiramos também a conclusão que há metodos de pesquisa melhores que os ifs e que mesmo com

Tiramos também a conclusão que nem sempre a abordagem mais simples é a mais efetiva, pois com outro tipo de metodo de pesquisa seria bem possivel obter-se uma maior pontuação.