IA- TPG - RUSH HOUR

Joaquim Andrade e Francisco Silva

UA 2022/23

Funcionamento ideal

Após a receção do state do nível um mapa é enviado para ser processado pela tree_search. A tree_search após descobrir a solução devolve uma lista de nós com as ações sobre peças (vetor e nome da peça) e mapas das ações necessárias para concluir o nível.

Após a conclusão do processamento, os states em atraso são então consumidos.

São então calculados todas as teclas necessárias para concluir o nível. Após calculadas são enviadas uma por uma até concluir o nível.

Caso Crazy Move

Caso aconteça um crazy move existem duas opções:

- ele é detetado através da comparação da grid teórica e a grid atual e ativa com sucesso a crazy move que calcula as teclas e desfaz os moves.
- ou ele é detetado, e, como pode acontecer mais do que um crazy move durante o processamento da solução ou em situações adversas, ele simplesmente refaz a solução. Este é o pior caso.

Classe Cursor

A classe cursor é iniciada com a posição do cursor do jogo e transforma um movimento de uma peça e um mapa nas teclas que serão fornecidas para efetuar o movimento e a translação da peça. Embora a posição do cursor seja atualizada, não está necessariamente igual ao cursor do jogo.

Tree_search e heurística

- O algoritmo de pesquisa é baseado no modelo que desenvolvemos nas aulas práticas.
- Para a encontrar uma solução utilizámos uma estratégia “greedy” com a heurística “blocking cars”. Esta heurística consiste em calcular o número de carros que se encontram em posição vertical e que estão a bloquear o carro A. Desta forma a heurística privilegia estados em que existem menos ou zero carros a bloquear a saída ao carro A, estando em teoria mais próximo de encontrar uma solução, reduzindo o número de nós expandidos e por consequente o tempo de pesquisa.

Justificação das decisões

As decisões tomadas embora tenham conseguido concluir a totalidade dos níveis com sucesso, eventualmente poderão ter ficado aquém dos nossos colegas. A falta de eficiência do tree_search pode ser um dos motivos, no entanto, penso que a originalidade do código e da aplicação do mesmo compensem o sucedido.

Poderiam existir alguns melhoramentos na função crazy_back e muitos melhoramentos na tree_search(eliminação de classes etc), no entanto, a carga de trabalho das outras cadeiras, bem como a conclusão de todos os níveis do jogo impediram tais mudanças.