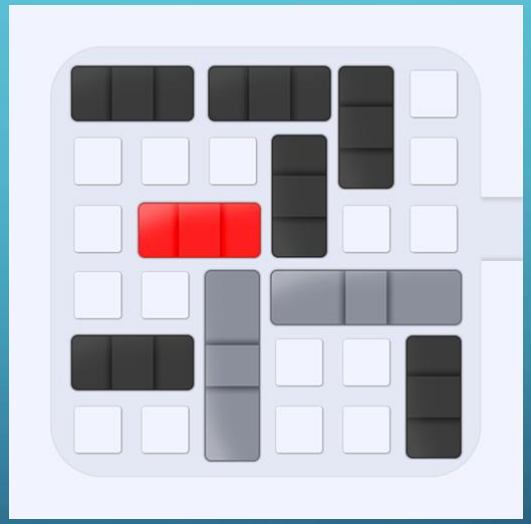
DESENVOLVIMENTO DE UM AGENTE AUTÓNOMO PARA O JOGO RUSH HOUR

João Ferreira - 103625



DETI – Universidade de Aveiro

Inteligência Artificial

Prof. Diogo Gomes

Prof. Luís Lopes

CLASSES DA TREE SEARCH

SearchDomain

Contém os métodos necessários para implementar um domínio. Todos os métodos desta classe são abstratos e serão mais tarde implementados no ficheiro domain.py.

SearchProblem

Esta classe é capaz de criar um problema, recebendo como argumento o domínio em questão e o ponto de partida.

SearchNode

Cria e implementa os nós necessários e específicos para a nossa árvore de pesquisa (SearchTree).

SearchTree

Cria e implementa a nossa árvore de pesquisa utilizando, no caso, a estratégia de pesquisa denominada por "**greedy**".

RushHourDomain

_actions(): Recebe um estado como argumento. Retorna uma lista de ações.

_results(): Dado um estado e uma ação, retorna o novo estado resultante.

_cost(): Define o custo de cada ação como 1.

_heuristic(): Dado um estado, calcula a distância entre o "nosso" carro e a distância ao goal.

Funções auxiliares

_do_action()	_same_coords()	_move_cursor_to_car()
• Faz a conecção entre as funções "same_coords" e "move_cursor_to_car" de forma a completar uma ação.	• Verifica se o cursor está nas mesmas coordenadas da peça que se pretende mover.	• Caso na função "same_coords" se verifique que o cursor não está nas mesmas coordenas do carro, esta função irá colocá-lo nessas coordenadas.

OBSERVAÇÕES

- Poderíamos ter feito algumas otimizações como, por exemplo, criar uma função que tivesse como objetivo fazer a verificação do crazy step e dessa forma, obrigar a parar as ações que estão programadas para acontecer e recalculá-las, tendo em conta esse movimento inesperado.