

Desenvolvimento de um Agente Autónomo para o jogo **Sokoban**

Inteligência Artificial / Introdução à Inteligência Artificial

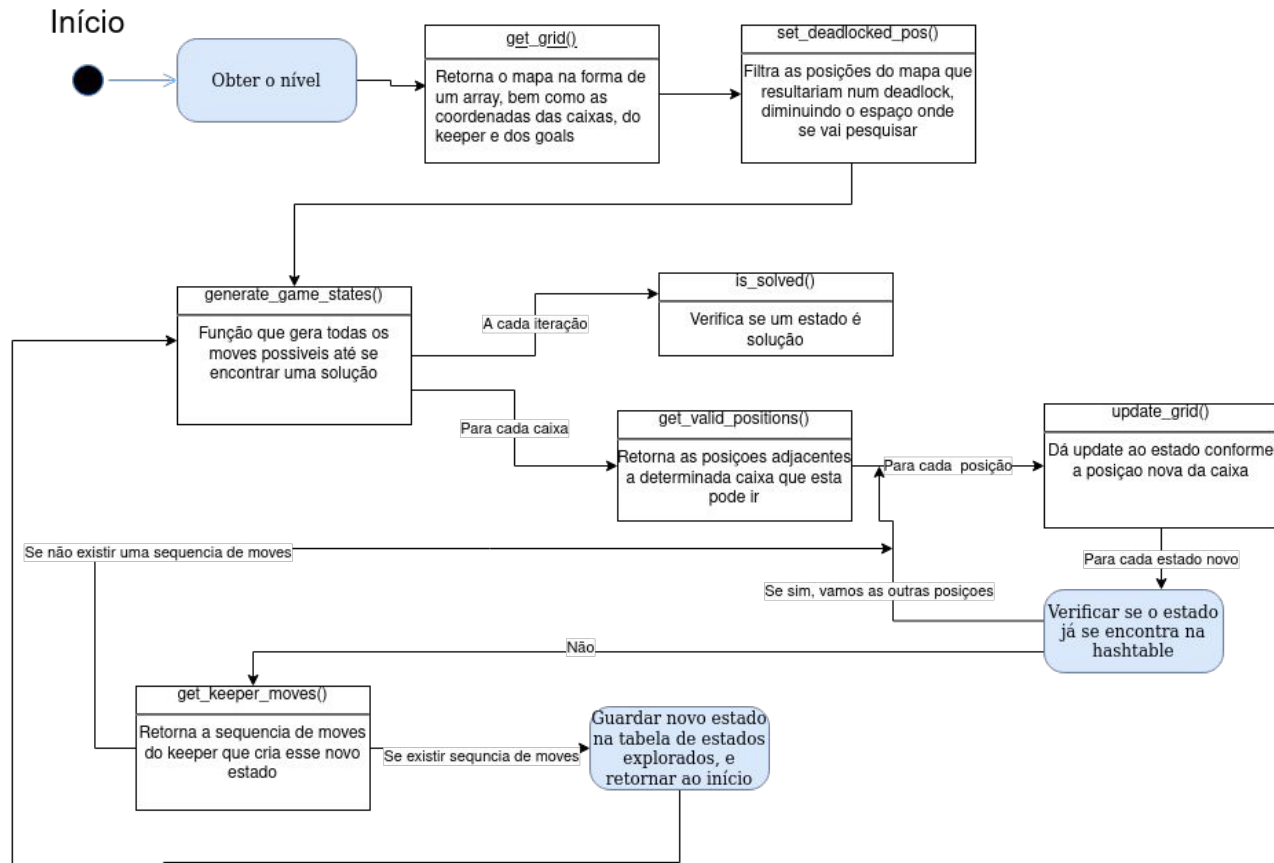
Afonso Domingos Cardoso - 88964

Tiago Alexandre Magalhães de Barros - 88963

Ano Letivo de 2020/2021



Descrição do algoritmo



Notas relevantes:

Na função `set_deadlocked_pos()`, quando testamos se existe um caminho desde uma coordenada até um goal, não nos interessa caminhos ótimos, pelo que usamos o algoritmo Breadth First, pois usa menos memória que o A*, e a velocidade para achar uma solução para sequências pequenas de movimentos é praticamente igual.

Para a função `get_keeper_moves()`, como os movimentos gerados podem ser os da solução final, é usado o algoritmo A* , para que se encontre um caminho ótimo para os empurrões nas caixas.

Elemento	Símbolo	Número
Floor	-	0
Wall	#	1
Goal	.	2
Box	\$	3
Man	@	4
Box_On_Goal	*	5
Man_On_Goal	+	6
Deadlocked	-----	9
Keeper_On_Deadlocked	-----	10

Exemplo de um array bidimensional de estado

Antes de ser filtrado:

```
[[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1],  
[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],  
[1, 2, 0, 0, 0, 2, 0, 0, 4, 1, 2, 1],  
[1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1],  
[1, 1, 0, 3, 0, 0, 1, 0, 0, 0, 0, 1],  
[0, 1, 0, 3, 0, 0, 0, 1, 1, 1, 1, 1],  
[0, 1, 0, 3, 1, 0, 0, 1],  
[0, 1, 1, 0, 1, 0, 0, 1],  
[0, 0, 1, 0, 0, 0, 1, 1],  
[0, 0, 1, 1, 1, 1, 1]]
```

Depois de ser filtrado:

```
[[9, 9, 9, 9, 9, 9, 9, 1, 1, 1, 1, 1],  
[1, 1, 1, 1, 1, 1, 1, 1, 9, 9, 9, 1],  
[1, 2, 0, 0, 0, 2, 0, 0, 4, 1, 2, 1],  
[1, 9, 0, 1, 1, 1, 9, 0, 0, 0, 0, 1],  
[1, 1, 0, 3, 0, 9, 1, 9, 9, 9, 9, 1],  
[9, 1, 0, 3, 0, 0, 9, 1, 1, 1, 1, 1],  
[9, 1, 9, 3, 1, 0, 9, 1],  
[9, 1, 1, 0, 1, 0, 9, 1],  
[9, 9, 1, 9, 9, 9, 1, 1],  
[9, 9, 1, 1, 1, 1, 1]]
```

Mapa: Nível 100



Observações

1º entrega - até ao nível 117

2º entrega - até ao nível 135

Poderíamos ter feito algumas otimizações como: a detecção de túneis, onde apenas é possível 1 movimento, pois nesses casos, ou se faz o movimento até a sua extensão máxima, ou não, e fazer o cálculo da heurística para cada estado gerado, a fim de verificar se determinado estado está mais próximo da solução.