

Desenvolvimento de um Agente autónomo para o jogo

Sokoban



Inteligência Artificial - 2020/2021

Diogo Gomes

Luís Seabra Lopes

90711 - Camila Uachave

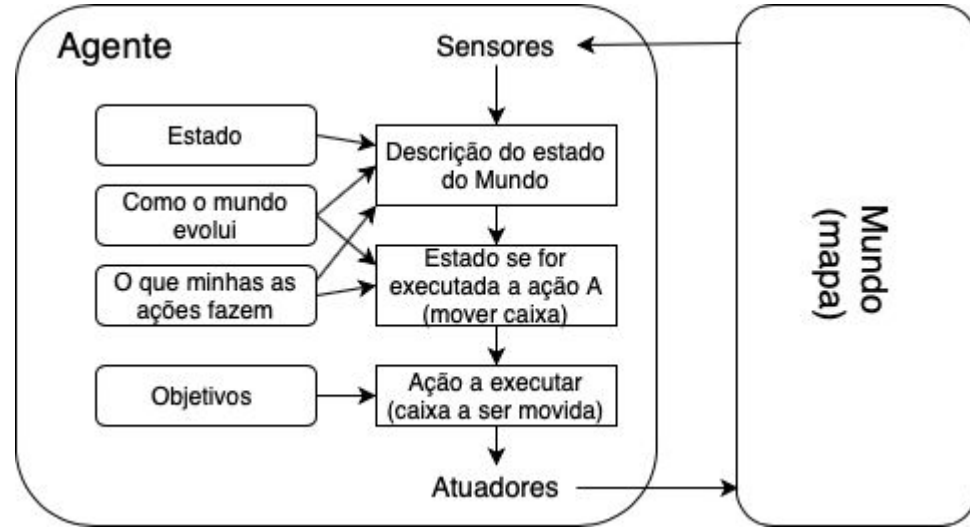
91322 - Isadora Loredó

92972 - Gonçalo Matos

Arquitetura do Agente

O agente é do tipo deliberativo, orientado por objetivos (colocar as caixas sobre os diamantes), baseado em quão longe está do seu objetivo:

- Calcula ações com o propósito de reduzir a distância até o objetivo;
- Exige pesquisa e planejamento;
- Este tipo de agente é mais flexível ao que diz respeito alcançar destinos diferentes;
- Ao especificar um novo destino, podemos fazer com que este tipo de agente apresente um novo comportamento.



Estratégias de pesquisa

1. **Árvore de pesquisa das caixas**

1.1. **Greedy**

Este método consiste na expansão do nó mais próximo do objetivo, sendo esta proximidade definida pela heurística. Para o caso particular da árvore de pesquisa das caixas, a nossa heurística é calculada tendo em conta o diamante mais próximo da caixa que está a ser avaliada.

2. **Árvore de pesquisa do agente**

2.1. **Greedy**

Faz uso do mesmo método da anterior. Contudo, calcula a heurística de uma forma diferente: por meio da distância de Manhattan entre a posição atual do agente e o objetivo que o mesmo deve alcançar.

Deadlocks

1. Na geração de ações do domínio das caixas:

- 1.1. Validação para verificar se a próxima posição da caixa a coloca por cima de uma parede ou caixa.
- 1.2. Validação para verificar se o próxima posição da caixa a leva para uma parede (*dead end*):
 - 1.2.1. Avaliamos se o movimento da caixa leva a uma esquina, verificando se a posição de cima (ou baixo) e da esquerda ou direita da próxima posição é uma parede.

2. Na geração de ações do domínio do agente:

- 2.1. Validação para verificar se o movimento do agente o leva para cima de uma parede ou caixa.

3. Na avaliação do resultado de uma ação no domínio das caixas,

antes de chamarmos a árvore de pesquisa do agente, temos de confirmar se o mesmo já não se encontra no estado de objectivo (goal state), caso não, verificamos se:

- 3.1. O goal não é uma parede;
- 3.2. O goal não é posição de alguma caixa;
- 3.3. Se o goal é acessível, se o mesmo não está em algum beco sem saída.

Otimização

1. RecoverSolutions

Esta otimização na árvore de pesquisa permite-nos recuperar os nós expandidos e descartados em pesquisas anteriores e é utilizada numa pesquisa dinâmica com uma profundidade inicial mais reduzida que é progressivamente incrementada caso não chegue a uma solução, até um limite definido.

2. Threshold

Tanto a árvore de pesquisa do agente como a das caixas recebem um parâmetro que corresponde ao limite de profundidade que a árvore pode expandir, em cada nível esse limite varia, sendo assim a atribuição foi feita de forma a esse limite crescer dinamicamente até atingir o threshold definido inicialmente.

3. GraphSearch

Através do graph search na árvore de pesquisa do agente e das caixas foi possível verificar se um novo nó já havia sido expandido anteriormente (se o mesmo fizer parte da lista nós fechados) , e isso leva a pesquisa a ser mais eficiente.