universidade de aveiro

Snake

Introdução à Inteligência Artificial

Diogo Ferreira 76504

Luís Leira 76514

universidade de aveiro

Jump Point Search (I)

➥ Algoritmo de pesquisa baseado em *Jump Point Search (JPS)*

A implementação baseia-se no algoritmo *JPS* para calcular o caminho para a comida. Este algoritmo é considerado uma otimização do algoritmo de pesquisa *A*,* com uma melhoria significativa de desempenho e rapidez de cálculo.

➥ Variações aplicadas ao *JPS (I)*

Após efetuarmos a divisão do mapa em 4 ou 16 quadrados (dependendo da dimensão do mapa), tentamos obter um caminho para a comida. Quando o algoritmo terminar com sucesso, retorna os pontos de salto necessários para chegar ao objetivo. Assim sendo, viajamos para os saltos heuristicamente de acordo com as nossas regras (descritas nos slides seguintes).

universidade de aveiro

Jump Point Search (II)

➥ Variações aplicadas ao *JPS (II)*

Se o algoritmo *JPS* detetar que está perto de exceder o tempo limite do agente calcular o caminho, guarda o seu estado atual e interrompe a sua execução, podendo ser retomada na ronda seguinte.

O algoritmo apenas é executado após todo o processamento da próxima direção, para garantir assim tempos de saída do algoritmo mais precisos. Assim sendo, o resultado do *JPS* apenas é aproveitado na ronda a seguir à ronda onde foi calculado.

Quando a comida muda de quadrado, caso a cobra esteja a três ou mais saltos do quadrado anterior, são mantidos os saltos que temos. Se estiver a menos do que três saltos, calculamos novamente o caminho através do *JPS*. Assim, evitamos situações onde estamos demasiado longe da comida e estamos sempre a voltar atrás para seguir os saltos porque a comida se moveu de quadrado.

Camada Reativa (I)

➥ Heurística baseada na distância de *Manhattan*

A primeira regra para determinar a próxima direção é a heurística, caso essa direção não viole nenhuma das condições seguintes.

➥ *Chicken Condition*

A cobra tem a capacidade de prever (heuristicamente) o próximo movimento da outra cobra. Quando esse movimento puder ser igual ao nosso e nos encontramos com igual ou inferior pontuação, a nossa cobra opta por escolher outra direção válida para evitar qualquer choque.

Camada Reativa (II)

➥ Forçar o choque de cabeça com cabeça

Pelo contrário, quando a nossa cobra deteta que o próximo movimento da outra cobra está ao alcance e, se a nossa cobra possuir maior pontuação do que a outra, tenta forçar o choque de cabeça com cabeça para precipitar o resultado do jogo.

➥ Encurralar cobra adversária

Caso a nossa cobra detete que está ao lado e à frente da outra cobra, e que a outra cobra está ao lado de um obstáculo, altera o seu caminho para a frente da outra cobra, obrigando-a a embater contra a cobra ou um obstáculo.

Camada Reativa (III)

➥ Deteção de caminhos sem saída

A cobra tem uma funcionalidade que permite detetar se um determinado caminho tem saída, verificando assim a viabilidade desse caminho, de forma a evitar uma situação de encurralamento/morte.

➥ Deteção de *loops*

É possível detetar quando a cobra entra num ciclo, permitindo assim optar por uma solução que permita quebrar o mesmo. Essa solução passa por contornar os obstáculos existentes na sua proximidade, e passado algumas rondas voltar a tentar calcular o caminho para a comida da sua posição atual.