

A-Constelação: Extensões do A* para aperfeiçoar a I.A. de NPCs em Jogos

Fernando Santos Osório, Sandro Ferreira, Vinicius Nonnenmacher

INTRODUÇÃO

Em jogos digitais complexos, o uso de Inteligência Artificial (I.A.) se faz necessário em vários momentos. Um deles é quando se faz necessária a solução do problema da busca de caminhos (*pathfinding*), ou seja, quando aplicamos um algoritmo de busca, que permita encontrar caminhos, dado um ponto inicial (origem) e um final (destino). Este problema pertence a uma área da I.A. denominada de busca em espaço de estados.

O algoritmo A* (A Star), é um dos algoritmos de *pathfinding* mais utilizados em jogos. O A* permite que sejam traçadas rotas baseadas em uma representação de grid (ocupação espacial, com custos de transposição para cada elemento da matriz), bem como em representações de grafos com custos associados às arestas que unem os vértices deste grafo.

DEFININDO REGIÕES IMPORTANTES

Definir regiões importantes é muito interessante quando se joga um jogo de estratégia. O agente deve ser capaz de analisar um caminho que é comum a outros agentes, e desse modo explorar esta propriedade.

Supondo o uso do A* padrão, um agente poderá analisar os terrenos em que está passando, e verificar se há uma área de baixo custo em meio a uma grande área de alto custo, como por exemplo, um vale em meio à montanhas. Para efeitos de um jogo, pode-se supor que aquela área é um caminho perigoso (ideal para emboscadas), por que vários agentes irão considerar aquele caminho como sendo um caminho ótimo (de baixo custo).

FUGA

Em um sistema multi-agentes muitas vezes é necessário criar uma estratégia para cercar uma determinada área importante, como por exemplo, a base dos inimigos, ou mesmo, um inimigo específico (e.g. predadores que cercam uma presa). Para se fixar o destino de cada agente, é definido um círculo ao redor do alvo a ser cercado. Este círculo será dividido, em partes iguais de modo que os agentes fiquem distribuídos ao redor do alvo. O raio será dado pela abrangência da área ao redor do alvo, exceto em casos mais específicos onde se deseja chegar a uma distância específica do ponto-alvo.

Para que os agentes cerquem o alvo vindos de diferentes regiões, é feito um novo círculo ainda maior da região alvo. Primeiramente a região alvo é dividida em quadrantes, e utilizando-se destes mesmos quadrantes podemos atribuir custos para distribuir o envio de agentes para cada um deles. Assim podemos aumentar o custo (camada dinâmica) para os quadrantes onde não se quer que um ou mais agentes passem. Desse modo o agente não passará pelos quadrantes que não foram atribuídos a ele, indo diretamente para o quadrante que lhe foi especificado, e desta forma cercando o alvo.

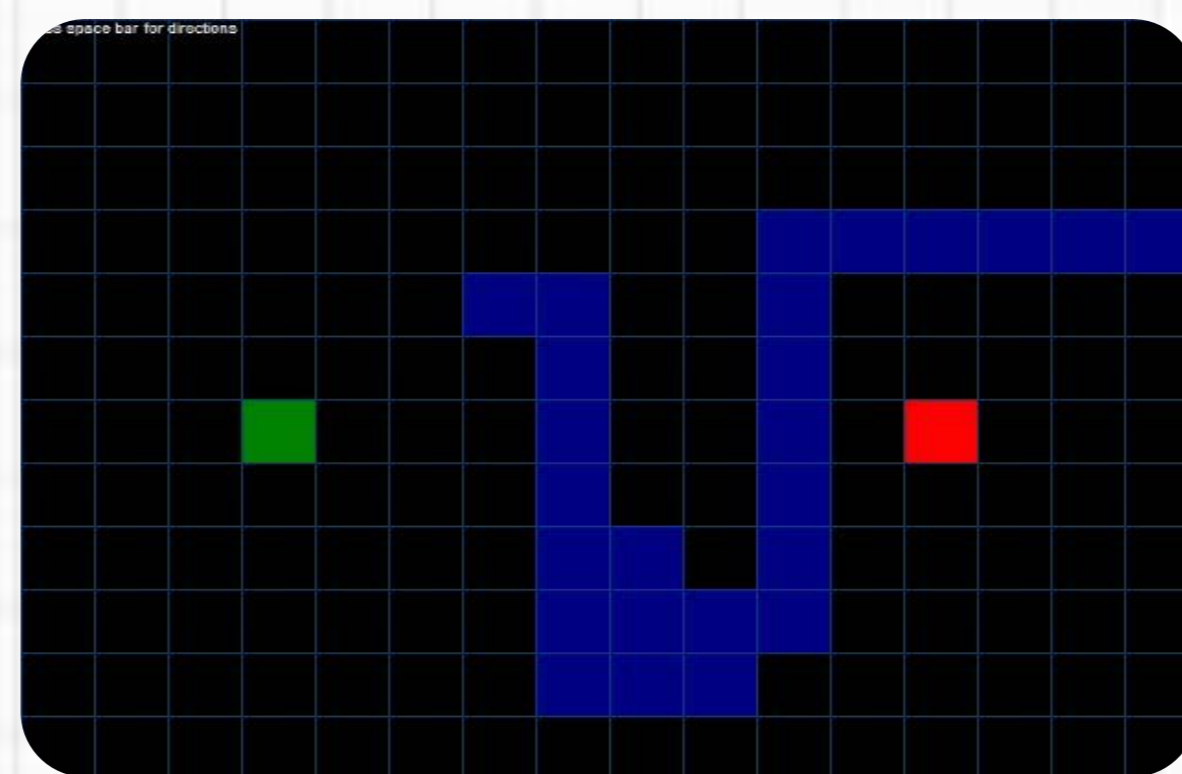


Figura 1: A*

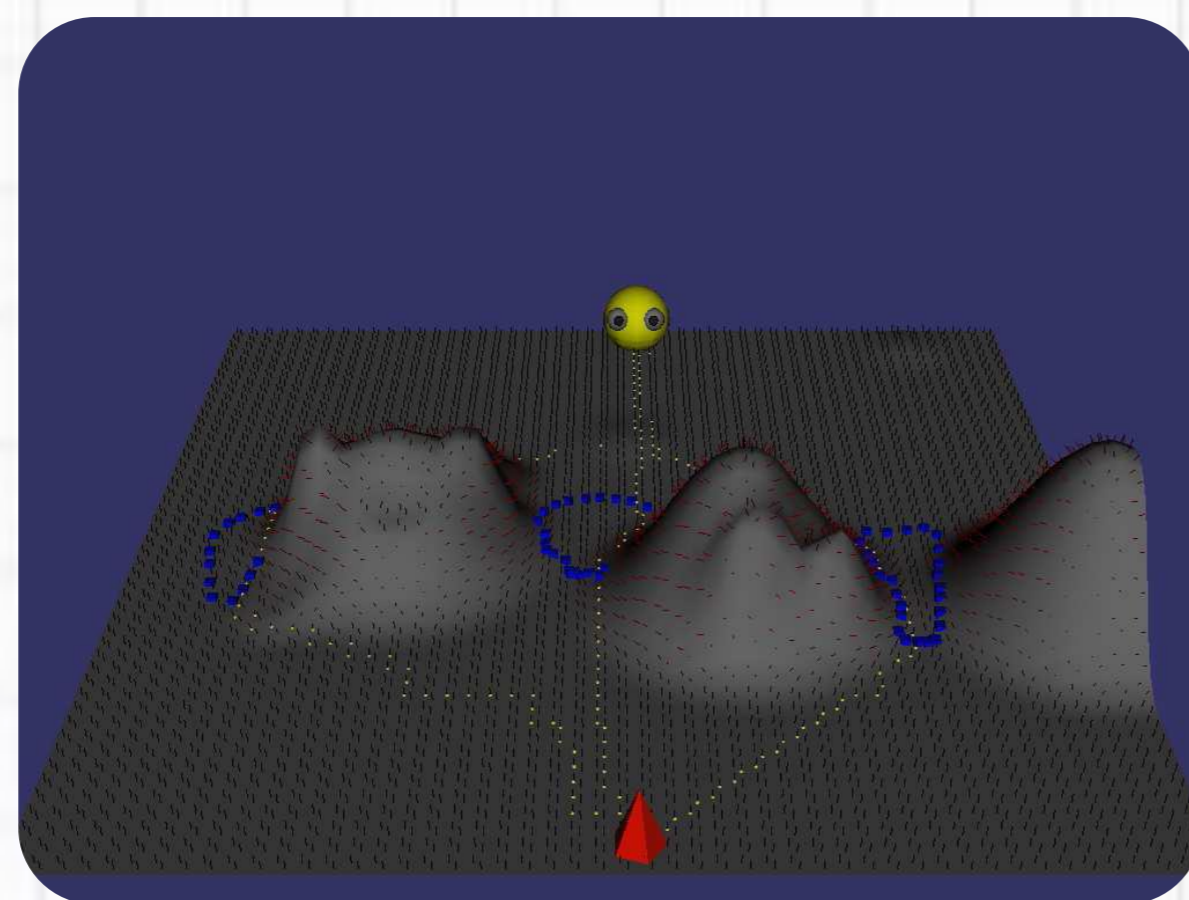


Figura 2: Definindo Regiões Importantes

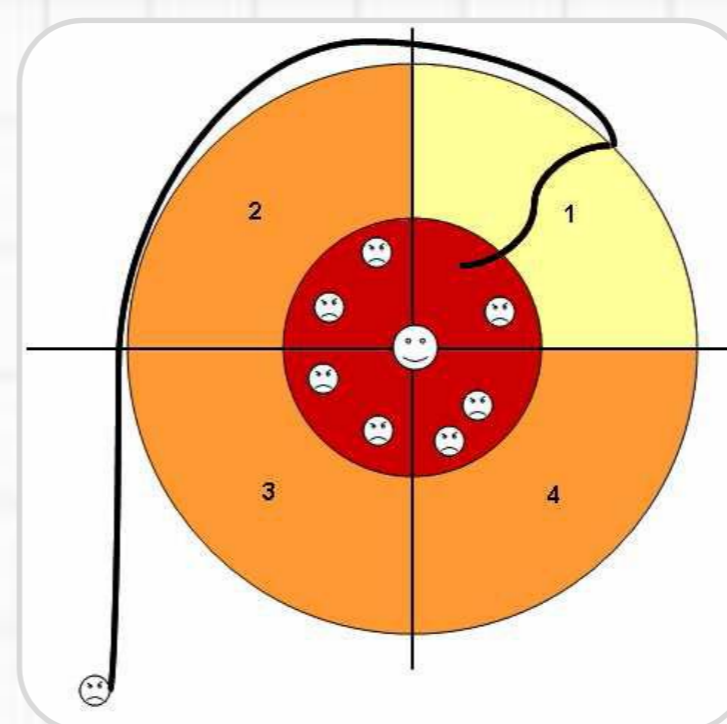


Figura 3: Alvo

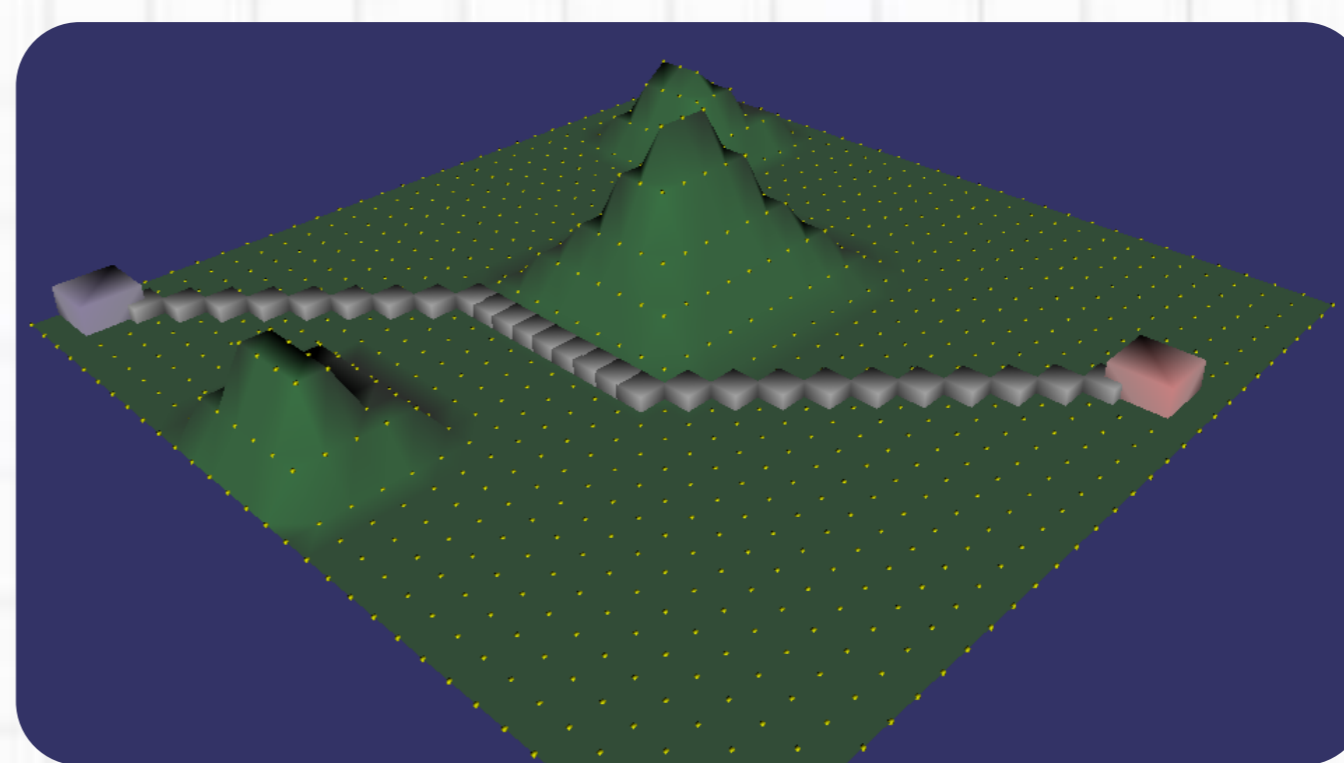


Figura 4: A* Em Terrenos 3D

A* EM TERRENOS 3D

A análise do terreno é uma grande ferramenta que pode ser usada para gerar dados que um algoritmo de encontrar caminho (*pathfinding*) de alto nível usa para encontrar caminhos interessantes. Para que isso seja possível o mais trivial é obter a inclinação dos polígonos (triângulos da malha) que definem o terreno, obtendo assim o vetor normal dos mesmos. Para isto, é feita uma análise prévia da malha (*mesh*) do terreno. Dada a normal dos polígonos, podemos estimar o grau de inclinação do terreno (individual ou média de uma certa região ou grupo de polígonos). Esta inclinação indicará o custo de locomoção, diretamente relacionado a inclinação do terreno no ponto em questão.

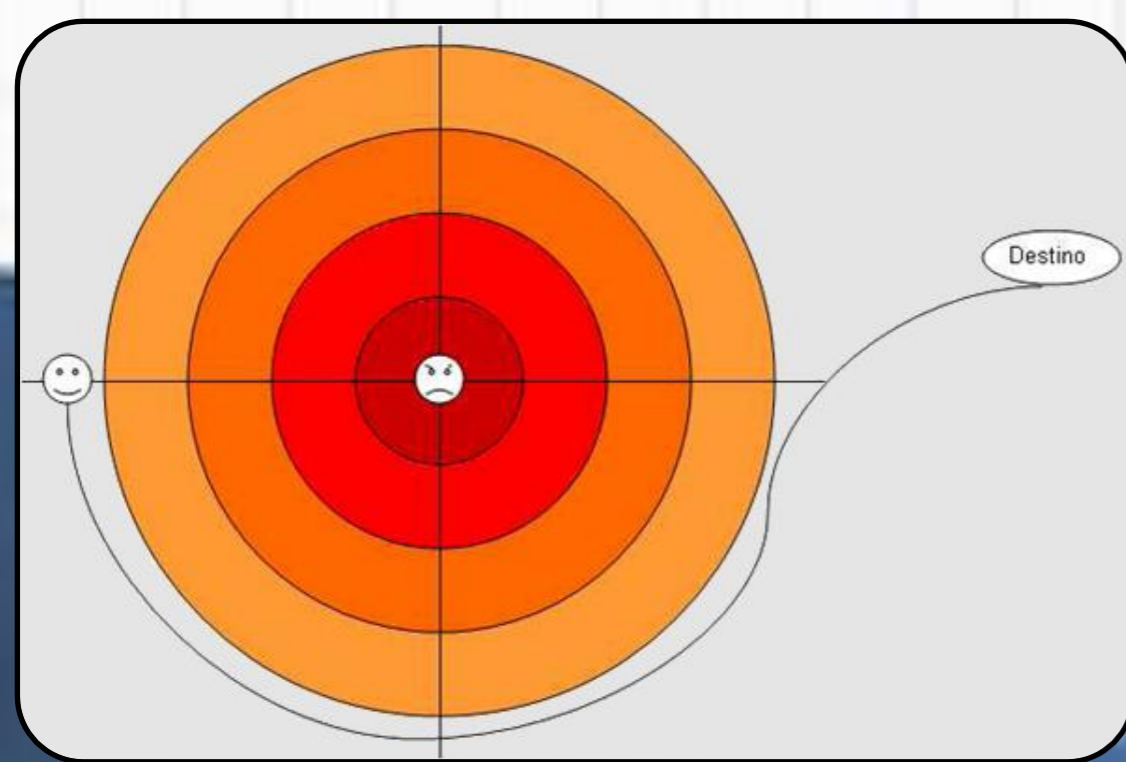


Figura 5: Fuga

OBJETIVO

O objetivo deste trabalho é, descrever técnicas e propostas baseadas no A* que permitam aos agentes inteligentes de jogos (NPCs) obter resultados bastante satisfatórios em algumas situações específicas que requerem o uso de algoritmos de *pathfinding*.

ALVO

O primeiro passo para obter um comportamento de fuga é definir prioridades em relação ao caminho a ser seguido, e também o caminho a ser evitado. Inicialmente devemos definir o ponto de destino do agente: a solução mais simples e rápida é determinar o ponto mais afastado (em linha reta) em relação ao(s) predador(es). Após ser definido o destino, a próxima prioridade é não passar por perto do(s) perseguidor(es).

Para que isso aconteça, devemos aumentar o custo de todos os nós que estejam próximos a um dos inimigos. Desta forma iremos criar uma camada dinâmica (*layer*) com custos que serão sobrepostos sobre os custos do mapa do ambiente, somando assim os custos referentes aos obstáculos estáticos definidos no mapa do ambiente original, junto com os custos que são determinados de modo dinâmico e que possuem uma relação direta com a proximidade em relação aos inimigos.