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

Vinícius Nonnenmacher Sandro Ferreira Fernando Osório

Universidade do Vale do Rios dos Sinos,
PIPCA – PPG de Computação Aplicada, GT-JEDi – Graduação em Jogos Digitais,
São Leopoldo, RS - Brasil

Abstract

The goal of this paper is to discuss about the use of the A* (A-Star) Search Algorithm in digital games, used to control NPCs (Intelligent autonomous agents). The A* algorithm has been used to find paths from one starting point to a destination point. In this paper we are proposing the A-Constellation which is based on the A-Star algorithm and allowed us to implement: different evasive and cooperative pursuit behaviors (pursuit-evade games), and also A* applied to 3D terrains. These new techniques are described in the paper.

Resumo

Este artigo tem como objetivo discutir sobre a utilização do algoritmo de procura (*pathfind*) A* (A-Star) em jogos digitais, usado para controlar NPCs (Agentes Autônomos Inteligentes – *Non Player Characters*). O algoritmo A* tem sido usado para encontrar caminhos de um ponto inicial até um destino. Neste artigo estamos propondo um conjunto de adaptações ao A* que nos permitem implementar: comportamentos de fuga e de perseguição cooperativa, assim como aplicar o A* na definição de rotas em terrenos 3D. Estas novas técnicas são descritas neste artigo.

Keywords: inteligência artificial, NPCs, agentes cooperativos, algoritmos de procura (*pathfinding*).

Authors' contact:
{vnonnenmacher, sandro.s.ferreira, fosorio} @ gmail .com

1. 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 [Russell & Norvig 95, Winston 92].

O uso de algoritmos de busca de caminhos geralmente pode adotar critérios de otimização e/ou heurísticas de busca, onde muitas vezes é necessário que o agente chegue ao ponto final: o mais rápido possível; percorrendo a menor distância; minimizando custos (e.g. combustível, dinheiro, equipamento); etc.

O algoritmo A* (A Star) [Russell & Norvig 95, Rabin 2002, cap.3], 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. Em sistemas com múltiplos agentes o objetivo de cada agente pode ser compartilhado por todos (capturar uma presa), isto pode implicar que as rotas sejam planejadas em conjunto, para que um resultado coletivo possa ser alcançado. Neste caso o A* deve ser adaptado para este fim, pois trata as rotas individualmente. Além de uma cooperação na perseguição, um agente pode buscar uma rota de fuga, onde deve evitar de se aproximar dos demais agentes que lhe perseguem (predadores). Também podemos destacar que o A* pode tirar proveito das rotas traçadas para definir estratégias de jogo (rotas a usar ou a evitar) em ambientes 3D.

O objetivo desse artigo é, portanto, 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.*

1.1 Algoritmo A* em Jogos

O A* (A Star) [Deloura 2000, Lester 2004, Lester 2007] é um dos algoritmos mais populares de *pathfinding* para jogos, sendo bastante flexível e podendo ser usado em uma grande variedade de contextos, apenas mudando a heurística básica (funções de custo) que auxilia na exploração e busca de trajetórias. Podemos associar/alterar os custos da matriz que descreve a ocupação do terreno e assim forçar um desvio na trajetória de um NPC. Nas próximas sessões iremos demonstrar exemplos práticos de como alterar as funções heurísticas e de custos para casos específicos, como fuga, perseguições e outros.

2. A* para Agentes Individuais

Nesta sessão apresentamos extensões do algoritmo A* em jogos, para agente individuais, obtendo assim um comportamento individual, como fuga ou perseguição.

2.1 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 (Fig. 1). 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).

Uma forma usada para encontrar essas "áreas de baixo custo" é executar o algoritmo várias vezes de lugares diferentes, porém com o mesmo destino, e analisar quais os nós que foram escolhidos pelo algoritmo A*, por onde os agentes irão passar mais seguidamente. Esta informação poderá ser usada do ponto de vista estratégico para o jogo, e não apenas para a movimentação dos agentes. Outro modo de definir regiões de possível tráfego intenso de agentes é executar o algoritmo várias vezes estabelecendo como origem um mesmo lugar e como destino também um mesmo ponto. Porém, para cada rota definida, os caminhos já percorridos passam a ter um custo muito alto. Assim ao final da geração de trajetórias teremos várias áreas marcadas sobre o mapa, e poderemos agrupar estas regiões como sendo regiões de alto tráfego e de relativo baixo custo. Caso essas áreas fiquem muito dispersas, constata-se então que nestas regiões existem diversas alternativas (rotas de fuga) possíveis.

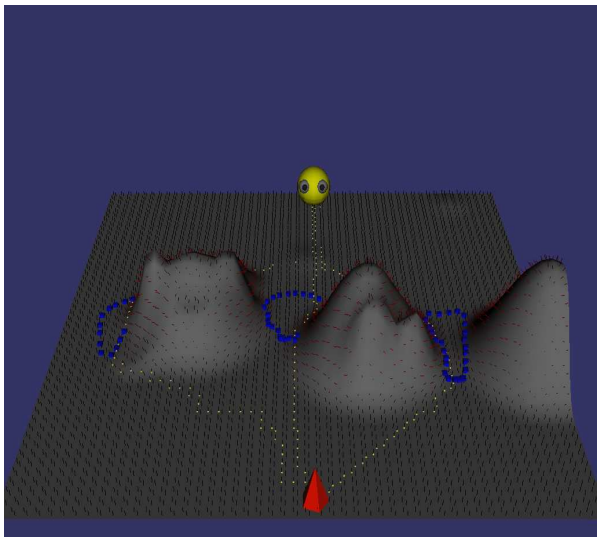

Figura 1: A-Constellation: Regiões de fluxo de agentes - Zonas tracejadas indicam as regiões de maior tráfego (vales).

Para que esta tarefa não fique muito cara computacionalmente, deve-se dar essa missão de mapear as regiões para um único agente ou para um grupo bastante limitado de agentes. Esses agentes fariam parte de um time de agentes exploradores, comuns em jogos de estratégia.

2.2 Fuga

O primeiro passo para obter um comportamento de fuga é definir prioridades em relação ao caminho a ser seguido, mas 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.

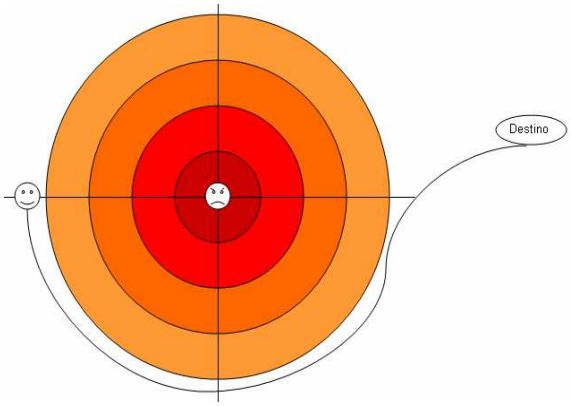

Figura 2: Área de influência e trajeto do agente

Cada inimigo ou objeto que o agente deseja desviar deve possuir definido um "raio de influência". Este raio de influência irá determinar uma região onde será aplicado um custo adicional aos nós consultados pelo algoritmo A*, estabelecendo um mínimo de distância que é seguro para o agente passar em relação a um inimigo. Desta forma o agente irá gerar uma trajetória que evita passar dentro do raio de influência do objeto a evitar, reduzindo os riscos de ser capturado (Fig. 2). Quanto maior o raio de influência, mais longe o agente irá passar do um inimigo.

Uma vez que também podemos definir regiões com um possível tráfego mais intenso de agentes (ver item 2.1), é interessante que o fugitivo possa avaliar as informações disponíveis sobre estas áreas, que poderão ser protegidas ou não por aliados, ou então, possuir uma maior chance de ter inimigos a espreita preparando uma emboscada.

Com o uso do A* e de mapas compostos por camadas (estática e dinâmica) de custos, podemos assim dar prioridade a outras questões referentes à estratégia de fuga. Concluindo, o agente será capaz de combinar em seu comportamento a geração de trajetórias que evitem obstáculos (mapa original), que evitem os predadores (camada dinâmica adicional) e que busque alcançar pontos específicos mais seguros do ambiente.

3. A* para multi-agentes

Em jogos de estratégia é comum ter sistemas multi-agentes sendo utilizados, onde os agentes devem executar tarefas em comum, ou seja, devem cooperar na execução de uma tarefa. Um dos métodos usados para que os agentes planejem em conjunto os seus caminhos é nomeando um deles como o agente líder. Assim esse agente será responsável por distribuir tarefas entre os demais agentes do grupo.

Não iremos entrar aqui em uma discussão mais aprofundada sobre os sistemas multi-agentes, mas buscamos demonstrar algumas técnicas, sempre baseadas no algoritmo A*, que poderão ser usadas para gerar caminhos e rotas que não fiquem sobrepostas (evitando comportamentos repetitivos de agentes que agem todos da mesma forma). Deste modo buscamos obter um comportamento simples, porém coordenado de um conjunto de agentes.

3.1 Definindo Caminhos

A base do A* multi-agentes é fazer com que os agentes não escolham rotas idênticas ou muito parecidas. Como já foi explicado acima, podemos definir uma camada dinâmica de custos a ser adicionada sobre o mapa original do ambiente. Assim atribuímos custos maiores em volta dos caminhos já escolhidos por outros agentes. Na prática cada agente analisa quais caminhos que os demais agentes do grupo já definiu, e assim os agentes aumentam o custo em volta dos caminhos já escolhidos, forçando estes a buscar caminhos alternativos. Portanto, o custo de passar por uma área já "tomada" por outro agente será superior aos dos outros nós que ainda não foram visitados.

A definição desses custos adicionados em volta das regiões por onde um agente já passou poderá variar. Quanto maior este custo, mais diferentes serão os caminhos definidos pelos diferentes agentes, e conseqüentemente, maior será a abrangência da área explorada pelos múltiplos agentes.

3.2 Alvo

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 (Fig. 3).

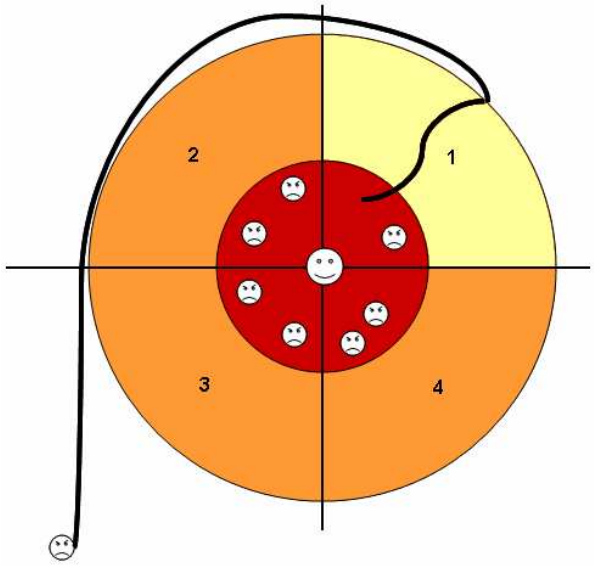

Figura 3: Agentes cercando o alvo

Na figura 3 pode-se ver claramente, o exemplo de um agente contornando os quadrantes que não são de sua responsabilidade, até chegar ao seu destino. Desse modo um grupo de agentes irá cercar o alvo, de modo coordenado, e possivelmente poderá evitar melhor que o agente alvo possa fugir.

4. A* em Terrenos 3D

O algoritmo A* em terrenos 3D é uma adaptação do A* em terrenos 2D, sendo que é adicionada a elevação (eixo Z), podendo a altitude ser considerada como um custo adicional. Entretanto o mais interessante é definir a inclinação do terreno como sendo um fator de dificuldade de movimentação: terreno plano é mais fácil, terreno íngreme é mais difícil. O primeiro passo será, portanto, fazer uma análise do relevo do terreno.

4.1 Análise do Terreno

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 [Pottinger 2000]. 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.

4.2 Adicionando Heurísticas

Após obter os dados necessários para aplicação do algoritmo A* em um terreno 3D, pode-se definir uma série de regras (heurísticas) que podem alterar o comportamento do algoritmo de planejamento de trajetórias e de navegação no ambiente.

Uma regra interessante é definir se o agente pode ou não andar naquele nó que está sendo verificado. Desta forma, se a inclinação for maior que um determinado valor (Figuras 4 e 5), ou seja, o terreno é muito íngreme, é definido que aquele nó da malha não pode ser percorrido pelo agente. Isso permite um aumento no desempenho da aplicação, pois não será necessária a utilização de motores de física ou cálculos matemáticos para verificar se o agente pode ou não passar por determinado local. Isso pode incluir qualquer obstáculo que faça parte do terreno.

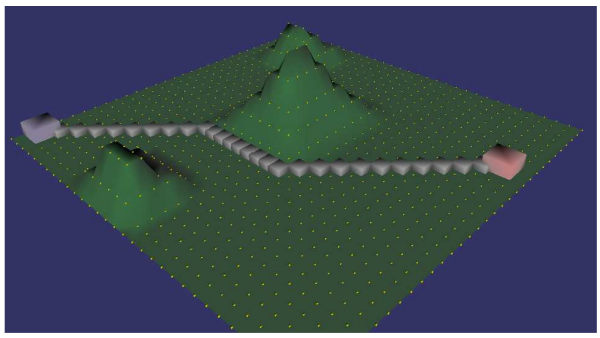

Figura 4: A-Constellation, terreno 3D com o A*

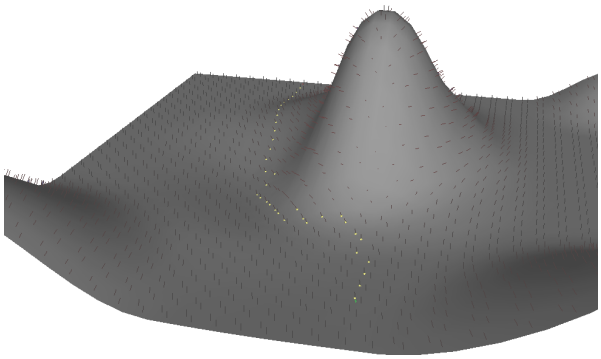

Figura 5: A-Constellation, usando o terreno 3D com normais

Outras estratégias também podem ser consideradas em jogos, por exemplo, um agente arqueiro poderá dar preferência para se movimentar no topo das montanhas e ter alto custo para andar em terrenos baixos.

5. Considerações Finais

Neste artigo apresentamos comportamentos de agentes com uso de técnicas de Inteligência Artificial, baseados em heurísticas aplicadas junto ao algoritmo A*. Podemos constatar que o comportamento dos agentes pode ser facilmente alterado, apenas alterando funções de custos que são atribuídas ao mapa que define o comportamento do agente no ambiente. Foi demonstrado que o A* possui uma característica de ser muito flexível, permitindo seu uso em diversas áreas. O A-Constelação explora isto e vem sendo usado para aperfeiçoar tanto jogos digitais, quanto simulações.

Agradecimentos

Os autores gostariam de agradecer a UNISINOS, CNPq e FAPERGS pela concessão de bolsas (IC) e de auxílios financeiros que permitiram o desenvolvimento deste trabalho.

Referências

DECHTER, R,. AND PEARL , J.,1985, *Generalized Best-First Search Strategies and the Optimality of A**. University of California.

DELOURA MARK. (ED.) . GAME PROGRAMMING GEMS (VOL.I). CHARLES RIVER MEDIA ED. 2000. 550 P. ISBN 1584500492.

LESTER, P., 2004. *A* Pathfinding for Beginners* [online], Disponível em: http://www.policyalmanac.org/games/aStarTutorial_port.htm [Acessado em 11/08/2007].

LESTER, PATRICK. *A* PATHFINDING FOR BEGINNERS*. [online] GAMEDEV–WEB: HTTP://WWW.GAMEDEV.NET/REFERENCE/ARTICLES/ ARTICLE2003.ASP [Acessado em 14/08/2007]

POTTINGER, D.C., 2000. *Terrain Analisys in RealTime Startegy Games* [online], Disponível em: www.gamasutra.com/features/gdcarchive/2www.gamasutra.com/features/gdcarchive/2000/pottinger.doc000/pottinger.doc [Acessado 14 de Agosto de 2007].

RABIN, STEVE (ED.). *AI GAME PROGRAMMNG WISDOM*. CHARLES RIVER MEDIA ED. 2002. 672P.

RUSSEL, R.; NORVIG, P. *ARTIFICIAL INTELLIGENCE: A MODERN APPROACH* ENGLEWOOD CLIFFS, PRENTICE HALL, 1995. 932P.

WIKIPEDIA - *A-Star Search Algorithm* [online], Disponível em: http://en.wikipedia.org/wiki/A*_search_algorithm [Acessado 5 de Agosto de 2007].

WINSTON, PATRICK H. *ARTIFICIAL INTELLIGENCE*. (3RD. EDITION) ADDISONS-WESLEY PUBLISHING, 1992, 737P.