

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Curso de Bacharelado em Ciência da Computação
Inteligência Artificial
Docente: Joseana Macedo Fachine
Discente: Paolo Victor Gonçalves Soares

Projeto de Inteligência Artificial

Relatório de avaliação final

Campina Grande

Junho de 2006

Sumário

Lista de figuras	3
Lista de tabelas	4
Resumo	5
1. Introdução	6
2. Metodologia	
2.1. Algoritmo de busca de caminhos	7
2.2. Cálculo do campo de visão dos agentes	8
2.3. Regras de interação entre os agentes	9
3. Resultados	11
4. Análise dos resultados	12
5. Considerações finais	13
6. Referências bibliográficas	14

Lista de figuras

Figura 2.1.1: Situação na qual o algoritmo de busca heurística por menor distância não encontra o caminho ótimo	6
Figura 2.2.1: Ilustração do cálculo do campo de visão	7

Lista de Tabelas

Tabela 3.1: Resultados da execução de 70 rodadas da simulação 11

Tabela 3.2: Dados sobre a cooperação entre agentes 11

Resumo

Este relatório aborda um projeto desenvolvido como avaliação para a disciplina Inteligência Artificial, da Universidade Federal de Campina Grande. O projeto busca simular um combate entre vários agentes divididos por times e que cooperam entre si, utilizando conceitos como busca heurística, representação do conhecimento em linguagens formais e raciocínio dedutivo para definir e executar as decisões tomadas pelos agentes.

1. Introdução

O objetivo deste projeto é simular um ambiente no qual vários agentes interagem entre si, competindo e cooperando. A função de cada agente é derrotar o time adversário, sendo que um time de agentes é considerado vencedor quando pelo menos um integrante deste está operante e nenhum outro agente de outros times está.

Os problemas inerentes à proposta são:

a) Como os agentes se movem pelo ambiente?

Os agentes devem poder se mover para qualquer parte acessível do ambiente, utilizando o menor caminho possível. O algoritmo de busca de caminho deve ser eficiente e rápido.

b) Como os agentes detectam a presença de outros agentes?

Os agentes devem poder “ver” outros agentes de forma realista – cada agente deve ter um campo de visão bem definido.

c) Como os agentes interagem entre si?

Agentes podem atacar uns aos outros e reagir aos ataques recebidos por outros agentes. Os agentes também devem reagir a estímulos como ataques de outros agentes.

d) Como os agentes podem cooperar para atingir seus objetivos?

Agentes devem trabalhar em times. Um agente deve ajudar agentes do mesmo time a atacar adversários em comum e devem andar em grupos.

Para resolver o primeiro problema, pode ser usado um algoritmo de busca heurística que apesar de não garantir o melhor caminho, satisfaz o requisito de performance. O problema b) é um problema matemático, enquanto os problemas c) e d) podem ser resolvidos estabelecendo um conjunto de regras para a interação entre os agentes.

2. Metodologia

2.1 Algoritmo de busca de caminhos

Para resolver o problema da busca de caminhos, foi utilizado um algoritmo de busca heurística, na qual a heurística é a distância cartesiana entre o próximo ponto do caminho e ponto de destino.

Esta solução tenha boa performance pois o primeiro caminho encontrado é escolhido, sendo desnecessário calcular todos os caminhos possíveis. Entretanto, ela não garante uma solução ótima.

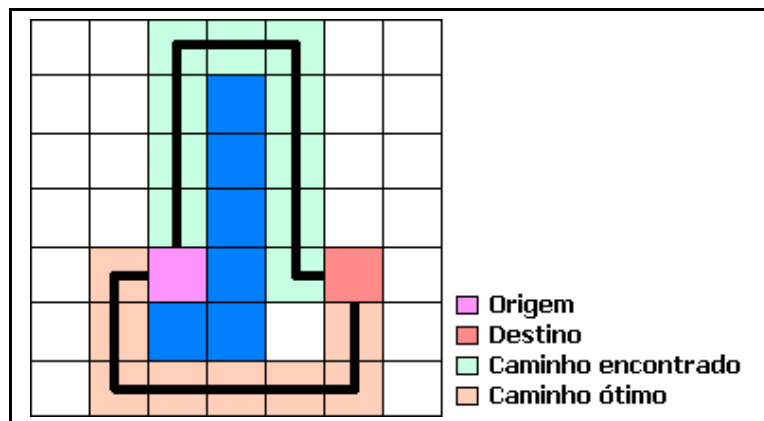


Figura 2.1.1: Situação na qual o algoritmo de busca heurística por menor distância não encontra o caminho ótimo.

2.2 Cálculo do campo de visão dos agentes

Dados dois agentes, A e B e sendo:

- $A(x1,y1)$ o ponto onde A está localizado
- $B(x2,y2)$ o ponto onde B está localizado
- θ o ângulo de visão de A
- $\theta_{A,B}$ o ângulo entre A e B
- θ_{FOV} o campo de visão de A
- $d(A, B)$ a distância entre os pontos $A(x1,y1)$ e $B(x2,y2)$
- D a distância máxima de visão de A

Se:

- i. $d(A, B) \leq D$
- ii. $|\theta_{A,B} - \theta| \leq \theta_{FOV} / 2$

Então B pode ser visto por A.

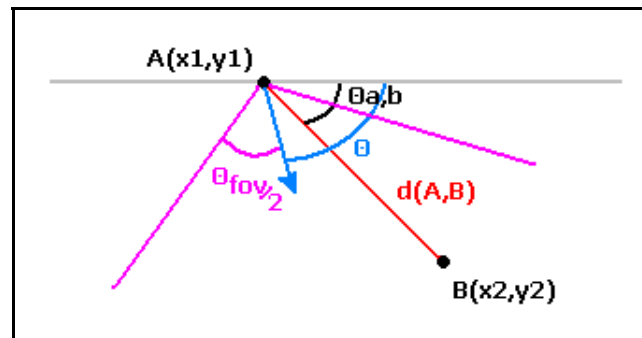


Figura 2.2.1: Ilustração do cálculo do campo de visão

2.3. Regras de interação entre agentes

Para simular a interação entre os agentes, foi utilizado um conjunto de regras baseados em estímulos vindos de outros agentes e informações relativas ao time do agente em questão.

Os agentes são divididos em times, podendo se subdividir em equipes, com um número máximo de integrantes pré-estabelecido. Agentes só cooperam com agentes da mesma equipe e podem atacar agentes de quaisquer outros times diferentes do seu.

Agentes podem ter âncoras, sendo uma âncora um ponto qualquer no ambiente, ou até mesmo um outro agente. Se o agente tem uma âncora, seus movimentos são limitados a uma área próxima à âncora e na qual o agente possa vê-la.

Cada agente tem uma energia associada. Um agente perde energia quando é atacado por outros agentes e ganha energia ao atacar outros agentes. A quantidade de energia ganha é igual à metade do dano aplicado ao agente inimigo. Um agente é considerado inoperante quando sua energia cai para um nível menor ou igual a zero.

Regras para a tomada de decisão de um agente:

- 1.** Se um agente tem alguma ordem a seguir então ele deve executar esta ordem.
- 2.** Se um agente não tiver uma ordem a seguir e não estiver executando uma ação, então:
 - 2.1.** Se existir alguma ação na pilha de ações pendentes então ele deve executá-la.
 - 2.2.** Se não existirem ações pendentes então ele deve se mover para um ponto aleatório no mapa.
- 3.** Dado um agente, para todos os outros agentes no ambiente:
 - 3.1.** Se o outro agente for inimigo, o agente não possuir um alvo e o agente pode vê-lo, então este agente deve ser o novo alvo.
 - 3.2.** Se o outro agente for amigo, então:
 - 3.2.1.** Se o agente não possuir uma equipe, ele pede ao amigo para entrar em sua equipe.

3.2.1.1. Se o amigo aceitar o pedido, o líder da equipe do amigo é a nova âncora do agente.

4. Se um agente torna-se inoperante e ele é líder de uma equipe, então a sua equipe é desfeita.
5. Se um agente não possui um alvo e um membro de sua equipe possui, então o alvo do companheiro de equipe deve ser escolhido como alvo.
6. Se o agente possui um alvo, então ele deve se mover para um lugar próximo do alvo, no qual possa vê-lo.
7. Se o agente não possui um alvo e possui uma âncora, então ele deve se mover para um lugar próximo da âncora, no qual possa vê-la.
8. Se um agente possui um inimigo como alvo e este torna-se inoperante, então este agente deve passar desconsiderar este inimigo como alvo.
9. Se um agente possui um alvo e este alvo pode ser visto, então este agente deve atacar o alvo.
10. Se um agente for atacado por um inimigo e este inimigo estiver mais próximo do que o alvo atual, então este inimigo deve ser o novo alvo.
11. Se um agente tiver um alvo e o alvo sair do campo de visão do agente, o agente deve mover-se para a última posição na qual o alvo foi visto.

3. Resultados

Foi desenvolvida um software Java que implementa uma simulação do ambiente proposto utilizando os conceitos e regras definidos anteriormente. Foram executadas 70 rodadas, sendo que em 10 delas foi também contabilizada a ajuda dada por agentes a outros agentes de suas equipes (o número de vezes que um agente tomou um inimigo como alvo por este ser um alvo de um agente da mesma equipe). Foi utilizada a seguinte configuração:

Número de times: 4 (Azul, Vermelho, Verde, Amarelo)

Número de agentes: 20 (5 por time)

Número máximo de agentes por equipe: 3

Energia inicial dos agentes: 50.0

Resultados das simulações:

Time	Vitórias	%
Azul	21	30%
Vermelho	18	26%
Verde	14	20%
Amarelo	17	24%

Tabela 3.1: Resultados da execução de 70 rodadas da simulação

Alvos escolhidos (A)	Alvos de amigos escolhidos (Am)	Proporção Am/A
1181	621	0,5258 (~53%)

Tabela 3.2: Dados sobre a cooperação entre agentes

4. Análise dos resultados

Os experimentos realizados apontam resultados satisfatórios e um dos pontos principais da proposta: a cooperação entre agentes. A fato de os agentes colaborarem com membros de sua equipe influi bastante na simulação, principalmente quando há desigualdade numérica entre as equipes.

Alguns pontos interessantes que faziam parte da proposta original mas que infelizmente não foram abordados foram a influência do contexto na tomada de decisões dos agentes, a implementação de ações evasivas e ofensivas, agentes controlados pelo jogador, a possibilidade de dar ordens aos agentes através de reconhecimento de fala e a limitação do conhecimento do mapa por times (no início da simulação, todos os agentes já conhecem o mapa inteiro).

5. Considerações finais

Este projeto representou uma boa oportunidade para aplicar os conhecimentos adquiridos durante a disciplina de Inteligência Artificial, tais quais métodos de busca em espaço de estados, linguagens e representação de conhecimento através de linguagens formais e raciocínio dedutivo.

Ainda há possibilidades de aperfeiçoamento, como a implementação das funcionalidades citadas na análise de resultados, que podem ser motivação para trabalhos futuros.

6. Referências bibliográficas

WEISSTEIN, Eric W. "Dot Product.". MathWorld - A Wolfram Web Resource. Online. Disponível em: <http://mathworld.wolfram.com/DotProduct.html>. Acesso em 10/06/2006

Game Programming Wiki. Tutorials and Source Code. Online. Disponível em: http://gpwiki.org/index.php/Tutorials_and_Source_Code. Acesso em 09/06/2006

BOURG, David M. AI for Game Developers. O'Reilly, Online. Disponível em: http://www.gamasutra.com/features/20041122/bourg_01.shtml. Acesso em 29/05/2006