

# Inteligência Artificial em Jogos

---



Marcos Romero  
Maurício Lacerda

---

Projeto JEDI  
Jogos Eletrônicos Distribuídos e Inteligentes  
[www.geocities.com/progjogos](http://www.geocities.com/progjogos)

# Sumário

## Fundamentos

- Máquinas de estados finitos
- Pathfinding

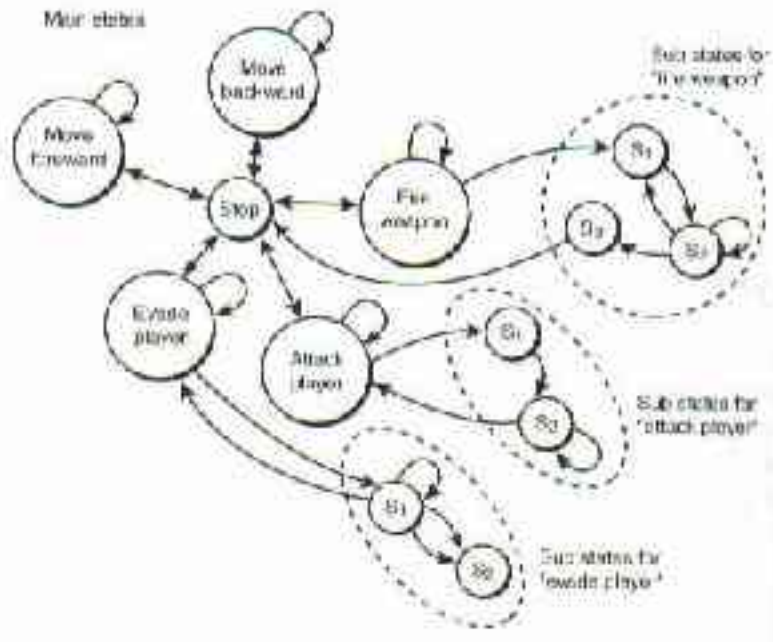
## Estilos de Jogos

- Ação
- Estratégia
- Esporte
- Vida Artificial

## Estudo de Casos

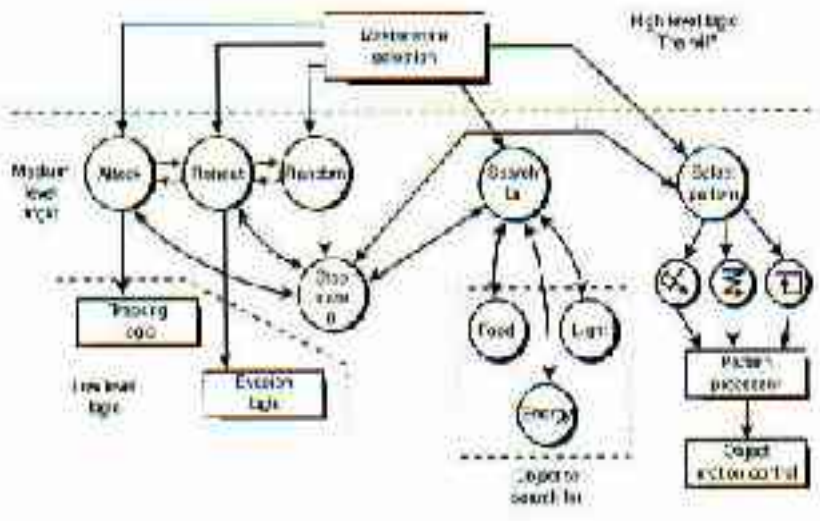
- Simulando os sentidos
- Correndo pela cidade

# Máquinas de Estados Finitos



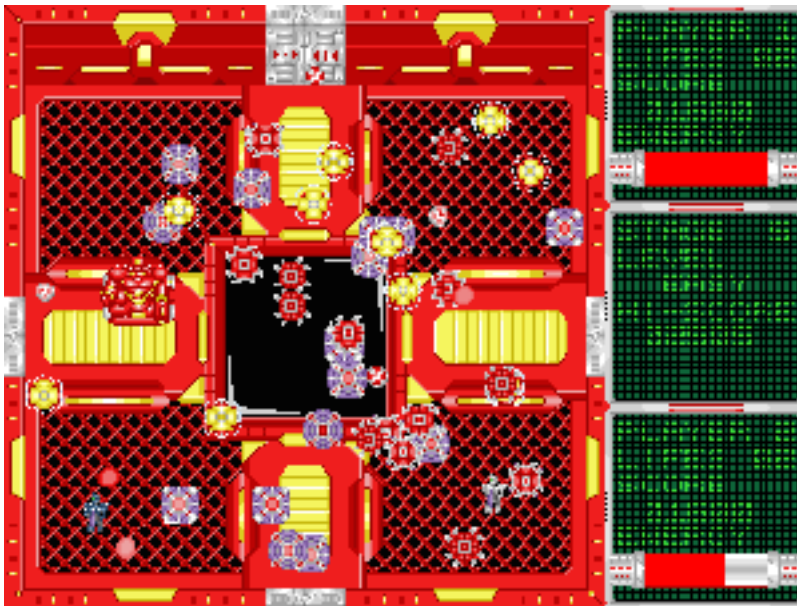
- É uma técnica muito usada em jogos por causa de sua facilidade de implementação e seu tempo de resposta rápido.
- Basicamente consiste na definição de diversos estados, onde cada estado representa uma ação do personagem.

# Níveis de Abstração



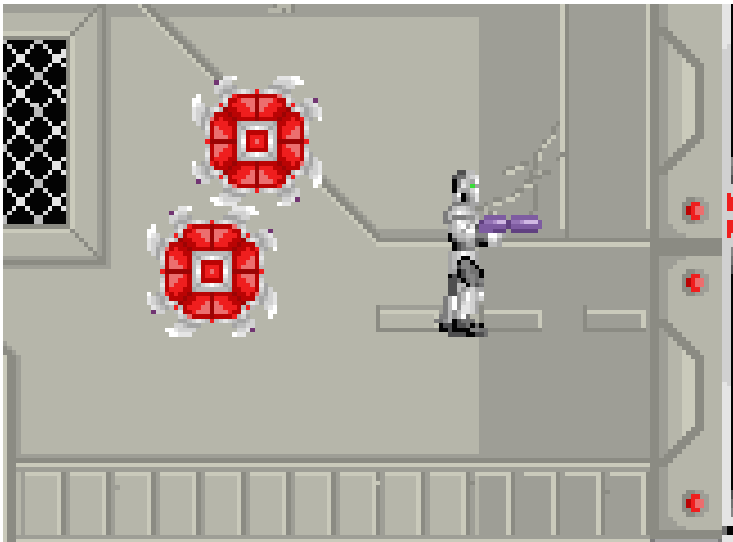
- A definição dos estados cria um alto nível de abstração que permite a fácil visualização das ações de um personagem.
- Cada estado poderá ser implementado usando diversas técnicas.

# Um exemplo simples



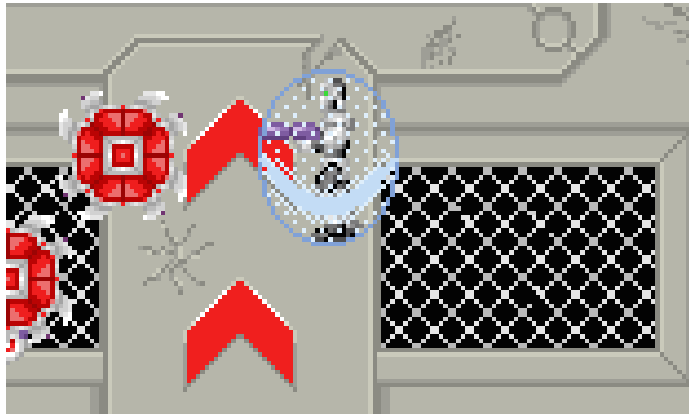
- O jogo Cyborg Arena da RH Games, utiliza máquinas de estados finitos para o controle dos robôs.
- Para cada tipo de robô existem diversos estados que definem o seu comportamento.

# Os estados



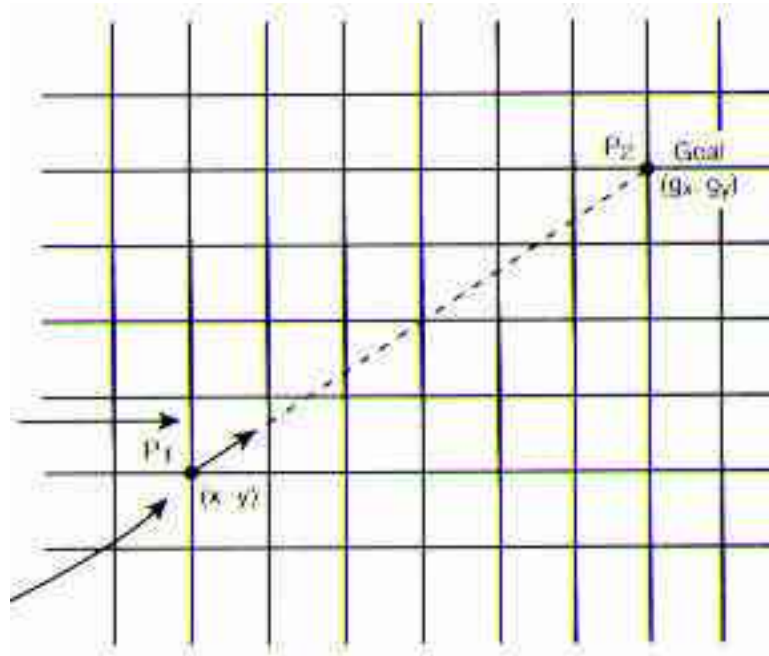
- A máquina de estado finito para um dos tipos de robôs contém os seguintes estados:
  - Perseguir o jogador;
  - Evitar o jogador;
  - Mover aleatoriamente;
  - Parar por um momento;

# Transição entre os estados



- Os principais fatores que determinam a transição de estados são o tempo e a distância ao jogador.
- Porém, se o jogador adquirir um campo de força que destrói o inimigo com o simples contato, então o robô passa imediatamente para o estado “Evitar Jogador”.

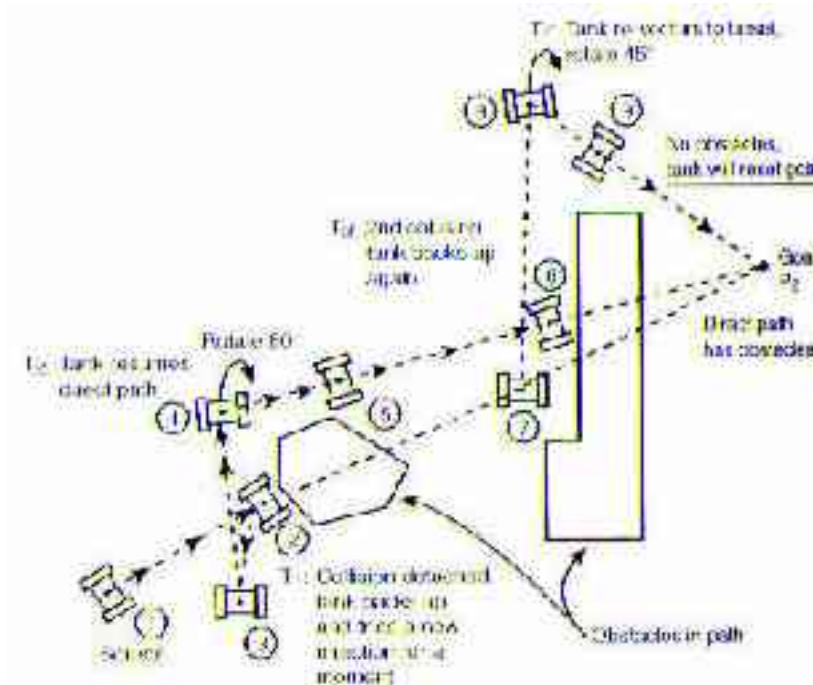
# Pathfinding



- Pathfinding (encontrando o caminho) é a computação e execução de um caminho do ponto  $p1$  até o objetivo  $p2$ , como exposto na figura.
- Se não houver obstáculos então um caminho direto será suficiente, porém se houver será preciso evitá-los.

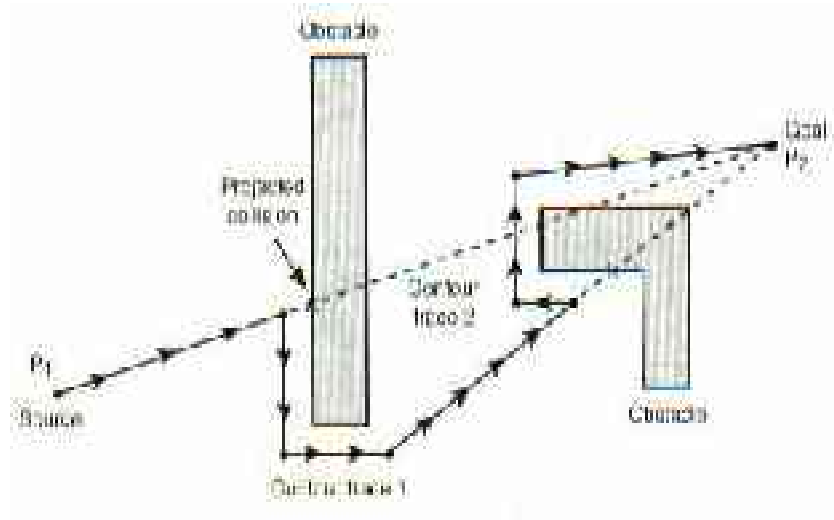


# Tentativa e erro



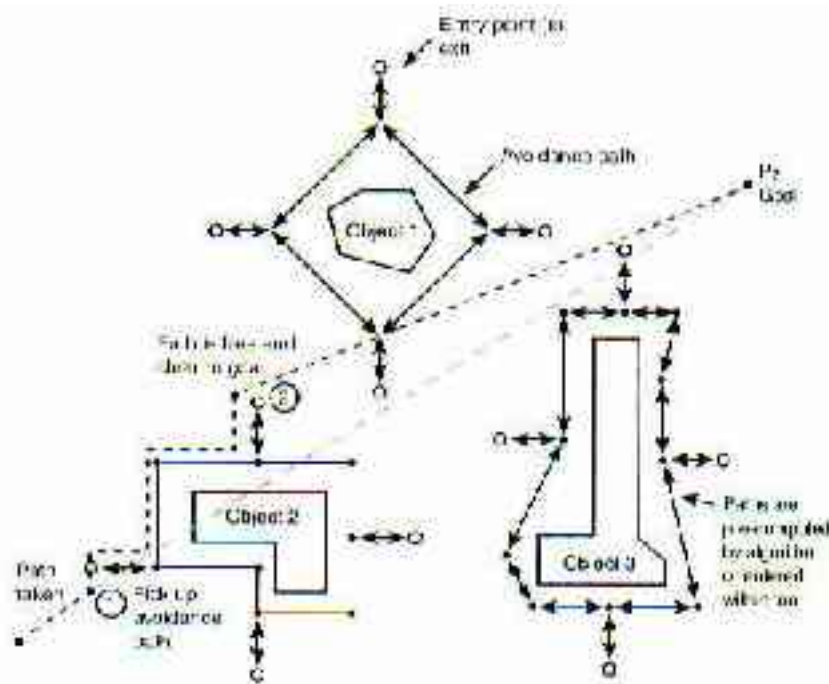
- Este algoritmo serve quando existem apenas simples obstáculos que não são grandes e são convexos.
- Quando o objeto encontra um obstáculo, ele recua um pouco, tenta uma nova direção por um momento e depois retoma a direção original.

# Seguindo o contorno



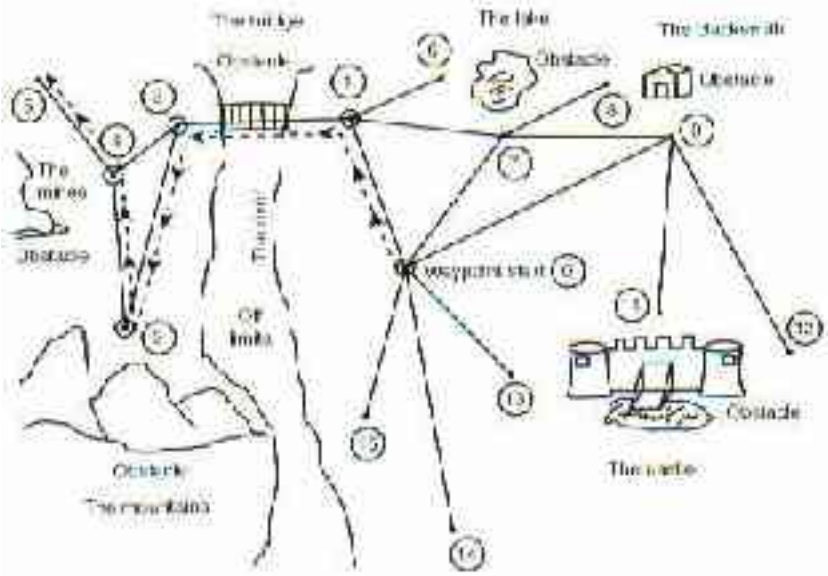
- Neste algoritmo o objeto segue o contorno do obstáculo que está interceptando o seu caminho.
- Enquanto o objeto estiver contornando é feito um teste periódico para verificar se o obstáculo ainda está interceptando.

# Rotas para evitar colisões



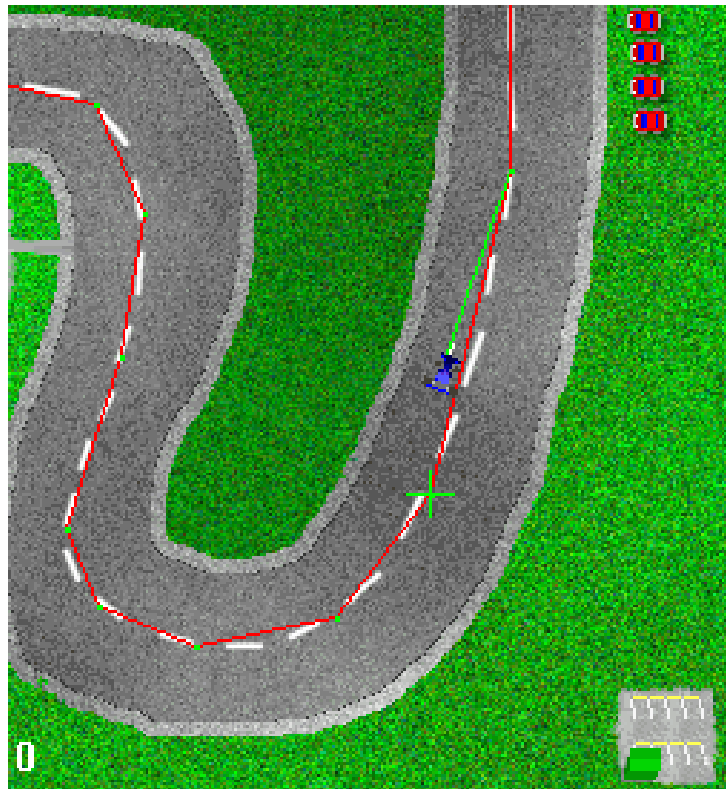
- Nesta técnica são criadas rotas virtuais, ao redor dos obstáculos, que consistem de uma série de pontos ou vetores e definem um caminho.
- Quando um objeto quiser passar por um obstáculo, ele verifica a rota mais perto e a segue.

# Waypoint Pathfinding



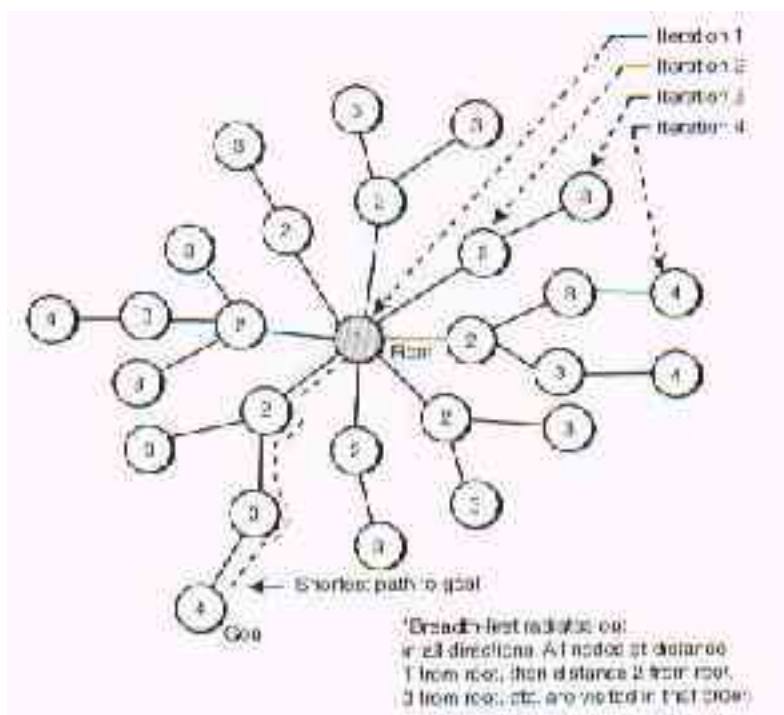
- A utilização de inúmeras rotas interligando todos os pontos principais em um mapa é a principal técnica usada em jogos.
- Cada nó do grafo representa um *waypoint* que contém informações sobre como chegar até os nós vizinhos.

# Exemplo de caminhos



- Um bom exemplo para o uso de rotas é em jogos de corrida. Em uma pista pode ser definidas diversas rotas com características diferentes.
- Cada carro começa seguindo uma rota e muda a medida que se fizer necessário, por exemplo quando existem carros na sua frente.

# Algoritmos de Busca



- Os algoritmos de busca da ciência da computação demandam um bom tempo de processamento tornando difícil o seu uso em tempo real em jogos.
- Porém, eles podem ser usados nos editores de níveis para gerar os mais diversos caminhos existentes no ambiente.

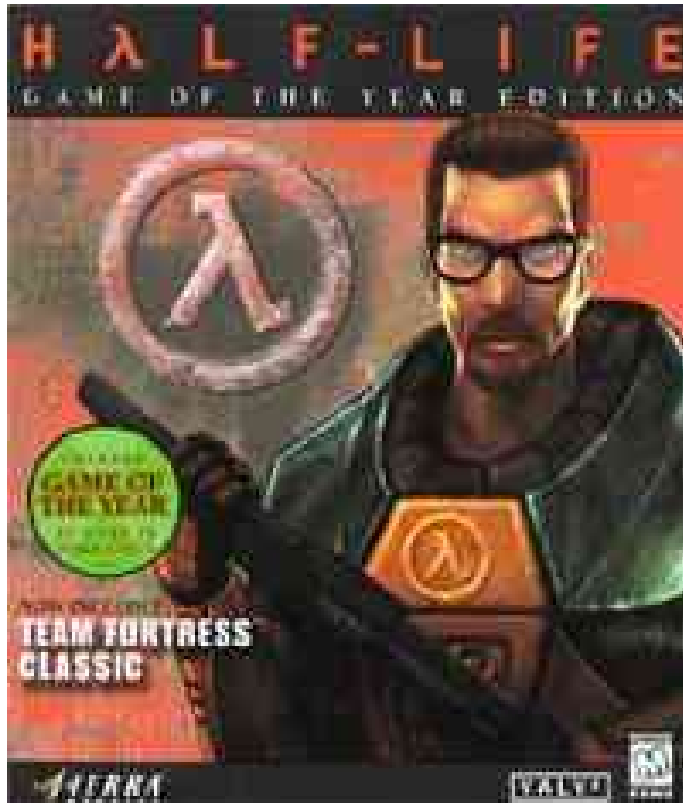
# A\*



[www.rpg-project.com](http://www.rpg-project.com)

- O algoritmo de busca A\* é o favorito para uso em jogos por causa de sua rápida resposta usando heurísticas para terrenos com custo associado.
- Ele pode até ser usado em tempo real com o objetivo de achar o caminho até o ponto mais perto de uma rota.

# Ação



- O tipo mais comum de jogos de ação para PC são o de “Tiro em 1ª pessoa”.
- O jogo Half-Life foi um dos maiores avanços no gênero devido a sua avançada implementação de Inteligência Artificial.

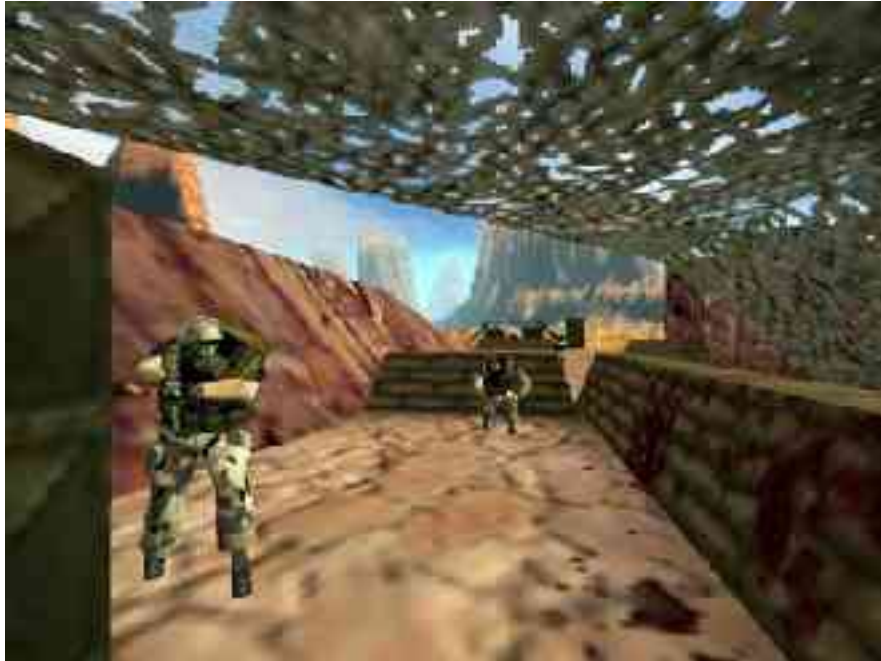


# Características do Half-Life



- Em Half-Life os inimigos só avançam quando isto faz sentido. Eles verificam a energia do jogador, para onde ele está mirando, quantos têm de seu próprio tipo e se eles têm energia suficiente para um combate.
- Enquanto em outros jogos os inimigos são basicamente suicidas, em Half-Life eles não querem morrer.

# Máquina de Estados



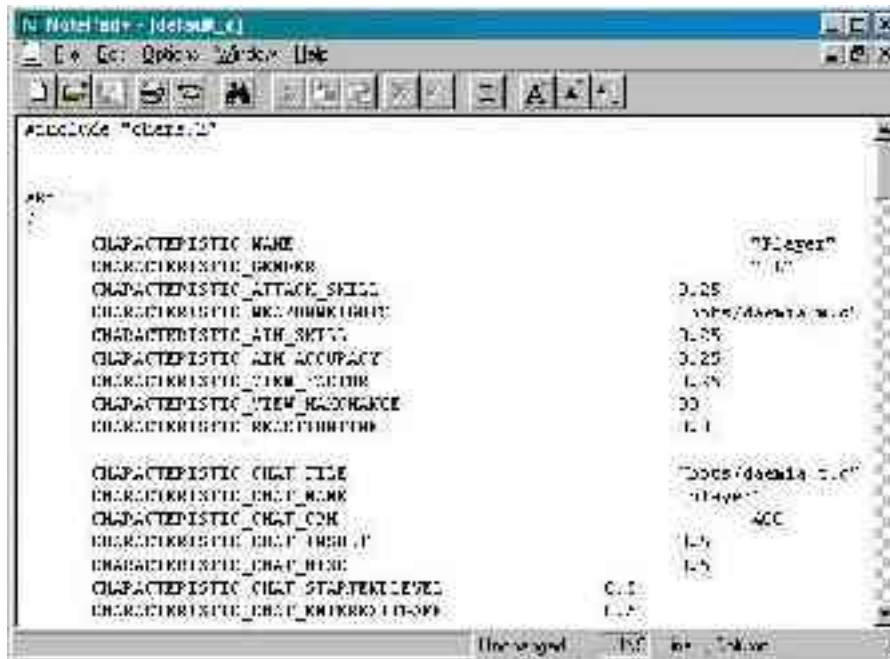
- A Valve, desenvolvedora do Half-Life, usou em sua arquitetura de IA máquinas de estados com listas que definem diversas ações.
- Cada estado define a lista de atitudes que estão disponíveis. Por exemplo, os inimigos podem decidir fugir, mas cada um deles pode fugir de uma forma diferente.

# Bots



- Atualmente o foco destes jogos de ação é o combate multiusuário em rede. Como exemplo temos o Quake 3 Arena.
- Em uma partida pode haver personagens controlados por humanos ou pelo computador (bots), sendo as vezes difícil de distingui-los.

# Ferramentas



- Outra característica comum nestes jogos é que eles fornecem ao usuário ferramentas para a programação dos Bots.
- É muito comum a existência de diversos projetos de IA envolvendo estes jogos.

# Estratégia



- Um gênero muito popular no PC. Starcraft é um bom exemplo de um jogo de estratégia.
- Os principais problemas de IA em jogos de estratégia são: movimentação; IA dos oponentes; IA de grupos e planejamento.

# Movimento e Patrulha



- A movimentação é vital em um jogo de estratégia pois o usuário controla o seu exército através do mouse, ou seja, ele apenas aponta o local para onde a unidade deve se mover.
- A patrulha consiste na movimentação ordenada por uma série de posições.



# Lidando com o oponente



- Cada unidade têm um alcance de visão que determina a percepção de seus inimigos. Além disso é preciso levar em conta os obstáculos que podem atrapalhar a visualização.
- Depois que uma unidade encontra um alvo ele poderá segui-lo para entrar em combate.

# Formando grupos



- É muito comum agrupar as unidades para que um comando seja atribuído a todos os integrantes do grupo.
- Uma das vantagens da implementação de grupos é que pode haver uma lista mestre com as informações de movimento que será acessada por cada unidade do grupo.

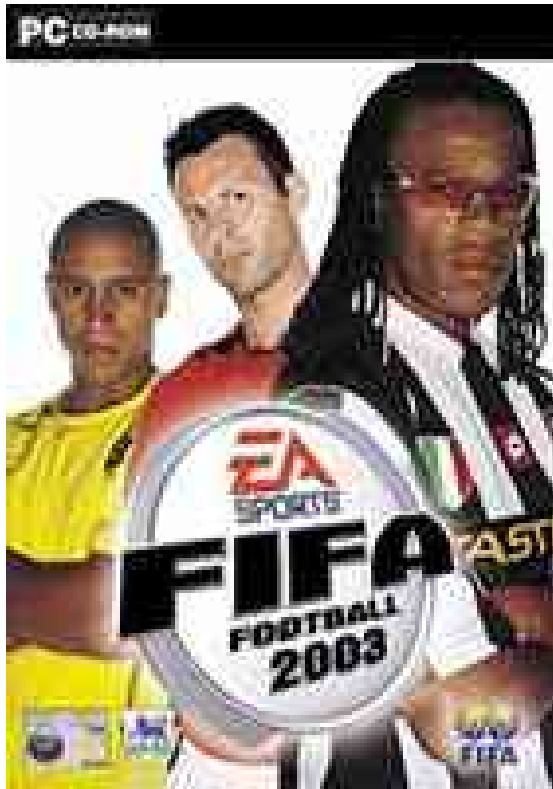


# Planejamento



- Um *plano* é um conjunto de ações em alto nível que devem ser executadas para alcançar um objetivo.
- Exemplo de objetivos:
  - Destruir o centro de comando;
  - Tomar posse da fonte de recursos do jogador.

# Esporte



- Este estilo têm o objetivo de simular esportes reais que geralmente envolvem times.
- Um bom exemplo são os jogos de futebol da série FIFA Soccer.

# Posicionamento



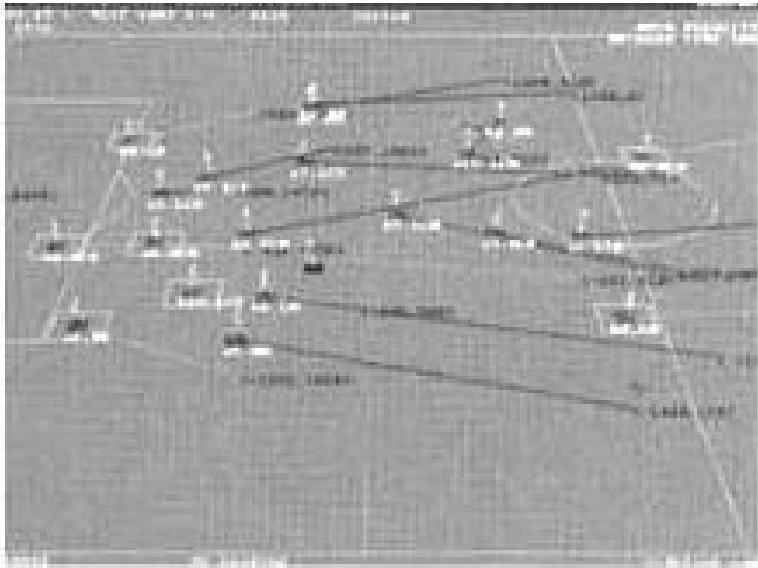
- O posicionamento no campo é fundamental para definir que tipo de estratégia de ser seguida por cada jogador.
- Quando um jogador é expulso, é um desafio fazer com que os outros jogadores ajustem suas posições para compensar a ausência.

# Trabalho em equipe



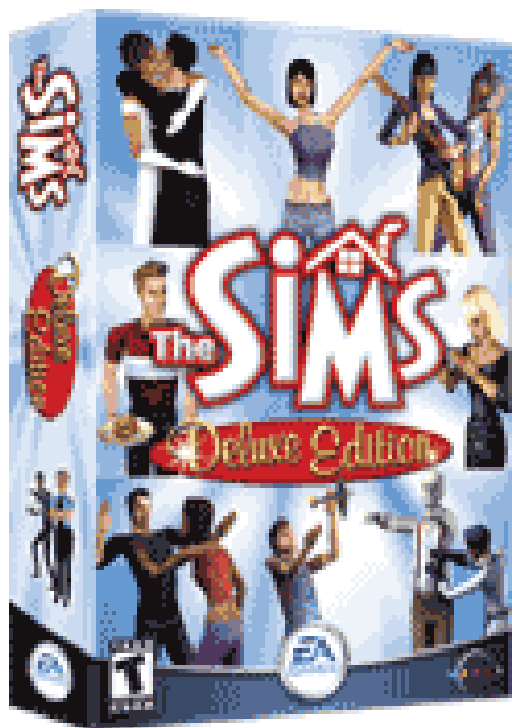
- Na IA de um jogo de futebol é preciso levar em conta a necessidade de cooperação e troca de informações entre os jogadores para alcançar o objetivo final.
- Este é um ambiente típico de multiagentes.

# Ambiente de teste



- Para testar a IA do FIFA, os desenvolvedores usam um ambiente simples sem gráficos bonitos ou som.
- São associados setas e linhas coloridas aos jogadores que ilustram sua IA para tudo, como passes, caminho, etc.

# Vida Artificial



- Os jogos deste tipo têm como objetivo imitar algum aspecto da vida real, seja dos humanos ou do mundo animal.
- O jogo “The SIMS” permite que você controle a casa de uma família e suas interações com os vizinhos.

# Características de um SIM



- Cada personagem virtual é conhecido como “SIM”. No momento de sua criação pode-se definir sua aparência, idade, sexo, cor e personalidade.
- A personalidade têm os atributos: asseio, gostar de sair, ativo, brincalhão, bonito.

# Levando a vida



- Os Sims vivem de acordo com a personalidade e conseguem sobreviver, mas para que eles prosperem é preciso uma orientação do jogador.
- Eles irão comer, beber, dormir, jogar, tomar banho, namorar, trabalhar, etc. Estas ações serão determinadas pelas suas necessidades.



# Necessidades



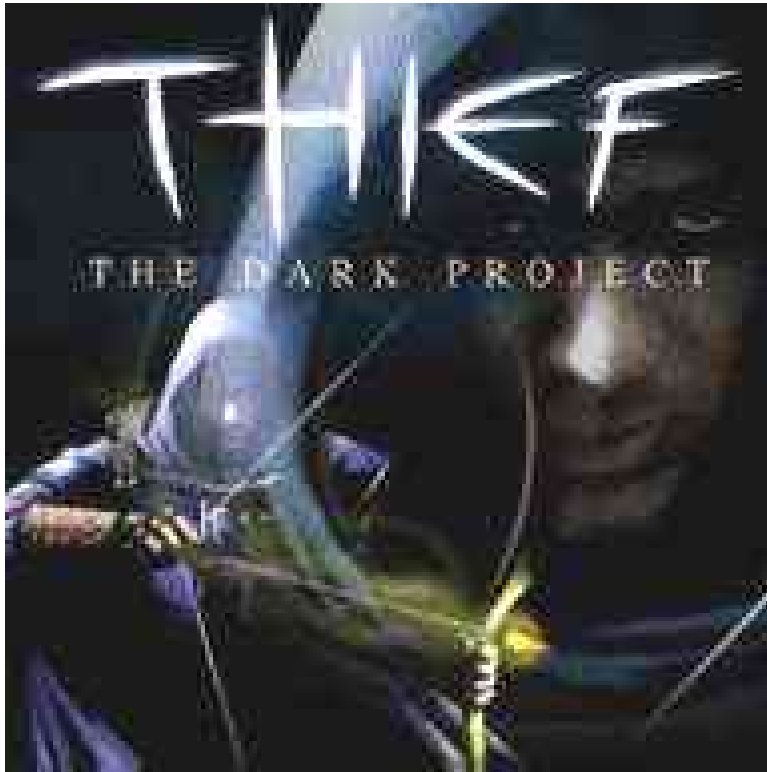
- Há diversas medidas que indicam as necessidades de um SIM e determinam seu comportamento.
- Como exemplo temos : fome, conforto, higiene, energia, diversão.

# A tecnologia



- “The SIMS” é um dos melhores exemplos de A-Life e Máquinas de Estado Fuzzy na indústria de jogos.
- As ações relacionadas a um item do jogo estão armazenadas no próprio item. Isto permite que outros itens sejam adicionados e os Sims saberão como usá-los.

# Simulando os sentidos



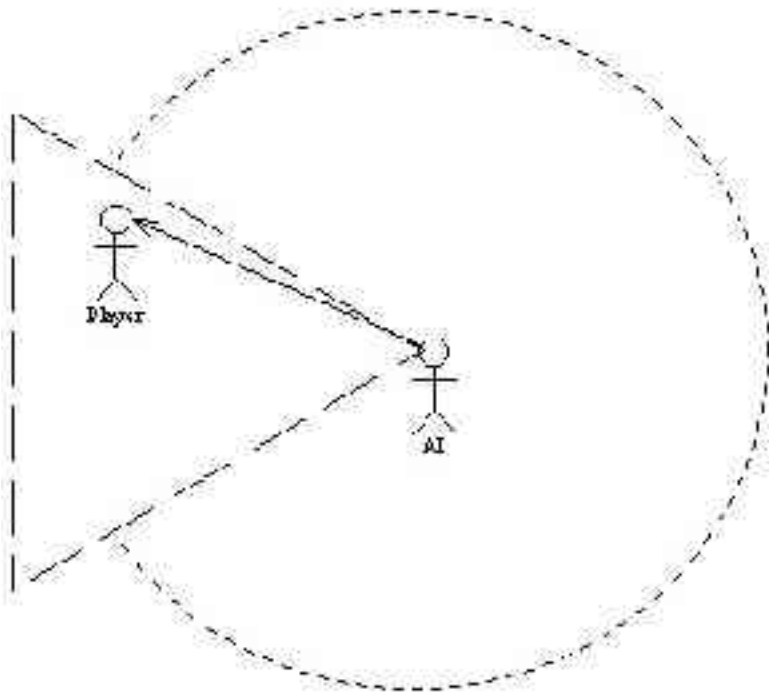
- “Thief” apresenta um ambiente de jogo cujo mecanismo central é a furtividade.
- O jogador deve se mover lentamente, evitar conflitos, andar pelas sombras e não fazer barulho.

# Os sentidos como foco do jogo



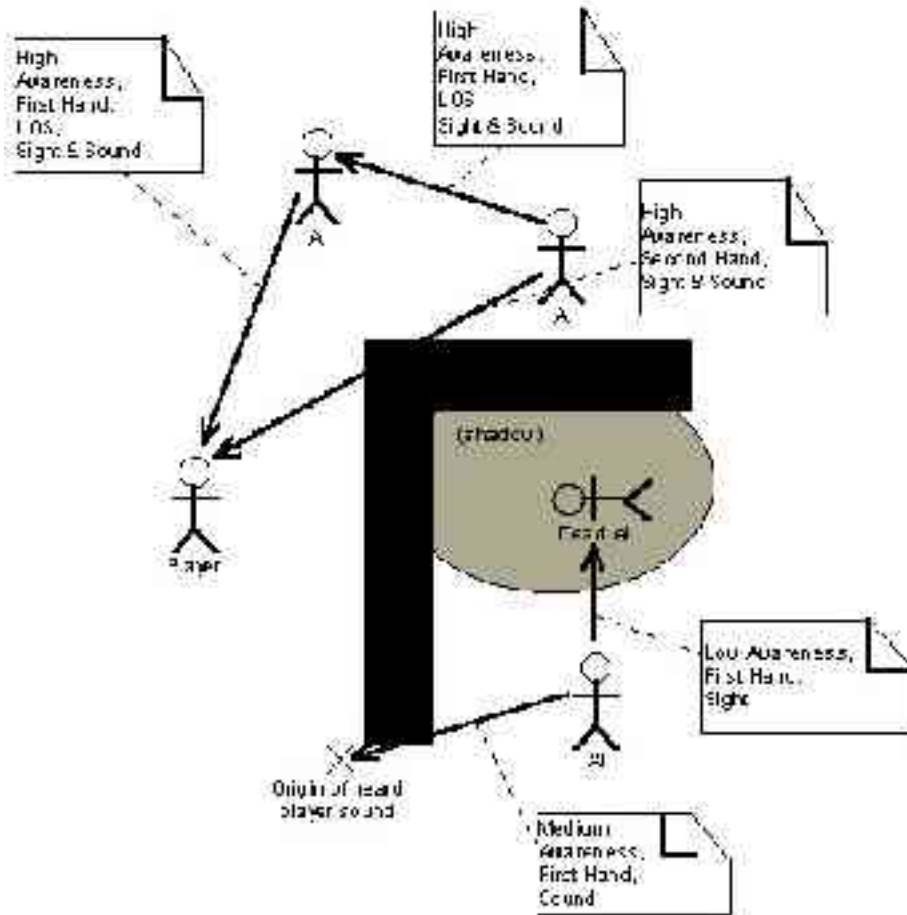
- O requisito primário era criar um sistema de sentidos altamente ajustável que operasse com um amplo espectro de estados.
- Dessa forma a faixa entre segurança e perigo no jogo é ampliada.

# Visão e Audição



- Os sentidos em jogos estão relacionados primariamente à visão e audição.
- Para a definição da visão é preciso os seguintes dados: alcance; posição; cone e linha de visão.
- O som é propagado no ambiente do jogo como no mundo real, levando em conta a direção e atenuação.

# Percepção



- A percepção é expressa como uma faixa de valores que representa a certeza sobre a localização, presença e identidade de um objeto de interesse.
- Por exemplo, se um elemento for visto no escuro, a percepção será bem menor do que se for visto diretamente no claro.

# Pulsos de percepção



- A descrição de uma percepção é organizada como um pulso que é analisado e atribuído um valor de percepção.
- Se o valor de um pulso for bem menor do que o anterior então é usado uma espécie de capacitor para suavizar a transição entre os estados.

# Correndo pela Cidade



- “Midtown Madness 2” é um jogo de corrida para PC que ocorre dentro de uma cidade.
- O ambiente contém várias entidades interativas que incluem os oponentes, tráfego e os pedestres.



# Componentes do Mapa



- A cidade é dividida em três componentes principais para o mapa de IA: ruas, interseções e áreas abertas.
- Atalhos são como as ruas só que sobrepostos sobre os outros componentes para ajudar os oponentes a navegar pelas áreas abertas.

# Criando o tráfego



- Cada carro segue um caminho predefinido e verifica apenas a distância ao carro a sua frente para evitar colisão.
- Quando um carro chega à uma interseção ele verifica o tipo de controle que pode ser : Não pare; Sempre pare; Semáforo; Alerta.

# Simulando os pedestres



- Os pedestres se movem pelas calçadas desviando de obstáculos estáticos, como postes, e dinâmicos, como outros pedestres e veículos.
- Para atravessar a rua, o pedestre verifica o semáforo e espera se for necessário.

# Evitando o jogador



- Os pedestres evitam os carros de duas formas. Se houver um muro por perto então ele corre até lá e se encosta.
- Se não houver, o pedestre fica de frente para o carro e espera até que esteja bem perto para pular para o lado.

# Os oponentes



- A simulação dos carros dos oponentes leva em consideração aspectos de motor, transmissão e direção, enquanto que os carros do tráfego têm uma física mais simplificada.
- Cada oponente têm um registro da rota que deve seguir para poder concluir a corrida.

# Escolhendo a melhor rota



- Se não houvesse o tráfego na cidade, então bastava uma rota para o oponente. Porém há tráfego por toda parte que precisa ser evitado.
- A melhor rota é tipicamente aquela que não está bloqueada, permanece na rua e vai o mais direto possível.

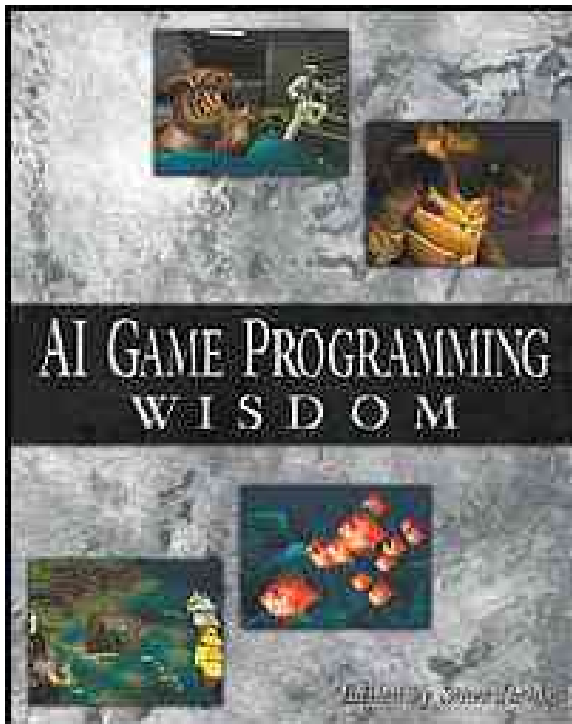
# Página de IA em jogos



- Há uma página específica sobre IA em jogos que mantém registros dos mais diversos projetos na área.
- Contém comentários sobre as implementações de IA em jogos atuais.
- Link : [www.gameai.com](http://www.gameai.com)



# Livro de IA em jogos



- Um dos principais livros sobre IA em jogos é o “AI Game Programming Wisdom”.
- Ele é formado por 71 artigos escritos por diversos especialistas da área.



# Referências

## Livros:

- LaMothe, André. *Tricks of the Windows Game Programming Gurus*. USA: Sams, 1999.
- Saltzman, Marc. *Game Design - Secrets of the Sages*. USA: Macmillan, 2000.

## Páginas da Internet:

- [www.gamedev.net](http://www.gamedev.net)
- [www.gamasutra.com](http://www.gamasutra.com)
- [www.gameai.com](http://www.gameai.com)

## Estudo de Casos:

- Leonard, Tom. *Building an AI Sensory System*. (junho/2003)  
URL : [www.gamasutra.com/gdc2003/features/20030307/leonard\\_01.htm](http://www.gamasutra.com/gdc2003/features/20030307/leonard_01.htm)
- Adzima, Joe. *Using AI to Bring Open-City Racing to Life*. (junho/2003)  
URL : [www.gamasutra.com/features/20010124/adzima\\_01.htm](http://www.gamasutra.com/features/20010124/adzima_01.htm)