

# Uma abordagem usando algoritmos genéticos em Agentes Inteligentes para Jogos

Felipe José Rocha Vieira      Christiano Lima Santos

Universidade Federal de Sergipe, Departamento de Computação, Brasil

## Resumo

É consenso que a tomada de decisões bem como o comportamento assumido pelos personagens em um jogo é uma das áreas mais discutidas em Inteligência Artificial aplicada a jogos. Este artigo tem como objetivo propor o uso de algoritmos genéticos na criação de agentes inteligentes em jogos, sendo este o passo inicial para aplicações mais complexas, bem como apresentar o FightManager, aplicação desenvolvida durante as pesquisas como estudo de caso.

**Palavras Chave:** Inteligência Artificial, Agentes, Algoritmos Genéticos, Máquina de Estados Finitos.

### Contato dos Autores:

{felipejrveira,christianolimasantos}@yahoo.com.br

## 1. Introdução

Com o passar do tempo observou-se a evolução da Inteligência Artificial como forma de buscar soluções a desafios ainda não resolvidos, muitos deles presentes na área de jogos [Rabuske 1995].

No início de uma aventura em um jogo o foco normalmente está em apresentar o mundo ao jogador: história, aliados e inimigos, dentre outros, e com o desenrolar da mesma, jogador e personagem começam a se entender melhor e juntos evoluem no jogo. Desta forma, seria muito decepcionante perceber-se que apenas estes evoluíram, permanecendo o mundo ao seu redor estático, com adversários que já não representam mais dificuldades e personagens autônomos demonstrando o mesmo comportamento repetitivo [Buckland 2002].

Nesse estágio, o jogo já não apresenta mais nenhum desafio. No entanto, o inverso também é desmotivante: inimigos muito fortes podem acabar levando o jogador a abandonar a aventura. Com isto, percebe-se que é de extrema importância uma correta evolução dos NPCs - Non-Player Character (personagem autônomo) que respeite o balanceamento do jogo, sendo este o foco do artigo.

No desenvolvimento de agentes inteligentes em um cenário, deve-se ter bastante cuidado, pois os jogadores esperam que os NPCs se comportem como outros jogadores, raciocinando e agindo de forma

similar a como um jogador humano comportar-se-ia diante de uma mesma situação.

Este trabalho apresenta uma abordagem ao uso de algoritmos genéticos em agentes inteligentes a fim de desenvolvê-los de forma evolutiva.

## 2. Técnicas empregadas

Existem diversas técnicas de algoritmos de IA aplicadas em jogos: minimax, técnicas de pathfinding, redes neurais, flocking, entre outros.

Neste artigo propõe-se uma arquitetura para o comportamento e tomada de decisões dos personagens focada em três das diversas técnicas existentes: máquinas de estados finitos, agentes inteligentes e algoritmos genéticos.

### 2.1 Máquina de Estados Finitos

Uma máquina de estados finitos (MEF) é uma modelagem do comportamento de um sistema que possui um número definido e limitado de condições, alternadas de acordo com os estímulos recebidos.

Segundo Brownlee [2002], as MEF são compostas de quatro elementos principais:

- Estados – definem comportamento e podem produzir ações;
- Transições de Estado – são movimentos de um estado para outro;
- Regras ou Condições – devem ser conhecidas para permitir uma transição de estado;
- Eventos de Entrada – são gerados externamente ou internamente, e podem ativar regras, podendo conduzir a transições.

A utilização desta técnica, apesar de possuir uma natureza previsível, é muito boa, por atribuir alguma inteligência para o personagem a priori e devido à sua simplicidade é rápida de projetar, implementar e em sua execução.

### 2.2 Agentes Inteligentes

Um agente inteligente é qualquer entidade que capte dados do ambiente e atue sobre este. Para Weiss [1999] um agente inteligente deve possuir as

seguintes características:

- **Reatividade:** percepção de seu ambiente, respondendo em tempo hábil às mudanças a fim de atingir seus objetivos;
- **Pró-Atividade:** capacidade de exibir comportamento meta-dirigido ao tomar a iniciativa;
- **Sociabilidade:** interação com outros agentes (e possivelmente humanos).

Através destas características define-se que um bom agente é aquele que a partir de um conjunto de percepções toma uma boa decisão que o leve ao objetivo que deseja.

### 2.3 Algoritmos Genéticos

A idéia do desenvolvimento dos algoritmos genéticos partiu dos conhecimentos da biologia, através da teoria da evolução de Darwin, daí a denominação desta abordagem de evolutiva. [Sobrinho e Girardi 2003]

Nesta técnica, diversos indivíduos diferentes são gerados aleatoriamente e somente os mais adaptados sobrevivem. O fundamento básico desta técnica é criar indivíduos, pontuá-los de acordo com suas ações e, após isso, escolher os mais aptos e fazer um “cruzamento”, gerando assim novos indivíduos possivelmente mais adaptados.

Assim, a cada geração espera-se ter seres mais aptos, podendo-se dizer que aquela população evoluiu. Para melhor compreensão é importante definir alguns termos [Buckland 2002]:

- **Crossover** – taxa de ocorrência de cruzamentos entre indivíduos selecionados, possivelmente os mais aptos. Altos níveis de crossover podem levar mais rapidamente à resposta, no entanto corre-se o risco de perder bons resultados encontrados anteriormente;
- **Mutação** – taxa de ocorrência de uma mudança aleatória na estrutura do indivíduo. Normalmente a taxa de mutação é bastante baixa a fim de evitar a perda das características herdadas que levaram à sua escolha sem desprezar a necessidade de variabilidade genética;
- **Fitness** – pontuação que o personagem adquiriu durante a geração, utilizada para identificar quem foi mais apto.

Testes demonstram que é possível também empregar algoritmos genéticos a fim de verificar se as regras que definem o universo de um jogo bem como sua população estão balanceadas, pois o uso de tais algoritmos permite que se perceba quais as características que mais podem afetar determinadas

situações, possivelmente quebrando o equilíbrio do jogo.

## 3. Arquitetura do Agente

No trabalho produzido por Monteiro [2005] foram utilizados sete módulos:

- **Perception** - Módulo de eventos responsável pela filtragem dos sensores do agente;
- **Location** - Centraliza informações sobre o modelo do ambiente para o agente;
- **Long-Term Decision** - Responsável pelo planejamento;
- **Short-Term Decision** - Responsável pela execução do plano atual;
- **Motion** - Manipula aspectos de roteamento, colisão e desvios de obstáculos;
- **Animation** - Determina a animação a ser exibida;
- **Behavior** - Atua sobre o ambiente de forma reativa.

Neste trabalho, será apresentado um modelo simplificado para a arquitetura do agente com apenas três módulos:

- **Percepção** – responsável pela análise da área onde o personagem está;
- **Decisão** – através da percepção que teve do ambiente o personagem escolhe qual melhor comportamento a se assumir;
- **Ação** – Depois de decidido qual comportamento utilizar o personagem executa sua ação.

A utilização de um modelo simplificado tem como intuito aumentar a abstração em relação à visualização que se tem do agente, assemelhando-se a como se toma uma decisão, observando o ambiente, decidindo-se e executando alguma ação.

### 3.1 Percepção

A percepção funciona recebendo estímulos do ambiente e levando o personagem a algumas conclusões, traçando planos de ação para serem comparados na próxima etapa de processamento.

Cada ser possui um nível de percepção, que influencia suas escolhas, por exemplo, ao observar um ônibus a certa distância, aquele que tiver melhor visão terá uma resposta mais precisa e mais rápida quando perguntado sobre que ônibus está vindo.

Isto leva a concluir que cada personagem deve ter uma percepção própria que dê a ele a possibilidade de ter opiniões diferentes dos outros, dando maior diversidade ao jogo.

### 3.2 Decisão

Depois de colher informações do meio e fazer uma análise prévia do que está acontecendo no ambiente o personagem pode tomar uma decisão, que acontece através do cruzamento dos dados dos oponentes com as informações do agente. Por exemplo, por mais que tenha identificado que exista um oponente fraco, o fato de ter poucos pontos de vida o impossibilita de decidir perseguir o inimigo, fazendo-o preferir fugir.

O momento da decisão é aquele em que o personagem decide qual das estratégias deve seguir entre a gama de possibilidades que sua percepção gerou. Então, por melhor que seja a percepção dele o que levará a tomar uma decisão é o conhecimento que possui de si mesmo, seu estado atual e seu objetivo.

### 3.3 Ação

Os dois módulos anteriores mostram ações mentais, já neste, o foco está nas ações físicas, executando o que foi decidido. A partir deste ponto é que o ambiente sofre mudanças, o personagem se move, muda sua animação e demonstra seu comportamento.

A cada turno todas as três etapas são executadas, pois só através delas é que se podem executar boas ações, já que o ambiente está em constante mudança e o que era bom anteriormente pode não ser mais.

## 4. Seleção e Evolução dos Agentes

Como já foi apresentado anteriormente como funciona cada agente, esta seção adentra na seleção e evolução dos mesmos. Cada unidade possui características que acabam diferenciando uma das outras, mas nem sempre estas são interessantes para o ambiente onde o agente está. E é nesta situação que se aplicam os algoritmos genéticos.

Para cada sucesso, o personagem recebe uma pontuação (*fitness*) que será utilizada para a seleção dos mais aptos, possibilitando a evolução da espécie, tendo assim oponentes e aliados mais adaptados. [Buckland 2002]

Quanto mais gerações forem geradas, encontrar-se-ão personagens mais aptos, e é importante ressaltar que este número de gerações deve possuir um limite que é quando o resultado converge para um grupo de personagens considerados os mais adaptados.

Desta forma, a seleção e evolução dos agentes podem ser empregadas como forma de criar diversos oponentes apresentando cada qual seu próprio nível de capacidade de raciocínio.

Isto levará a soluções que possuem certa variação, gerando dificuldades para o jogador que, quando bem

balanceadas, representarão obstáculos com nível de complexidade bem mensurado.

## 5. Estudo de Caso – FightManager

FightManager trata-se de um jogo onde o usuário deve participar de lutas em arenas (áreas fechadas onde vários lutadores confrontam-se) e treinar seu personagem de forma a buscar ser o melhor.

O jogo possui uma fase não jogável de evolução que mostra combates entre NPCs, buscando, através das interações, pontuar os agentes a fim de classificá-los e identificar os mais adaptados, possibilitando a geração de novos combatentes através das informações adquiridas com as gerações anteriores.

A figura 1 apresenta a interface gráfica de FightManager, atualmente em modo texto:



Figura 1: Interface gráfica.

Para a realização dos combates é necessário que os BOTs tenham atributos que definirão as características mentais e físicas de cada um. No FightManager, um personagem é constituído das seguintes características:

- Força (F) – Atributo relacionado a quanto de dano um personagem pode causar;
- Destreza (D) – Identifica a capacidade de deslocamento e a habilidade;
- Vigor (V) – Capacidade de absorver os danos sofridos;
- Percepção (P) – Influi diretamente na análise que o personagem faz de cada oponente;
- Intimidação (I) – Nível de medo que um oponente causa a um personagem
- Coragem (C) – Indica os pontos de vida para um NPC começar a fugir, calculado através da seguinte fórmula:  $(30 - \text{coragem})/3$ .

A implementação do NPC está distribuída da seguinte forma:

- Percepção – Escolhe o oponente aparentemente mais fácil de derrotar;
- Decisão – Através da primeira análise muda

seu estado para o mais adequado entre: atacar, perseguir, fugir, descansar e morto, a Figura 2 mostra como são tomadas estas decisões.

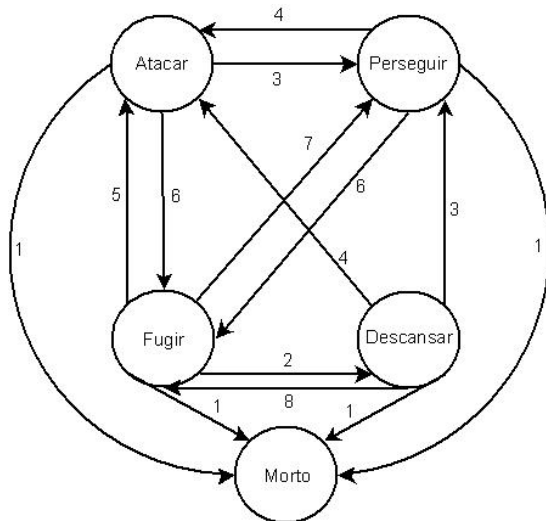


Figura 2: Todas as regras de mudança de estado.

Segue abaixo a explicação dos códigos para ocorrer transição entre os estados.

1. Pontos de Vida menor que zero
2. Oponentes estão distantes
3. Oponente alvo fora de alcance
4. Oponente alvo próximo
5. Oponente alvo em piores condições que o personagem está próximo
6. Pontos de Vida baixo e oponente alvo em melhores condições que o personagem
7. Oponente alvo em piores condições que o personagem está distante
8. Oponentes próximos;

- Ação – Executa o estado escolhido.

Ao ser executado, criaram-se inicialmente 10 personagens com os seguintes atributos:

Nome: 0 F: 7 D: 6 V: 3 P: 9 I: 5 C: 11  
 Nome: 1 F: 3 D: 7 V: 8 P: 2 I: 10 C: 2  
 Nome: 2 F: 7 D: 5 V: 4 P: 5 I: 9 C: 10  
 Nome: 3 F: 10 D: 1 V: 6 P: 7 I: 6 C: 11  
 Nome: 4 F: 8 D: 5 V: 5 P: 9 I: 3 C: 16  
 Nome: 5 F: 4 D: 2 V: 10 P: 6 I: 8 C: 0  
 Nome: 6 F: 10 D: 10 V: 1 P: 3 I: 6 C: 2  
 Nome: 7 F: 9 D: 2 V: 2 P: 7 I: 10 C: 26  
 Nome: 8 F: 4 D: 6 V: 6 P: 8 I: 6 C: 17  
 Nome: 9 F: 8 D: 3 V: 7 P: 6 I: 6 C: 27

Após 30 gerações, os últimos 10 personagens tinham adotado a seguinte configuração:

Nome: 290 F: 6 D: 5 V: 8 P: 6 I: 5 C: 3  
 Nome: 291 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 292 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 293 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 294 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3

Nome: 295 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 296 F: 6 D: 5 V: 8 P: 6 I: 5 C: 3  
 Nome: 297 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 298 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3  
 Nome: 299 F: 6 D: 6 V: 8 P: 5 I: 5 C: 3

Em outras palavras, os personagens evoluíram até a constituição que eles encontraram como sendo ideal para os combates, demonstrando assim o aprendizado propiciado bem como a eficácia do método.

## 6. Considerações Finais

Este trabalho é o primeiro passo para a busca de novas soluções no desenvolvimento de jogos com um melhor balanceamento, bem como personagens capazes de aprender e ter comportamentos distintos, uma vez que muitas das técnicas de IA atualmente empregadas são pouco flexíveis ou possuem taxas de aprendizado muito baixas ou mesmo inexistentes.

Outro ponto importante foi desmistificar diversos aspectos da área de jogos, permitindo assim aos pesquisadores o acúmulo de experiência na mesma.

Como futura extensão, pretende-se estudar a API gráfica OpenGL e implementar uma versão demonstração jogável do mesmo e distribuir livremente o mesmo a fim de avaliar com jogadores o potencial dos agentes empregados.

## Referências

- BROWNLIE, Jason, 2002. *Finite State Machine (FSM)*. [online]. Disponível em <http://ai-depot.com/FiniteStateMachines/FSM.html>. [Acessado em 23/08/2008].
- BUCKLAND, Mat, 2002. *AI Techniques For Game Programming*. Ohio: Premier Press.
- MONTEIRO, Ivan Medeiros, 2005. *Uma Arquitetura Modular para Desenvolvimento de Agentes Cognitivos em Jogos de Primeira e Terceira Pessoa*. Universidade Federal da Bahia, Instituto de Matemática.
- RABUSKE, Renato, 1995. *Inteligência Artificial*, Editora da UFSC.
- SOBRINHO, Antonio Carlos C. da S. E GIRARDI, Maria Del Rosário, 2003. *Uma Análise das Aplicações dos Algoritmos Genéticos em Sistemas de Acesso à Informação Personalizada*. Universidade Federal do Maranhão.
- WEISS, Gerhard, 1999. *Multiagent Systems, A Modern Approach to Distributed Modern Approach to Artificial Intelligence*. Massachusetts Institute of Technology. Editado por Gerhard Weiss, Inglaterra.
- WIKIPEDIA, *Máquina de Estados Finitos*, disponível em [http://pt.wikipedia.org/wiki/M%C3%A1quina\\_de\\_estado\\_finito](http://pt.wikipedia.org/wiki/M%C3%A1quina_de_estado_finito) [Acessado em 25/08/2008].