



UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE CIÊNCIAS EXATAS

**INTELIGÊNCIA ARTIFICIAL APLICADA NO DESENVOLVIMENTO DE JOGOS DE
COMPUTADOR**

Carlos Henrique Silva Galdino

**Trabalho da Disciplina Metodologia de Pesquisa aplicada a Computação apresentado em 26
de outubro de 2007**

**Itajubá
2007**

RESUMO

Os jogos digitais são uma das formas de entretenimento de mais sucesso no mundo. Desde a sua criação eles estão em um processo de constante evolução, onde os fabricantes procuram melhorar a sua qualidade empregando técnicas de diversas áreas de pesquisa da computação no seu desenvolvimento.

A Inteligência Artificial é uma área da computação que é bastante empregada nos jogos, pois ela é responsável por deixar os personagens e objetos do jogo inteligentes, proporcionando maior realismo ao jogo e atraindo mais jogadores.

O artigo apresenta as principais técnicas de Inteligência Artificial utilizadas nos jogos, mostrando o seu funcionamento e discutindo a causa que levam elas a serem as técnicas mais empregadas. No fim são apresentadas as tendências da Inteligência Artificial para o mercado dos jogos de computador.

Palavras-chave: jogos de computador, inteligência artificial, aplicações de inteligência artificial, desenvolvimento de jogos.

ABSTRACT

The computer games are one of the most successful ways of entertainment in the world. Since its beginning, they are in continuous evolution, and the manufacturers aim to improve its quality on its development by using techniques from several areas of research in computing.

The Artificial Intelligence is an area of computing widely used on games, once it is responsible for making characters and objects intelligent, providing more reality and attracting more players.

This article presents the main techniques of Artificial Intelligence used in the games, demonstrating and discussing the causes that make them the most used techniques. Finally, it is presented the tendencies of Artificial Intelligence in the field of computer games.

Keywords: computer games, artificial intelligence, artificial intelligence applications, games development.

Sumário

1 – Introdução	4
2 – Revisão Bibliográfica	6
3 – Técnicas de IA utilizadas no desenvolvimento de jogos de computador	8
3.1 – Máquinas de Estado Finito	8
3.2 – Path-Finding	10
3.3 – Padrões de Movimento.....	10
3.4 – Sistemas baseados em regras.....	11
3.5 – Lógica Fuzzy	12
4 – Futuro da IA nos jogos de computador.....	13
5 – Conclusão.....	13
6 – Referências bibliográficas.....	13

1 – Introdução

Segundo Luz (2004), uma das primeiras tentativas de se criar um jogo eletrônico foi a de Willy Higinbotham, que em 1958 criou um protótipo de jogo de tênis usando um osciloscópio como display. Essa foi uma das primeiras tentativas de se criar um jogo eletrônico, mas para os amantes de jogos clássicos o primeiro jogo eletrônico chama-se *Spacewar!*, criado por Steve Russel durante a sua graduação no MIT (Massachusetts Institute of Technology).

Desde então, a criação de jogos evoluiu, segundo PJMOO (2007), ao ponto da indústria de jogos superar a indústria do cinema, movimentando em 2003, US\$ 28 bilhões no mundo inteiro. Essa grande movimentação de dinheiro é fruto da criação de jogos cada vez mais sofisticados, aplicando novas tecnologias de computação, e a inteligência artificial é uma delas.

Na computação existem várias áreas de pesquisa como computação gráfica que são aplicadas nos jogos de computador e este artigo tem como objetivo principal estudar as aplicações da IA (Inteligência Artificial) no desenvolvimento dos jogos de computador.

Os jogos de computador procuram se tornar cada vez mais realistas, o que atrai novos jogadores e aumenta o lucro dos fabricantes. Nesse processo de busca por mais realismo pode-se citar a melhoria dos gráficos do jogo, o que implica em melhoria dos componentes de hardware, melhoria dos enredos e jogabilidade, dentre outros. Os jogadores sempre buscam um jogo não muito fácil, mas nem tão difícil ao ponto de ser impossível de se completar o objetivo, eles buscam um jogo onde predomine o equilíbrio de dificuldade. Para se chegar ao equilíbrio é necessário utilizar a IA, pois é ela que proporciona inteligência aos NPC's (*non-player characters*, personagens do jogo que não são controlados pelo humano). A IA fica responsável de possibilitar aos NPC's a inteligência que somente os humanos possuem proporcionando ao jogador inúmeras situações diferentes no jogo, o que deixa o jogo mais real e interessante para o jogador.

A seguir é mostrada uma tabela que mostra a história do uso de IA nos jogos eletrônicos.

Tabela 1: Linha de tempo da IA em jogos.

Ano	Descrição	IA utilizada
1962	Primeiro jogo de computador, <i>Spacewar</i> , para 2 jogadores.	Nenhuma
1972	Lançamento do jogo <i>Pong</i> , para 2 jogadores	Nenhuma
1974	Jogadores tinham que atirar em alvos móveis em <i>Pursuit</i> e <i>Qwak</i> .	Padrões de Movimento
1975	<i>Gun Fight</i> lançado, personagens com movimentos aleatórios.	Padrões de Movimento
1978	<i>Space Invaders</i> contém inimigos com movimentos padronizados, mas também atiram contra o jogador.	Padrões de Movimento
1980	O jogo <i>Pac-man</i> conta com movimentos padronizados dos inimigos, porém cada fantasma (inimigo) tem uma “personalidade” sobre o modo em que caça o jogador.	Padrões de Movimento
1990	O primeiro jogo de estratégia em tempo real, <i>Herzog Wei</i> , é lançado. Junto, os jogadores puderam noticiar uma péssima busca de caminho.	Máquina de Estados
1993	<i>Doom</i> é lançado como primeiro jogo de tiro em primeira pessoa.	Máquina de Estados
1996	<i>BattleCruiser: 3000AD</i> é publicado como o primeiro jogo a utilizar redes neurais em um jogo comercial	Redes Neurais
1998	<i>Half-Life</i> é lançado e analisado como a melhor IA em jogos até a época, porém, o jogo utiliza IA baseada em <i>scripts</i> .	Máquina de Estados / <i>Script</i>
2001	O jogo <i>Black & White</i> é alvo da mídia a respeito de como as criaturas do jogo aprendem com as decisões feitas pelo jogador. Utiliza redes neurais, <i>reinforcement</i> e <i>observational learning</i> .	Diversos

Fonte: Schwab, 2004.

O objeto de estudo deste artigo são as técnicas de IA mais utilizadas pelos desenvolvedores, quais são, como funcionam, porque são mais utilizadas e o futuro da IA nos jogos de computador.

2 – Revisão Bibliográfica

A definição de inteligência artificial segundo o dicionário *The American Heritage Dictionary of the English Language* é: “A habilidade de um computador ou outra máquina executar atividades que normalmente se acredita requerer inteligência.”

Segundo Bourg e Seemann (2004), “é mais apropriado pensar que IA é o comportamento inteligente demonstrado pela máquina que foi criada, ou talvez o cérebro artificial por trás daquele comportamento inteligente.” Ainda Bourg e Seemann citam que “para outros estudiosos o estudo de IA não necessariamente tem o propósito de criar máquinas inteligentes, mas o propósito de conseguir o melhor entendimento da natureza da inteligência humana. Ainda outros estudiosos estudam os métodos de IA para criar máquinas que demonstram alguma forma limitada de inteligência.”

Segundo Russel e Norvig (2004), os trabalhos em torno da IA começaram logo após a Segunda Guerra Mundial, e o próprio nome foi cunhado em 1956. Mas o primeiro trabalho reconhecido como IA foi realizado por Warren McCulloch e Walter Pitts em 1943, eles propuseram um modelo de neurônios artificiais, onde cada neurônio tinha dois estados: “ligado” ou “desligado”. Um neurônio mudava para o estado “ligado” em resposta ao estímulo de um número suficiente de neurônios vizinhos. Em 1949, “Donald Hebb demonstrou uma regra de atualização simples para modificar as intensidades de conexão entre neurônios. Essa regra, batizada de aprendizagem de Hebb, continua a ser um modelo influente até hoje.” (Russel e Norvig, 2004).

“Vários trabalhos que podem ser caracterizados como IA surgiram desde então, mas Alan Turing ao publicar em 1950 o seu artigo “*Computing Machinery and Intelligency*” articulou uma visão completa da IA, onde apresentou o teste de Turing, aprendizagem por reforço, aprendizagem de máquina e algoritmos genéticos.”(Russel e Norvig, 2004).

Já a definição de jogos de computador, segundo Battaiola (2000), “pode ser definido como um sistema interativo que permite ao usuário experimentar, uma situação de conflito. Quase sempre, como plano de fundo dessa situação, existe um enredo, que define do que se trata o jogo, as regras que o jogador deve seguir e os objetivos que ele deve se esforçar para atingir.”

Nos jogos de computador, os desenvolvedores e os jogadores sempre estão interessados em atribuir aos personagens que não são controlados por um humano, características de inteligência humana, para melhorar assim a qualidade do jogo e atraírem mais consumidores.

Para isso são utilizadas diversas técnicas de IA, segundo Karlsson (2005), algumas das principais são:

- **Máquinas de Estado Finito:** “são as porcas e parafusos da IA dos jogos” (Bourg e Seemann, 2004). As máquinas de estado finito são compostas de um conjunto de estados predefinidos e um conjunto de regras que definem a transição entre os estados, geralmente os estados são alguma ação ou tipo de comportamento que o personagem deve executar e podem também representar algum evento a ser executado no ambiente do jogo;
- **Path-Finding:** é a busca de caminho entre dois pontos do cenário. Para isso utilizam-se algoritmos específicos, como o A* (lê-se a-estrela) que determinam os caminhos a serem percorridos para o desvio de possíveis obstáculos, inimigos etc.;
- **Padrões de Movimento:** são técnicas que determinam o comportamento de objetos e/ou personagens do jogo durante a movimentação dos mesmos, essas técnicas pertencem ao conjunto de técnicas que são chamadas de A-Life (*Artificial Life*);
- **Sistemas baseados em regras:** são sistemas onde existe um conjunto de parâmetros e um conjunto de regras, que trabalham sobre os parâmetros de modo a processarem essas regras e exibirem uma saída que pode ser uma ação do personagem;
- **Lógica Fuzzy:** a lógica *fuzzy* ou difusa é uma lógica que admite valores intermediários entre o falso e o verdadeiro, como o “talvez”. Esse tipo de lógica analisa as sentenças de forma incerta e procura representar as tomadas de decisão humana;

Muitos estudos foram realizados na área, e segundo Russel e Norvig (2004), IA continua a ser estudada pelos pesquisadores com diversas aplicações como: Planejamento autônomo e escalonamento, controle-autônomo, diagnóstico médico, planejamento logístico, robótica, reconhecimento de linguagem e resolução de problemas. Há uma diferença entre a IA estudada e aplicada nas áreas citadas acima e a que é aplicada nos jogos. Segundo Kishimoto (2004), a inteligência artificial aplicada nos jogos de computador tem um objetivo mais prático que a IA aplicada no meio acadêmico, enquanto no meio acadêmico busca-se utilizar IA para resolução de problemas complexos, como reconhecimento de sinais e imagens, no

desenvolvimento de jogos de computador a inteligência artificial é utilizada para se obter um jogo mais interessante e divertido para o consumidor.

3 – Técnicas de IA utilizadas no desenvolvimento de jogos de computador

Neste capítulo serão apresentadas as principais técnicas de inteligência artificial utilizadas nos jogos de computador, será explicado o funcionamento de cada uma delas e também serão discutidas as causas do sucesso do seu uso.

3.1 – Máquinas de Estado Finito

As máquinas de estado finito (*Finite State Machines – FSM's*) são a forma mais comum e utilizada para representar o comportamento dos personagens de um jogo.

Segundo Bourg e Seemann (2004), uma máquina de estado finito é uma máquina abstrata que pode existir em um ou vários diferentes e predefinidos estados. Máquinas de estado finito também definem um conjunto de condições que determinam quando o estado deve mudar. O estado atual determina como a FSM deve se comportar, os estados são as ações possíveis que o personagem pode executar, e as transições entre os estados são feitas quando a condição de mudança de estado é alcançada, assim ocorre a transição e a máquina passa a operar a partir daquele estado vigente.

A seguir é apresentada uma figura que ilustra uma máquina de estado finito utilizada para representar o comportamento de um personagem em jogos de FPS (*First Person Shooter – Tiro em primeira pessoa*).

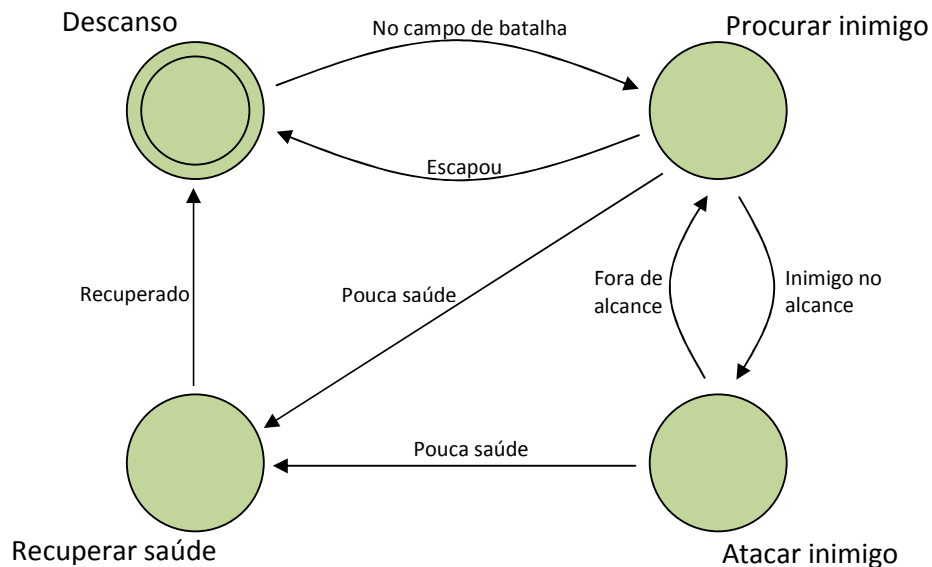


Figura 1: Máquina de estados finitos para um personagem de jogo FPS.

No início da execução do comportamento, a máquina encontra-se no seu estado inicial; tomando como exemplo a FSM representada pela figura acima, o estado inicial é “Descanso”; a cada iteração as condições de transição são avaliadas, se alguma delas for satisfeita a transição baseada nesta regra é efetuada, o novo estado passa a ser o estado vigente e a máquina passa a operar a partir daquele estado.

Exemplificando com base na Fig. 1, o personagem encontra no seu estado inicial que é “Descanso”, quando ele entra no campo de batalha a condição para procurar o inimigo é satisfeita e ocorre a transição do estado de “Descanso” para “Procurar o inimigo”, assim o agente passa a perseguir o oponente. Quando o inimigo entra no alcance do personagem uma nova condição é satisfeita e ocorre uma nova mudança de estado, desta vez de “Procurar o inimigo” para “Atacar o inimigo”, a cada transição a máquina passa a operar daquele estado que ela se situa, voltando ao raciocínio, quando o inimigo fica fora de alcance ocorre a transição do estado “Atacar o inimigo” para “Procurar o inimigo” e o agente novamente passa a perseguir o seu oponente.

Segundo Karlsson (2005), as FSM's são bastante utilizadas porque são capazes de controlar o comportamento de agentes necessitando de pouco poder de processamento além de “ser fácil e intuitivo definir comportamentos por meio dessa abordagem, principalmente com o auxílio de ferramentas visuais.”. Elas também são fáceis de entender, programar e depurar, por isso são frequentemente usadas no desenvolvimento de jogos de computador.

As máquinas de estados finitos apresentam limitações bem conhecidas para o seu uso. O número de estados e transições possíveis são diretamente proporcionais à complexidade do ambiente, por isso quando há um ambiente complexo fica difícil para a FSM prever todos os casos e situações presentes. Outra limitação das FSM's segundo Karlsson (2005) é o fato de ter comportamentos potencialmente repetitivos, isso surge da definição da própria máquina possuir estados finitos, ou seja, um número fixo de estados, assim quando uma mesma situação acontece duas vezes, o estado vigente é o mesmo nas duas ocasiões.

Para amenizar o problema quando há um aumento da complexidade do ambiente, Karlsson (2005) menciona uma técnica já utilizada, a criação de HFSM's (*Hierarchical Finite State Machines*); que são máquinas de estados finitos hierárquicas; onde cada novo estado pode representar uma nova FSM. Contudo esse problema não é eliminado e o problema da repetição de comportamentos não é tratado por essa técnica.

3.2 – Path-Finding

Nos jogos de computadores os NPC's se movimentam de um ponto ao outro e durante essa movimentação eles têm de se desviar de obstáculos, demonstrando um sinal de inteligência. Para se determinar qual caminho que o NPC deve seguir faz-se o uso de algoritmos de *path-finding* (busca de caminho).

Segundo Karlsson (2005), o modo como é feita essa busca de caminho é executando um algoritmo de busca sobre os dados do cenário para encontrar o melhor caminho entre a posição de origem e posição de destino.

O algoritmo A* foi descrito pela primeira vez em 1968 por Peter Hart, Nils Nilsson, e Bertram Raphael e foi chamado inicialmente de algoritmo A. Por ser um algoritmo fácil de implementar e também por ser rápido para calcular o menor caminho entre dois pontos, o A* é um dos algoritmos de busca de caminho mais utilizados no desenvolvimento dos jogos.

Para se efetuar a busca de caminho os cenários geralmente são organizados em formas de nós no grafo em que os objetos e personagens podem ocupá-los, mas essa representação pode variar dependendo do tipo de jogo. Quando o jogo utiliza um ambiente contínuo, o número de nós é muito grande e a busca de caminho toma muitos ciclos de CPU (*Central Processing Unit* – unidade central de processamento). Por isso Karlsson (2005) sugere a realização de “um balanceamento entre a qualidade do caminho encontrado e o tempo de processamento”. Ainda segundo Karlsson (2005), o A* demonstra dificuldades ao lidar com mudanças e objetivos dinâmicos no cenário. Para melhorar o desempenho do A* é feita a simplificação da área de busca diminuindo o número de nós existentes que são marcações no ambiente que ajudam os NPC's a se moverem e os nós, também chamados de *waypoints*, podem representar pontos de interesse para o NPC, como a entrada de uma casa. Essa representação através de *waypoints* é bastante útil para diminuir as chances de ocorrerem problemas como personagens fiquem vagando pelo cenário ou presos nos cantos do mesmo.

3.3 – Padrões de Movimento

Em um jogo de computador os NPC's estão constantemente se movimentando no cenário, e isso é feito com o uso dos algoritmos de busca de caminho citados anteriormente. Durante essa movimentação é necessário que cada NPC se comporte de uma certa maneira para aquela determinada situação, e os padrões de movimento são uma maneira de gerar a ilusão de um comportamento inteligente durante o trajeto.

Karlsson (2005) fala da necessidade de se minimizar a carga da CPU e maximizar a aparente inteligência do movimento. Além de implementar comportamentos inteligentes que possam ser reutilizados melhorando a experiência do jogador.

Segundo Bourg e Seemann (2004), o modo como os padrões de movimento são implementados funciona da seguinte maneira: é escolhido o padrão desejado e este é codificado na central de dados dentro de uma matriz ou conjunto de matrizes. A central de dados consiste em instruções específicas de movimento, tais como andar pra frente e girar e a intensidade de que o NPC ou objeto deve se mover utilizando aquele padrão. Utilizando esses algoritmos é possível criar diversos tipos de movimentos como movimento em círculo, em forma de um quadrado, ziguezague e qualquer tipo de movimento que o desenvolvedor consiga programar no conjunto de instruções de movimento.

Segundo Karlsson (2005), “os esforços de pesquisas sobre essa área têm relacionamento e implicações em robótica, IA e vida artificial, sendo geralmente voltadas para animação, ou seja, modelagem de comportamento está localizada na interseção entre animação e os três campos citados.”.

Ainda segundo Karlsson (2005), o uso combinado entre *path-finding* e os padrões de movimento permite criar jogos onde os NPC's e objetos tenham uma autonomia para a sua movimentação, tratando também de problemas de desvio de obstáculos dinâmicos, fazendo com que o jogo fique mais real e interessante.

3.4 – Sistemas baseados em regras

Os sistemas baseados em regras; RBS (*Rule Based Systems*); segundo Bourg e Seemann (2004), são provavelmente os sistemas de IA mais amplamente utilizados tanto para o mundo real quanto para as aplicações em IA de jogos.

Conforme dito anteriormente, os RBS's consistem em um conjunto de regras que são utilizadas para fazer inferências ou decidir a tomada de ações. Uma FSM pode parecer com um RBS, pois as FSM's se baseiam em regras para a ocorrência das transições, mas veremos que existem diferenças entre uma máquina de estado finito e um RBS. Já a lógica *fuzzy*; assunto do próximo tópico; é um tipo de sistema baseado em regras.

O funcionamento de uma RBS é simples, ela faz uso de um conjunto de parâmetros que são trabalhados por um conjunto de regras que no final exibem uma saída, geralmente uma ação do personagem ou comportamento que ele ou algum objeto deverá ter.

Para entender melhor, usar-se-á o exemplo visto em Karlsson (2005) de um jogo FPS. Supondo a existência de um conjunto de parâmetros: {saúde do personagem, saúde do

inimigo, distância do inimigo, quantidade de munição} e também a existência de um conjunto de ações que o agente possa executar: {atacar o inimigo, fugir, procurar o inimigo}.

A ação que o agente poderá executar será tomada ao se verificar as condições que as regras impõem para a execução daquela ação. Um exemplo seria:

SE (distância do inimigo < 50 **E** saúde do agente > 50)
ENTÃO atacar o inimigo

As condições são interligadas por conjunções lógicas como o E e OU. A ação “atacar o inimigo” somente será executada se ambas as condições forem satisfeitas. Se uma delas não for verdadeira, a ação não é executada e então são verificadas as outras regras.

Os sistemas baseados em regras segundo Bourg e Seemann (2004), são utilizados em diagnósticos médicos, proteções contra fraudes e também na análise de falhas de engenharia. Uma das vantagens apresentadas por esse sistema é “a capacidade deles imitarem o modo como as pessoas tendem a pensar e a razão de um conjunto de fatos conhecidos e os seus conhecimentos sobre o problema específico do domínio.” (Bourg e Seemann, 2004). Além disso, os RBS’s são razoavelmente fáceis de programar e gerenciar. Segundo Karlsson (2005), os sistemas baseados em regras, permitem a modelagem de comportamentos complexos e são mais concisos que máquinas de estados finitos.

Apesar de ser vantajoso usar os RBS’s, eles podem apresentar problemas, podendo “necessitar de muito espaço em memória além de muito poder de processamento.” (Karlsson, 2005).

3.5 – Lógica Fuzzy

As pessoas geralmente analisam as situações de um modo impreciso, sem ter absoluta certeza. Um exemplo de afirmação que ilustra essa incerteza é: “Pedro é muito baixo” ou “Denise é pouco confiável.”. A lógica *fuzzy* é capaz de representar essas situações de incerteza, pois ela permite o uso de valores intermediários entre o que é totalmente verdadeiro e o totalmente falso. Karlsson (2005) exemplifica o uso de lógica *fuzzy*, em um suposto jogo onde um certo personagem pode ter seu estado emocional pertencente ao conjunto feliz com grau de pertinência 0.7 e ao conjunto frustrado com grau 0.5.

A tradicional lógica booleana só admite valores como “sim” ou “não”, “verdadeiro” ou “falso”, e por isso fica difícil, mas não impossível, de representar situações naturais onde não é possível ter certeza absoluta. O problema de utilizar a lógica booleana é que para apresentar

repostas intermediárias é necessário o uso de muitos estados, o que leva ao perigo de “haver explosões da quantidade de regras” (Karlsson, 2005).

A lógica difusa por ser capaz de tratar o grau de “verdade” das situações permite lidar com problemas complexos com custo computacional relativamente baixo, (Karlsson, 2005) além de possuir outras vantagens como “síntese, adaptação, flexibilidade e versatilidade, importantes na modelagem da inteligência artificial em um jogo” (Karlsson, 2005).

Karlsson (2005), cita que pesquisadores buscaram resolver os problemas de comportamentos previsíveis das FSM's criando as *Fuzzy* FSM's (FuSM's), onde nas regras de transição existissem valores *fuzzy* para determinar a mudança ou não de estado. Segundo Karlsson (2005), alguns jogos fizeram uso de FuSM's dentre eles: “Unreal”, “S.W.A.T. 2” e “Civilization: Call to Power”.

4 – Futuro da IA nos jogos de computador

Segundo Bourg e Seemann (2004), o próximo grande passo da IA nos jogos é a aprendizagem. Ao invés de se ter o comportamento do NPC predefinido, os jogos devem envolver, aprender e adaptar os comportamentos na medida em que ele é mais jogado. Isso resulta em um jogo que cresce e melhora a sua qualidade e dificulta pro jogador prever as situações, aumentando assim o tempo de vida do jogo.

5 – Conclusão

Ao final do trabalho obteve-se o conhecimento das principais técnicas de IA aplicadas no desenvolvimento dos jogos de computador e foi constatado que essas técnicas são de extrema importância para a qualidade final do jogo, visto que a IA é o ramo da computação que tem mais poder de prover realidade aos elementos do jogo. Essa “realidade” gerada pela IA é o que prende o jogador, pois quanto mais situações imprevisíveis o jogo possui mais próximo da realidade ele se situa e o tempo de vida do mesmo aumenta e como foi visto no final do artigo, o futuro da IA nos jogos de computador é tentar fornecer o maior número de situações imprevisíveis para o jogador, aproximando-se cada vez mais da realidade, uma vez que todos são sujeitos à situações adversas na vida real.

6 – Referências bibliográficas

Luz, M. H. G. C. da. *Desenvolvimento de Jogos de Computador – Trabalho de Conclusão de Curso*, Universidade Federal de Itajubá, 2004.

Bourg, D. M.; Seemann, G. *AI for Game Developers*. – O'Reilly, 2004.

Schwab, B. *AI Game Engine Programming* - Hingham: Charles River Media, 2004.

PJMOO, Disponível em:

<http://pjmo0.codigolivre.org.br/wiki/index.php/Industria_de_Jogos>, visitado em 24/10/2007.

Russel, S. J.; Norvig, P. *Inteligência Artificial* – Editora Campus, 2004.

Battaiola, A. L. *Jogos por Computador – Histórico Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação* – Anais da XIX Jornada de Atualização em Informática, 2000.

The American Heritage Dictionary of the English Language – Houghton Mifflin Company.

Karlsson, B. F. F. *Um Middleware de Inteligência Artificial para Jogos Digitais* – Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, 2005.

Kishimoto, A. *Inteligência Artificial em Jogos Eletrônicos*, 2004. Disponível em: <http://www.programadoresdejogos.com/trab_academicos/andre_kishimoto.pdf>, visitado em 24/10/2007.