

Inteligência Artificial em Jogos Eletrônicos

André Kishimoto
<http://www.tupinihon.com>
<http://www.tupinihon.info>
kishimoto@tupinihon.com

RESUMO

Pesquisa sobre o uso da inteligência artificial em jogos eletrônicos para computadores e videogames, desde o início da indústria de jogos até os dias atuais, incluindo a história, evolução e estado da arte da inteligência artificial aplicada aos jogos eletrônicos; quais técnicas e algoritmos são utilizados para a resolução dos problemas encontradas nos jogos eletrônicos e que benefícios (e conseqüências) a inteligência artificial traz a eles.

Palavras-chave

Inteligência artificial, jogos eletrônicos, computadores, videogames, indústria de jogos.

ABSTRACT

Research about the use of artificial intelligence in electronic games for computers and consoles, since the beginning of the game industry until recently, including the history, the evolution and the state of the art of artificial intelligence applied in electronic games; which techniques and algorithms are used to solve the problems found in electronic games and what benefits (and consequences) the artificial intelligence bring to it.

Keywords

Artificial intelligence, electronic games, computers, consoles, game industry.

1 – Introdução

De acordo com o dicionário *Oxford* (WEHMEIER, 2000), *artificial intelligence* (inteligência artificial, ou simplesmente IA) corresponde a uma área de pesquisa sobre computadores simulando o comportamento humano inteligente. Para a grande maioria da população, IA é o cérebro por trás de máquinas poderosas, como as encontradas em filmes de ficção científica, enquanto que para os acadêmicos, é uma fonte infinita de desafios e estudos sobre como recriar um ser inteligente através do uso de computadores (CHAMPANDARD, 2003).

1.1 – *Game AI* (IA para jogos eletrônicos)

Para os desenvolvedores de jogos eletrônicos, as aplicações computacionais de IA e o significado do termo IA são diferente dos encontrados no meio acadêmico. Para distinguir a inteligência artificial utilizada em jogos e no meio acadêmico, os desenvolvedores adotaram o termo *Game AI* (FUNGE, 2004).

A principal diferença entre a IA acadêmica e a IA para jogos é o objetivo que cada uma busca. No primeiro caso, o objetivo é buscar a solução para problemas extremamente difíceis, como imitar o reconhecimento que os humanos são capazes de realizar (reconhecimento facial e de imagens e objetos, por exemplo), entender e construir agentes inteligentes (SCHWAB, 2004). No

segundo caso, o objetivo de usar inteligência artificial é a diversão. Sua importância é quanto aos resultados que o sistema irá gerar, e não como o sistema chega até os resultados; ou seja, o problema não é como o sistema pensa, mas sim como ele age. Isso se deve pelo fato que jogos eletrônicos são negócios – os consumidores desses produtos os compram em busca de diversão, e não lhes interessa como a inteligência de um personagem no jogo foi criada, desde que ela transforme o jogo divertido e desafiador, além, claro, de tomar decisões coerentes com o contexto do jogo (TOZOUR, 2002) (SCHWAB, 2004).

Um dos problemas encontrados sobre IA na indústria de jogos eletrônicos é a grande variedade de gênero dos jogos existentes e os comportamentos dos personagens, resultando numa interpretação ampla do que é considerada IA para jogos. Há desenvolvedores que consideram a interface do jogo com o usuário parte da área de IA, assim como outros consideram algoritmos de movimento e colisão também como IA (BOURG, 2004). Para (TOZOUR, 2002), é até vergonhoso que *Game AI* seja chamada e considerada inteligência artificial, uma vez que no campo de IA para jogos é necessário criar agentes com comportamentos apropriados num determinado contexto, embora a adaptabilidade da inteligência humana nem sempre é necessária ou desejada para produzir tais comportamentos.

Embora exista esse problema, um fato que surpreendeu muitas pessoas da indústria ocorreu durante a mesa-redonda de IA na *Game Developers Conference* (GDC) de 1999: diversos membros da parte de pesquisa acadêmica sobre IA estavam presentes no evento e puderam compartilhar e discutir idéias sobre o

assunto com os desenvolvedores de jogos. Durante o evento, alguns desses pesquisadores admiraram a rápida evolução de desenvolvimento da indústria de jogos, dizendo que a indústria de jogos parecia estar anos-luz à frente do meio acadêmico com relação à construção de soluções práticas de IA para certos problemas, que estudos formais de IA podem levar anos para formular teorias de comportamento, examinar possíveis soluções e desenvolver protótipos para testes. Porém, a falta de uma metodologia rigorosa faz com que muitas das soluções encontradas pelos desenvolvedores de jogos não sejam aceitas como um apoio aos estudos formais de IA (WOODCOCK, 1999).

2 – Breve História dos Jogos Eletrônicos

A história dos jogos eletrônicos envolve milhares de pessoas, empresas, produtos e acontecimentos, conforme pode ser visto em extensas literaturas, como (KENT, 2001), (KUSHNER, 2003) e (DEMARIA, 2004). Visto que o foco desse artigo é a IA utilizada em jogos, será abordada apenas uma breve história dos jogos eletrônicos, citando os principais acontecimentos ao longo dos anos.

A história dos jogos pode ser dividida em: antes da década de 70, décadas de 70, 80, 90 e 2000, e através das diferentes tecnologias utilizadas em diferentes épocas. Antes da década de 70, algumas empresas e pessoas já começavam a dar os primeiros passos na criação de jogos. Uma das principais empresas da indústria, a *Nintendo*, começou como uma empresa que fabricava cartas de baralho em 1889. Segundo a linha de tempo encontrada em (KENT, 2001), esse é o primeiro marco da história dos jogos eletrônicos (pelo

fato que a *Nintendo* hoje está no mercado de jogos eletrônicos). Ainda, desde o início do século XX, (KENT, 2001) e (DEMARIA, 2004) consideram máquinas como a *Gottlieb Baffle Ball* os precursores das máquinas de *pinball*, que hoje em dia são máquinas de jogos eletrônicos.

O primeiro jogo eletrônico interativo criado na história foi o *Spacewar*, um jogo onde duas pessoas controlavam dois tipos diferentes de espaço-ave que deveriam combater entre si. Esse jogo foi programado por um estudante do MIT, *Steve Russell*, em um computador PDP-1 em 1961 (DEMARIA, 2004). Existem alguns historiadores, porém, que argumentam que o primeiro jogo eletrônico foi criado por *Willy Higinbotham*, um cientista do *Brookhaven National Laboratory*. Higinbotham programou, em 1958, um osciloscópio onde era possível jogar uma partida de tênis interativa (KENT, 2001).

Em 1970, *Nolan Bushnell* começou a trabalhar em uma versão fliperama do jogo *Spacewar*, chamada *Computer Space*. No ano seguinte, a empresa *Nutting Associates* comprou o jogo de Bushnell, colocando no mercado a primeira máquina de fliperama da história. Em 1972, Bushnell abre sua própria empresa, a *Atari*, famosa pelo jogo *Pong* (criado pelo engenheiro *Al Alcorn*). Até o fim da dessa década, muitas empresas entraram no mercado de jogos, como *Taito*, *Midway* e *Capcom*, além da *Magnavox* lançar em 1972 o computador *Odyssey* (KENT, 2001).

Seguindo a linha de tempo dos jogos em (KENT, 2001), nos anos 80 as máquinas de fliperama estavam em seu auge, com muitos jogos sendo lançados (*Donkey Kong*, *Tron* e *Q*Bert* são alguns

exemplos) ao mesmo tempo em que surgiram os primeiros videogames 8-bit: *Famicom*, da *Nintendo* e *Master System*, da *SEGA* (*Service Games*). Na área de jogos para computador, houve também um grande lançamento de jogos e criação de empresas, sendo a *On-Line Systems* (atual *Sierra Online*) uma das pioneiras no setor de jogos para computador (DEMARIA, 2004). Em março de 1986, *Chris Crawford* (CRAWFORD, 2003) reuniu alguns amigos desenvolvedores em sua casa para a primeira *Computer Game Developers Conference* (atual GDC), onde discutiram assuntos sobre *game design* e negócios.

A década de 90 foi marcada pelo lançamento e batalhas de videogames de 16-bit (*Sega Genesis* e *Super Famicom*, da *Nintendo*), de 32-bit (*PlayStation* da *Sony* e *Sega Saturn*) e o lançamento de videogames como *3DO* da *Panasonic* e *Nintendo64* da *Nintendo* (esse de 64-bit) (KENT, 2001). Um grande marco para a história de jogos para computador foi o lançamento do jogo *Wolfstein 3D* pela *id Software* em 1991, o primeiro jogo de tiro em primeira pessoa, atualmente um dos gêneros de jogos mais famosos entre os jogadores (KUSHNER, 2003).

No final da década de 90 para 2000, *Sony* e *Nintendo* divulgaram seus novos videogames de 128-bit (*PlayStation 2* e *GameCube*, respectivamente), enquanto a *Microsoft* também entrou para esse segmento. Atualmente, a atenção da mídia e dos jogadores está voltada aos jogos *online multiplayer* massivos (centenas a milhares de pessoas jogando ao mesmo tempo num mundo virtual via internet) e aos videogames portáteis *Sony PSP* e *Nintendo DS* (esse inovando com duas telas, sendo uma delas sensíveis ao toque, como em PDA's) (EGM, 2004).

3 – IA e Jogos Eletrônicos

Conforme já mencionado, a IA para jogos não é considerada a mesma que estudada e pesquisada no meio acadêmico. No começo do desenvolvimento de jogos eletrônicos, a programação de IA era mais usualmente conhecida por “programação

de jogabilidade”, pois não havia nada de inteligente sobre os comportamentos exibidos pelos personagens controlados pelo computador (SCHWAB, 2004). A tabela 1 contém alguns exemplos de como a IA foi utilizada em jogos com o passar do tempo.

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

Tabela 1: Linha de tempo da IA em jogos.
Fonte: (SCHWAB, 2004)

3.1 – O início da IA em jogos

Muitos programadores do início da era de jogos eletrônicos implementavam padrões de movimentos ou movimentos repetitivos e/ou aleatórios para os personagens controlados pelo computador (como *Galaga* e *Donkey Kong*) como sendo a inteligência existente no jogo. Esse fato foi principalmente causado pela falta de memória e limitação existente na

velocidade de processamento (SCHWAB, 2004).

Os jogos de estratégia (*Civilization* de 1991, por exemplo) estão entre os pioneiros em IA para jogos, uma vez que tais jogos necessitam de uma boa IA para que sejam jogáveis, pois requerem que o computador controle unidades (grupo de personagens) com estratégias e táticas

complexas. Uma extensão dos jogos de estratégia são os jogos de estratégia em tempo real, onde toda a ação acontece em tempo real (ao contrário de outros jogos de estratégia, que ocorre em turnos). A IA para esse gênero de jogo deve realizar buscas de caminhos (*pathfinding*) para centenas de unidades em tempo real (TOZOUR, 2002).

Jogos do gênero “sims” (simuladores de gestão de cidades, fazendas, relações pessoais, entre outros), como o clássico *SimCity*, lançado pela empresa *Maxis* em 1989, foram os primeiros a provarem o potencial dos métodos de *Artificial Life (A-Life)*. Outro jogo famoso da *Maxis*, *The Sims* (2000), conta com personalidades profundas em seus agentes inteligentes. Tal jogo é exemplo do potencial uso de máquinas de estado *fuzzy (fuzzy-state machines* ou *FuSM*) e *A-Life*. Outro exemplo do uso de *A-Life* em jogos é o título *Creatures* (criado pela *CyberLife* em 1996), que simula a psicologia e fisiologia dos personagens do jogo, incluindo um “DNA digital” único de cada personagem (TOZOUR, 2002).

Em jogos de tiro de primeira pessoa (*first-person shooter* ou *FPS*), como *Half-Life (Valve)* e *Unreal: Tournament* (lançado em 1999 pela *Epic MegaGames*), a IA ficou conhecida pelo excelente nível tático dos inimigos, desenvolvida através do uso de máquinas de estado finito (*finite state machines* ou *FSM*) e *scripts* que determinam como um agente inteligente deve agir em várias situações (WOODCOCK 1999) (TOZOUR, 2002).

3.2 – Técnicas e algoritmos de IA implementada em jogos

Existem diversas técnicas e algoritmos utilizados pelos desenvolvedores de jogos

para dar aos personagens uma certa inteligência (ou ao menos fazer com que os personagens pareçam ser inteligentes) e uma personalidade (BOURG, 2004). Segundo (LAMOTHE, 1999), um dos princípios básicos de IA para jogos são os algoritmos de IA determinísticos e padrões de movimento, onde os comportamentos são pré-programados ou pré-processados. Ainda, (DALMAU, 2004) cita quatro tipos principais de IA que são implementadas em jogos: máquinas de estado, sistemas baseados em regras, algoritmos de busca e algoritmos genéticos.

3.2.1 – Algoritmos de IA determinísticos e padrões de movimento

Os algoritmos de IA determinísticos, junto com padrões de movimento, foram utilizados nos primeiros jogos eletrônicos da história, e são compostos por movimentos aleatórios, algoritmos de perseguição e evasão. Movimentos aleatórios podem ser implementados simplesmente obtendo um valor aleatório e incrementando a posição de um personagem com tal valor. O algoritmo de perseguição verifica a posição de um personagem 1 em relação à posição de um personagem 2, e avança em direção a ele. O algoritmo de evasão faz o personagem 1 se distanciar do personagem 2. Os padrões de movimento fazem com que um personagem se movimente em um determinado padrão, por exemplo, um personagem pode fazer uma ronda em uma área retangular (LAMOTHE, 1999).

3.2.2 – Máquinas de estado

Uma máquina de estado finita é uma máquina abstrata que define os estados em que um personagem pode se encontrar e quando o mesmo muda de estado. O estado atual da máquina determina como o personagem deve atuar no jogo.

Máquinas de estado foram usadas no início da criação de jogos (com IA) e são usadas até hoje por serem de fácil entendimento, implementação e depuração de erros. No jogo *Pac-man*, por exemplo, uma máquina de estado é implementada para cada fantasma do jogo. Um fantasma pode estar nos seguintes estados: “procurando jogador”, “perseguido jogador” e “fugindo do jogador”. Quando o fantasma está procurando o jogador, ele apenas se movimenta pelo labirinto até encontrar o jogador. Quando ele se depara com o jogador, verifica se ele pode perseguir o jogador ou se precisa fugir (isso acontece quando o jogador obtém poder de “engolir” o fantasma), e troca de estado conforme a situação. Se o fantasma pode seguir o jogador, ele muda seu estado para “perseguido jogador” e tenta alcançar o jogador. Se durante esse tempo o jogador ganha a habilidade de engolir o fantasma, o fantasma muda seu estado para “fugindo do jogador” (BOURG, 2004).

Os desenvolvedores também utilizam a lógica *fuzzy* em máquinas de estado *fuzzy* para criar resultados de ações que são menos previsíveis e para reduzir o grande trabalho de enumerar a grande quantidade de regras if-then. A lógica *fuzzy* permite criar regras usando condições menos precisas, criando agentes com um conhecimento imperfeito, uma vez que essa lógica é baseada em níveis de incerteza e verdades em uma sentença (WOODCOCK, 1999) (BOURG, 2004).

3.2.3 – Sistemas baseados em regras

Alguns fenômenos não são fáceis de serem modelados em termos de estados e transições. Considerando como exemplo os seguintes fenômenos de um cachorro virtual:

- Se há um osso por perto e o cachorro está com fome, ele irá comê-lo;
- Se o cachorro está com fome, mas não há nenhum osso por perto, ele procura por um;
- Se o cachorro não está com fome, mas está com sono, ele irá dormir;
- Se o cachorro não está com fome e não está com sono, o cachorro irá andar e latir.

Essas quatro sentenças são difíceis de serem representadas através de uma máquina de estados, pois cada sentença leva a um estado da máquina e cada estado pode transitar para qualquer um dos outros estados. Esse tipo de problema é conhecido por comportamento global. Máquinas de estado são úteis para situações locais (onde dado um estado, apenas algumas condições podem ser aplicadas como saída).

Nesse exemplo, o cachorro se comporta de acordo com um conjunto de prioridades ou regras. Um sistema baseado em regras tem a forma “Condição → Ação”. No exemplo citado, as regras teriam a seguinte forma:

- Fome & osso por perto → comer;
- Fome & não osso por perto → procurar;
- Não fome & com sono → dormir;
- Não fome & sem sono → andar e latir.

Dessa maneira, são especificadas as condições que ativam as regras assim como quais ações devem ser tomadas caso a regra é ativada (DALMAU, 2004).

3.2.4 – Algoritmos de busca

Busca é um dos problemas mais básicos de IA para jogos. Quando um jogo implementa uma busca pobre (ou “burra”), o resultado é personagens que

parecem totalmente artificiais e sem inteligência de navegar entre locais e desviar de obstáculos, o que acaba com a imersão do jogo e a diversão (BOURG, 2004).

Para solucionar o problema de busca (sair de um ponto e chegar a um destino), diversos algoritmos podem ser utilizados, sendo o algoritmo A^* o mais famoso e implementado em jogos, embora soluções como o algoritmo de *Dijkstra* e *waypoints* também são utilizados (LAMOTHE, 1999) (DALMAU, 2004). Em muitos jogos, os desenvolvedores representam o mundo virtual por onde um personagem caminha através de “grades” (*grids*), onde cada célula pode representar um nó de um grafo. Um custo é associado para cada célula do *grid*, utilizado pela heurística do A^* (LAMOTHE, 1999).

Como o uso de busca pode consumir muito tempo do processador, é possível contornar esse problema através de caminhos pré-calculados, chamados de *waypoints*, quando o jogo permite esse tipo de solução. Os *waypoints* são nós em locais do mundo virtual que auxiliam no deslocamento de um lugar (nó atual) para outro (nó destino) através de caminhos pré-calculados ou métodos de busca de baixo custo (BOURG, 2004).

3.2.5 – Algoritmos genéticos

(DALMAU, 2004) cita que um dos usos de algoritmos genéticos em jogos pode ser a geração de uma população, criando diferentes indivíduos de acordo com um DNA virtual, sendo esse representado por um vetor de valores, cada um sendo um parâmetro da espécie a ser modelada. Essa técnica pode ser utilizada para a criação de pedestres em um jogo onde o mundo virtual seja uma cidade. (DALMAU, 2004) e (LAMOTHE, 1999) também citam o uso de algoritmos

genéticos para mutação ou evolução de personagens.

3.2.6 – Outras técnicas

A IA para jogos não pára por aqui. Existem outras técnicas que não foram discutidas e que são aplicadas nos jogos. O uso de *A-Life* é uma delas. Alguns jogos utilizam algoritmos de *flocking* para simular o movimento em grupo de monstros, pássaros, peixes, entre outros (WOODCOCK, 1999). A implementação de redes neurais em jogos também é realizada, onde os personagens necessitam de aprendizado através das escolhas do jogador, como no jogo *Black & White* (um gênero onde o jogador assume a posição de deus e controla o ambiente do jogo) (BOURG, 2004). Ainda, há jogos (por exemplo, *Baldur's Gate* e *Unreal*) que implementam a IA através de *scripts*, possibilitando que qualquer pessoa possa criar novos tipos de NPC's (*non-player characters*) ou modificar um personagem já existente de acordo com o seu estilo de jogo. Esse tipo de IA (também conhecido por *Extensible AI*) é baseada fortemente em sistemas de regras (WOODCOCK, 1999).

3.3 – Benefícios do uso de IA em jogos

O principal benefício que o uso de IA em jogos pode propiciar ao desenvolvimento de jogos é o fator diversão. Os personagens de um jogo devem simular inteligência e erros humanos e ter personalidades, devem ser capazes de fornecer diferentes níveis de dificuldade ao jogador, para que o mesmo se sinta desafiado. Além disso, os jogadores cada vez mais demandam melhores oponentes em jogos mais complexos (SCHWAB, 2004). A IA em jogos aumenta a experiência e imersão do jogo, melhorando sua jogabilidade. NPC's inteligentes fazem com que a criação de jogos *single-player* (para um jogador)

seja possível, além de melhorar a experiência em jogos *multiplayer* (para vários jogadores) sem a necessidade de se ter uma comunidade de pessoas (reais) atuando durante o jogo. Os *NPC's* inteligentes são necessários a qualquer gênero de jogo para criar a ilusão que o jogador está num mundo com outros jogadores inteligentes, além de adicionar uma profundidade ao jogo (CHAMPANDARD, 2003).

O uso da IA também pode trazer vantagens de desenvolvimento de jogos; segundo (CHAMPANDARD, 2003), o jogo *Colin McRae Rally 2* utiliza redes neurais e aprendizado, não sendo necessário assim a programação manual da IA (uma vez que o jogo aprende como os carros devem se comportar durante as corridas). Outros exemplos de como a IA pode ser aplicada em jogos é encontrada nos artigos dos livros da série *AI Game Programming Wisdom* e da série *Game Programming Gems*, ambos da *Charles River Media*. No volume 4 da série *Gems* (KIRMSE, 2004), pode-se estudar o uso de IA para realizar a navegação tridimensional de uma câmera, assim como o uso da IA pode aumentar a tensão em um jogo de ação e realizar decisões feitas por *NPC's*.

3.4 – Problemas envolvendo IA e jogos

Embora os desenvolvedores tenham encontrado muitas soluções através da implementação de IA nos jogos, muitos problemas também aparecem pelo uso de IA em jogos. Quatro fatores podem ser citados como principais problemas da IA para jogos:

- Período de desenvolvimento: o curto período de desenvolvimento dos jogos dificulta o aprendizado dos desenvolvedores para tecnologias de ponta sobre IA e

suas aplicações em um jogo comercial (BOURG, 2004);

- Algoritmos de aprendizado: os resultados produzidos por algoritmos de aprendizado são difíceis de serem testados e depurados contra erros, visto que os resultados não são previsíveis. Por essa razão, muitos desenvolvedores têm preferência por técnicas de IA determinísticas, e também por serem mais conhecidas pelos mesmos e de fácil implementação e depuração (WOODCOCK, 1999);
- Poder de processamento: jogos são *softwares* com execução em tempo real, onde informações são processadas e atualizadas a cada ciclo de máquina. Algoritmos com alto custo de processamento (ainda) não podem ser implementados em jogos que necessitam de respostas em tempo real (TOZOUR, 2002);
- Jogos são baseados em *game design*: durante o desenvolvimento de um jogo, toda equipe baseia-se no documento de *game design* do jogo. No entanto, existe um confronto entre *game designers* e *game AI*, pois os *game designers* constroem a narrativa do jogo e definem a jogabilidade e eventos do jogo, tendo um controle explícito do jogo e dos *NPC's*. Surge então o seguinte conflito: os *designers* controlam o comportamento dos *NPC's*. É preciso então o uso de IA? Em contrapartida, se a IA estiver presente (*NPC's* inteligentes podem se comportar com autonomia), é preciso ter *designers*? A solução para esse caso é os *game designers* e programadores de IA entrarem em

um acordo sobre o controle sobre os NPC's do jogo (CHAMPANDARD, 2003).

3.5 – Estado da arte

O estado da arte de IA para jogos é um assunto de certa forma difícil de ser discutido. Conforme (TOZOUR, 2004) explica em seus seminários sobre *Introduction to Game AI*, a inteligência artificial em jogos não é um problema único, é um vasto conjunto de problemas. Não somente isso, como também é um conjunto diferente de problemas para cada jogo. (TOZOUR, 2004) cita ainda que é surpreendentemente fácil encontrar 50 jogos diferentes aonde o termo IA significa 50 coisas completamente diferentes. Porém, é possível citar as principais técnicas mais utilizadas entre os desenvolvedores de jogos.

De acordo com a pesquisa realizada por (WOODCOCK, 1999), as técnicas mais populares e utilizadas pelos desenvolvedores de IA são as máquinas de estado finito (*FSM*) e as máquinas de estado *fuzzy* (*FuSM*). (TOZOUR, 2002) indica que o algoritmo de busca A^* é a principal técnica de busca que está aprofundada e sendo utilizada em jogos de todos os gêneros. Para (WOODCOCK, 1999) e (CHAMPANDARD, 2003), o uso de *scripts* para gerar IA nos jogos também encontra-se em um nível que gera excelentes avaliações da mídia sobre a jogabilidade dos jogos que implementam tal técnica e é de grande aceitação por parte dos jogadores, que podem personalizar certas características do comportamento dos personagens. Com o forte uso de IA através de *scripts*, os jogos conseguem transmitir um alto nível de realismo durante as partidas dos jogos.

Embora técnicas como redes neurais e algoritmos genéticos ainda não estejam

totalmente difundidas na *Game AI*, todas as referências desse artigo prevêem que o futuro da IA para jogos esteja relacionado com tais técnicas, com personagens aplicando conceitos de aprendizado, criando um mundo virtual totalmente dinâmico e imprevisível para o jogador, o que irá proporcionar maior diversão e imersão, além de estender o tempo de jogo.

4 – Conclusão

Após realizar a pesquisa sobre o uso de Inteligência Artificial em jogos eletrônicos, foi constatado que existem grandes diferenças entre IA para jogos e IA no meio acadêmico. Enquanto no meio acadêmico é necessário o estudo formal, baseando-se principalmente em teorias, a aplicação de IA para jogos é totalmente prática e, de certa forma, mais simplificada (dado que as técnicas utilizadas para obter o resultado esperado não são tão importantes como em um estudo acadêmico), embora ainda existam dúvidas quanto o que é considerado IA ou não (como por exemplo, colisões, ao meu ver, pertence à física).

Essa diferença deve-se pelo fato que jogos são um meio de entretenimento (embora haja casos de “jogos sérios” voltados para a educação), e que a indústria de jogos tem um potente mercado que fatura bilhões de dólares por ano, portanto, é fácil compreender a informalidade na área de IA, dado os cenários que um desenvolvedor de jogos é submetido durante seu trabalho (principalmente pelo curto prazo para criar um produto onde uma boa quantidade de capital foi investida).

Durante anos, as empresas desenvolvedoras de jogos estavam preocupadas com a “corrida tecnológica”,

tentando criar jogos com os mais belos e realistas gráficos; porém, a decisão (hoje e no futuro) se um título é ótimo ou não será feito através do quão bom está a inteligência e interatividade (fator imersão) dos personagens com o jogador. De fato, a inclusão de agentes inteligentes em um mundo virtual fornece aos jogadores uma experiência muito diferente em relação aos antigos (e aos não tão antigos) jogos eletrônicos.

Caso a previsão dos autores pesquisados seja realizada, estaremos num futuro próximo interagindo com agentes inteligentes que aprenderão nosso modo de jogar, de como resolver problemas e criar táticas durante uma partida, que poderão ser utilizadas em nosso benefício para atingir o objetivo do jogo, ou contra nós, como uma forma de aumentar o desafio inicial proposto pelo jogo.

Falta agora os computadores ganharem uma maior capacidade de processamento de dados, para que os algoritmos com alto custo de processamento sejam viáveis em tempo real, além dos desenvolvedores resolverem os problemas que foram citados nesse artigo, principalmente o fator conflitante de *game design* e IA – caso haja compreensão entre os *designers* e os programadores de IA, a IA será melhor utilizada nos jogos, possibilitando uma melhor experiência e jogabilidade para os jogadores.

5 – Referências Bibliográficas

(BOURG, 2004) BOURG, David M.; SEEMAN, Glenn. *AI for Game Developers*. Sebastopol: O'Reilly. 2004.

(CHAMPANDARD, 2003) CHAMPANDARD, Alex J. *AI Game Development – Synthetic Creatures with*

Learning and Reactive Behaviors. Indianapolis: New Riders. 2003.

(CRAWFORD, 2003) CRAWFORD, Chris. *Chris Crawford on Game Design*. Indianapolis: New Riders. 2003.

(DALMAU, 2004) DALMAU, Daniel Sánchez-Crespo. *Core Techniques and Algorithms in Game Programming*. Indianapolis: New Riders. 2004.

(DEMARIA, 2004) DEMARIA, Rusel; WILSON, Johnny L. *High Score! The Illustrated History of Electronic Games, 2nd edition*. Emeryville: McGraw-Hill/Osborne. 2004.

(EGM, 2004) EGM. *Capa: PSP vs. DS*. Electronic Gaming Monthly Brasil. Conrad Editora. Jul 2004. Págs. 52-58.

(FUNGE, 2004) FUNGE, John David. *Artificial Intelligence for Computer Games: An Introduction*. Natick: AK Peters. 2004.

(KENT, 2001) KENT, Steven L. *The Ultimate History of Video Games: From Pong to Pokémon and Beyond – The Story Behind the Craze that Touched Our Lives and Changed the World*. New York: Three River Press. 2001.

(KIRMSE, 2004) KIRMSE, Andrew. *Game Programming Gems 4*. Hingham: Charles River Media. 2004.

(KUSHNER, 2003) KUSHNER, David. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*. New York: Random House. 2003.

(LAMOTHE, 1999) LAMOTHE, André. *Tricks of the Windows Game Programming Gurus – Fundamentals of*

2D And 3D Game Programming. Indianapolis: Sams. 1999.

(SCHWAB, 2004) SCHWAB, Brian. *AI Game Engine Programming*. Hingham: Charles River Media. 2004.

(TOZOUR, 2002) TOZOUR, Paul. *The Evolution of Game AI from AI Game Programming Wisdom*. Hingham: Charles River Media. 2002.

(TOZOUR, 2004) TOZOUR, Paul. *Artificial Intelligence – Introduction from Game Programming Gems 4*. Hingham: Charles River Media. 2004.

(WEHMEIER, 2000) WEHMEIER, Sally. *Oxford Advanced Learner's Dictionary*. Oxford: Oxford University Press. 2000.

(WOODCOCK, 1999) WOODCOCK, Steven. *Game AI: The State of the Industry*. Game Developer Magazine. CMP Media. Aug 1999. Pages 34-43.