Inteligência Artificial em Jogos Digitais

Bruno Ribeiro, Fabiano Lucchese, Maycon Rocha e Vera Figueiredo

FEEC / Universidade Estadual de Campinas
Cidade Universitátia Zeferino Vaz, Campinas, SP, Brasil

Resumo. *neste artigo apresentamos uma breve descrição dos principais mecanismos de inteligência artificial utilizados em jogos digitais. Em seguida, discutimos algumas das aplicações destes mecanismos em jogos, procurando ainda apontar as possibilidades futuras de uso de inteligência artificial nesta área.*

1. Introdução

Neste artigo discutimos brevemente a aplicação de inteligência artificial em jogos digitais. Para isso, o documento encontra-se separado em duas seções; na seção a seguir, descrevemos alguns dos principais mecanismos utilizados em inteligência artificial para dotar sistemas computacionais de um comportamento "inteligente". A seguir, utilizamos esta descrição como o embasamento necessário para ilustrar algumas das principais aplicações de IA em jogos digitais.

Além de descrever o estado da arte de IA em jogos digitais, este documento se propõe também a apontar algumas das tendências do uso destas técnicas para possíveis aplicações futuras em jogos digitais.

2. Mecanismos de IA

O termo "Inteligência Artificial" foi cunhado em 1956 por John McCarthy, que o definiu como "a ciência e engenharia capaz de construir máquinas inteligentes". Apesar desta terminologia ser amplamente aceita, há ainda muita controvérsia relacionada ao conceito de "inteligência": se inteligência artificial é a inteligência aplicada ás máquinas, o que é exatamente "inteligência", independentemente dela estar associada ou não a seres vivos?

Em [3] são apresentados diversos propostas para formalização do conceito de inteligência artificial e nelas podemos observar a constâncias de duas idéias fundamentais sobre inteligência:

1) a capacidade de aprendizagem,
2) a manifestação de "comportamento inteligente"

A primeira idéia é suficientemente universal para poder ser aplicada a máquinas e a seres humanos. Mais ainda, pode inclusive ser aplicada a animais, como cães que, condicionados por rotinas de treinamento, desenvolvem comportamentos que imitam certos aspectos do comportamento humano. Já a segundo idéia é mais dificilmente aplicável a nosso contexto por sua característica recursiva: como pode-se definir

“comportamento inteligente” sem que o conceito de “inteligência” tenha sido estabelecido?

Outra tentativa de definição para inteligência foi proposta em um relatório denominado “Mainstream Science on Intelligence”, assinado por 52 pesquisadores, onde se afirma que “inteligência é uma capacidade mental bastante geral que, entre outras coisas, envolve a habilidade de raciocinar, planejar, resolver problemas, pensar de forma abstrata, compreender idéias complexas, aprender rapidamente e aprender a partir de experiências”. Nesse sentido, é possível perceber a ampliação dos conceitos apresentados anteriormente, criando uma forte relação entre a inteligência e as capacidades, em sua maioria, predominantemente humanas.

Russel & Norvig [3] afirmam que, em geral, as definições de Inteligência Artificial são determinadas em duas dimensões: centrada nos seres humanos e a racionalista. A centrada nos seres humanos consiste numa abordagem mais empírica, envolvendo hipóteses e confirmação experimental, e medem o nível de sucesso em termos de fidelidade em relação ao desempenho humano. Já a racionalista envolve a combinação de matemática e engenharia, sendo sustentada por modelos formais. Os autores destacam, entretanto, que essa distinção não sugere que os humanos sejam irracionais, mas sim que não são perfeitos. Um exemplo apresentado pelos autores foi o fato de nem todos os humanos serem grandes enxadristas, até mesmo alguns que têm pleno conhecimento das regras do jogo.

Russel & Norvig descrevem ainda quatro categorias de definições de inteligência artificial, relacionadas às dimensões anteriormente apresentadas:

- **Sistemas que agem como seres humanos:** nessa categoria se enquadram as iniciativas em criar sistemas que apresentam comportamento similar ao dos seres humanos. Em geral, tais sistemas estão relacionados ao teste de Turing, que consiste em desafiar a inteligência de uma máquina através de um teste onde a máquina deve ser interrogada por um humano e tal interrogador deve ser incapaz de distinguir que está interagindo com uma máquina [2]. Nesse sentido a máquina deve possuir algumas capacidades específicas para apresentar tal comportamento humano: processamento de linguagem natural, a fim de permitir interpretar e responder às mensagens fornecidas pelo interrogador; representação de conhecimento, para armazenar seu conhecimento prévio e o conhecimento a ser adquirido; argumentação automatizada, que consiste na capacidade de usar o conhecimento armazenado para fornecer respostas e tirar conclusões; e aprendizado de máquina, que permite adaptar-se a novas circunstâncias e extrapolar o conhecimento atual.
- **Sistemas que pensam como seres humanos:** nessa categoria estão as iniciativas em criar sistemas que tentam simular a capacidade de pensar dos seres humanos. Dessa forma, fica evidente a necessidade de se obter modelos teóricos sobre o funcionamento da mente humana para que, então, tais modelos possam ser adaptados ao contexto dos computadores.
- **Sistemas que pensam racionalmente:** categoria que engloba as iniciativas baseadas em processos de argumentação irrefutável, conceito introduzido pelo filósofo grego Aristóteles. Nessa abordagem, considera-se que a mente trabalha através de leis do pensamento, isto é, premissas que permitem inferir conclusões corretas a respeito de fatos. O estudo deu início ao campo

da lógica. Entretanto, essa abordagem traz consigo duas barreiras: primeiramente, não é simples transformar conhecimento informal em termos formais requeridos pela notação lógica, principalmente quando a informação não é totalmente certa; em segundo lugar, mesmo problemas com dezenas de fatos podem exigir processamento a ponto de exaurir os recursos computacionais caso não haja uma forma de estipular quais passos da argumentação devem ser tentados primeiro.

- **Sistemas que agem racionalmente:** trata-se da categoria de iniciativas que fazem uso de agentes, ou seja, programas com controle autônomo, capazes de perceber o ambiente e que se adaptarem a mudanças. Mais precisamente, são baseadas em agentes racionais, que agem a fim de obter o melhor resultado ou, quando há incerteza, o melhor resultado esperado. Ao contrário dos sistemas que pensam racionalmente, que enfatizam inferências corretas, sistemas que agem racionalmente têm a vantagem de serem mais genéricos, pois a capacidade de inferir corretamente é apenas um dos possíveis mecanismos de alcançar racionalidade. Além disso, tal abordagem é mais favorável para o desenvolvimento científico do que as abordagens de pensamento e ação humana, pois a racionalidade é mais claramente definida e completamente genérica.

Nas seções que seguem procuraremos analisar diversos mecanismos tradicionalmente associados à inteligência artificial que estão presentes nos jogos digitais, destacando em que medida eles se encaixam nestes mecanismos. Desta forma esperamos traçar uma clara fronteira para aquilo que pode ser considerado inteligente no mundo das máquinas, ainda que isso viole o senso comum sobre o que é "realmente inteligente".

2.1. Sistemas de Busca em Árvore

Um dos primeiros tratamentos teóricos dados aos jogos foi apresentado em [1]. Neste trabalho, os autores introduzem uma modelagem capaz de representar uma grande variedade de jogos em função de suas representações de estado, as regras pelas quais mudanças de estados podem ocorrer e o custo ou benefício que cada participante consegue depreender de cada estado. Além de permitir a representação de jogos, esta modelagem permitiu também a realização de estudos e análises importantes sobre suas propriedades.

Um dos benefícios da representação matemática de um jogo é a possibilidade de utilizarmos esta representação em um ambiente computacional para efetuar pesquisas de soluções a uma velocidade muito superior àquela realizável por um ser humano. Como exemplo, imaginemos um tabuleiro de "jogo da velha" 3 x 3; o estado deste jogo pode ser representado por uma matriz bidimensional 3 x 3 onde cada posição pode assumir um de três valores: vazio, X ou O. A partir desta representação, podemos mapear todos os estados possíveis e, dado um estado, quais são as possíveis transições dele para os estados atingíveis. A figura a seguir ilustra esta forma de representação.

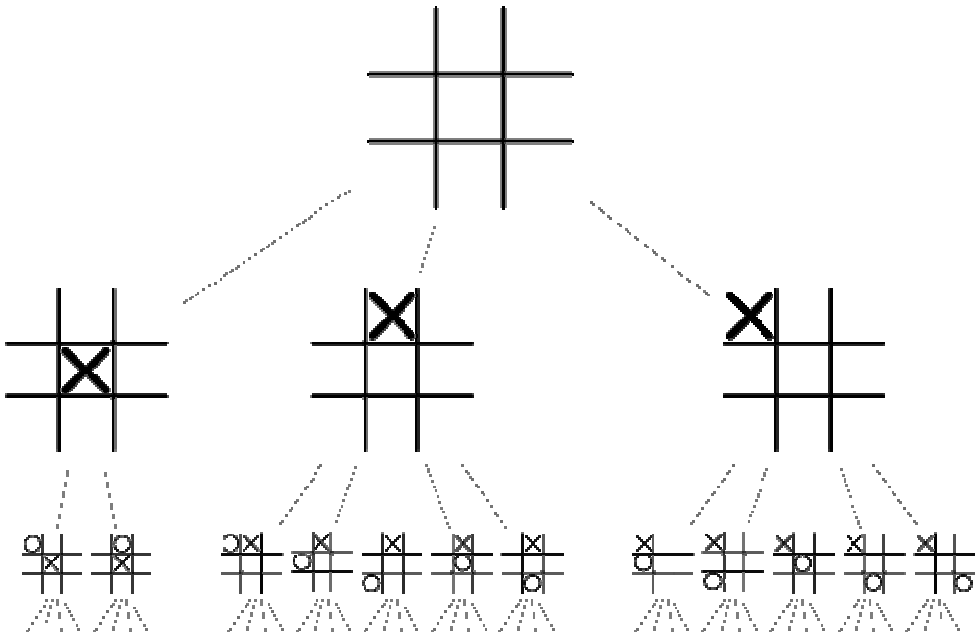

Figura 1 – Arvore de busca de estados do jogo da velha 3 x 3

Esta capacidade estendida de trabalhar com muitas possibilidades num curto espaço de tempo nos permite, por exemplo, avaliar todas as possíveis jogadas a partir de um determinado estado considerando todas as possíveis ações do adversário. A partir desta análise, um programa de computador seria capaz de fazer "jogadas perfeitas", no sentido de que ele sempre escolheria o "caminho" mais provável de levá-lo ao melhor resultado possível. Isto obviamente só é possível atribuindo-se valores de ganho a cada estado possível, mas no caso de um jogo simples como este não seria uma tarefa difícil.

Esta descrição ilustra ser possível dotar um programa de "inteligência" – no sentido que, aos olhos de um jogador, este apresentará um comportamento semelhante ao de um hábil oponente – ainda que sem utilizar técnicas sofisticadas de tomada de decisão. Neste caso, foi empregada uma técnica comumente referenciada como "força bruta", já que analisa todas as possibilidades futuras e seleciona a melhor.

O emprego da "força bruta" pode ser viável em uma situação como a descrita anteriormente, em que o número de "caminhos percorríveis" a cada interação ainda é computacionalmente tratável, mas se mostra completamente inadequado para a maioria dos jogos. Para estes casos, a busca de estados organizados em uma árvore ainda é um modelo interessante, mas requer alterações que o tornem aplicável a casos mais complexos.

Há inúmeros métodos para a simplificação da explosão combinatorial de estados atingíveis em uma árvore de busca. A estas simplificações, que podem ou não comprometer a qualidade do resultado final gerado, chamamos *heurísticas*. Heurísticas podem levar em conta aspectos relacionados às regras do jogo mas que, por si só, não indicam necessariamente o melhor caminho a ser seguido no momento de uma jogada.

Em [2] é apresentada uma das principais estratégias para simplificação das árvores de busca no contexto de jogos digitais, conhecida como A* (A estrela). Esta estratégia se resume na seguinte fórmula:

$$f(x) = g(x) + h(x)$$

Nesta fórmula, f(x) representa o custo associado a uma transição, que por sua vez é composto pelo custo da transição em si (g(x)) e pelo custo estimado de se atingir o objetivo a partir do estado obtido por esta transição (h(X)). De volta a nosso caso simples, podemos assumir que g(x) = 0 já que não há nenhum custo associado a uma transição; situações em que esta função não é nula são aquelas em que, por exemplo, há o consumo de recursos (como tempo, gasolina, energia etc) para se chegar a um determinado estado. Já para a função h(x), podemos estimar que o custo para se chegar ao objetivo do jogo da velha 3 x 3 (enfileirar 3 símbolos iguais) é igual ao número de símbolos necessários para completar a maior sequência livre do jogo.

A função h(x) proposta é uma *heurística* já que representa uma suposição capaz de simplificar grandemente o processo de busca de estados. As escolhermos os próximos estados a partir desta heurística, não temos a garantia de seguir pelo melhor caminho, mas certamente seguiremos por um caminho "com uma certa inteligência", que pode, em uma boa quantidade de casos, nos conduzir à vitória com custo computacional reduzido.

O uso de árvores de busca depende portanto da representação do estado do jogo, da associação de um valor a cada estado que indique quais são aqueles onde cada jogador obtém o maior benefício e, por fim, da definição de heurísticas que permitam percorrer as transições de estados com critérios que levem em conta o custo de cada transição e a "qualidade" de cada estado intermediário.

À luz do conceito de inteligência apresentado na parte introdutória desta seção, podemos perceber que este é um tipo de IA não necessariamente aderente ao primeiro critério. De fato, um jogo que utiliza buscas em árvore de estado puras e simples não possui nenhum mecanismo de aprendizagem e adaptação; seu comportamento será sempre o mesmo e condicionado a programação prévia que este recebeu. Entretanto, existem inúmeras heurísticas que procuram simplificar o espaço de buscas pela análise de padrões de jogos observados dos adversários. Em [4] é descrito um destes casos, em que a complexidade inerente ao jogo de xadrez foi mitigada pelo uso de redes neurais e outros mecanismos de adaptação.

2.2. Sistemas Baseados em Regras

Sistemas baseados em regra constituem uma forma alternativa de se especificar um mecanismo de busca de soluções. Estes sistemas são comumente formados por dois elementos básicos:

1) um conjunto de regras, também denominado base de conhecimento,
2) um motor (engine) capaz de processar regras à luz das informações disponíveis.

A filosofia por trás de um sistema baseado em regras, também chamado de sistema especialista, está na elaboração de um conjunto de regras, geralmente estruturadas como cláusulas do tipo "*se... então*", que, ao serem processadas uma a uma

pelo motor de execução, procurarão reproduzir a forma natural como se dá o raciocínio de um ser humano detentor do conhecimento específico relevante ao problema.

O principal ponto que distingue os sistemas especialistas dos algoritmos tradicionais é o fato de que a qualidade do resultado gerado não está associada à natureza de seu processamento, mas sim à quantidade de informações acerca do problema e de suas possíveis soluções. Por isso, sistemas especialistas complexos são caracterizados por motores simples associados a extensas bases de conhecimento.

Vejamos abaixo um exemplo simples de uma base de regras:

1) Se carro == pequeno, então carro == econômico
2) Se carro == econômico, então carro == barato
3) Se item == barato E item == vermelho, então item == interessante
4) Se carro == flex, então carro = ecológico
5) Se item == ecológico E item = interessante, então devo comprá-lo

Esta base de regras descreve o conhecimento detido a respeito dos critérios para a compra de um carro, associado a dois critérios aplicáveis a itens genéricos. Há essencialmente duas formas de se processar estas regras, descritas a seguir.

- **Forma direta**: nesta forma, o motor de processamento percorre a base de regras elaborando questões que objetivam expressar uma conclusão relacionada às informações fornecidas pelo usuário. A seguir ilustramos este tipo de aplicação.

Q) O carro é pequeno?
R) Não
C) Então você não deve comprá-lo.

Q) O carro é pequeno?
R) Sim
Q) O carro é vermelho?
R) Sim
Q) Este carro é interessante. O carro é flex?
R) Sim
C) Então você deve comprá-lo.

- **Forma reversa**: nesta forma, o usuário especifica um objetivo e o motor de processamento de regras procura definir os critérios necessários para que este objetivo seja alcançado, conforme ilustrado no exemplo a seguir.

O) Desejo comprar um carro

R1) O carro deve ser ecológico
R1.1) O carro deve ser flex
R2) O carro deve ser interessante
R2.1) O carro deve ser vermelho
R2.2) O carro deve ser barato
R2.2.1) O carro deve ser econômico

R2.2.1.1) O carro deve ser pequeno

Neste exemplo as conclusões depreendidas do processamento da base de regras podem parecer ingênuas, mas isto se deve unicamente à simplicidade da base utilizada. Sistemas especialistas reais empregam centenas ou milhares de regras que, quando encadeadas, formam estruturas de conhecimento que demandam intenso processamento computacional. Sob essa ótica, sistemas especialistas podem ser vistos como sistemas de busca otimizados através do direcionamento proporcionado pela especificação de regras claras.

Uma das aplicações mais tradicionais dos sistemas especialistas está na área médica; sistemas fundamentados em bases de conhecimento se mostraram particularmente adequados para auxiliar processos diagnósticos, que geralmente são formados por um conjunto de informações a respeito do paciente – obtidos através de exames clínicos – e um apanhado de regras que procuram estabelecer relações de causa e efeito na saúde do paciente.

No campo dos jogos digitais, sistemas especialistas podem ser utilizados no processamento de linguagem natural em jogos que envolvam uma interação mais humanizada com o jogador. Nota-se, no entanto, que, a exemplo do ocorrido com árvores de busca sem qualquer capacidade de aprendizagem, este sistema imita um comportamento inteligente, pré-programado e condicionado pela qualidade de sua base de regras.

Sistemas baseados em regras não devem ser confundidos com máquinas finitas de estado, descritas em seção futura.

2.3. Sistemas Multi-Agentes

Um dos elementos principais dos sistemas multi-agentes são os agentes em si. Não existe um consenso sobre a definição de que é um agente, sobretudo por que diferentes domínios de aplicação podem valorizar diferentes aspectos deforma que determinados conceitos associados podem receber maior ou menor importância. Russel e Norvig [3] definem que os agentes são entidades que podem perceber um ambiente, através de sensores, e agir sobre ele, através de atuadores. No contexto da computação e da inteligência artificial, tem-se os agentes inteligentes, que por sua vez são programas de computador com controle autônomo, capazes de perceber o ambiente e se adaptarem à mudanças. Além disso, tais agentes possuem um conjunto de capacidades, que designam o que são capazes de realizar, e um conjunto de objetivos a serem alcançados. Numa abordagem aplicada a jogos digitais, os personagens não controlados pelos jogadores poderiam ser modelados considerando o uso de tais agentes inteligentes de forma que seja possível o surgimento de comportamentos complexos, como trabalho em equipe, e fornecer uma experiência mais rica aos jogadores.

Segundo Demazeau [17], sistemas multi-agentes são grupos de agentes, também conhecidos como sociedades de agentes, que interagem entre si em um ambiente comum. Através dessas interações, realizadas entre os próprios agentes e com o ambiente, os agentes conseguem fornecer um comportamento global inteligente [3]. Um dos aspectos fundamentais dos sistemas multi-agentes é a capacidade destes poderem interagir direta ou indiretamente através de um mecanismo de comunicação.

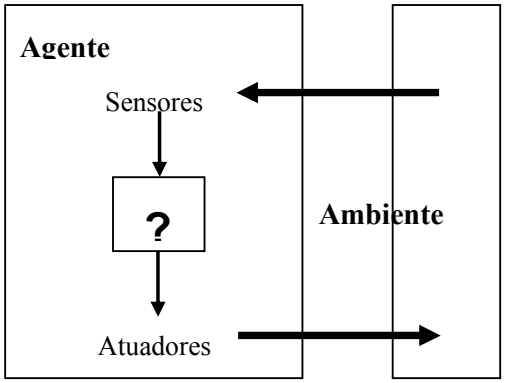

Figura 2 – Modelo geral da relação entre um agente e o ambiente.

Norvig e Russel [3] destacam que a modelagem de problemas em sistemas multi-agentes são frequentemente diferentes de modelagens para sistemas com um único agente principalmente devido à capacidade de comunicação, que em geral emerge como um comportamento racional. Outro aspecto importante destacado pelos autores a respeito da modelagem de tais sistemas é o tipo de relação existente entre os agentes. Convencionalmente, pensa-se no uso de sistemas multi-agentes na resolução de problemas onde os agentes precisam interagir para executarem uma tarefa complexa demais para um único agente, seja por falta de informação ou por capacidade de realizar as atividades necessárias. Nesse cenário surge uma relação de cooperação, ou seja, os agentes sinalizam suas necessidades aos demais a fim de receber ajuda para executar uma tarefa maior. Comumente tal cooperação ocorre através de duas formas: compartilhamento de tarefas, que ocorre quando um agente necessita realizar uma determinada tarefa e requer auxílio de outros agentes, ou compartilhamento de resultados, que ocorre quando um agente disponibiliza informações aos demais agentes para que possam utilizá-las na execução de suas tarefas.

Outro cenário que pode surgir é uma situação em que um agente, na medida em que tenta alcançar seus objetivos, acaba impedindo que outro agente possa fazer o mesmo. Em resumo se trata de problemas onde as regras que o regem fornecem uma situação em que maximizar a performance de um agente representa minimizar a performance de outro, como no jogo de xadrez ou jogos qualificados como soma-zero pela Teoria dos Jogos. Existem ainda casos onde existe o misto das duas relações, onde os agentes podem cooperar e competir até certo nível para que possam alcançar seus objetivos. Um exemplo dessa relação mista seria um jogo onde existem dois motoristas de taxi trabalhando na mesma região: eles competem por que disputam os passageiros, mas cooperam por que precisam evitar colisões entre si, o que trará benefício para ambos.

Os sistemas multi-agentes possuem aplicações muito diversificadas e têm um papel relevante nos jogos digitais. Além das aplicações convencionalmente associadas ao conceito, como o controle de grupos de entidades em jogos de estratégia ou tiro em primeira pessoa, é possível introduzir tal técnica na construção de jogos menos usuais. Weiss et al [15] afirmam que os sistemas multi-agentes permitem, entre outras possibilidades, desenvolver e analisar modelos e teorias sobre a interação em sociedades

humanas. Assim, no que se refere aos jogos digitais, é possível introduzir comportamentos complexos em indivíduos computadorizados de forma a melhorar significativamente a experiência dos jogadores.

2.4. Redes Neurais Artificiais

Uma rede neural artificial (RNA) é um modelo matemático inspirado no funcionamento dos neurônios presentes no cérebro humano. Segundo [9], uma RNA pode ser definida como uma estrutura de processamento, passível de implementação em dispositivos eletrônicos, composta por um número de unidades interconectadas (neurônios artificiais), sendo que cada unidade apresenta um comportamento específico, determinado pela sua função de transferência, pelas interconexões com outras unidades e possivelmente pelas entradas externas.

A figura a seguir apresenta o modelo de uma rede neural onde as conexões entre os neurônios artificiais procuram simular as conexões sinápticas biológicas fazendo uso de uma variável chamada peso. A função de soma acumula os dados recebidos de outros neurônios e a função de transferência, também denominada função de ativação, processa a função soma transformando-a. Em outras palavras, um neurônio corresponde a uma soma ponderada de entradas, soma esta aplicada a uma função de transferência que vai determinar sua ativação.

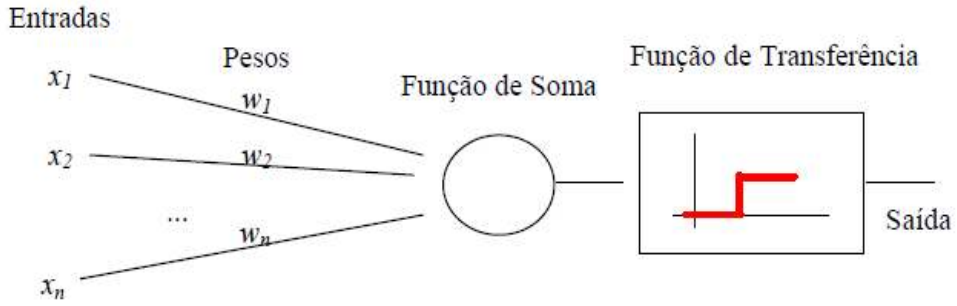

Figura 3 - Rede neural artificial

Existem basicamente três camadas em uma rede neural artificial, uma *camada de entrada*, uma *camada intermediária* (opcional) e uma *camada de saída*. Existem também três tipos de arquiteturas de RNAs:

- *Redes feedforward de uma única camada*: consiste em uma camada de entrada e uma camada de saída, onde a propagação do sinal ocorre apenas da entrada para a saída.
- *Rede feedforward de múltiplas camadas:* possuem uma ou mais camadas intermediárias. A saída da camada intermediária é utilizada como entrada para a camada seguinte. Em geral, o algoritmo de treinamento para esse tipo de rede envolve a retropropagação do erro entre a saída da rede e uma rede desejada conhecida.
- *Redes recorrentes:* esses tipos de rede possuem pelo menos um laço realimentando a saída de neurônios para outros neurônios da rede (conexão cíclica).

O processo de aprendizagem das RNAs é uma das importantes qualidades dessas estruturas. O termo "aprendizagem" corresponde ao processo de ajuste de parâmetros livres da rede através de um mecanismo de apresentação de estímulos ambientais, conhecidos como padrões (ou dados) de entrada ou treinamento:

Estimulo -> adaptação -> novo comportamento da rede

Existem basicamente três paradigmas de aprendizado:

- *Aprendizado supervisionado*: também conhecido como aprendizado com professor, em que o professor tem o conhecimento do ambiente e fornece o conjunto de exemplos entrada-resposta desejada. O treinamento é feito usando a *regra de aprendizagem por correção de erro*.
- *Aprendizado não-supervisionado*: não existe supervisor para avaliar o desempenho da rede em relação aos dados de entrada. Nenhuma medida de erro é utilizada para realimentar a rede. Geralmente elas empregam um algoritmo *competitivo de aprendizagem* (os neurônios de saída da rede competem entre si para se tornarem ativos, com um único neurônio sendo o vencedor da competição).
- *Aprendizado por reforço*: não existe uma interação direta com um supervisor ou modelo específico do ambiente. Geralmente a única informação disponível é um valor escalar que indica a qualidade do desempenho de RNAs. Durante o processo de aprendizagem, a rede testa algumas ações (saídas) e recebe um sinal de reforço (estimulo) do ambiente que permite avaliar a qualidade de sua ação.

2.5. Logica Fuzzy

A lógica fuzzy é a lógica baseada na teoria dos conjuntos fuzzy. Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, sendo interpretado como um processo de composição de relações nebulosas.

Na lógica fuzzy, o valor verdade de uma proposição pode ser um subconjunto fuzzy de qualquer conjunto parcialmente ordenado. Na lógica fuzzy, os valores verdade são expressos linguisticamente (e.g.: verdade, muito verdade, não verdade, falso, muito falso etc), onde cada termo linguístico é interpretado como um subconjunto fuzzy do intervalo unitário.

Outras características da lógica fuzzy podem ser sumarizadas como segue: na lógica fuzzy os predicados são nebulosos (e.g.: alto, baixo, ...). e há uma variedade de modificadores de predicados possíveis (e.g.: muito, mais ou menos). Estes modificadores são essenciais na geração de termos linguísticos (e.g. : muito alto, mais ou menos perto etc).

A lógica fuzzy admite, em adição, uma ampla variedade de quantificadores (e.g.: pouco, vários, usualmente, frequentemente, em torno de cinco etc).

Na lógica fuzzy existe a opção adicional de se empregar probabilidades linguísticas (e.g.: provável, altamente provável, improvável etc), interpretados como números fuzzy e manipuladas pela aritmética fuzzy [10] e o conceito de possibilidade é interpretado utilizando-se subconjuntos fuzzy no universo dos reais [11].

A modelagem e o controle fuzzy [12] são técnicas para se manusear informações qualitativas de uma maneira rigorosa. Tais técnicas consideram o modo como a falta de exatidão e a incerteza são descritas e, fazendo isso, tornam-se suficientemente poderosas para manipular de maneira conveniente o conhecimento. A sua utilização em sistemas de controle de processos em tempo real, em computadores ou micro-controladores, é das mais convenientes, dado que, geralmente, não envolvem nenhum problema computacional sério. A teoria de modelagem e controle fuzzy trata do relacionamento entre entradas e saídas, agregando vários parâmetros de processo e de controle. A grande simplicidade de implementação de sistemas de controle fuzzy pode reduzir a complexidade de um projeto a um ponto em que problemas anteriormente intratáveis passam agora a ser solúveis.

2.5.1. Variáveis Linguisticas

As variáveis lingüísticas são elementos que permitem a descrição de informações que estão normalmente disponibilizadas de forma qualitativa. Por exemplo, a temperatura de um determinado processo pode ser uma variável lingüística assumindo valores baixa, média, e alta [13]. Estes valores são descritos por intermédio de conjuntos fuzzy, representados por funções de pertinência, conforme mostrado na figura a seguir.

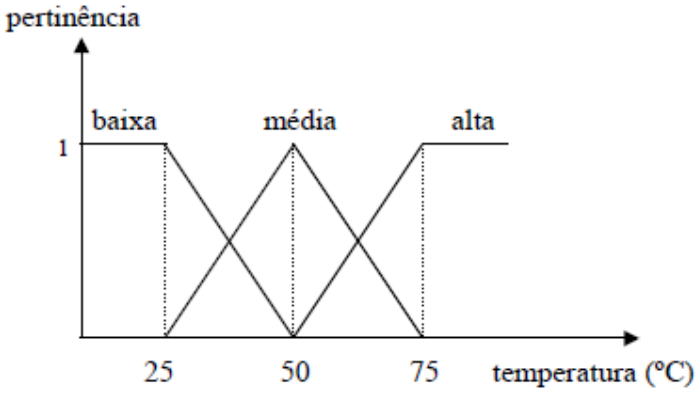

Figura 4 – Variáveis linguìsticas

Em termos gerais, os valores de uma variável linguística podem ser sentenças em uma linguagem especificada, construídas a partir de termos primários (alto, baixo, pequeno, médio, grande, zero, por exemplo), de conectivos lógicos (“não”, “e” e “ou”), de modificadores (muito, pouco, levemente, extremamente) e de delimitadores (como parênteses).

A principal função das variáveis linguísticas é fornecer uma maneira sistemática para uma caracterização aproximada de fenômenos complexos ou mal definidos. Em essência, a utilização do tipo de descrição linguística empregada por seres humanos, e não de variáveis quantificadas, permite o tratamento de sistemas que são muito

complexos para serem analisados através de termos matemáticos convencionais. Formalmente, uma variável linguística é caracterizada por uma quíntupla (N, T(N), X, G, M), onde:

- N: nome da variável
- T(N): conjunto de termos de N, ou seja, o conjunto de nomes dos valores linguísticos de N
- X: universo de discurso
- G: regra sintática para gerar os valores de N como uma composição de termos de T(N), conectivos lógicos, modificadores e delimitadores
- M: regra semântica, para associar a cada valor gerado por G um conjunto fuzzy em X.

No caso da variável temperatura da Figura XXX, teriamos:

- N: temperatura
- T(N): {baixa, média, alta}
- X: 0 a 100 ºC (por exemplo)
- G: temperatura não baixa e não muito alta (por exemplo)
- M: associa o valor acima a um conjunto fuzzy cuja função de pertinência exprime o seu significado.

2.5.2. Funções de Pertinência

As funções de pertinência podem ter diferentes formas, dependendo do conceito que se deseja representar e do contexto em que serão utilizadas. Para exemplificar o quanto o contexto é relevante na definição de funções de pertinência e de sua distribuição ao longo de um dado universo, considere-se a variável linguística estatura (de pessoas) constituída dos seguintes termos: T(estatura) = {baixa, média, alta}. A esses valores faz-se corresponder os conjuntos fuzzy A, B e C, respectivamente), definidos por suas funções de pertinência [13]. Uma escolha possível de funções de pertinência seria:

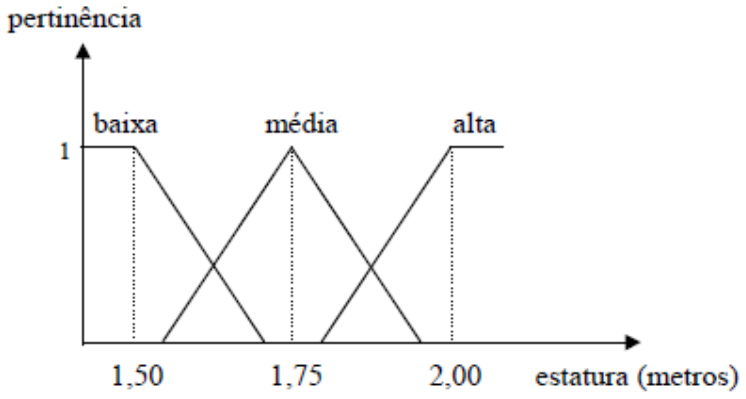

Figura 5 – Funções de pertinência

Na definição acima, estaturas de até 1,5 metros apresentam grau de pertinência igual a 1 no conjunto A; o grau de pertinência neste conjunto decresce à medida que a estatura aumenta.

Considera-se que uma estatura de 1,75 metros é "totalmente compatível" com o conjunto B, ao passo que estaturas acima de 1,8 metros (aproximadamente) apresentam grau de pertinência diferente de zero em C. Pessoas com estatura acima de 2 metros são "definitivamente" altas. Observe-se que, nesta definição das funções de pertinência, estaturas no em torno de 1,75 metros têm grau de pertinência diferente de zero somente no conjunto B, o que poderia parecer inadequado para alguns observadores. Estes prefeririam que as funções de pertinência de A e B se interceptassem em 1,75 metros (com graus de pertinência nulos, a exemplo daquelas da Figura 1), por exemplo.

Funções de pertinência podem ser definidas a partir da experiência e da perspectiva do usuário, mas é comum fazer-se uso de funções de pertinência padrão, como, por exemplo, as de forma triangular, trapezoidal e Gaussiana. Em aplicações práticas as formas escolhidas inicialmente podem sofrer ajustes em função dos resultados observados.

2.5.3. A Inferência Fuzzy

O processo de inferência fuzzy ou raciocínio aproximado permite o mapeamento do conhecimento a respeito de um determinado sistema através de regras fuzzy do tipo "se-então" podendo determinar o comportamento de saída do mesmo [13]. As regras associadas ao processo possuem a seguinte forma:

Se (condição) → antecedente

Então (ação) → conseqüente

Assim, dado que x e y são variáveis lingüísticas compostas respectivamente por um conjunto de termos. A = {A1, A2,...., An} e B = {B1, B2, ...,Bn}. Então, o problema básico do processo de inferência é encontrar uma função de pertinência B` que representa a conseqüência da aplicação simultânea de regras da forma "se (condição), então (ação)".

Normalmente, os processos de inferência fuzzy aplicados nas regras acima são baseados nas regra "*modus ponens*" generalizada que é definida por:

Fato: X é A`

Regra : X é A, entao y é B

Consequencia : y é B`

A regra de inferência anterior pode ser interpretada por: se x é A` e sabe-se que quando X é A então y é B, é verdade que y é realmente B`. Logo se o conjunto "A" implica diretamente no conjunto "B" então essa operação de implicações pode ser transformada em uma relação de implicação R (x,y).

Alternativamente ao modelo *ponens*, pode–se utilizar nos processos de inferência fuzzy a regra de "*modus-tolens*" generalizada. Este procedimento é aplicado em regras do tipo:

Fato: y é B`

Regra : Se x é A, então y é B

Conseqüência: x é A`

Utilizando a mesma análise feita anteriormente para a regra de "*modus ponens*', obtém-se o conjunto A` que é dado por:

$$A`(x) = R(x, y) \circ B`(y)$$

A regra "*modus ponens*" é bastante utilizada em processos de estimação/ controle fuzzy, enquanto que a regra de "*modus tolens*" é aplicada em processos envolvendo sistemas especialistas.

2.5.4. Controlador Fuzzy

Segundo Pedriycz e Gomide [18], o diagrama esquemático de um controlador fuzzy pode ser representado, como na figura abaixo:

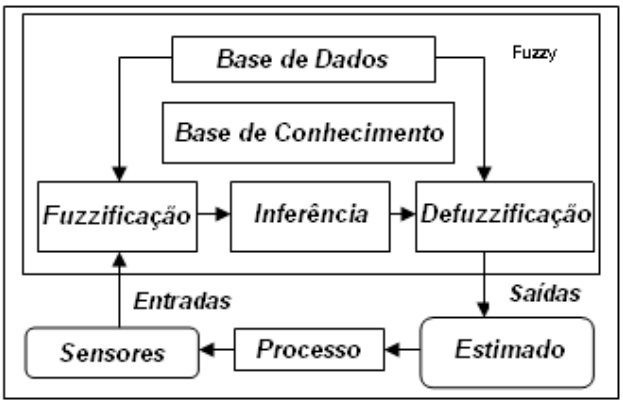

Figura 6 – Funções de pertinência

A descrição dos blocos que compõem o diagrama da Fig. 6 é dada a seguir:

- **Interface de Fuzzificação**: obtém os valores das variáveis de entrada, faz um escalonamento para condicionar os universos de discurso, transformando números em conjuntos fuzzy.
- **Base de Conhecimento**: consiste de uma base de regras, caracterizando a estratégia de estimação e suas metas.
- **Base de Dados**: armazenam as definições necessárias sobre discretização, definição de funções de pertinência e etc.
- **Procedimento de Inferência**: processa os dados fuzzy de entrada, juntamente com as regras, de modo a inferir as ações de saída fuzzy.
- **Interface de Defuzzificação**: Transforma as ações de saída fuzzy inferidas em ações/respostas não fuzzy. Em seguida, efetua um escalonamento de modo a compatibilizar os valores normalizados vindos do passo anterior com os valores reais dos universos de discurso das variáveis.

Como mencionado, os sistemas fuzzy permitem a manipulação de informações inexatas como também a modelagem de processos descritos qualitativamente. Estas informações compõem uma família de conjuntos fuzzy que representam as entradas e saídas do controle através de variáveis lingüísticas.

2.6. Máquinas de Estado Finitos

Historicamente, uma Máquinas de Estados Finitos (Finite State Machine) também denominada FSM, é um dispositivo com uma formalização rígida, usado por matemáticos para resolver problemas baseados em eventos discretos. Dentre as FSMs historicamente conhecidas está a célebre Máquina de Touring, que definida em um artigo de 1936 que ainda propõe a seguinte definição geérica de FSM: "uma FSM é um dispositivo, ou um modelo de um dispositivo, no qual temos um número finito de estados num dado momento qualquer e que pode ser operado efetuando-se transições de um estado para outro. Uma FSM pode ter somente um estado por vez."

Segundo Bourg [5], uma FSM é uma abstração de uma máquina que conduz diversos estados predefinidos, definindo condições que determinam quando um estado muda. O estado atual determina como a máquina se comporta.

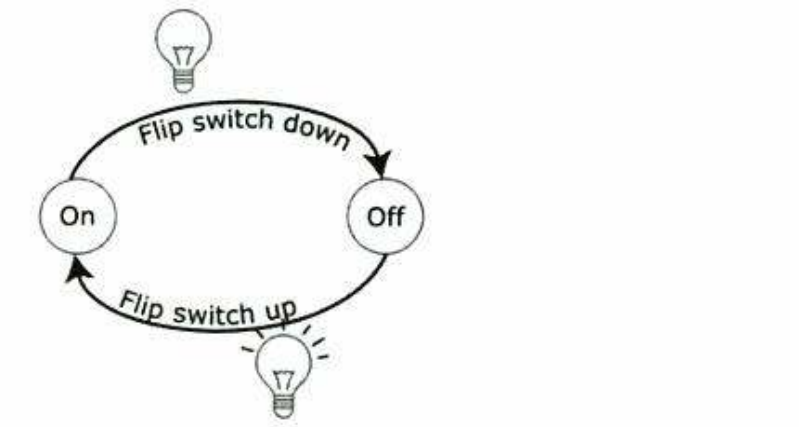

Figura 7 – Máquina de estado simples

A idéia de uma FSM é decompor um comportamento de um objeto em estados facilmente gerenciáveis. Podemos compreender facilmente a idéia com um simples exemplo proposto por Buckland [6]: Um interruptor de luz possui dois estados: ligado e desligado. Transições entre os estados são feitas por uma ação de um dedo considerado com uma entrada. Clicando para cima é feita uma transição de desligado para ligado, e clicando para baixo é feita uma transição de desligado para ligado. Não há saída associada com o estado desligado, mas quando o estado é ligado, a eletricidade é conduzida através do interruptor clareando uma sala através de um filamento de uma lâmpada.

Em jogos as FSMs são aplicadas de maneiras abrangentes e já eram utilizadas por exemplo com os fantasmas de "Pac Man". Estes podem perambular livremente, perseguir o jogador, ou fugir do jogador. Em cada estado há um comportamento diferente, e estas transições são determinadas pelas ações do jogador. Por exemplo, se o

jogador comer uma pílula de força, os estados dos fantasmas mudam de "perseguir" para "fugir". O diagrama da figura a seguir ilustra um simples modelo de uma FSM.

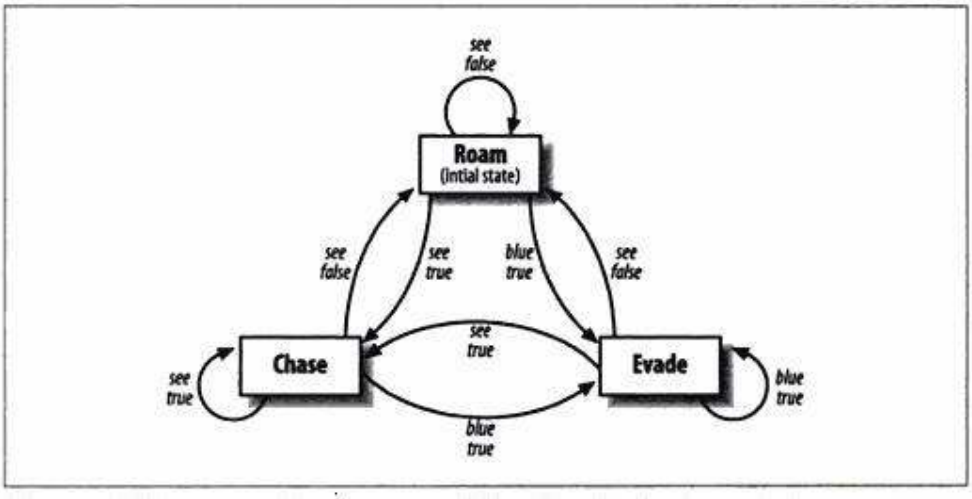

Figura 8 – FSM dos fantasmas de PAC-MAN

A figura apresenta uma Maquina Finita de estados que demonstra um modelo similar ao comportamento dos fantasmas de "Pac man". Cada retângulo representa um possível estado (perambular, caçar e fugir). A condição "blue" atende a ação de o jogador utilizar uma pílula de força, enquanto a condição "see" identifica a condição de o fantasma encontrar o jogador.

Buckland [6] apresenta algumas vantagens na utilização de Máquina Finita de Estado:

- **Facilidade e rapidez**: existem diversas formas para programar uma máquina finita de estados, e quase todas elas são simples de programar.
- **Facilidade de depurar**: quando um comportamento de um agente estiver sendo interpretado de forma incorreta, este pode ser facilmente depurado rastreando o código entre os estados.
- **Intuitivo**: os estados são interpretados de maneira similar ao pensamento humano.
- **Flexibilidade**: um agente pode ser facilmente ajustado e evoluído em um projeto de jogo digital, podendo receber novos estado e condições. Técnicas de lógica fuzzy e redes neurais podem ser combinadas à lógica de uma Máquina finita de estados.

Uma desvantagem da utilização de Máquinas Finitas de Estados são as ações repetitivas quando uma determinada condição é atendida, os estados podem se tornar limitados ("quanto mais complexo for o ambiente, maior o número de estados e transições na qual a máquina terá de prever" (Karlsson, 2006)).

3. Aplicações em Jogos

Nesta seção descrevemos algumas das principais aplicações de IA em jogos digitais. Para cada uma delas, procuraremos apresentar uma descrição que claramente referencia o mecanismos de IA (dentre os apresentados na seção anterior) que podem ser empregados para se obter o resultado esperado.

3.1. Estratégia

Um jogo de estratégia tem como característica explorar as habilidades lógicas de um jogador, tais como gerenciamento de recursos, análise de cenários e oportunismo. Segundo [7] a inteligência em jogos de estratégia pode ser dividida em dois níveis:

- **Navegação de unidades**: geralmente utilizado algoritmo de vida artificial (A*), ou alguma técnica que considere planejamento em tempo real.
- **Planejamento estratégico**: determina caminhos de árvores, ritmos de produção, freqüência de ataques, deslocamentos e posicionamento.

Nestes níveis podemos encontrar métodos aplicados como: máquinas de estados combinados com sistemas baseados em regras, permitindo ajuste de dificuldade e jogabilidade.

Do gênero estratégia, em destaque estão os jogos de tabuleiro, normalmente jogados por duas pessoas, mas que podem admitir um número arbitrário de participantes. Como jogos de tabuleiro mais populares podemos citar xadrez, damas, ludo e gamão.

Tomando como base um jogo de xadrez, a metodologia para se desenvolver esse tipo de jogo (programa) pode ser dividida em três partes, sendo a primeira, um código deve ser escolhido para o posicionamento das peças no tabuleiro. Uma estratégia deve ser encontrada para escolher os movimentos a serem feitos e essa estratégia deve ser traduzida numa seqüência de ordens elementares. O movimento especificado pelo código do lugar onde a peça está e pelo código da localização de destino. O segundo problema é a definição da estratégia do jogo, um processo direto deve ser encontrado para calcular um movimento razoavelmente bom para qualquer posição corrente. O programa pode usar o principio das jogadas registradas pelos jogadores de xadrez, que podem dar alguma ordem ao labirinto de possibilidades de variações do jogo.

No momento de montar a estratégia, deve existir um método de avaliação para qualquer posição do xadrez. Um jogador pode estimar qual dos lados tem a vantagem (branco ou preto), além disso pode fazer julgamentos sobre o valor de cada peça. Ele faz isso baseado em uma longa experiência. Por exemplo, pode-se encontrar escrito que uma rainha vale nove peões. Em um primeiro momento, uma posição pode ser avaliada medindo-se as forças de cada lado em termos da unidade de peão. No entanto outros fatores devem se levados em conta: A mobilidade e a colocação das peças, a fragilidade na proteção do rei, a formação dos peões e outros. Esses podem ter seus pesos e serem combinados na avaliação. Por isso o conhecimento dos jogadores deve ser incorporado em regras heurísticas.

O gênero de estratégia evoluiu de jogos de tabuleiro, como WAR, para os computadores. Como os computadores são capazes de anotar dados e dar informações

em tempo real para o jogador, os jogos puderam atingir um novo grau de maturidade e dinamismo, tornando o gênero único.

Cronologicamente, os primeiros jogos de estratégia eram adaptações simples de jogos de tabuleiro e, por isso, ainda mantinham a característica de jogos baseados em turnos, ou seja, cada jogador pode realizar suas ações separadamente, enquanto o outro jogador aguarda. Jogos baseados em turnos permitem ao jogador analisar o impacto de cada um de seus movimentos, olhar com cuidado o que seus oponentes estão fazendo e planejar meticulosamente as suas ações.

Estratégia baseada em turnos, um dos jogos mais famosos do gênero, é o Civilization, uma adaptação de um jogo de tabuleiro chamado Advanced Civilization. O jogo permite mais de uma forma de vitória, uma dela por meio militares e outra por meios tecnológicos. Também inclui meios políticos de resolver conflitos, um sistema de comércio entre as civilizações e as unidades não agressivas, diplomatas.

Os jogos de estratégia em tempo real (RTS) empregam um modelo de inteligência artificial hierárquico tendo seus programas divididos em camadas de decisão de estratégia e tática. A camada de estratégica possui módulos de planejamento, usando arvores para navegação entre recursos, matrizes e processamento de imagens para localização de problemas na formação do jogador ou mesmo partes do código desenhadas para definir derrota ou vitória. Já a camada tática, pode-se basear em máquinas de estado, lógica fuzzy e árvores de decisão para definir possíveis situações de ataque e defesa.

Um dos jogos de estratégia em tempo real (RTS), Dune – the building of a dinasty, da Westwood Studios, ou simplesmente Dune 2, com tema baseado nos livros de Frank Herbert, o jogo definiu o gênero e tinha todos os elementos que encontramos em um bom RTS de hoje: diferentes unidades militares, necessidade de mineração de recursos emissões entre fases.

Certamente os jogos de estratégia tiveram grande impacto com as series Warcraft, Starcraft , Command e Conquer. Houve melhorias na capacidade das unidades de se locomoverem no mapa (pathfinding), e nas estratégias desempenhadas pelo computador.

3.2. Sensoriamento do Ambiente

Atualmente a modelagem de percepção é chave para a sustentação da ilusão de atenção de um agente [6]. Estas modelagens tem o intuído de simular decisões lógicas dos sentidos, como audição, visão e olfato, de modo que um agente possa perceber um ambiente de forma similar à humana. Alguns comportamentos inadequados são claros nos jogos digitais, como por exemplo:

- Um jogador se aproxima silenciosamente de um inimigo, este só é identificado quando se encontra no raio de ação do agente.
- Quando o jogador corre e se esconde, o inimigo vai diretamente ao ponto onde o jogador se escondeu.
- Holofotes de luzes cruzam um campo de visão, deixando pontos cegos, o jogador passa pelos pontos cegos sem ser visto, mesmo assim ele se depara com um guarda que desceu a torre a seu encontro.

Estes eventos são acionados porque os dados do jogo são acessíveis pela inteligência artificial, entregando aos agentes um sensoriamento onipotente. Isto acontece pela não separação da percepção x realidade [6]. Para prevenir estas inconsistências, os sentidos dos agentes devem ser filtrados, capacitando estes de forma consistente, por exemplo:

- Se um jogador humano tem a capacidade de visão de 90 graus, a capacidade de visão de um agente deverá compartilhar a mesma restrição.
- Se o jogador não pode ouvir um agente piscar, um agente também não deve ouvi-lo.
- Um agente não deve enxergar um jogador na escuridão.

Em outros jogos, o erro não está no excesso de informações dos agentes, mas sim na falta de variáveis para interpretação de uma ação, causando uma ineficiência aos agentes para identificar eventos e entidades, como por exemplo:

- Um jogador controla uma nave, duas naves inimigas atiram em direção ao jogador, uma delas é atingida e ocorre uma ampla explosão, capaz de danificar a outra nave inimiga, esta por sua vez sai intacta da explosão.
- O jogador atinge um guarda, o corpo dele fica estendido no chão, com o movimento aparecem novos guardas, estes não tomam conhecimento do corpo estendido, chegando até mesmo pisar no corpo.
- Em uma batalha com um inimigo, o jogador está com pouca energia, este sai da batalha para obter suprimentos, o inimigo não sai da sala em sua procura, e ao retornar tudo se recomeça.

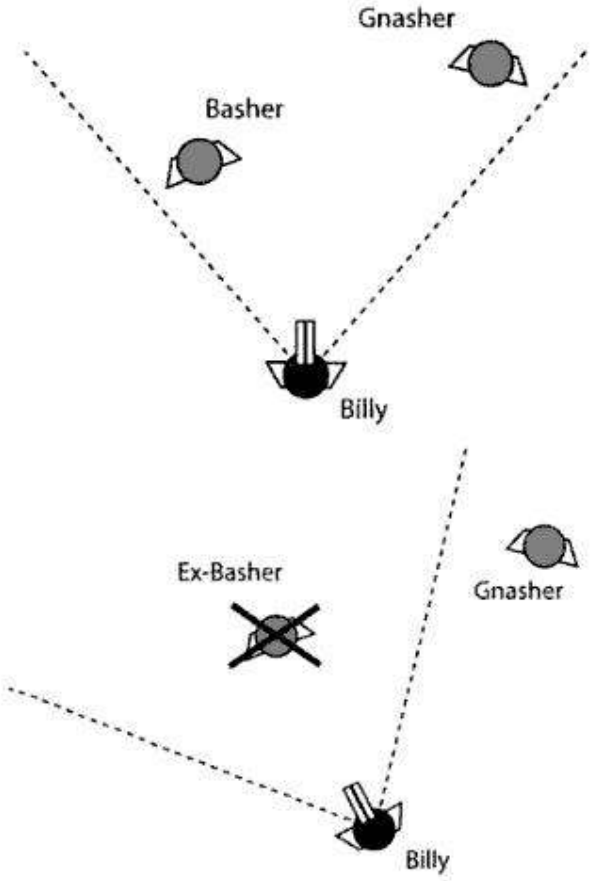

Figura 9 – Campos de visão de agentes

Na figura 9, observamos Basher e Gnasher no campo de visão de Billy. No segundo momento Billy ataca Basher, este perde a visão de Gnasher, ficando exposto ao ataque [14].

De acordo com as situações apresentadas, a ilusão de inteligência é quebrada, os agentes não se comportam de maneira esperada, em acordo com uma capacidade de percepção consistente. Segundo [6] estas inconsistências podem ser evitadas, isolando tarefas, filtrando percepções e simulando lembranças sensoriais.

3.3. Path-Finding

Path-finding, ou simplesmente navegação, é uma das aplicações mais corriqueiras em jogos eletrônicos. De fato, a construção de um mundo virtual habitado por agentes autônomos que se movem em busca de algum objetivo – frequentemente à caça do personagem controlado pelo jogador – requer a introdução de mecanismos que suportem essa navegação.

Os métodos mais comuns de navegação são aqueles baseados em regras simples e com características reativas. Um exemplo deste tipo de mecanismo, conhecido como "*crash and turn*" pode ser descrito pelas seguintes regras:

1) Verifique se a próxima posição em direção ao alvo está vazia; se estiver, mova-se.
2) Senão, vire-se na direção cuja movimentação o coloque mais próximo de seu alvo. Se não houver uma escolha única, selecione uma aleatoriamente. Mova-se.

Não há nada realmente inteligente nesta técnica. Segundo ela, agentes adquirem a propriedade da navegação sem que tenham qualquer conhecimento prévio do terreno por onde deverão se movimentar, o que demonstra a ausência da característica de aprendizagem. Este mecanismo pode, no entanto, ser estendido e sofisticado com o auxílio de técnicas de IA para se tornar verdadeiramente inteligente.

Uma pequena alteração que proporciona uma significativa melhora no processo de navegação é a introdução de uma *memória de mapeamento* em cada agente. Através desta técnica, cada agente mantém em memória um mapa que é construído à medida que se movimenta pelo terreno ou labirinto. Com isso, seu mecanismo de navegação pode ser incrementado com processos de busca semelhantes aos vistos na seção 2.1 para selecionar previamente o melhor caminho rumo ao objetivo. Neste caso, o algoritmo de navegação seria semelhante a:

1) Considerando-se a posição atual do alvo, procure no mapa em memória o melhor caminho até ele.
2) Se o próximo movimento não estiver mapeado em memória, faça:
    a. Verifique se a próxima posição em direção ao alvo está vazia; se estiver, mova-se.
    b. Senão, vire-se na direção cuja movimentação o coloque mais próximo de seu alvo. Se não houver uma escolha única, selecione uma aleatoriamente. Mova-se.
    c. Marcar a posição atual no mapa em memória.

Este tipo de abordagem emula um processo hipotético de aprendizagem, mas ainda está distante de algo verdadeiramente inteligente porque as informações disponíveis ao agente incluem elementos que, numa situação real, não existiriam, como o conhecimento da posição de seu alvo em um labirinto. Pode-se modelar este conhecimento através da abordagem sensorial apresentada na seção anterior, mas neste caso o nível de percepção seria limitado por conta das barreiras físicas existentes no cenário.

Num jogo digital o desenvolvedor dispõem de todas as informações que precisa para definir o mecanismo de navegação a ser empregado pelos agentes. Desta forma, é possível construir um mecanismo relativamente simples e que "caça" seus alvos com precisão e eficiência absolutas. Essa onipresença de informações acerca do estado do jogo torna o uso do termo IA questionável, porque não estamos emulando um comportamento verdadeiramente humano em nosso agente.

Mecanismos sofisticados de navegação, como os empregados, por exemplo, para a definição de rotas em mapas de cidades, podem sim utilizar técnicas de IA para contornar a complexidade computacional de se lidar com representações de estado

muito detalhadas (mapas com informações de trânsito em tempo real, por exemplo). Entretanto, no campo dos jogos digitais essas abordagens são não só desnecessárias como também proibitivas, já que a navegação deve-se dar em tempo real e não se dispõe de recursos computacionais de alto desempenho para realizá-la.

Técnicas de IA, como a combinação de um sistema especialista auto-alimentado com modelos sensoriais, podem ser empregados em jogos que se proponham a simular o comportamento dos agentes tal como o de seres vivos. Entretanto, voltamos a colocar que, para isso, técnicas mais simples e computacionalmente mais econômicas podem ser utilizadas. Acreditamos que, com a contínua evolução do poder computacional disponível aos consoles de vídeo game, poderá se imaginar uma modelagem genérica de agentes totalmente autônomos que agirão como seres naturalmente inteligentes na descoberta de terrenos e na aplicação de técnicas de navegação.

3.4. Combate

Uma situação muito comum em jogos, principalmente nos de ação e estratégia, é a presença de combates entre grupos de entidades onde, em geral, têm-se entidades controladas pelo jogador combatendo entidades computadorizadas. Nesses casos, os jogos precisam dispor de mecanismos que permitam controlar a forma como as entidades computadorizadas irão coordenar as ações e cooperar entre si, visando o objetivo maior que é frequentemente vencer o grupo inimigo. Um exemplo desse cenário pode ser encontrado no jogo Half-life 2, onde o protagonista precisa combater entidades computadorizadas. Neste jogo, as entidades computadorizadas apresentam um comportamento aparentemente inteligente e através de uma simples análise, puramente empírica, é possível identificar alguns comportamentos básicos:

- **Cobertura**: quando uma entidade computadorizada está recarregando sua arma, as demais entidades fornecem cobertura, continuando a disparar contra o protagonista. Isso ocorre também quando uma entidade inimiga está "ferida", situação onde entidades mais "saudáveis" tomam a frente do combate. Os inimigos tendem a se protegerem atrás de paredes, barris ou outros elementos do cenário durante o combate e enquanto recarregam suas armas a fim de evitar serem atingidos. Além disso, raramente os inimigos recarregam suas armas ao mesmo tempo ou em momentos muito próximos.
- **Recuo**: apesar do caráter agressivo, entidades computadorizadas ocasionalmente recuam, principalmente em situações onde detectam que estão em desvantagem. Outra situação onde as entidades recuam é quando estão feridas.
- **Avanço**: ao contrário do recuo, o avanço é caracterizado pelo aumento da agressividade e tomada de posição realizada pelas entidades computadorizadas quando detectam que estão em vantagem. Nota-se que o recuo do jogador reforça o caráter de avanço das entidades inimigas.

O jogo Half-life 2 faz uso de sistemas multi-agentes para prover os comportamentos identificados. Nota-se que é natural modelar a problemática de combate entre entidades, abordada no jogo, através de sistemas multi-agentes, uma vez que é clara a necessidade de que as entidades precisam efetuar algum tipo de

comunicação para que possam coordenar suas ações e trocar informações. Uma possível descrição do sistema multi-agentes adotado no jogo pode considerar que as entidades computadorizadas descrevem um agente inteligente que possui o objetivo de vencer o combate através da cooperação com outros agentes parceiros. Através de comunicação poderiam expor informações sobre seu estado (ferido, recarregando a arma, disparando etc), além de poderem interpretar essas informações a fim de realizarem ações que eventualmente determinariam o sucesso coletivo, como dar cobertura a outra entidade.

Outro aspecto relevante em mecanismos de coordenação de grupos, como os sistemas multi-agente, que está presente no jogo Half-life 2 é a introdução de uma entidade controlada pelo jogador em um grupo de entidades computadorizadas. Em Half-life 2 ocasionalmente o jogador possui entidades computadorizadas como aliados. Dessa forma, o mecanismo de inteligência artificial precisa lidar com as decisões do jogador e considerá-las como peso na definição do comportamento das demais entidades computadorizados. Millington [16] afirma que em geral, mecanismos que coordenam ações em grupo não lidam muito bem com o fato do jogador tomar decisões que diferem grandemente com o consenso do grupo formado pelas entidades computadorizadas. Como exemplo, ele cita o caso de um grupo formado pela entidade do jogador e outras entidades computadorizadas, onde o jogador decide seguir um caminho e as entidades planejaram seguir outro. Como saída para o problema, o autor afirma que muitos jogos, principalmente os de tiro em primeira pessoa, tendem a deixar a cargo do jogador o controle de decisões de alto nível permitindo, inclusive, que o jogador possa dar instruções às entidades computadorizadas. Em Half-life 2 tal característica fica muito evidente, pois as entidades computadorizadas aliadas tem o comportamento padrão de seguir o jogador e podem ainda serem ordenadas a irem a determinado local através de instruções do jogador. Entretanto, o comportamento inteligente ainda permanece para decisões mais simples, como a de se defender quando surgem inimigos e recuar quando a situação fica desfavorável, até mesmo em alguns casos em que o próprio jogador não recua.

4. Considerações de Desempenho

O estudo da literatura acerca de IA nos mostra claramente que sua definição está longe de ser um consenso e, mais ainda, têm mudado significativamente com o tempo. Ao longo de suas quase oito décadas de vida, o conceito de inteligência artificial migrou de algo informalmente definido como “o comportamento humano nas máquinas” para um conjunto de técnicas formais que, quando aplicadas nos mais diversos contextos, simulam diferentes aspectos do comportamento e inteligência humana.

A aplicação de IA em jogos digitais pode igualmente ser vista como um campo de estudo em constante mutação. O uso de simples sistemas baseados em regras, presentes nos primeiros videogames para emular comportamentos aparentemente inteligentes, são raramente aceitos hoje em dia como IA. Esta evolução tem culminado na recente disponibilização de jogos que empregam avançadas técnicas de IA para implementar a aprendizagem e o raciocínios em seres virtuais.

Essa evolução – observada tanto na IA quanto em sua aplicação nos jogos digitais - deve-se não só aos avanços conceituais da pesquisa nesta área, quanto ao incrível crescimento da disponibilidade de poder computacional em computadores de

consoles de vídeo game. De fato, a implementação de complexas redes neurais artificiais de processamento em tempo real em jogos seria algo inimaginável há 15 anos, mas já é uma realidade em títulos como Creatures e Black & White.

Acreditamos que a possibilidade de se aplicar técnicas sofisticadas de IA em jogos digitais nos auxiliará no processo de definição do que é efetivamente IA e o que são simples técnicas matemáticas para a resolução de problemas de tomada de decisão. Acreditamos que a aparição de agentes virtuais dotados de inteligência – na mesma acepção utilizada corriqueiramente quando nos referimos a pessoas – nos permitirá classificar no futuro um jogo como "IA enabled" com muito mais propriedade.

Neste ponto, cabe mencionar Weiss et al [10], que apontam as implicações da gradativa migração das atuais plataformas monolíticas para plataformas paralelas (multi-core e distribuídas). Os autores destacam que técnicas como a de sistemas multi-agentes se adequam facilmente a este novo paradigma, o que deve incentivar o desenvolvimento de jogos e aplicações que empregam este modelo.

Por fim, cabe citar os MMORPG (Massive Multiplayer Online Role-Playing Game), que já tem se beneficiado desses aspectos, pois são jogos que podem apresentar alto grau de paralelismo e usufruir de arquiteturas de software distribuídas para gerenciar e manter a coerência de ambientes complexos.

5. Conclusões

Neste trabalho analisamos algumas das principais técnicas de inteligência artificial utilizadas em jogos digitais. Pudemos perceber que, independentemente do aparente comportamento das entidades presentes em um jogo, poucos são os usos legítimos de IA neste campo. De fato, comportamentos aparentemente inteligentes podem ser obtidos através de técnicas simples de busca em árvores de estado, ou através da implementação de máquinas finitas de estado. Nenhum destes casos requer necessariamente a introdução da capacidade de aprendizagem.

Jogos digitais são um grande sucesso comercial e, como tais, estão sujeitos a todo o tipo de recurso de marketing que suportem suas vendas e seu apelo ante ao público-alvo. Uma das "killer features" de todos os jogos candidatos a vendas maciças é a presença de IA. Nosso objetivo neste relatório foi fornecer uma visão crítica que permitisse ao leitor discernir o que pode e o que não pode ser classificado como IA num jogo, independentemente do que diga seu material publicitário. Infelizmente esta análise requer o conhecimento de detalhes de desenvolvimento que normalmente não estão disponíveis ao usuário normal.

Por fim, esperamos que, com este texto, tenhamos oferecido uma contribuição ao desenvolvedor de jogos que considerar a introdução de mecanismos de IA em seus jogos como alternativa a outras formas de implementação de comportamento inteligente.

Referências

[1] von Neumann. J., Morgenstem, O.. *Theory of Games and Economic Behavior*, Princeton University Press, 1944

[2] Deloura, M., *Game Programming Gems* (Vol. I), Charles River Media Ed.. 2000. 550 P. ISBN 1584500492.

[3] Russel, S., *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2a edição, 2002, ISBN-10: 0137903952

[4] Campbell, M., Hoane, A. J., Hsu, F-h., *Deep Blue*, Artificial Intelligence, Volume 134, Issues 1-2, Janeiro 2002, Páginas 57-83

[5] Bourg, D. M., Seemann, G., *AI for Game Developers*, California, 2004

[6] Buckland, M., *Progamming game AI by example*, Texas, 2005

[7] Tatai, V. K.. *Técnicas de Sistemas Inteligentes Aplicadas ao Desenvolvimento de Jogos de Computador*. Tese de Mestrado, FEEC/UNICAMP, 2003.

[8] Ramsey, M, *Designing a Multi-Tiered AI Framework*, AI Game Programming Wisdom 2 (ed. S. Rabin), Charles River Media. (2004)

[9] Von Zuben, F. J.; *Introdução a computação natural,* DCA/FEEC/UNICAMP, 2008.

[10] Kaufmann, A. ; M. Gupta (1985) - *Introduction to Fuzzy Arithmetic* - Van Nostrand, NY.

[11] Zadeh, L. (1988) - Fuzzy Logic - *IEEE Computer*, April, pp. 83-92

[12] Lee, C.C (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller, part I and II. *IEEE Trans. onSystems, Man and Cybernetics*, vol. 20, pp 404-435

[13] Tanscheit, R; *Sistemas Fuzzy*, DEE-PUC-Rio, Rio de Janeiro (2004)

[14] Buckland, M., *Progamming game AI by example*, Texas, 2005

[15] Weiss, G, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Massachusetts,

[16] Millington, I., *Artificial Intelligence for Games,* Elsevier, California, 2006.

[17] Demazeau, Y., Muller, J. P., *Descentralized articifial intelligence*, Amsterdam, Elsevier Science Publishers, 1991

[18] Pedrycz, W.; Gomide, F., *An Introduction to Fuzzy Sets – Analysis and Design*. MIT Press: Cambridge, USA, 1998