UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE SISTEMAS DE INFORMAÇÃO

JOÃO ANTÔNIO PRESTES MATIUZZI

UMA PRPOSTA PARA APLICAÇÃO DE SISTEMAS FUZZY EM AGENTES PARA JOGOS DIGITAIS

Porto Alegre
2014

João Antônio Prestes Matiuzzi

UMA PRPOSTA PARA APLICAÇÃO DE SISTEMAS FUZZY EM AGENTES PARA JOGOS DIGITAIS

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação, pelo Curso de Sistemas de Informação da Universidade do Vale do Rio dos Sinos – UNISINOS

Orientador: Vinícius Jurinic Cassol

Porto Alegre

2014

AGRADECIMENTOS

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com a crescente demanda por jogos que ofereçam cada vez mais interação entre o sistema e o jogador, se faz necessário que o jogo disponibilize agentes automatizados cada vez mais realistas. Com isso, se torna importante o estudo e o aprimoramento das técnicas de Inteligência Artificial que podem ser aplicadas em jogos.

Nos últimos anos, tem sido observado cada vez mais avanços na área da inteligência artificial aplicada a jogos, sendo muitas vezes um aspecto crítico em relação ao sucesso alcançado pelo jogo (TOZOUR,2003). Portanto, fica evidente que entre todos os demais aspectos importantes do jogo, como qualidade de gráficos e som, o motor de inteligência dos agentes também deve ser trabalhado em busca do melhor resultado possível.

A inteligência artificial acompanha o desenvolvimento dos jogos desde o seu início, nos anos 1970, sendo importante da definição dos objetivos e da jogabilidade dos jogos. Mesmo assim, durante um determinado tempo, a criação do motor de inteligência foi considerado como um aspecto secundário do desenvolvimento dos jogos (TOZOUR,2003), deixando uma maior quantidade de tempo e trabalho disponíveis para o desenvolvimento dos aspectos mais impactantes no jogo, como os gráficos.

Com o desenvolvimento constante dos sistemas de informação e do avanço da capacidade computacional dos video games e computadores, foi possível criar sistemas cada vez mais complexos de agentes automatizados, criando assim a possibilidade de existir um trabalho mais dedicado a criação do motor de inteligência dos jogos.

Com o aumento do interesse no desenvolvimento da AI em jogos, várias técnicas passaram a ser aplicadas na busca de agentes cada vez mais realistas, técnicas como Sistemas Especialistas e Raciocínio baseado em casos, dão ao agente automatizado a possibilidade de responder às entradas do usuário de maneira similar a uma decisão humana.

Este trabalho propõe a aplicação da Lógica Fuzzy na implementação dos agentes de um jogo no estilo “Estratégia”, apresentando quais características da Lógica Fuzzy são importante de serem aplicadas em um jogo, de maneira que irá impactar positivamente na qualidade dos agente automatizados.

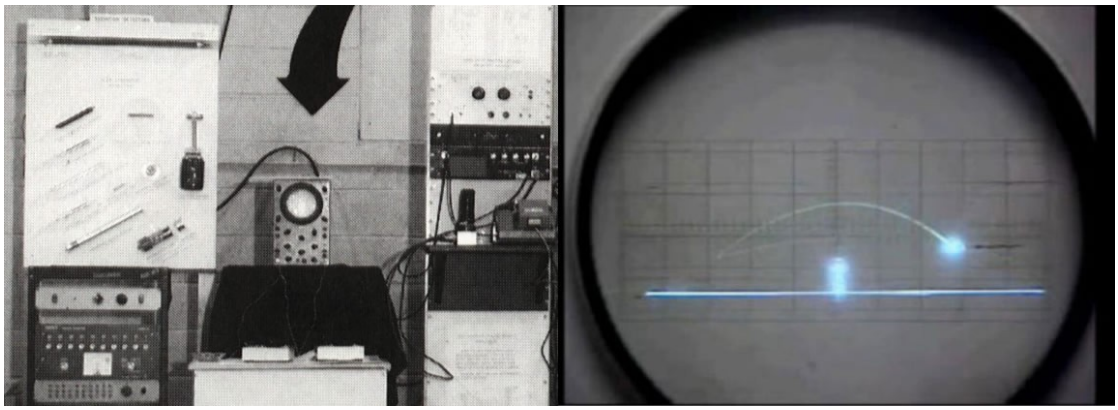
2 JOGOS DIGITAIS

Neste capítulo, será feito um breve histório do surgimento dos primeiros jogos eletrônicos e também um resumo do mercado atual de video games.

2.1 Os Primeiros jogos digitais

Os jogos como conhecemos atualmente, começaram a tomar forma ainda na década de 1950, nos Estados Unidos. Podemos considerar como o início do conceito de jogos digitais, um projeto criado pelo cientista William Higinbotham, o qual, na época, era líder da divisão de instrumentos do Brookheaven National Laboratory, um laboratório do governo Norte-Americano. (TRISTAN, 2002). Tal projeto, era, na realidade, uma maneira de de agradar pessoas que visitavam o o Brookheaven National Laboratory anualmente, o jogo consistia numa simulação de uma partida de Tênis feita em um osciloscópio, vista pela perspectiva de lado em relação a rede, onde existia um traço no meio da tela do osciloscópio, representando a rede.

O jogo era controlado por dois controles, cada um com um botão pressionável e um giratório, no qual o jogador movia a “raquete” girando um botão e arremeçava a bola pressionando o outro.

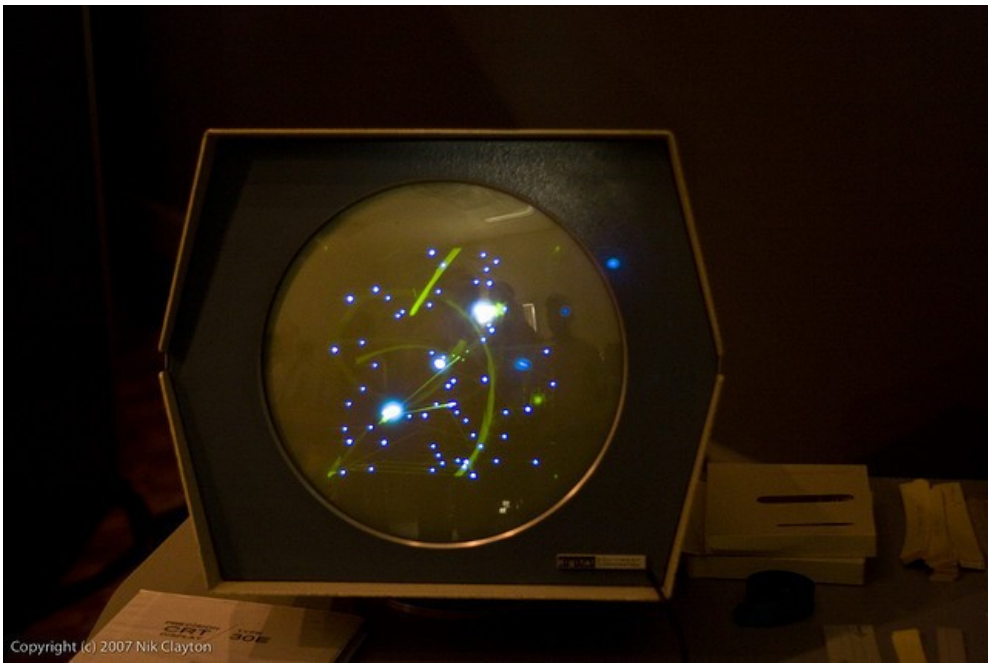


Embora, segundo Willian Higinbotham, muitos jovens tivessem gostado da ideia do jogo, o jogo foi utilizado apenas mais uma vez nas visitas do laboratório em 1959, e após isto, ele foi desmontado para que suas peças fossem utilizadas em outros projetos, pois acreditava-se que não havia motivo para continuar desenvolvendo o jogo, terminando aquilo que foi o primeiro videogame da história, por haver o entendimento que aquilo se tratava de perda de tempo(TRISTAN, 2002).

Já nos anos 1960, a ideia de que os computadores poderiam ser utilizados em atividades

consideradas “menos sérias”, surgiu no Massachusetts Institute of Technology, Estados Unidos, com o jogo “SpaceWar!”. Tratava-se de um jogo desenvolvido por um grupo de estudantes em um computador PDP-1, um computador que possuia o tamanho de um carro. (TRISTAN, 2002)

O grupo decidiu que o jogo deveria proporcionar ação através da interação do jogador com o computador, com isso, decidiram que o jogo seria um jogo de batalha entre naves espaciais, embora o jogo fizesse sucesso entre os usuários do PDP-1, o alto custo do computador, $120.000,00 tornou impossível qualquer forma de comercialização do jogo.



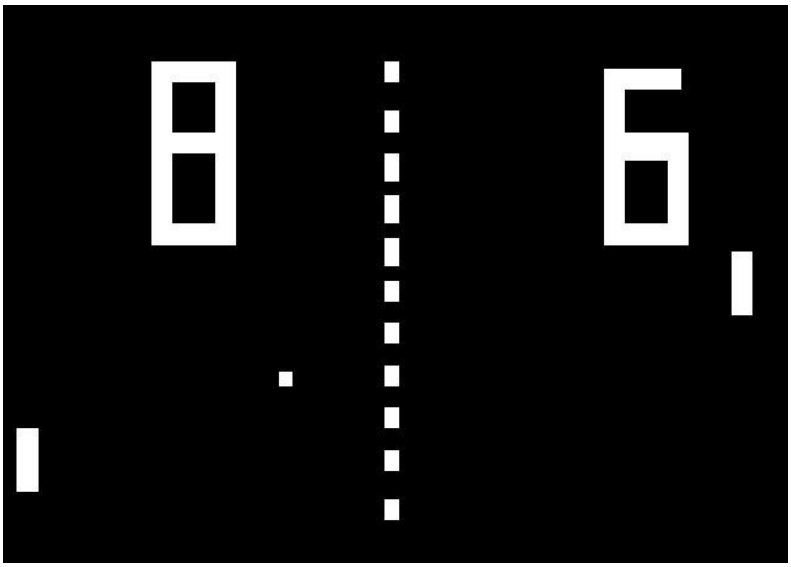
Já no final dos anos 1960, e anos 1970, o primeiros Arcades, máquinas especializadas que executavam um jogo, começaram a ser desenvolvidos, pelo fato de que os computadores ainda eram caros demais para serem adquiridos pelos usuários alvo do jogos. Nolan Bushnell, um engenheiro que havia jogado o SpaceWar! Desenvolvido no MIT, desenvolveu o Space Computer, um Arcade que executava o SpaceWar!, mais tarde, Nolan Bushnell viria a fundar a empresa Atari.(TRISTAN, 2002)

2.2 A época dourada dos Arcades

Conforme KENT(2001), os anos entre 1979 e 1983 foram os principais na popularidade dos Arcades, nesta época foram lançados jogos como Pong, Space Invaders e o Arcade mais popular do mundo Pac Man, desta forma, várias empresas voltaram a sua atenção para o mercado de Arcades, como as empresas MidWay, Taito e Atari.

Nesta época, os lucros obtidos da comecialização de jogos vinha em grande parte dos Arcades, a sua grande popularidade fazia com que milhares de máquinas fossem vendidas anualmente nos Estados Unidos.

O grande sucesso incial dos Arcades se deu pelo jogo Pong, jogo que teve diversas variaçoes, mas que com o tempo se tornou obsoleto por volta de 1975, dando espaço para novos jogos com qualidade gráfica mais avançadas, e com temas diferentes, como corridas de carro.(WOLF,2009,pg 38) .



2.3 A primeira geração de Consoles

Com o lançamento do Atari 2600 em 1977, os consoles domésticos ganharam a atenção do mercado, já no ano de 1985, o Nes(Nintendo Entertainment System), console da empresa japonesa Nintendo, já era um sucesso nos Estados Unidos, com processadores menores e mais rápidos, nesta época os videogames e computadores já possuiam preços mais acessíveis para o público.

O sucesso do Nes recuperou o mercado de consoles nos Estados Unidos, que vinham de uma crise causada pelo desinteresse do público em relação aos consoles(WOLF,2009). Com o grande sucesso de jogos como Mario, o Nes provou que o mercado de consoles ainda era viável, com isso empresas como a Atari, lançaram novos consoles, por exemplo o Atari 7800, que era um console destinado a consumidores com poder aquisitivo mais baixo(WOLF,2009). Mesmo assim, a Atari não foi capaz de competir com os produtos da Nintendo, que já dominavam o mercado de consoles.

2.4 O mercado atual de jogos no Brasil

O Brasil é um grande mercado para a produção de jogos, possuindo expectativa de grande crescimento do setor. Com expansão de 60% em 2012, foi o mercado que mais cresceu no mesmo ano (LANNOY, 2013). Este seria o reflexo, entre outros, da dimuição da pirataria no país, pois o dinheiro que é investido no desenvolvimento de software estaria trazendo mais retorno às empresas.

Além do aumento significativo do setor, o governo brasileiro demonstra interesse em estimular o setor com o objetivo de aquecer o mercado, aumentando assim a criação de empregos na área. Segundo a Associação Brasileira dos Desenvolvedores de Games (Abragames), muitas empresas tem sido criadas na área de desenvolvimento de jogos, segundo uma pesquisa realizada pelo consultor Marcos Cardoso, a pedido da Abragames, existem, atualmente, cerca de 220 empresas que atuam na área no Brasil, um grande aumento em relação ao ano de 2008, quando haviam apenas 48 empresas.

Tais dados evidenciam que o mercado brasileiro está aquecendo em relação a produção de jogos, embora ainda não seja o ideal pois, segundo Vitor Severo, que é sócio em uma empresa nacional de desenvolvimento de jogos, o governo está começando a estimular o mercado de games no país, mas o estímulo ainda é considerado modesto.

Sobre o futuro dos jogos no Brasil, foi realizada uma audiência pública em 27 de maio de 2014, a pedido da Associação Comercial, Industrial e Cultural de Games (ACIGAMES), onde foi citado que o Brasil atualmente é o 4º maior mercado de jogos no mundo, tendo crescido 38% em 2013. Mesmo assim, ainda há falta de ações governamentais para fazer com que o mercado cresça ainda mais, não apenas consumindo jogos fabricados no exterior, mas sim desenvolvendo e exportando. Mas para isso é necessário que a carga tributária do jogos seja revista, para assim aumentar a competitividade dos jogos desenvolvidos no país.

Embora a audiência não tenha produzido resultados práticos, é importante notar que há o interesse em fazer com que a indústria de jogos no Brasil tenha a importância devida, para que assim mais empresas sejam criadas, criando assim empregos e renda.

3 INTELIGÊNCIA ARTIFICIAL APLICADA A JOGOS

Neste capítulo será abordada a aplicação de técnicas de inteligência artifical em jogos, citando técnicas que são aplicadas atualmente no desenvolvimento de jogos.

3.1 Inteligência Artificial e jogos

A aplicação de IA em jogos vem desde os primeiros jogos produzidos, muitas vezes, eram simples motores de tomada de decisões aleatórias, em jogos como PacMan e Space Invaders. Com a evolução da capacidade de processamento, como nos jogos de estratégia, o desenvolvimento dos sistemas de IA ganhou importância no processo de desenvolvimento do jogo, com isso, a qualidade das interações entre o jogador e os agentes automatizados recebeu uma melhoria considerável(TOZOUR,2003). A partir deste momento várias técnicas foram aplicadas para a criação dos motores de inteligência artificial

Segundo (MILLINGTON,2006), inteligência artifical é "..possibilitar computadores realizarem tarefas que seres humanos e animais são capazes". Ou seja, em um ambiente de um jogo, a inteligência artifical seria aplicada, por exemplo, no controle de um agente, que seria um subsistema do jogo, que realizaria tarefas pré-determinadas, como mover-se em um mapa, realizar uma determinada jogada, ou um ataque, seguindo as regras definidas no jogo.

Os computadores apresentam boa capacidade de resolução de problemas complexos de aritmética, por exemplo, mas apresentam dificuldade em sistemas que exigem a aplicação de criatividade e tomada de decisões, tarefas que geralmente são consideradas simples pelos seres humanos, é neste aspecto que aplicamos técnicas de inteligência artifical, para possibilitar ao computador possuir um comportamento similar ao dos seres humanos (MILLINGTON,2006).

Conforme (KIRBY,2011), a inteligência artificial aplicada em um jogo, deve possuir três aspectos:

- ser notada pelo usuário, ou seja, deve ficar claro em que momento os agentes automatizados estão agindo de maneira inteligente, com isso, a inteligência deve ser aplicada a um agente somente no momento na qual a interação com o usuário irá ocorrer.
- O segundo aspecto, é de que as decisões do motor de inteligência devem ser inteligentes, mas não serem totalmente inteligente todo o tempo e repetir as mesmas decisões sempre; também previnir que a IA tenha atitudes consideradas "bobas".

- O terceiro aspecto, é de que as para um jogo parecer divertido e com boa interatividade, o usuário precisa ter a sensação que as suas ações estão influenciando no ambiente do jogo, ou seja, os agentes devem responder aos estímulos do usuário de maneira satisfatória.

Encontrar o modelo certo de IA a ser utilizada é vital para o bom funcionamento do motor de inteligência de um jogo, pois muitas vezes um sistema complexo de IA pode ser menos eficiente do que um projeto mais simplificado, que possua um comportamento adequado para um determinado sistema.

Há uma variedade enorme de técnicas descritas para a criação de sistemas inteligentes, cada uma possuindo determinadas características que a fazem ser úteis em determinadas situações. Dentre as principais técnicas podemos citar algumas que, segundo (TOZOUR,2003), são as mais relevantes atual e futuramente na produção de motores de inteligência artificial para jogos:

Sistemas Especialistas: Consiste na modelagem baseada no conhecimento de uma pessoa experiente em um determinado assunto, ou seja, é criada uma base de conhecimento na qual o sistema irá se basear para produzir uma resposta similar ao que um ser humano experiente responderia.

Árvore de Decisão: É um sistema onde é formado um grafo onde os nós filhos representam respostas que o sistema pode reproduzir a partir de uma entrada de dado.

Algorítimo Genético: É uma técnica que tenta reproduzir o processo biológico de evolução, aplicando os conceitos reais de hereditariedade e mutação. O algoritmo genético consiste em definir um grupo de possíveis respostas para um problema e submetê-lo a um processo evolutivo onde é calculado o nível de aptidão de cada membro da populão e somente os membros com maior nível de aptidão serão mantidos.(SOUSA,2002)

Redes Neurais: Redes Neurais é uma técnica de implentação de sistemas de IA onde o objetivo é reproduzir o comportamento de um cérebro na resolução de problemas. A partir de modelos baseado no funcionamento de neurônios, chamados de *perceptron*, o algoritmo procura reproduzir o comportamento de um neurônio, embora seja uma simplificação, o algoritmo reproduz comportamentos que podem ser aplicados na simulação de memória e de aprendizado.(SOUSA,2002)

Máquina de Estados Finitos: Uma máquina de estados finitos é um modelo onde é definido um grafo com uma determinada quantidade de estados que o agente pode possuir durante a execução do sistema, também é definido quais ações que podem fazer com que o agente abandone um estado e entre em um novo estado. (KIRBY,2011)

3.2 Lógica Fuzzy

A Lógica Fuzzy expande o conceito utilizado na lógica booleana onde um objeto deve pertencer a um determinado grupo. Em um sistema fuzzy, um objeto tem um valor que determina o grau de exatidão em relação a sua participação a um determinado grupo, chamado grau de pertinência, quanto maior for o grau de pertinência de um objeto em relação a um grupo, maior será a sua relação com o grupo.(SOUSA,2002)

A lógica Fuzzy é explicada por (MILLINGTON, 2006) através do seguinte exemplo:

Imagine em um jogo, um personagem percorrendo um mapa em um ambiente perigoso, em uma máquina de estados finitos, por exemplo, o personagem poderia estar apenas em um dos seguintes estados: “Cauteloso” ou “Confiante” e de acordo com estes estados, definir que ações o agente iria realizar:

Se (Estado_Agente == “Cauteloso”) “Caminhar Lentamente”

Já na lógica Fuzzy, o estado do agente pode ser representando por um intervalo de estados possíveis, fazendo com que exista um espectro entre os estados, permitindo que respostas mais específicas sejam produzidas como:

Se (Estado_Agente == “Bastante Cauteloso”) “Caminhar Muito Lentamente”

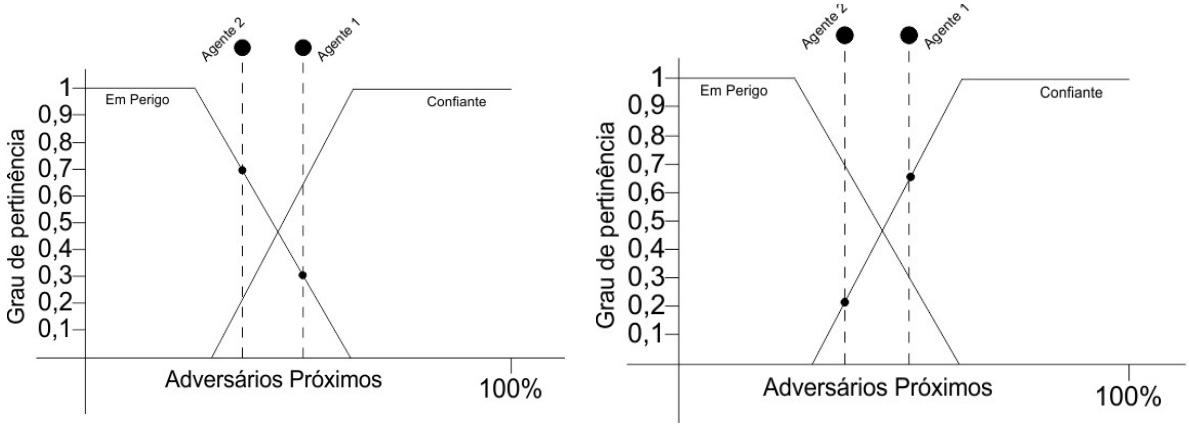
Podemos concluir então, que um estado na lógica fuzzy corresponde a um determinado peso de um determinado estado, ao invés do tradicional estado de verdadeiro ou falso, definindo o quão verdadeiro ou falso um estado é. Ou seja, em um jogo onde um agente está no estado “Perigo”, ele terá um peso relacionado ao nível de perigo no qual ele se encontra, como por exemplo:

Agente	Estado	Grau de Pertinência
Agente 1	Perigo	0.3
Agente 2	Perigo	0.7
Agente 3	Confiante	0.5

No quadro acima, é exemplificado um grupo de agentes e de estados que um agente pode se encontrar em um determinado momento de um jogo, se considerarmos a lógica booleana, o Agente 1 e o 2 estão no mesmo estado, mas ambos estão em estados diferentes em

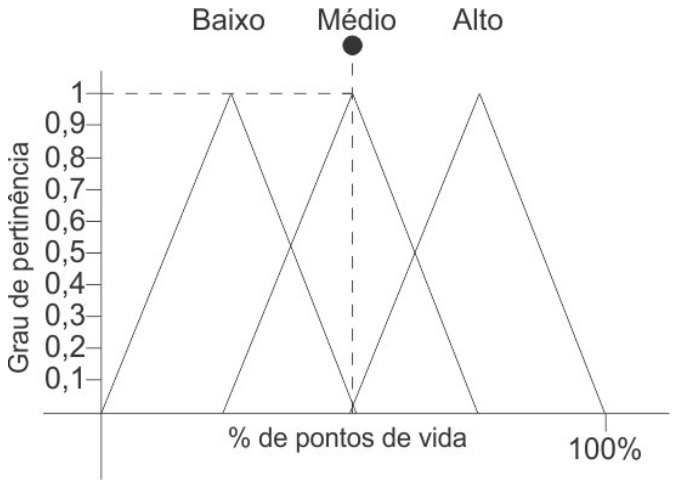
relação ao Agente 3, já na lógica Fuzzy, é levado em consideração o grau de pertinência de cada estado, com isso, teríamos cada agente em um estado diferente, sendo que o Agente 1, está em um estado que pode ser defindo como de “Menos Perigo” em relação ao Agente 2.

Nos gráficos a abaixo, é demonstrado como o grau de pertinência define o estado no qual um agente se encontra:



Na imagem, há a relação do grau de pertinência dos estados “Em Perigo” e “Confiante” dos Agentes 1 e 2 em relação a quantidade de inimigos que estão próximos a ele. Analisando a imagem, podemos definir que o agente possui um grau de pertinência menor do que o Agente 2 no estado “Em Perigo”, também é possível defnir que em relação ao estado “Confiante”, o Agente 1 possui um maior grau de pertinência em relação ao Agente 2, ou seja, na Lógica Fuzzy existe a possibilidade de os agentes serem partes de conjuntos diferentes ao mesmo tempo, desde que a soma dos seus graus de pertinência seja igual a 1, (MILLINGTON, 2006).

Como na lógica Fuzzy os valores são tratados como graus de um determinado conjunto, os dados de entrada no processo de inferência precisam ser transformados, este processo é conhecido como fuzzificação, onde um determinado dado de entrada será convertido em graus de um determinado qualificador. Ou seja, se no jogo em desenvolvimento existir um atributo chamado “vida”, que pode ser um número no intervado de 100 até 300 pontos, e o agente possuir 150 pontos de vida, a representação do atributo “vida” após o processo de fuzzificação será a seguinte:



No exemplo, se o agente possuir 50% do atributo “Vida”, possuirá grau de pertinência 1 para o conjunto chamado “Médio”, após o processo de fuzzificaçao, com o valor fuzzificado será possível realizar a aplicação dos demais aspectos da a lógica Fuzzy, como a Inferência.

Na etapa de inferência, a variável que passou pelo processo de fuzzificação será, testada de acordo com as regras definidas na base de conhecimento.

As regras que serão utilizadas na etapa de inferência podem ser definidas como condicionais no seguinte formato:

SE X ENTAO Y

Contextualizando com o exemplo utilizado na fase de fuzzificação, a base de conhecimento de um sistema fuzzy para o atributo “vida” poderia ter as seguintes regras:

SE VIDA == MÉDIO ENTÃO ACAO = LUTAR
SE VIDA == BAIXO ENTÃO ACAO = FUGIR

Após o processo de inferência, a variavel de entrada será relacionada com as regras definidas na base de conhecimento, neste momento, é necessário que este valor seja transformado novamente em um valor útil para ser aplicado no sistema (MILLINGTON, 2006), um valor da mesma natureza do dado de entrada utilizado no processo de fuzzificação, este processo é chamado de defuzzificação.

Segundo (MILLINGTON,2006), este processo, ao contrário do processo de fuzzificação que é direto, é mais complicado por não existir um método único para a defuzzificação, existindo algumas técnicas para realizar a tarefa. Como por exemplo:

- Maior Grau de Pertinência: Neste o valor considerado será o do grupo que possuir o maior grau de pertinênia.

- Centro de Gravidade: Neste método, ao contrário da última técnica, não considera somente o maior grau de pertinência, mas sim todos os valores resultantes do processo de inferência, encontrando um ponto médio referente a o conjunto de valores produzidos.

4 PROPOSTA

Neste capítulo será proposto o desenvolvimento de um jogo no qual os agentes serão controlados utilizando a lógica Fuzzy. Será definida a estrutura necessária para a implementação e as suas principais características de funcionamento.

4.1 Aspectos gerais do jogo

O jogo proposto terá o estilo conhecido como estratégia baseado em turnos. (BARTON, 2008), define os jogos neste estilo como jogos fortemente baseados em estratégia-tática, exploração e constrole de recursos.

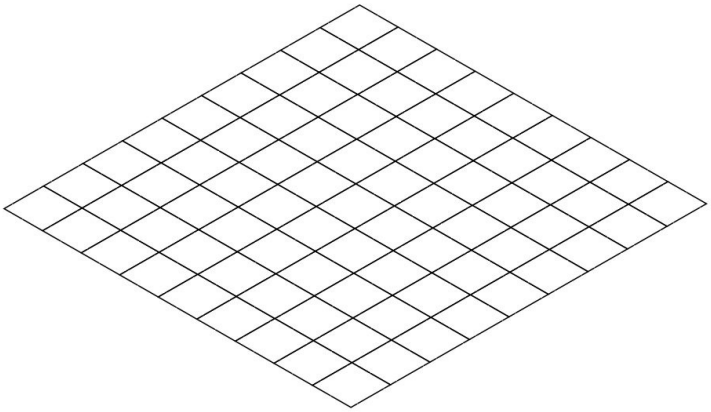
O jogo em questão consistirá em duas equipes: uma controlada pelo usuário e outra controlado pelo sistema, ambas equipes irão possuir o mesmo objetivo: chegar a uma base, que é definida em um determinado local do mapa, para isso, cada equipe deverá desenvolver uma estratégia, balanceando os momentos onde irá atacar as peças do inimigo e os momentos onde usará o seu turno para mover uma peça em direção a base.

Para alcançar o objetivo, cada equipe irá possuir 4 peças, as quais podem a cada turno do jogo, mover-se pelo mapa ou atacar uma peça inimiga.

Tais características irão evidenciar a aplicação da lógica Fuzzy no controle dos agentes no momento em que é necessário que os agentes analisem o ambiente, reagindo com as iterações do jogador.

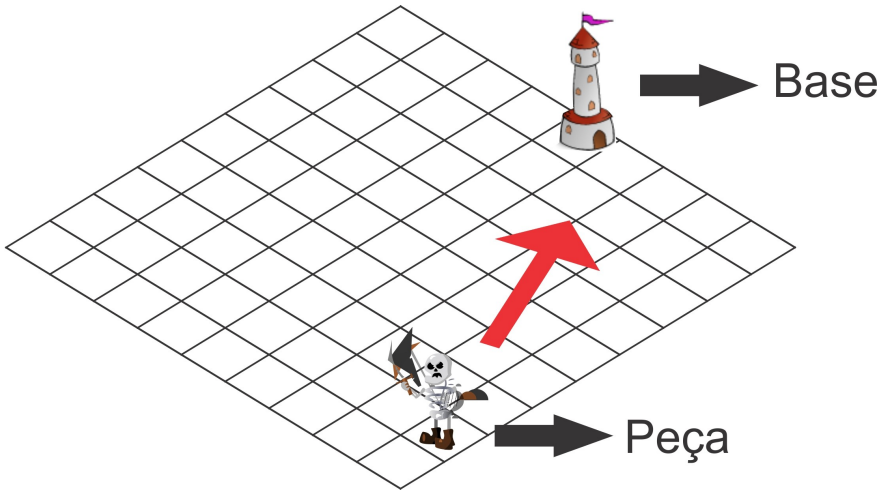
4.1.1 Ambiente do jogo

O ambiente do jogo, consistirá em um mapa no estilo isométrico,(PAZERA,2001) define o mapa que será utilizado nesta proposta como, mapa isométrico estilo diamante.



4.1.2 Objetivos

O objetivo do jogo define quais ações o jogador deve realizar para ser considerado o vencedor da partida, neste jogo o objetivo principal é alcaçar a “base”, que consiste em um determinado local no mapa, a primeira equipe que mover uma peça para casa que possui a base, será definida como a vencedora, também será considerada vencedora a equipe que eliminar todas as peças adversárias.



4.1.3 Regras

As regras especificam os conceitos utilizados no jogo e o que cada equipe pode fazer durante uma partida:

Turno: Consiste no momento em que é possível um jogador realizar uma jogada, cada jogador irá possuir um turno, e os turnos serão intercalados entre as duas equipes. A cada turno cada jogador pode realizar apenas uma ação: Atacar ou mover;

Peça: Peça são os componentes das equipes, que serão controladas pelos jogadores, no início da partida, cada jogador possuirá 4 peças. Cada peça possuirá dois atributos: Ataque e Vida

Atributo Vida: É um atributo que cada peça no mapa possui, no início da partida, as peças possuirão o valor máximo do atributo, se durante a partida a peça receber um ataque que fará a sua vida chegar a zero, a peça atacada é eliminada.

Atributo Ataque: É um valor inteiro que representa o valor que será subtraído do atributo “vida” da carta inimiga ao receber um ataque.

Ação Ataque: Ataque é uma ação que o usuário poderá realizar, ao atacar uma peça

adversária o usuário gasta o seu turno, e é subtraída da vida do adversário o valor do atributo “ataque” da peça.

Mover: É a ação que o jogador pode realizar para fazer com que a peça no mapa troque de casa, cada peça pode mover no máximo 3 casas por vez.

Mapa: É o ambiente onde a partida será desenvolvida, consiste em um mapa isométrico com tamanho de 16 casas, onde em determinadas casas o usuário pode mover-se e em outras, que representarção obstáculos, não poderá mover-se.

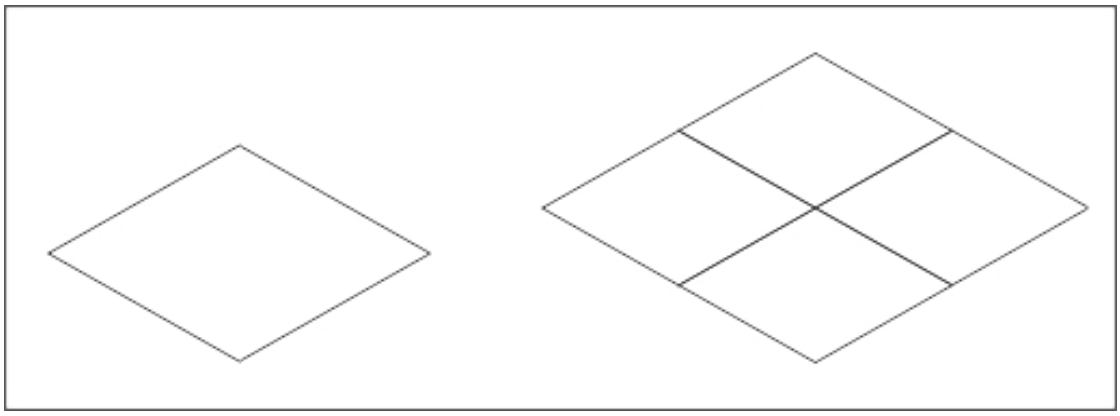
4.2 Módulos básicos

Os módulos são as estruturas que permitirão o funcionamento adequado do jogo. Cada módulo irá conter propriedades específicas em relação ao seu papel no funcionanemento do jogo. Cada módulo também deve disponibilizar um meio de conexão com os demais módulos do sistema.

4.2.1 Módulo de Interface

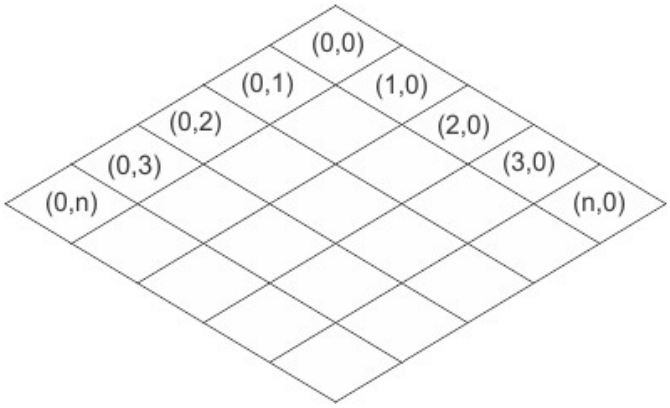
O módulo de interface, é o módulo responsável por manipular todos os aspectos visuais do jogo, permitindo que o usuário visualize o estado atual do mapa.

Pelo fato do jogo em questão se tratar de um mapa isométrico, composto de casas, será adotada a técnica chamada de *tile-based game*, que segundo (PAZERA,2001), é um tipo de jogo onde todos seus aspectos visuais são constituídos por pequenas imagens, que formam o um contexto maior, como o cenário, personagens e detalhes do mapa. Com isso, o mapa será constítuído por uma matriz de *tiles* que representarão cada parte do cenário:



A matriz de *tiles* fornecerá os mecanismos para navegação dos agentes controlados

pela IA e também pelo usuário, através dos índices da matriz, será possível deslocar as peças através do mapa, a organização da matriz se dará como mostra a figura:

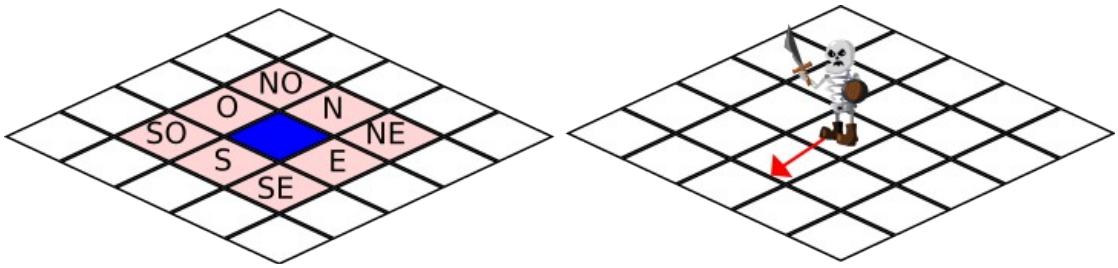


Deste modo será possível controlar a exibição de cada parte do mapa, através do acesso a matriz que o representa, cada célula da matríz conterá uma estrutura de dados que trará as características visuais de cada casa(tile), defindo a imagem que a casa deve exibir e se casa é apenas parte do cenário ou se possui um agente, o objeto a seguir define tais parâmetros:

```
class Casa {
        Object imagemCenario;
        Object imagemPersonagem;
        Boolean possuiAgente;
}
```

Através da matriz de casas, será definido também o sistema de navegação do mapa, ou seja a maneira na qual as peças irão se mover.

Cada casa, representada pelos campos da matriz, obedecerá a seguinte definição de coordenadas:



Na imagem, são exibidas as coordenadas que serão utilizadas na navegação do jogo, para indicar a direção na qual a peça irá se mover, serão utilizadas as nomeclaturas do pontos cardeais: Norte, Sul, Leste, Oeste, Nordeste,Noroeste,Sudeste e Sudoeste.

Na imagem acima, também é exibido um exemplo da requisição de movimentação de uma peça que está se deslocando uma casa a partir da sua posição original em direção ao Sul.

A partir do momento que a matriz é iniciada com os valores padrões, com a definição de cada casa, e do local onde cada agente se encontra do mapa, o módulo visual estará renderizando dos os aspectos visuais necessários para que seja possível iniciar uma partida.

Após iniciada a partida, o módulo de interface irá prover acesso às sua estrutura interna(matriz), disponibilizando aos demais módulos, informações para que realizem suas respectivas tarefas. No módulo de interface também estarão os manipuladores de eventos que irão fornecer informações para o módulo de controle.

Ao receber um determinado evento, o módulo de interface irá processá-lo e fazer uma requisição ao módulo de controle, após o processamento destes dados, o módulo de interface receberá uma requisição vinda do módulo de controle.

Para que a manipulação do módulo de interface seja possível, deverá ser disponibilizado os métodos que irão de fato manipular os aspectos visuais do jogo. Tais métodos irão receber como parâmetros dados vindos da camada de controle, ficando a cargo do módulo de interface executar a movimentação das peças no mapa.

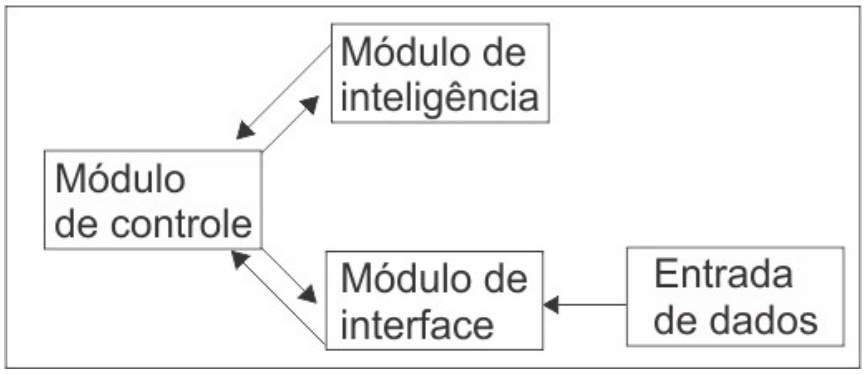
Os métodos disponibilizados para manipularem o mapa, normalmente irão seguir o padrão de possuir em sua nomeclatura a ação que irá executar, e como parâmetro o objeto no qual a ação será realizada:

```
...
public void movePeca(Peca peca) {
        ...
}
...
```

4.2.2 Módulo de Controle

O módulo de controle, será responsável pelas tarefas de manipulação do fluxo de dados entre os dados de entrada do usuário, o módulo de interface e o módulo de inteligência.

A partir deste módulo, serão acessados os dados provenientes do módulo de interface bem como o envio de tais dados para o processamento no módulo de inteligência. De maneira geral, o fluxo de dados que será manipulado pelo módulo de controle seguirá o seguinte modelo:

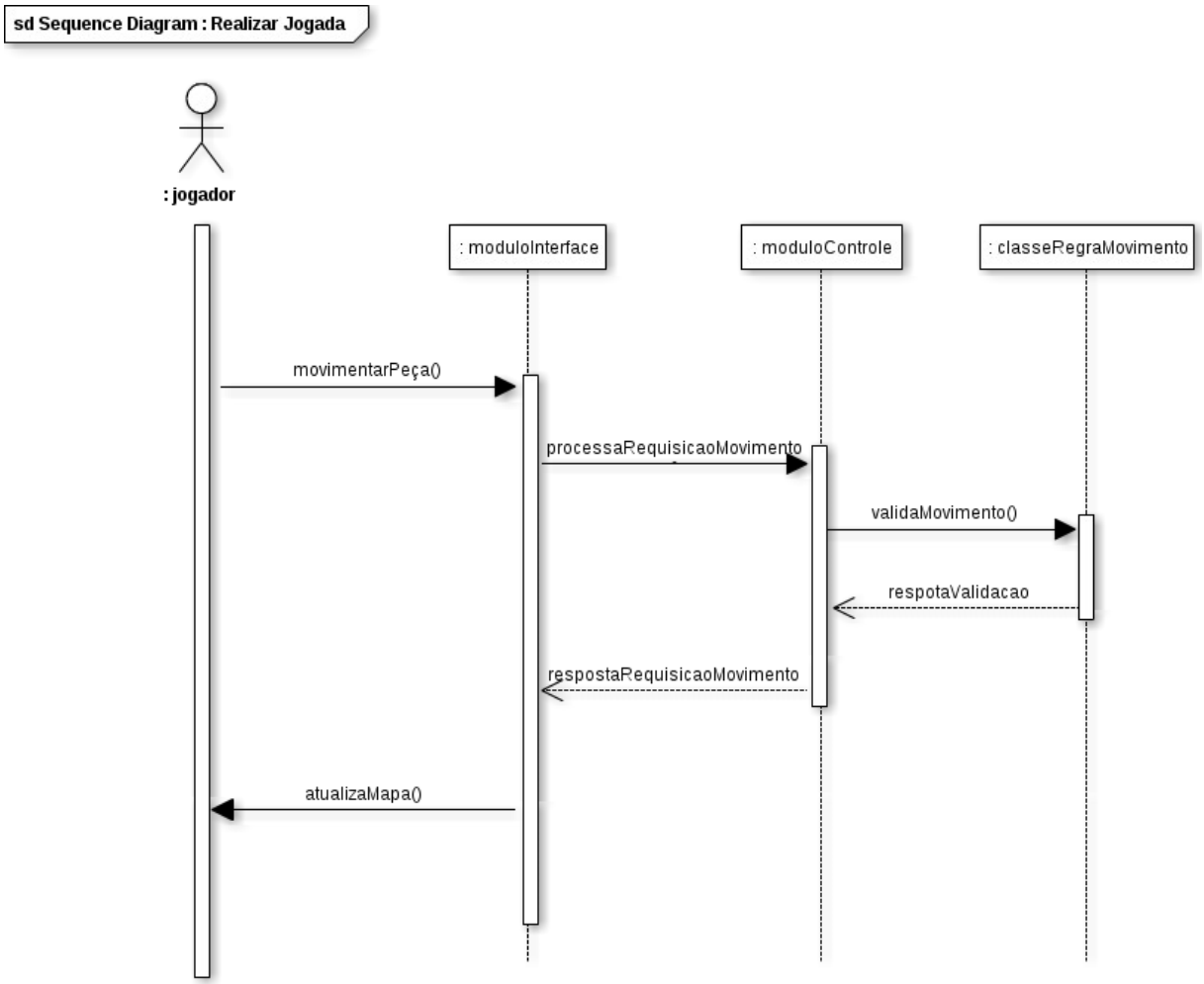


Todas entradas de dados feitas pelo usuário, como por exemplo, a solicitação de movimento de uma peça, será processado no módulo de controle. Em suas estruturas internas, o módulo de controle irá manipular classes especializadas em determinadas tarefas, como as regras do jogo.

Nas classes de regras, estarão definidos todos os conjuntos de especificações a cerca do funcionamento de cada aspecto do jogo, após o processamento e validação das requisições, o módulo de controle poderá acessar o módulo de interface para que seja exibido ao usuário o resultado da sua interação. Portanto, qualquer classe que controle qualquer aspecto do jogo, deve ser implementada no módulo de controle, cabendo a esse módulo a tarefa de receber uma requição, identificar quais regras devem ser aplicadas e devolver ao usuário a resposta adequada.

Durante a execução do sistema, a classe de controle irá verificar cosntantemente o estágio atual do jogo, pois a partir da leitura correta destes dados, irá definir, baseado em suas classes de regras, qual ação deve ser executada em um determinado momento. Todas a requisições que serão feitas ao módulo de inteligência, partirão da classe de controle.

No diagrama a baixo, é exibido como o fluxo de uma requisição de movimentação de uma peça será tratada pela estrutura definida no jogo:



4.2.3 Módulo de Inteligência

No módulo de inteligência, serão processadas todas as informações relativas ao controle dos agentes automatizados do jogo.

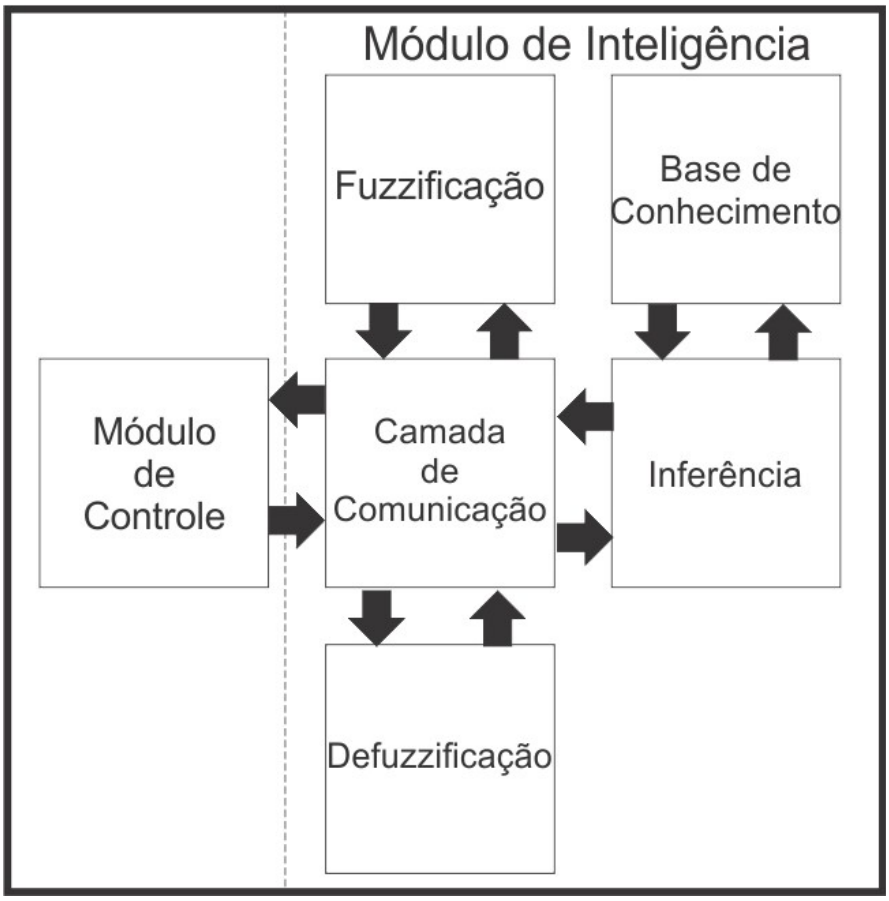
A partir do módulo de controle, serão requisitadas chamadas a camada de IA, estas requisições consistirão em enviar a camada de inteligência o pedido para a resolução de algum problema relativo ao funcionamento do jogo.

O módulo de inteligência possuirá toda a estrutura necessária para processar a requisição aplicando a lógica Fuzzy na resolução dos problemas. Tais funcionalidades devem ficar organizadas em classes especializadas dentro do módulo de inteligência.

Cada classe será responsável por realizar uma parte específica do processamento das requisições feitas ao módulo de inteligência.

Basicamente, o módulo de inteligência ficará organizado da seguinte forma:

A seguir será desenvolvida uma explicação sobre o papel de cada parte do módulo de inteligência.



4.2.3.1 Camada de Comunicação

A camada de comunicação funcionará como uma “interface” para acesso(a partir do módulo de controle) à camada de inteligência, nesta classe serão disponibilizados métodos para a requisição das funcionalidades do módulo.

Estes métodos serão responsáveis por realizar todas as tarefas para que o processamento necessita para formular uma resposta à camada de controle.

Basicamente, as requisições ao módulo de inteligência serão realizadas após uma interação realizada pelo usuário. Após realizada uma jogada, a camada de controle irá atualizar a matriz do mapa, de maneira que ela represente o estágio atual do jogo(ex. posição de peças), pois o mapa será a estrutura que será utilizada pela camada de inteligência para que sejam tomadas decisões.

A estrutura básica de um método da classe que irá gerenciar a Camada de

Comunicação será a seguinte:

```
...
public class camadaInterface {
        ...
        Mapa[] mapaAtual;

        public camadaInterface(Mapa[] mapaAtual){
                this. MapaAtual = mapaAtual;
        }

        public processaJogada(){
                // Aqui serão chamadas as demais classe
                // que participam do processamento da
                //requisição

                ...
        }

        ...
}
...
```

No exemplo acima, o método “processaJogada” será responsável por tratar todas as fases do processamento, desde a chegada do pedido vindo da classe de controle até as chamadas e tratamento de respostas vindas das demais classes que participam do fluxo.

4.2.3.2 Fuzzificação

Nesta estrutura, será realizada a transformação dos dados vindos da camada de controle em dados que possam ser utilizados pelas demais estruturas do módulo de inteligência.

De maneira geral, serão definidas variáveis que serão utilizadas pelo módulo de inteligência, estas variáveis serão processadas sempre que uma requisição for feita, como por exemplo:

- Quantidade de vida de cada peça do mapa

- Distância das peças adversárias em relação ao objetivo do jogo(base)
- Distância das peças do agente automatizado em relação ao objeitvo do jogo
- Quantidade de peças dos dois times
- Perigo atual de cada peça do tabuleiro(Calculado a partir da quantidade de peças adversárias próximas)
- etc

Após a definição de cada uma das variáveis citadas, o sistema poderá utilizar a camada de inferência para que seja definido um grau de saída baseado nas regras definidas na base conhecimento.

4.2.3.2 Inferência

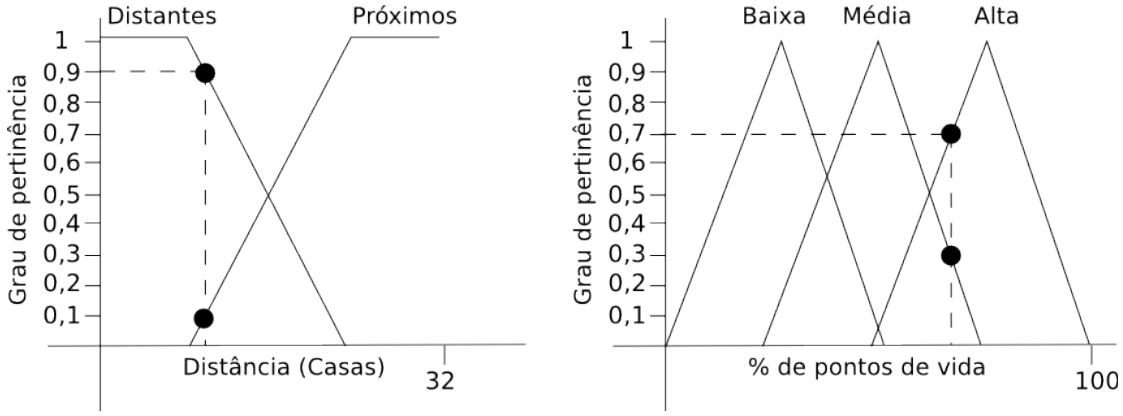
Nesta estrutura, serão aplicadas as regras às variáveis definidas na fase de fuzzificação. Estas regras irão, de fato, produzir os resultados que irão definir a decisão que será tomada pelo mecanismo de IA do sistema.

Segundo (MILLINGTON,2006), as regras são atividades que consistem em transformar os membros de um determinado conjunto, em membros de outro conjunto calculando assim a sua pertinência em tal grupo.

O formato geral de um regra fuzzy é a seguinte:

SE <ATECENDENTE> ENTÃO <CONSEQUENTE>

A partir das variáveis que sofream processo de fuzzificação, é possível conhecer o seu grau de pertinência para cada conjunto no qual o processo foi realizado, com estes graus, será possível aplicar os operadores lógicos Fuzzy, para que assim seja possível definir a resposta para a regra:



Na figura acima, é ilustrado o processo de fuzzificação de variável que possui o grau de inferência 0,7 no conjunto “Vida Alta”, e grau 0,3 no atributo “Vida Média”, também é ilutrada uma variável que possui grau de pertinência 0,9 no conjunto “Adversários próximos” e 0,1 no cojunto “Adversários Distantes”.

De posse destes dados de exemplo, a inferência das regras sobre a variavel pode ser realizada. Estes são exemplos de regras que serão aplicadas no jogo:

Regra1:

SE VIDA_MEDIA E ADVERSARIO_DISTANTES ENTAO MOVER_DIRECAO_BASE

Regra 2:

SE VIDA_MEDIA E ADVERSARIO_PROXIMOS ENTAO MOVER_DIRECAO_TIME

Regra 3:

SE VIDA_ALTA E ADVERSARIO_PROXIMOS ENTAO LUTAR

É enumerado por (MILLINGTON,2006) as seguintes regras para a realização das operações lógicas básicas em sistemas Fuzzy:

Expressão	Equação Fuzzy
NOT A	1 - Ma
A AND B	min(Ma, Mb)
A OR B	max(Ma, Mb)
A XOR B	min(Ma, 1 - Mb)
A NOR B	1 – max(Ma, Mb)
A NAND B	1 – min(Ma,Mb)

Onde Mx é o grau de pertinência da váriavel no conjunto X. As funcões max() e min() retornam, respectivamente, o maior e o menor valor recebido via parâmetros.

Pela tabela acima, podemos perceber que o operdor lógico AND, é definido pela seguinte equação Fuzzy:

A AND B => min(Ma,Mb), onde Ma é o valor da variável no grupo A e Mb é o valor da variável no grupo B

Com isso, podemos realizar o processo de inferência sobre os dados adquiridos:

M(regra 1) = min(0,3;0,2) = 0,2;
M(regra 2) = min(0,1;0,9) = 0,1;
M(regra 3) = min(0,7;0,9) = 0,7;

Após realizado este processo para cada aspecto que deve ser levado em consideração pelo sistema de inteligência, será possível realizar o próximo passo do processamento.

4.2.3.3 Defuzzificação

A fase de defuzzificação é onde os valores processados na atividade de inferência são analisados e será definida qual ação resultou de todo o processo de aplicação da lógica Fuzzy, ou seja, o valor de cada regra é analisado com o objetivo de encontra qual regra melhor se encaixou com os dados de entrada fornecidos.

Segundo (MILLINGTON, 2006), existem diversas maneiras de realizar a defuzzificação, entre elas, a considerada mais simples, que é a escolha do conjunto com o maior grau de pertinência, embora seja considerado um método de rápido processamento e de fácil implementação, ele é também considerado simplificado demais.

Utilizando os dados anteriores, e aplicando a defuzzificação nos resultados das regras, teríamos o seguinte resultado:

LUTAR = min(0,7;0,9) = 0,7;
MOVER_DIRECAO_TIME = min(0,1;0,9) = 0,1;
MOVER_DIRECAO_BASE = min(0,3;0,2) = 0,2;

Neste caso, o resultado final do processo de aplicação da lógica fuzzy a ação Lutar, com um grau de 0,7.

4.2.4 Finalização da Requisição

Após todo o processo fuzzy, o controlador possuirá uma ação que pode ser aplicada a a uma peça. Neste momento, o controlador irá fazer uma requisição ao módulo de interface, exibindo o resultado do processamento, e assim, dando continuidade ao ciclo do jogo.

REFERÊNCIAS

TOZOUR, Paul. The Evolution of Game AI. In: Rabin, Steve. **AI Game Programming Wisdom**. Rockland, MA, USA: Charles River Media, 2002. p. 03-15.

TRISTAN, Donavan. **Replay: history of video games**. Lewes: Yellow Ant, 2010.

WOLF, Mark. **The Video Game Explosion: A History from PONG to PlayStation® and Beyond** . Westport, Greenwood Press , 2008.

KIRBY, Neil. **Introduction to Game AI** . Boston, Cengage Learning , 2011.

SOUSA, Bruno. **Game Programming All in One** . USA, Premier Press. , 2002.

BARTON, Matt. **Dungeons and Desktops The History of Computer : Role-Playing Games** . Natick, A K Peters , 2008.

PAZERA, Ernest. **Isometric Game Programing With DirectX 7.0,** USA, Prima Publishing , 2001.

KENT, Steven. **The Ultimate History Of Video Games: From Pongo to Pokémon and beyond,** Nova Iorque, Three Rivers Press, 2001.

MILLINGTON, IAN. **ARTIFICIAL INTELLIGENCE FOR GAMES** San Francisco, Elsevier., 2006.

NUNES,Douglas. **Mercado de jogos eletrônicos explode no Brasil.** Portal IG. 2013. Disponível em: <http://economia.ig.com.br/empresas/2013-12-26/mercado-de-jogos-eletronicos-explode-no-brasil.html>. Acesso em: 29 de Maio. 2014

LANNOY, Carlos. **Brasil lidera crescimento do mercado de jogos eletrônicos em 2012.** Portal G1. 2013. Disponível em: <http://g1.globo.com/jornal-da-globo/noticia/2013/05/brasil-lidera-crescimento-do-mercado-de-jogos-eletronicos-em-2012.html >. Acesso em: 26 de Maio. 2014

PETRÓ, Gustavo. **Pesquisa do BNDES mostra potencial do mercado de games no Brasil Gustavo Petró .** Portal G1. 2013. Disponível em: <http://g1.globo.com/tecnologia/games/noticia/2014/04/pesquisa-do-bndes-mostra-potencial-do-mercado-de-games-no-brasil.html>. Acesso em: 26 de Maio. 2014

YUGE, Claudio. **Audiência pública discute futuro dos games no Brasil.** Portal G1. 2014. Disponível em: <http://corporate.canaltech.com.br/noticia/games/Audiencia-publica-discute-futuro-dos-games-no-Brasil/>. Acesso em: 29 de Maio. 2014