

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

**JOÃO ANTÔNIO PRESTES MATIUZZI**

**DESENVOLVIMENTO DE UM JOGO DIGITAL APLICANDO LÓGICA FUZZY**

**Porto Alegre**  
**2017**

João Antônio Prestes Matiuzzi

## DESENVOLVIMENTO DE UM JOGO DIGITAL APLICANDO LÓGICA FUZZY

Artigo apresentado como requisito parcial  
para obtenção do título de Bacharel em  
Sistemas de Informação da Universidade,  
pelo Curso de Sistemas de Informação da  
Universidade da Universidade do Vale do  
Rio dos Sinos - UNISINOS

Orientador(a): Prof. Dr. Vinícius Jurinic Cassol

Porto Alegre

2017

## **DESENVOLVIMENTO DE UM JOGO DIGITAL APLICANDO LÓGICA FUZZY**

João Antônio Prestes Matiuzzi

Prof. Dr. Vinícius Jurinic Cassol

**Resumo:** O sistema de Inteligência Artificial de um jogo é um dos principais aspectos que definem o desafio e o entretenimento que um jogo proporciona ao usuário, pois a implementação do sistema de IA irá definir características como a imprevisibilidade das ações que os agentes irão realizar, fazendo com que o usuário sinta que o jogo está reagindo de maneira inteligente ao momento atual do jogo, fazendo com que o usuário sinta-se desafiado a jogar. Com isso, torna-se importante o estudo das técnicas existentes para o desenvolvimento dos sistemas de IA dos jogos. Neste trabalho, será apresentado um modelo que utiliza a Lógica Fuzzy para o controle dos agentes automatizado, a partir deste modelo, é então desenvolvido um jogo de estratégia, no estilo RPG em turnos, para a validação do modelo proposto. Utilizando o jogo desenvolvido, será testada a eficiência do modelo, comparando o mesmo com um modelo de IA baseado em decisões aleatórias.

**Palavras-chave:** Lógica Fuzzy. Jogos Digitais. Inteligência Artificial.

### **1 INTRODUÇÃO**

Os jogos digitais possuem relação com a inteligência artificial desde o seu surgimento, na década de 1970 (TOZOUR,2003). Sendo assim, nos dias atuais, o estudo da inteligência artificial é um dos principais aspectos do processo de desenvolvimento de jogos. Os agentes controlados pelo computador são vistos como substitutos de outros jogadores humanos pelo usuário, com isso, é importante trazer o comportamento dos agentes o mais próximo possível ao comportamento humano: ser imprevisível em relação a tomada de decisão em uma determinada jogada, mas ser previsível quando uma jogada se mostra claramente a melhor opção (TOZOUR,2003). Com isso, podemos destacar que o sistema de Inteligência Artificial deve proporcionar variações de jogadas para chegar aos objetivos do jogo, fazendo que os agentes tenham diferentes ações e que as mesmas não se repitam de maneira a oferecer ações previsíveis demais, desta maneira, o jogador terá o sentimento de que o sistema está reagindo às suas jogadas utilizando diferentes estratégias, e portanto realizando ações coerentes e que proporcionam desafio ao usuário.

Como exemplos de sistemas de Inteligência Artificial aplicadas em jogos, podemos citar os jogos de xadrez, quem tem sido objeto de estudo há muito tempo, jogos como *Chessmaster 2000*<sup>1</sup>, implementaram sistemas de AI notáveis, proporcionando ao usuário jogos considerados desafiantes. (TOZOUR,2003). Podemos também citar jogos no estilo *RTS (Real Time Strategy)* como *WarCraf II*<sup>2</sup> e *Age of Empires 2*<sup>3</sup>, que, apesar de possuírem a dificuldade de existir uma grande quantidade de agentes a serem controlados simultaneamente, conseguiram proporcionar bons resultados (TOZOUR,2003).

Atualmente, as produtoras de jogos desenvolvem sistemas de inteligência de artificial cada vez mais desafiadores e realistas, a partir do momento em que existem, nos dias atuais, melhores hardwares e técnicas de programação em relação aos disponíveis nos primórdios do desenvolvimento de jogos. Com isso, tornou-se possível a implementação de sistemas de IA mais complexo, tornando a inteligência artificial um componente importante na qualidade final do jogo, e por consequência no entretenimento que o jogo proporciona ao usuário.

Dessa forma, é importante o estudo das diversas técnicas utilizadas para a implementação destes sistemas de IA. Neste artigo, é estudada a Lógica Fuzzy (ZADEH, 1965), que pode ser definida como uma técnica que expande o conceito da lógica booleana. Na Lógica Fuzzy, uma determinada variável possui um valor que determina o grau da participação da mesma em um determinado grupo, ou seja, ao invés de definirmos se um valor pertence ou não a um grupo, definiremos nível de participação do valor em um determinado grupo.

A partir do modelo proposto, será descrito o desenvolvido um jogo estilo RPG, onde os agentes serão controlados por um motor de IA que irá aplicar os conceitos da Lógica Fuzzy. Demonstrando os principais aspectos da Lógica Fuzzy, será descrito como o motor de IA do jogo foi implementado, e a maneira pela qual ele interage com o restante do sistema. Também será descrito como os módulos do jogo foram organizado, e a função de cada um como, por exemplo: módulo de controle e o módulo de inteligência.

---

<sup>1</sup> Jogo de Xadrez desenvolvido pela Software Tool Works:  
<http://www.old-games.com/download/3205/chessmaster-2000>

<sup>2</sup> *RTS (Real Time Strategy)* desenvolvido pela Blizzard: <http://us.blizzard.com/en-us/games/legacy/>

<sup>3</sup> *RTS (Real Time Strategy)* desenvolvido pela Ensemble: <https://www.ageofempires.com/games/aoeii/>

## 2 JOGOS DIGITAIS E INTELIGÊNCIA ARTIFICIAL

A inteligência artificial está ligada aos jogos digitais desde os primeiros jogos desenvolvidos, jogos como Pac-Man<sup>4</sup>, *Pong*<sup>5</sup> e *Space Invaders*<sup>6</sup> já utilizavam de técnicas simples baseadas em regras e tomada de decisões aleatórias. O jogo *Pac-man* foi implementado utilizando uma técnica muito simples de Inteligência Artificial: Máquina de Estados Finitos, onde cada agente controlado pelo sistema possuía dois estados: Perseguindo ou Fugindo, e em cada estado o agente poderia escolher rotas aleatórias tanto em direção ao agente controlado pelo usuário ou para longe do mesmo, como é exibido na Figura 1. Já no jogo *Pong* o sistema simplesmente seguia a posição atual da bola. (MILLINGTON,2006).

Figura 1 – Jogo Pac-Man: Agentes automatizados em estados distintos: Perseguindo ou Fugindo



Fonte: <https://www.bandainamcoent.com/games/pac-man-official>

Embora fosse uma parte importante da diversão que o jogo proporciona ao usuário, o sistema de inteligência artificial era, no princípio, uma tarefa secundária em relação aos demais aspectos do jogo, como som e gráficos. Mas atualmente esse pensamento tem mudado, sendo até considerado que o sistema de IA pode ser um fator que poderá definir o sucesso de um jogo (TOZOUR,2003).

---

<sup>4</sup> Jogo desenvolvido pela Namco, publicado em 1980: <https://pt.wikipedia.org/wiki/Pac-Man>

<sup>5</sup> Jogo desenvolvido pela Atari, publicado em 1975: <https://pt.wikipedia.org/wiki/Pong>

<sup>6</sup> Jogo desenvolvido pela Taito Corporation, publicado em 1978:  
[https://pt.wikipedia.org/wiki/Space\\_Invaders](https://pt.wikipedia.org/wiki/Space_Invaders)

Podemos definir a inteligência como a “habilidade de agir de maneira inteligente frente a condições distintas” (KIRBY,2011), ou seja, se aplicarmos o conceito aos jogos, os agentes automatizados devem perceber o ambiente nos quais estão inseridos e tomar decisões baseadas nele. Desta forma, o usuário irá perceber que os agentes funcionam de maneira inteligente, correspondendo ao cenário atual do jogo e às suas ações. Neste ponto, fica evidente que o principal objetivo do sistema de IA em um jogo, deve ser apresentar um comportamento similar a de um ser humano (MILLINGTON,2006).

Considerando a importância do sistema de IA em jogo, torna-se importante o estudo das diversas técnicas para a implementação do mesmo. Atualmente, existem uma grande quantidade de técnicas para o desenvolvimento da IA:

- a) **Sistemas Especialistas:** É criada uma base onde são criadas regras com base no conhecimento de um especialista em um determinado assunto. O objetivo deste modelo é selecionar a regra que um especialista selecionaria.
- b) **Máquinas de Estado Finito:** Nesta técnica, é definida uma estrutura onde um agente pode estar em um número determinado de Estados que representam diferentes ações que o agente pode realizar, também são definidas Transições que alteram o estado atual da Máquina.
- c) **Árvores de Decisão:** É um modelo onde é definido um conjunto de condicionais em forma de uma estrutura de dados de árvore, onde o resultado da avaliação de um nó pai define qual nó filho será executado, até que seja selecionada uma ação que o agente irá realizar.
- d) **Algoritmos Genéticos:** Neste modelo, é simulado o mecanismo de evolução e genética onde soluções são aleatoriamente selecionadas em busca das melhores soluções.

Ao decidirmos qual abordagem devemos aplicar na hora da modelagem de um sistema de IA, é necessário compreender quais as características do sistema que irão trazer interatividade e diversão para o jogador. Com isso, é importante perceber que nem sempre os sistemas mais complexos de IA são os mais apropriados, mas sim os que são codificados de maneira mais simples e direta e que

apresentam o melhor comportamento dos personagens controlados pelo sistema de IA (MILLINGTON, 2006; KIRBY, 2011).

É importante também compreender como o sistema de IA pode impactar negativamente um jogo. Segundo Kirby (2011), o mais complexo em um sistema de IA em um jogo é o modelo decidir quando o seu comportamento é apropriado para o ambiente e as condições nas quais o agente se encontra. Esta tarefa pode ser difícil de ser executada, pois o sistema de Inteligência Artificial é utilizado na resolução de diversos problemas relacionados a mecânica do jogo, como por exemplo: Realizar tomada de decisões de quando movimentar um personagem, encontrar o melhor caminho para o movimento do agente ou escolher uma estratégia de jogo.

Por possuir tais responsabilidades, o modelo de IA de um jogo deve considerar o contexto atual do jogo, ou seja, o sistema de IA deve fornecer ao usuário um ambiente com agentes inteligentes que tomam suas decisões baseadas nas regras do jogo levando em consideração qualquer cenário em que o jogo pode se encontrar, fornecendo ao usuário percepção de que ele está participando de um jogo onde os agentes automatizados tomam decisões similares as de que um humano poderia tomar, oferecendo assim um nível satisfatório de diversão e desafio.

### **3 LÓGICA FUZZY**

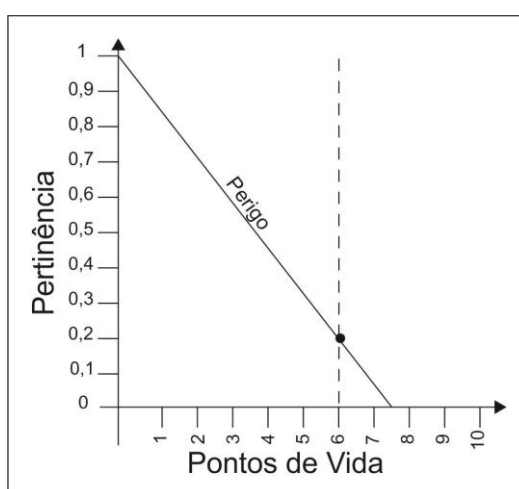
A Lógica Fuzzy (ZADEH, 1965), diferentemente da noção clássica de conjuntos onde um elemento pertence ou não pertence a um conjunto, define o grau de pertinência que um elemento possui em relação a um determinado conjunto. O termo pertinência se refere o quão próximo do conceito definido do conjunto o elemento em discussão está. A Lógica Fuzzy tem sido utilizada para diferentes tipos de tomada decisão, podemos citar situações onde é aplicada para o controle de temperatura em um determinado ambiente, utilizando como parâmetro a humidade e a temperatura do ambiente para determinar a velocidade em que um ventilador deve ser acionado; em sistemas espaciais pode ser aplicada para o controle da altitude de satélites. No desenvolvimento de jogos, por exemplo, podemos definir o estado em que um determinado agente automatizado se encontra, para que a partir disto sejam tomadas decisões sobre as ações que o agente deve tomar.

Utilizando a definição clássica da teoria dos conjuntos, poderíamos ter um conjunto chamado "Perigo", que define se um agente se encontra em uma situação

de perigo baseado na quantidade de “Pontos de Vida” que possui no momento. Com base nos valores de entrada, o elemento em discussão poderia pertencer ou não ao conjunto "Perigo". Já na Lógica Fuzzy, seria aplicado o conceito de conjunto difuso, onde seria definido o conjunto "Perigo", mas ao invés de apenas definirmos se um elemento pertence ou não ao conjunto, iríamos definir o grau de exatidão que o valor de entrada tem em relação ao conceito do conjunto.

A Figura 2 a seguir ilustra como ficaria o grupo difuso "Perigo":

Figura 2 – Grau de Pertinência por Pontos de Vida



Fonte: Elaborado pelo autor

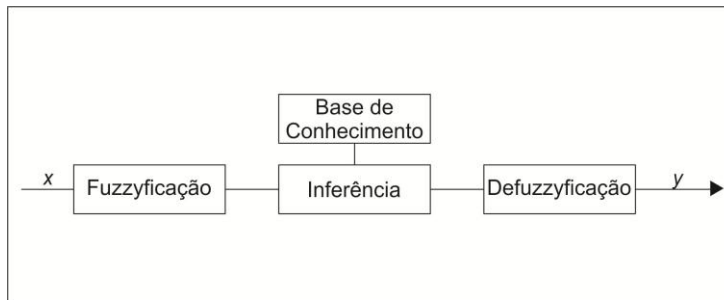
Interpretando a figura, podemos perceber que o elemento tem pertinência diferente em relação a variação da quantidade de vida fornecido pelo dado de entrada, no exemplo, é fornecido o dado de entrada 6 “Pontos de Vida”, chegando a um valor de 0,2 de pertinência.

É possível observar que a mudança de estado do elemento em relação ao conjunto é mais suave do que se utilizássemos um conjunto clássico, ou seja, há uma faixa de valores nas quais podemos encaixar cada valor de entrada, para assim definirmos o grau de pertinência de um elemento em relação ao conjunto em questão.

Para realizar uma tomada de decisão, a Lógica Fuzzy define um processo que deve ser seguido, definindo uma sequência de manipulação de dados, a Figura 3 a seguir ilustra um sistema fuzzy básico:



Figura 3 – Base de Conhecimento



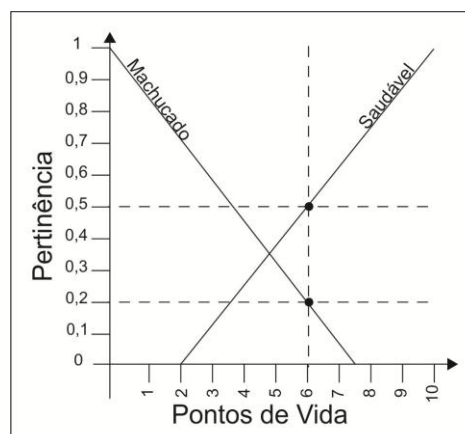
Fonte: Elaborado pelo autor

Na figura, temos o valor de entrada  $x$ , e um valor de saída  $y$ . Para chegarmos ao valor  $y$  foram processados os passos de Fuzzificação, Inferência e Defuzzificação, que serão explicados a seguir.


**Fuzzificação:** Para podermos trabalhar com conjuntos Fuzzy, os valores de entrada, que são os valores fornecidos ao sistema Fuzzy para que seja realizada a inferência, devem ser transformados em valor de grau de pertinência. Um elemento pode pertencer a diferentes conjuntos ao mesmo tempo, para um mesmo valor de entrada, este elemento possuirá diferentes graus de pertinência em determinados conjuntos, por exemplo:

Em um determinado jogo, um agente pode estar no estado "Saudável" e "Machucado", dependendo da quantidade de vida que possui. O valor de entrada neste caso, é a quantidade de vida e os conjuntos Fuzzy são "Saudável" e "Machucado", após o processo de fuzzificação, chegaremos ao grau de pertinência que o dado de entrada tem em relação a cada conjunto Fuzzy, como mostra a Figura 4:

Figura 4 – Grau de Pertinência por Pontos de Vida



Fonte: Elaborado pelo autor

A partir da figura  podemos perceber que para a quantidade de vida 6, o elemento possuirá um grau de pertinência de 0,5 no conjunto "Saudável", e 0,2 no conjunto "Machucado", a partir destes valores Fuzzyficados será possível realizar o próximo passo do processo da Lógica Fuzzy.

**Inferência:** Neste passo, serão utilizadas regras obtidas através de um banco de conhecimento. Estas regras relacionarão os conjuntos em questão e fornecerão um resposta de acordo com os dados de entrada.

As regras possuem o seguinte formato:

**SE condição ENTÃO ação**

Estas regras irão definir o comportamento que o agente terá durante o jogo. Como mostra o exemplo a seguir:

**SE** *vida == baixa* **E** *inimigos\_proximos == muitos* **ENTÃO** *fugir*

Neste momento são aplicadas as operações lógicas sobre as regras. Na lógica tradicional, utilizamos uma tabela verdade para chegarmos a uma resposta de uma proposição, já na Lógica Fuzzy, são utilizadas regras numéricas para calcular o grau de verdade de uma proposição, baseado no grau de verdade dos valores de entrada. A seguir, são enumeradas as operações lógicas Fuzzy básicas:

**Tabela:** Relação das operações Fuzzy básicas.

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	$1 - \min(Ma, Mb)$

Fonte: Elaborado pelo autor

**Defuzzyficação:** Após o processo de inferência das regras da base de conhecimento do jogo, é necessário que seja fornecido um dado de saída útil para o sistema, ou seja, um dado que possa ser utilizado pelo sistema para que a ação do agente automatizado seja realizada.

A partir dos resultados obtidos após a inferência das regras, é possível realizar o processo de defuzzificação. Uma das técnicas de defuzzificação é a simples escolha da regra que possui o maior grau de pertinência que irá selecionar qual regra que possui maior grau de pertinência, chegando assim a um resultado.

A partir do resultado, o sistema pode realizar as demais tarefas relativas ao controle do agente, como movimentar ou realizar um ataque.



## 4 PROPOSTA DO JOGO

O modelo de jogo proposto foi criado visando a aplicação da Lógica Fuzzy no sistema de IA. Para isso o jogo possuirá agentes automatizados, e que a tomada de decisão dos mesmos fique clara para o usuário. Nas próximas seções serão abordadas as características de funcionamento do jogo.

### 4.1 Aspectos gerais

O estilo de jogo que será desenvolvido neste trabalho é conhecido como RPG por turnos. Neste estilo, dois jogadores se enfrentam realizando uma ação por vez, ou seja, após um dos jogadores efetuar uma jogada, o outro jogador pode efetuar a sua jogada, alterando os turnos até que o jogo chegue ao fim. Desta maneira, ficará claro a tomada de decisão do agente, que terá que analisar o cenário atual do jogo e reagir às jogadas do usuário.

Basicamente o jogo possuirá dois times compostos por 3 personagens, um dos times será controlado pelo jogador e o outro time pelo sistema de IA do jogo. Cada time possuirá a representação de uma Base, que deve ser defendida pelo jogador. Tanto os personagens como também a Base terão uma determinada quantidade de “vida” que é removida quando o personagem é atacado por uma unidade inimiga, ao ter toda a sua vida removida, a peça é removida do jogo. A Figura 5 ilustra os aspectos gerais do jogo:

Figura 5 – Aspectos gerais do jogo



Fonte: Elaborado pelo autor.

## 4.2 Objetivos do jogo

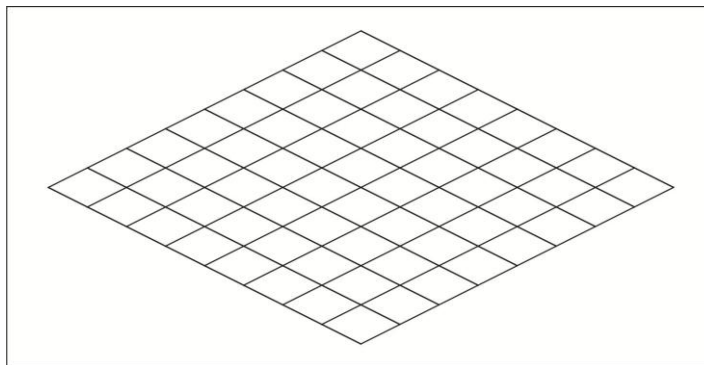
Para vencer uma partida, os jogadores devem utilizar as mecânicas de jogo disponíveis, que são o movimento das peças e o ataque às peças inimigas. Cada mecânica de jogo deve ser realizada levando em consideração a limitação de intervalo de casas nas quais cada peça pode realizar suas ações: cada peça só pode realizar suas jogadas nas 16 casas ao redor da casa onde ela se encontra. Por se tratar de um jogo no estilo RPG por turnos, cada jogador poderá executar uma ação a cada turno, devendo escolher entre atacar uma peça inimiga ou mover uma de suas peças.

O jogo possui dois objetivos, sendo que o jogador que atingir ao menos um dos objetivos primeiro será considerado o vencedor da partida. Um dos objetivos consiste em eliminar todas as peças do time inimigo, ou seja, o jogador deve, através do ataque às peças inimigas, remover toda a vida do personagem. O outro objetivo é atacar a base do time inimigo, ao remover todos os pontos de vida da base inimiga, o jogador será considerado vencedor.

## 4.3 Ambiente do jogo

O jogo será baseado em um mapa isométrico composto por casas, onde o jogador controlará um dos times na partida e o sistema de IA irá controlar o outro, na imagem a seguir é ilustrado como o mapa será representado visualmente, na Figura 6:

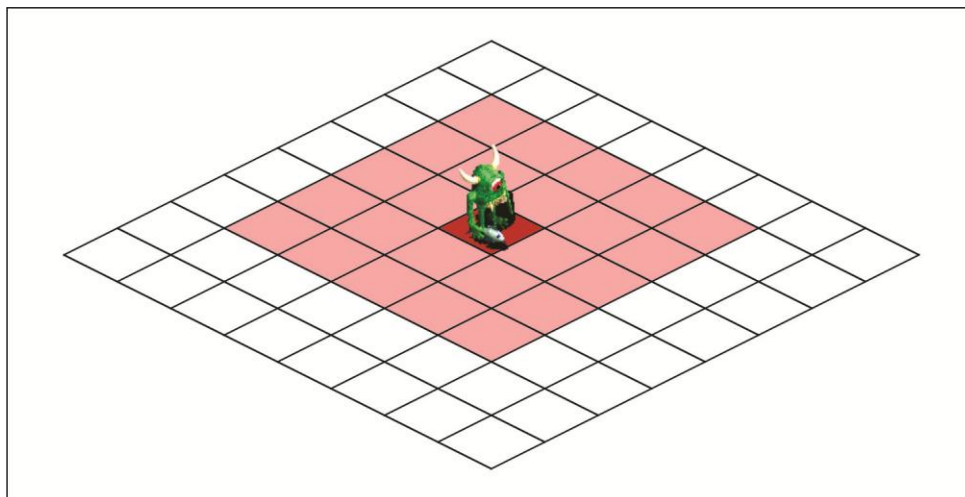
Figura 6 – Ambiente do jogo



Fonte: Elaborado pelo autor

As mecânicas do jogo são realizadas através do clique do mouse. Ao clicar em uma peça, serão exibidas as casas que estão dentro do limite possível de jogada, após selecionada a peça, se o usuário clicar em uma peça inimiga dentro do intervalo de casas, ele irá realizar um ataque a peça ou a base inimiga, se caso o usuário clicar em uma casa vazia, a peça irá se mover até a casa selecionada. Na Figura 7, é demonstrado o funcionamento das mecânicas básicas do jogo:

Figura 7 – Limite possível para realizar jogada



Fonte: Elaborado pelo autor

#### 4.4 Regras e definições do jogo

A seguir será enumerado os conceitos e regras que serão aplicadas ao jogo:

- **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.

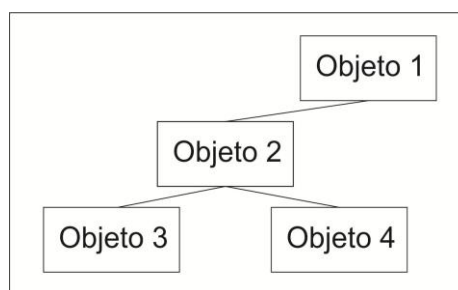
## 5 DESENVOLVIMENTO DO JOGO “CAÇANDO MONSTROS”

A partir do modelo definido neste trabalho, foi implementado um jogo como forma de validar a eficiência da aplicação da Lógica Fuzzy em jogos digitais. A seguir, será explicado como foi implementado o modelo proposto e o funcionamento de cada módulo desenvolvido.

## 5.1 Tecnologia

O jogo foi desenvolvido utilizando o *Starling Framework*<sup>7</sup>, que é uma biblioteca *Open Source* para desenvolvimento de jogos baseada na *API Stage3D*<sup>8</sup>, disponível em computadores desktop através da tecnologia *Adobe Air*<sup>9</sup>. Por se baseada na linguagem *ActionScript*<sup>10</sup>, facilita o desenvolvimento de jogos oferecendo suporte a uma hierarquia de objetos em forma de árvore, onde um objeto está ligado a um objeto pai, como mostra a Figura 8:

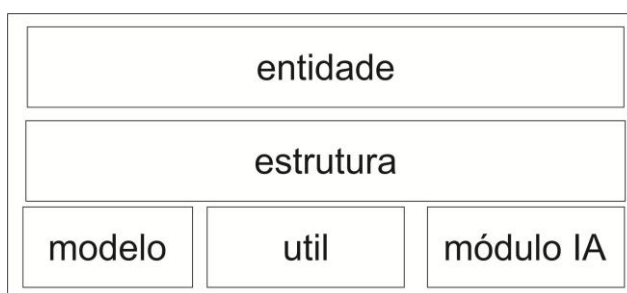
Figura 8 – Hierarquia de objetos



Fonte: Elaborado pelo autor

Por possuir esta característica, facilita a criação de estrutura em blocos, que conseqüentemente facilita a criação da estrutura do jogo em módulos responsáveis por cada aspecto do jogo. O jogo em questão foi desenvolvido em diferentes módulos que são demonstrados na Figura 9.

Figura 9 – Módulos do jogo



Fonte: Elaborado pelo autor

<sup>7</sup> Disponível em: <https://gamua.com/starling/>

<sup>8</sup> Mais informações: <http://www.adobe.com/devnet/flashplayer/stage3d.html>

<sup>9</sup> Disponível em: <https://get.adobe.com/br/air/>

<sup>10</sup> Documentação disponível em: <http://www.adobe.com/devnet/actionscript.html>

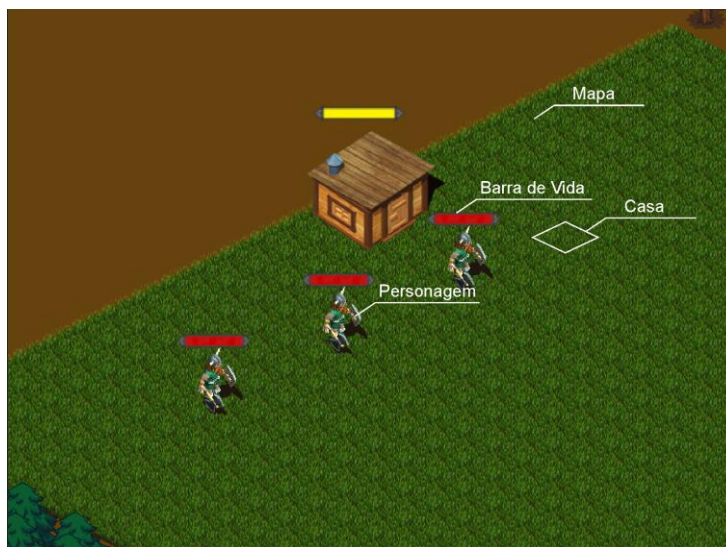


## 5.2 Módulo entidade

Neste módulo, estão localizadas as classes que controlam os objetos que serão exibidos na tela, fornecendo os métodos necessários para a manipulação das suas propriedades. A principal função do módulo entidade é a de fornecer uma interface para possibilitar que os demais módulos do sistema possam alterar o estado atual da tela do jogo. A partir da manipulação dos eventos de entrada realizada pelo usuário, este módulo irá realizar as devidas validações (de acordo com as regras do jogo) e efetuar a alteração dos elementos visuais do jogo.

Este módulo também é utilizado pelo módulo de Inteligência artificial para acessar as informações sobre o cenário atual do jogo, fornecendo assim uma maneira de extrair informações do estado atual do jogo com a finalidade de processar as informações e realizar a inferência. No desenvolvimento do jogo em questão, foram definidas as seguintes classes no módulo de entidades, ilustradas na Figura 10:


Figura 10 – Classes definidas no jogo




Fonte: Elaborado pelo autor

**Mapa:** Esta classe é responsável por exibir o mapa do jogo, ela contém todas as propriedades e parâmetros necessários para que sejam renderizados todos os aspectos relativos ao mapa do jogo, como as imagens utilizadas, as Torres e também as casas do mapa.



Além dos aspectos visuais do mapa, a classe  acesso às estruturas de dados necessárias para a manipulação das casas e das torres. Nesta classe, está localizada a matriz de casas, que é a estrutura responsável por armazenar uma referência a cada casa do mapa.

**Casa:** A classe Casa é responsável por processar todas as propriedades de cada casa do mapa, também possui variáveis que controlam o estado atual da casa. Nesta classe também são gerenciados os eventos que são realizados sobre cada casa.

**Personagem:** Nesta classe, estão os métodos responsáveis por manipular os personagens disponíveis para cada time de jogador. Esta classe oferece, por exemplo suporte ao movimento e ao controle das animações de movimento do  Também são controlados os eventos de entrada realizados pelo jogador.

**Barra de Vida:** Esta classe controla a exibição da barra de vida, que representa a quantidade de pontos de vida que cada personagem possui no momento.

### 5.3 Módulo estrutura

O módulo estrutura é responsável por armazenar as classes que são utilizadas como estrutura de dados para a troca de mensagem entre os módulos do sistema.

### 5.4 Módulo Útil

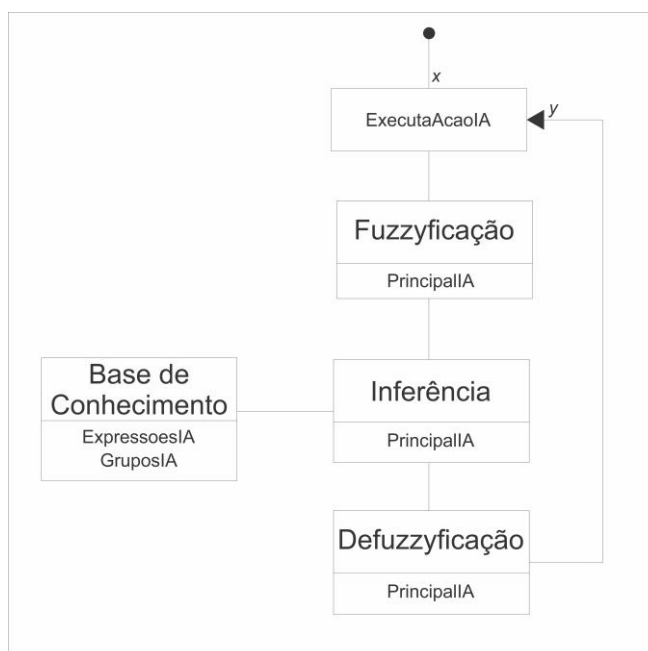
O módulo útil é responsável pelas classes de suporte ao sistema, ou seja, ele reúne funcionalidades que podem ser utilizadas em diferentes módulos, com o objetivo de gerar um reaproveitamento de códigos.

## 5.5 Módulo de Inteligência Artificial

Neste módulo estão localizadas todas as classes que controlam o motor de inteligência artificial do jogo que implementa a Lógica Fuzzy. Este módulo é responsável por realizar as tarefas referentes às atividades necessárias para a realização da inferência e tomada de decisão dos agentes automatizados do jogo.

A Figura 11 ilustra a relação de cada classe do módulo de Inteligência Artificial com o processo de inferência definida neste trabalho:

Figura 11 – Comunicação entre os módulos




Fonte: Elaborado pelo autor

A partir da figura acima, podemos observar que cada passo do processo definido na Lógica Fuzzy está relacionado a uma classe implementada no jogo. A estrutura do módulo de IA é acessado pelo sistema do jogo toda vez que o usuário realiza uma jogada; nesta hora, pelas regras do jogo, é o turno do sistema, para poder realizar a sua jogada, o sistema deve analisar o cenário atual do jogo e a partir da inferência realizada, tomar a sua decisão.

A seguir é explicada a função de cada classe do módulo de inteligência artificial do jogo:

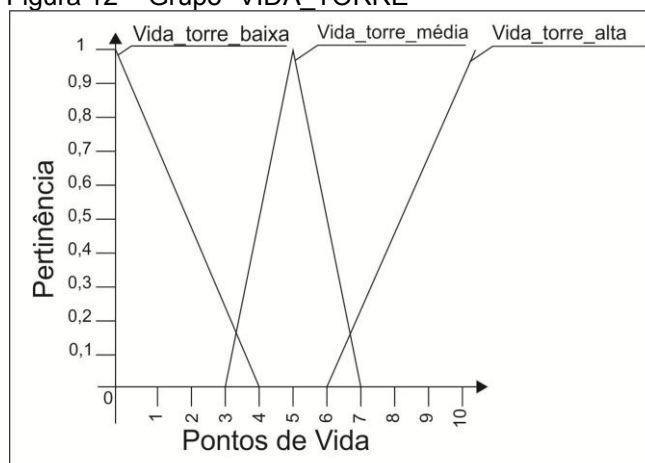
**Classe ExecutaAcaolA:** O início do processamento do módulo de IA se dá nesta classe. Ela é responsável por reunir as informações necessárias para que seja realizada a inferência. A partir dos dados fornecidos pelo módulo de Entidades é realizado o sensoriamento do cenário atual do jogo e a partir das informações adquiridas é então iniciado o processo do Sistema Fuzzy, sendo feita uma chamada a classe PrincipallA.

**Classe PrincipallA:** Nesta classe estão reunidas as funcionalidades que irão realizar todo o processo do Sistema Fuzzy. A partir dos dados fornecidos pela classe ExecutaAcaolA, são executados os métodos responsáveis pela realização da Fuzzyficação, Inferência e Defuzzyficação dos dados de entrada. No processo de Inferência são utilizadas as classes GruposIA e ExpressoesIA. 

**Classe GruposIA:** Esta classe é responsável por definir os conjuntos Fuzzy que serão utilizados no jogo, a partir deste grupo serão realizadas as inferências das regras definidas no banco de conhecimento do jogo. Os grupos são criados fornecendo as informações do intervalo de pertinência, neste trabalho foram utilizadas os atributos qualitativos “Baixo”, “Médio” e “Alto” para cada grupo, de maneira que, para cada conjunto, existirá o intervalo definido dos atributos qualitativos citados. A seguir, será ilustrado na Figura 12 o resultado da criação de um Grupo através da classe GruposIA:

**Grupo Vida da Torre** – Define o conjunto Fuzzy a partir da variável vida da torre a partir da quantidade de pontos de vida que a mesma possui no momento, comando `iniciaGrupo("VIDA_TORRE")`:

Figura 12 – Grupo “VIDA\_TORRE”



Fonte: Elaborado pelo autor

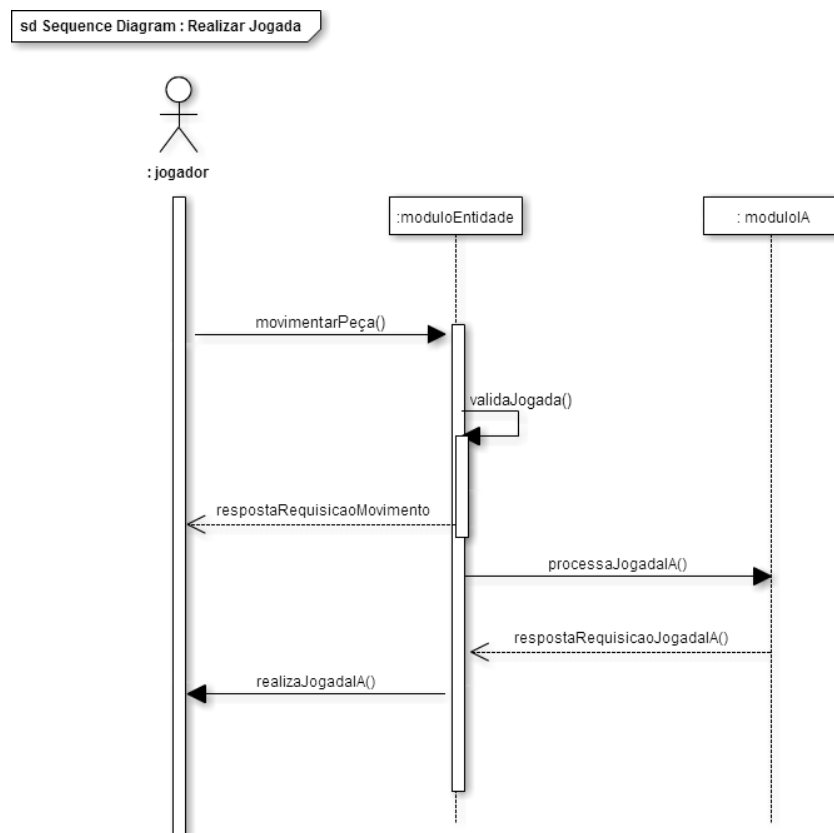
**Classe ExpressoesIA:** Nesta classe são definidas as regras que utilizarão os conjuntos definidos na classe GruposIA. As regras são definidas seguindo o modelo:

**SE condição ENTÃO ação**

Esta classe funciona como o banco de conhecimento do jogo, com isso, todas as possibilidades e cenários no qual o agente automatizado poderá estar inserido devem estar relacionadas nesta classe, de maneira que possibilite que o módulo de IA tenha condições de avaliar o cenário que encontrar uma boa jogada.

Finalizado o processamento das informações pelo módulo de Inteligência artificial, o controle do sistema é devolvido para o módulo de Entidade, para que seja dada a continuação da execução do jogo. O processo é ilustrado na Figura 13 a seguir:

Figura 13 – Processo “Realizar Jogada”

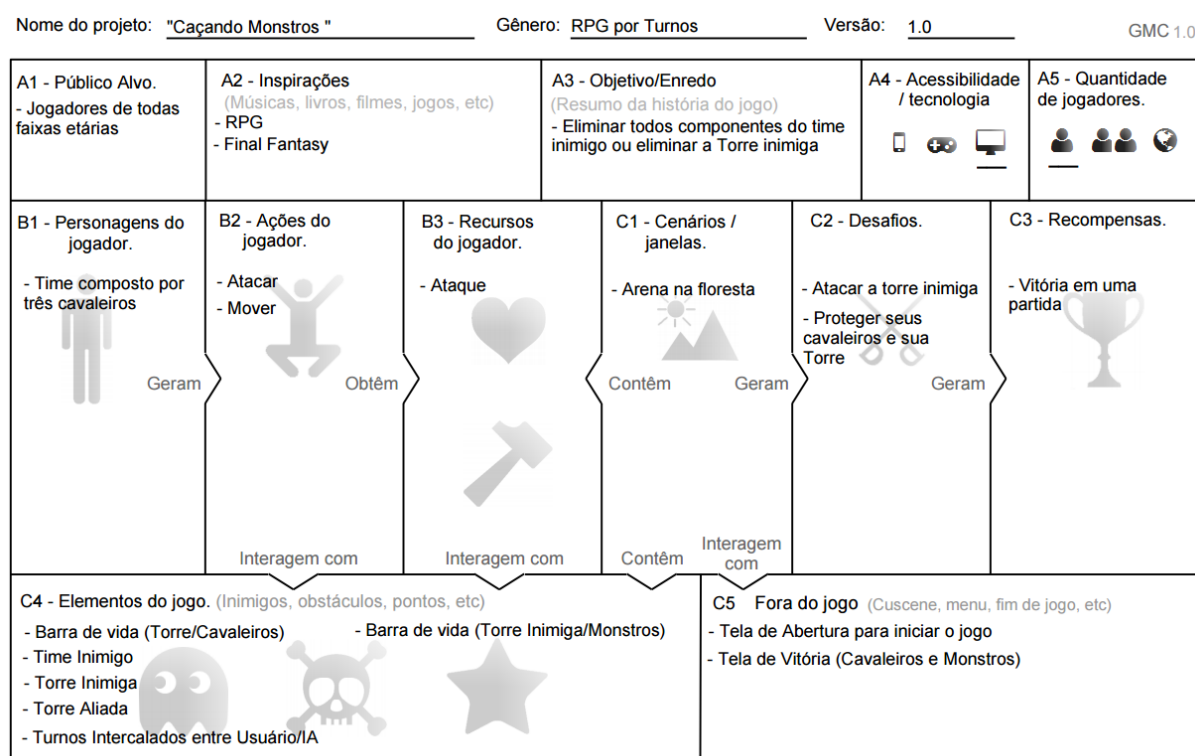


Fonte: Elaborado pelo autor

## 5.6 Protótipo do Jogo

Para este projeto, foi implementado o jogo “Caçando Monstros”<sup>11</sup> baseado no modelo proposto anteriormente. O objetivo do jogo desenvolvido foi validar as técnicas apresentadas neste trabalho, de modo a ser observada a tomada de decisão nos agentes automatizados em determinadas situações de jogo. Os principais aspectos do jogo estão listados no *Game Model Canvas* exibido na Figura 14:

Figura 14 – Game Model Canvas do jogo “Caçando Monstros”



O Game Model Canvas foi criado pela Drecon Entretenimentos, encontrado pelo site: [drecon.com.br/GMC](http://drecon.com.br/GMC), sob licença Creative Commons.

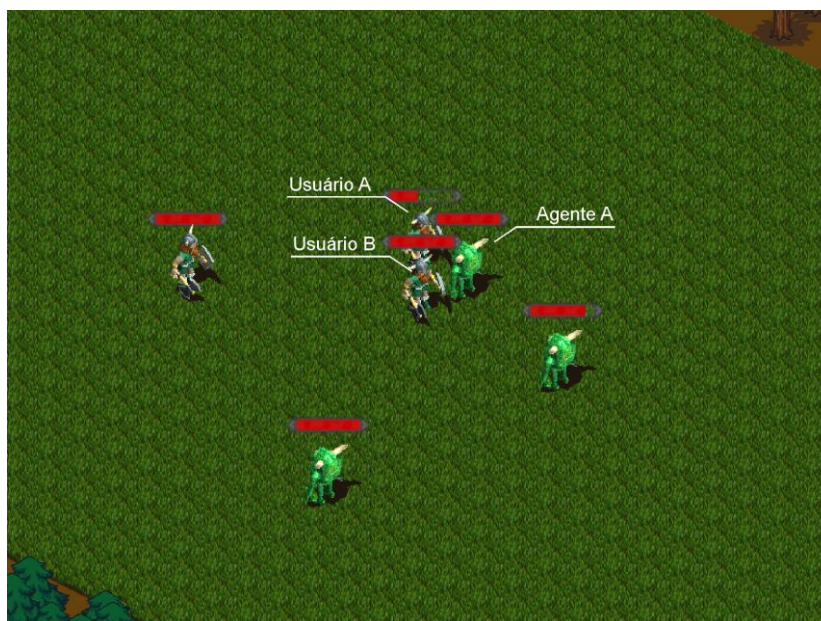
Fonte – Elaborado pelo Autor com base no modelo disponível em

[http://www.drecon.com.br/GMC/Game\\_Model\\_Canvas\\_1\\_0\\_0.pdf](http://www.drecon.com.br/GMC/Game_Model_Canvas_1_0_0.pdf)

Embora tenha sido definida uma quantidade relativamente baixa de regras, o sistema consegue manter ações pertinentes de acordo com o momento atual que se encontra, em relação a base de conhecimento definida. No exemplo a seguir, é demonstrado um determinado momento do jogo, onde o agente automatizado “Agente A”, possui ao seu redor dois agentes controlados pelo usuário: “Usuário A” e “Usuário B”, como é demonstrado na Figura 14 a seguir:

<sup>11</sup> Jogo disponível *on-line* em: <https://tcc2joaoant.github.io/>

Figura 14 – Exemplo do momento atual de jogo



Fonte: Elaborado pelo autor

Como é possível observar na imagem, o “Agente A” encontra-se isolado em relação aos agentes controlados pelo usuário. Para uma melhor tomada de decisão, foi implementado no sistema uma estrutura que permite o jogo decidir suas ações em dois momentos distintos:

- a) Momento A - Decisão relacionada às distâncias e quantidade de inimigos: Neste momento, o sistema decide, baseado na distância do agente selecionado para realizar a jogada em relação a torre inimiga e na quantidade de inimigos ao redor, se deve realizar um movimento em direção a torre ou se deve escolher atacar.
- b) Momento B - Decisão baseada na quantidade de vida: Caso no momento anterior o sistema tenha escolhido a opção de ataque, o sistema irá avaliar, baseado na vida dos agentes inimigos ao redor, se irá realmente atacar algum agente controlado pelo usuário que esteja dentro do seu intervalo de ataque, caso avalie que não deve atacar, o sistema irá definir que o usuário deverá realizar uma ação de movimento.

Para realizar estes dois momentos distintos de tomada de decisão, foram criados grupos de regras distintas para cada momento, a seguir são demonstrados exemplos das regras definidas:

a) Regras relacionada às distâncias:

SE PERSON\_AOREDOR = ALTO E DISTANCIA\_T1\_TORRE1 = ALTO  
ENTAO MOVE\_ATACA\_INI;

SE PERSON\_AOREDOR = MEDIO E DISTANCIA\_T1\_TORRE1 = MEDIO  
ENTAO MOVE\_ATACA\_INI;

a) Regras relacionada a quantidade de vida:

SE VIDA\_P\_T1 = ALTO E VIDA\_P\_T2 = ALTO  
ENTAO ATACA\_INI;

SE VIDA\_P\_T1 = BAIXO E VIDA\_P\_T2 = MEDIO  
ENTAO MOVE\_TORRE\_INI

Em cada momento de tomada de decisão apresentadas anteriormente, o sistema irá realizar todo o ciclo de tomada de decisão em um sistema Fuzzy, ou seja, no momento A, o sistema irá Fuzzyficar as entradas, neste caso, os valores de entrada são as distância (em casas) entre o agente e a torre inimiga, e a quantidade de inimigos ao redor do agente. A partir das variáveis Fuzzyficadas, o sistema realizará a fase de Inferência, onde serão avaliadas as regras definidas no bando de conhecimento, após esta fase, é realizada a fase de Defuzzyficação, onde a regra de maior pertinência é selecionada. Com a regra do Momento A definida, caso tenha sido selecionada a ação de atacar, o sistema irá realiza novamente o processo de tomada de decisão do sistema Fuzzy, para finalmente finalizar o processo de tomada de decisão e retornar a ação a ser realizada.

Portanto, o sistema interage com o sistema Fuzzy em casa ciclo de jogada, sendo que em cada ciclo de jogada o sistema Fuzzy é consultado uma ou duas vezes para que seja definida a ação a ser realizada.

## 8 CONCLUSÃO

O objetivo deste trabalho foi validar o funcionamento da Lógica Fuzzy aplicada em um jogo no estilo RPG em turnos. Foi definida uma estrutura que contemplasse todas as fases do processamento de uma tomada de decisão relacionada a realização de uma jogada pelo sistema de Inteligência Artificial implementado no jogo. O banco de regras do jogo, consultado em todas as jogadas realizadas pelo sistema, foi construído de maneira a oferecer ao Módulo de Inteligência Artificial subsídios suficientes para que fosse possível, através dos dados de entrada, a inferência sobre qual jogada seria apropriada para determinados cenários nos quais o jogo poderia se encontrar.

A capacidade da Lógica Fuzzy de se adaptar ao cenário atual do jogo se mostrou evidente neste projeto a partir do momento em que na maior parte das vezes o sistema selecionou a melhor alternativa entre as expressões definidas no banco de conhecimento do jogo, ou seja, levando em consideração os dados de entrada, foram escolhidas as ações otimizadas para o momento do jogo em questão.

Podemos exemplificar tal afirmação observando o conjunto de expressões definidas para a definição de qual ação o agente iria tomar: Atacar ou Movimentar, a partir do dados de entrada, o sistema analisa sua distância para a torre inimiga e a quantidade de inimigos ao redor do agente, e define se deve atacar ou movimentar. Além disso, o sistema ao escolher o ataque também usa o banco de conhecimento para definir se deve atacar de fato o inimigo, baseado na quantidade de vida que tanto o agente possui quanto o personagem controlado pelo usuário.

Também foram realizados testes comparativos entre o motor Fuzzy e um motor totalmente aleatório. Os testes realizados consistiram em preparar uma versão do jogo onde foi implementado um modelo onde as regras existentes no banco de conhecimento eram selecionadas de maneira aleatória, nesta versão do jogo, o sistema avalia o momento atual do jogo e indica uma jogada aleatória a ser realizada pelo usuário do jogo, permitindo assim que sejam aplicadas jogadas baseadas no Motor Aleatório ao Motor Fuzzy.



Ao testar o motor aleatório contra o motor Fuzzy, foi verificado que o sistema Fuzzy possui uma tendência de vencer a partida, sendo que em 10 partidas realizadas, o motor fuzzy teve 70% de vitória.

A principal diferença entre o motor aleatório e o motor Fuzzy se dá no fato de, no motor Fuzzy, o sistema considerar o cenário atual em que o jogo se encontra através do processo de Fuzzyficação das variáveis de entrada, e ao aplicar tais valores das variáveis, era possível calcular qual ação era mais apropriada de acordo com a base de conhecimento.

Também foi verificado um aumento da dificuldade do jogo ao adicionar mais regras ao banco de conhecimento, proporcionando ao sistema encontrar melhores ações de acordo com o estado atual do jogo.

Por tanto, é possível concluir que a aplicação da Lógica Fuzzy para o controle de Agente Automatizados em jogos, pode trazer ganhos em termos de dificuldade de jogo e facilmente pode ter seu banco de conhecimento aumentado, a partir do momento que as regras são simples, com o seguinte formato:

**SE condição ENTÃO ação**

Desta forma, o motor desenvolvido pode facilmente ser adaptado para um diferente estilo de jogo, sendo necessário apenas atualizar as Variáveis e Expressões Fuzzy.

## 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

ZADEH, L.A. **FUZZY SETS** Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California, USA, 1964.