

**RAFAEL BALLOTTIN MARTINS**

**COMPARATIVO DE UMA ABORDAGEM BASEADA NO  
COMPORTAMENTO HUMANO ATRAVÉS DO USO DE RNA FRENTE A  
ABORDAGENS DETERMINÍSTICAS EM UM AMBIENTE DE  
ESTRATÉGIA**

Itajaí (SC), Fevereiro de 2013



**UNIVALI**

**UNIVERSIDADE DO VALE DO ITAJAÍ**  
**CURSO DE MESTRADO ACADÊMICO EM**  
**COMPUTAÇÃO APLICADA**

**COMPARATIVO DE UMA ABORDAGEM BASEADA NO**  
**COMPORTAMENTO HUMANO ATRAVÉS DO USO DE RNA FRENTE A**  
**ABORDAGENS DETERMINÍSTICAS EM UM AMBIENTE DE**  
**ESTRATÉGIA**

por

Rafael Ballottin Martins

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre em Computação  
Aplicada.

Orientador: Rudimar Luís Scaranto Dazzi, Dr.

Itajaí (SC), Fevereiro de 2013

## **FOLHA DE APROVAÇÃO**

Esta página é reservada para inclusão da folha de assinaturas, a ser disponibilizada pela Secretaria do Curso para coleta da assinatura no ato da defesa.

*Dedico este trabalho a meus pais, que tanto amo, Eros Aristeu Martins e Heloisa Ballottin Martins, que me deram força, incentivo e amparo durante toda minha vida, com quem divido minhas conquistas e minhas derrotas.*

## **AGRADECIMENTOS**

Inicialmente agradeço a minha família, pelo apoio, compreensão e incentivo durante o desenvolvimento desta dissertação.

Agradeço da mesma forma a meu orientador Rudimar Luís Scaranto Dazzi pelo acompanhamento não só durante o período de desenvolvimento desta pesquisa, mas por todos os anos em que vem guiando minha formação e também pela confiança em meu trabalho durante todo este tempo.

Também agradeço aos professores Raimundo Celeste Ghizoni Teive, Eros Comunello, Anita Maria Fernandes da Rocha e Fernando Azevedo pelas importantes contribuições dadas a versão final deste trabalho.

Aos Professores Nilmar de Souza e Leo Lynce Valle de Lacerda pela importantíssima contribuição na realização das análises estatísticas.

Agradeço a todos os colegas da Gerência de Tecnologia da Informação da Univali que auxiliaram na pesquisa realizada.

Aos meus colegas de mestrado Andrey Kuehlkamp, Lennon Bisolo, Marcelo Shmidt, Cristiano Roberto Franco e Paulo Viniccus, pela amizade e companheirismo.

Agradeço a todos os demais que tiveram alguma participação direta ou indireta e que eu tenha esquecido de agradecer.

# **COMPARATIVO DE UMA ABORDAGEM BASEADA NO COMPORTAMENTO HUMANO ATRAVÉS DO USO DE RNA FRENTE A ABORDAGENS DETERMINÍSTICAS EM UM AMBIENTE DE ESTRATÉGIA**

Rafael Ballottin Martins

Fevereiro / 2013

Orientador: Rudimar Luís Scaranto Dazzi, Dr.

Área de Concentração: Computação Aplicada

Linha de Pesquisa: Inteligência Aplicada

Palavras-chave: RNA, Jogos, Comportamento Humano

Número de páginas: 164

## **RESUMO**

Muitas aplicações de Inteligência Artificial na área de jogos tem como foco a melhora na qualidade do comportamento de personagens, apesar de muitas soluções ainda se focarem em abordagens paliativas visando apenas à resolução de tarefas com base em regras determinísticas limitadas. A criação de agentes para operar em ambientes complexos de uma maneira que se permita observar comportamentos inteligentes é um fator fundamental para os jogadores, principalmente quando o agente inteligente é comparado a agentes que apenas realizam ações otimizadas a partir de regras determinísticas. Apesar da melhora na qualidade com soluções de IA, o comportamento inteligente é difícil de especificar e principalmente de se avaliar. Sendo que uma abordagem baseada em exemplos gerados por jogadores humanos é uma solução promissora para realizar o mapeamento de suas ações. Este trabalho buscou verificar através de experimentos se o investimento na construção de uma IA baseada em ações do jogador humano tem resultados significativos frente a abordagens determinísticas normalmente utilizadas em jogos. Para tanto a técnica de Redes Neurais Artificiais foi utilizada por seus benefícios já reconhecidos para aplicações em jogos e principalmente em trabalhos na área de imitação do comportamento humano. O contexto da pesquisa foi direcionado a um ambiente de estratégia, proporcionando a verificação do comportamento dos agentes para esta categoria. A partir dos confrontos diretos do agente utilizando RNA frente ao agente com aplicação determinística e o próprio jogador, foi utilizada a interface de interação do Robocode para que jogadores avaliassem as diferentes abordagens, permitindo verificar a percepção de inteligência obtida em relação ao agente mapeado com o comportamento humano. Os resultados desta avaliação de percepção demonstraram que é possível mapear ações executadas pelo jogador humano em agentes virtuais e que estas ações são percebidas

pelos jogadores como ações mais próximas ao esperado por um jogador humano, melhorando a qualidade destes agentes e conseqüentemente o nível de entusiasmo dos jogadores.

# **COMPARISON OF AN APPROACH BASED ON HUMAN BEHAVIOR THROUGH THE USE OF ANN WITH DETERMINISTIC APPROACHES IN A STRATEGY ENVIRONMENT**

Rafael Ballottin Martins

February / 2013

Supervisor: Rudimar Luís Scaranto Dazzi, Dr.

Area of Concentration: Applied Computer Science

Line of Research: Applied Intelligence

Keywords: ANN, Games, Human Behavior

Number of pages: 164

## **ABSTRACT**

Many artificial intelligence applications in the area of games are focused on improving the behavioral quality of the characters, although many solutions still focus only on palliative solutions to solving tasks, based on limited deterministic sets of rules. Creating agents to operate in complex environments in a way that enables intelligent behaviors to be observed is a key factor for game players, especially when the intelligent agent is compared to agents that only perform actions with optimized deterministic rules. Despite the improvements in the quality of AI solutions, intelligent behavior is difficult to specify, and even harder to evaluate. An approach based on examples generated by human players offers a promising approach to mapping the player's actions. This paper seeks to determine, through experiments, whether investments in building an AI based on human behavior has significant results, when compared with the deterministic approaches typically used in games. The technique of Artificial Neural Networks was used for this purpose, due to its recognized benefits when applied to games, particularly in works in the area of imitation of human behavior. The context of the research was directed towards a strategy environment, enabling the behavior of agents to be determined, for this category. Based on direct comparisons of the agent using ANN with the agent with deterministic application, and the player himself, the Robocode interaction interface was used to enable the players to evaluate these different approaches, enabling comparison of the perception of intelligence obtained with that of the agent mapped with human behavior. The results of this evaluation of perception demonstrate that it is possible to map the actions carried out by the human player to virtual agents, and that these actions are perceived by the players as being closer to what would be expected of a human player, improving virtual agents quality and consequently enhancing players enthusiasm.

## LISTA DE ILUSTRAÇÕES

Figura 1. Modelo não-linear de um neurônio. ....	29
Figura 2. Camadas de um Rede Neural Artificial. ....	31
Figura 4. Diagrama em blocos do modelo de aprendizagem supervisionada. ....	33
Figura 5. Interface de configuração das entradas. ....	44
Figura 6. Interface demonstrativa da Rede Neural. ....	44
Figura 7. Interface inicial do assistente. ....	45
Figura 8. Interface para seleção da regra de aprendizado da rede MLP. ....	45
Figura 9. Exemplo de um diagrama de estados finitos. ....	47
Figura 10. Modelo geral de agente. ....	49
Figura 11. Ambiente do Robocode e personagens. ....	60
Figura 12. IDE de implementação do Robocode. ....	61
Figura 13. Quantidade de trabalhos obtidos por base de dados. ....	73
Figura 14. Proporção de artigos entre o total de selecionados por base. ....	74
Figura 15. Arquitetura do robô. ....	89
Figura 16. Modelo do método COCOM. ....	90
Figura 17. Exemplo de sensores de movimentação. ....	93
Figura 18. Exemplo de sensores para controle da torre. ....	96
Figura 19. Topologia da rede responsável pela movimentação. ....	110
Figura 21. Topologia da rede responsável pelo controle da torre. ....	111
Figura 23. Tela de exemplo do questionário de avaliação. ....	119

## LISTA DE TABELAS

Tabela 1. Amostra de trabalhos envolvendo análise comportamental.....	39
Tabela 2. Comparativo entre ferramentas de simulação.....	59
Tabela 3. Fontes de Dados.....	65
Tabela 4. Extração dos dados para cada questão.....	69
Tabela 5. Seleção por base.....	71
Tabela 6. Referências dos estudos selecionados por base.....	71
Tabela 7. Estudos selecionados por base.....	72
Tabela 8. Estudos selecionados por ano.....	74
Tabela 9. Ações Mapeadas.....	77
Tabela 10. Ações Mapeadas por categoria de jogos.....	78
Tabela 11. Método de Mapeamento das ações do jogador.....	78
Tabela 12. Técnicas de IA utilizadas.....	79
Tabela 13. Jogos e respectivas categorias utilizadas nos estudos.....	79
Tabela 14. Resultados Alcançados.....	80
Tabela 15. Métodos utilizados.....	82
Tabela 16. Relação de potência, dano causado, dano recebido e velocidade do projétil.....	88
Tabela 18. Redes com melhores resultados para controle da torre.....	109
Tabela 19. Resultados das tarefas treinadas para 50 batalhas.....	113
Tabela 20. Respostas relativas a avaliação do comportamento.....	123
Tabela 21. Resultados do teste de Spearman.....	126
Tabela 22. Distribuição dos jogadores por nível.....	127
Tabela 23. Grupos definidos.....	127
Tabela 24. Resultado das comparações entre as diferentes abordagens para jogadores menos experientes.....	128
Tabela 25. Resultado das comparações entre as diferentes abordagens para jogadores mais experientes.....	129
Tabela 26. Média das notas para atribuição artificial de comportamento.....	130
Tabela 27. Totais de comentários similares para cada abordagem entre jogadores menos experientes.....	131
Tabela 28. Totais de comentários similares para cada abordagem entre jogadores mais experientes.....	133
Tabela 29. Comentários atribuídos a abordagem de RNA frente ao determinístico DrussGT.....	136
Tabela 30. Comentários atribuídos ao determinístico DrussGT frente a abordagem de RNA.....	137
Tabela 31. Comentários atribuídos ao jogador humano frente a abordagem de RNA.....	138
Tabela 32. Comentários atribuídos a abordagem de RNA frente ao jogador humano.....	139
Tabela 33. Comentários atribuídos a abordagem de RNA frente ao determinístico DustBunny ....	139
Tabela 34. Comentários atribuídos ao determinístico DustBunny frente a abordagem de RNA ....	140

## LISTA DE ABREVIATURAS E SIGLAS

ACM	Association for Computing Machinery
BP	BackPropagation
FPS	First-Person Shooter
IA	Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
L2S	Laboratório de Informática na Educação
LIA	Laboratório de Inteligência Aplicada
LSED	Laboratório de Sistemas Embarcados e Distribuídos
MCA	Mestrado em Computação Aplicada
MEF	Máquinas de Estados Finitos
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NEAT	NeuroEvolution of Augmenting Topologies
RBF	Radial Basis Function Network
RBS	Rule Based Systems
RNA	Redes Neurais Artificiais
TI	Tecnologia da Informação
UNIVALI	Universidade do Vale do Itajaí

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
<b>1.1 PROBLEMA DE PESQUISA.....</b>	<b>17</b>
1.1.1 Solução Proposta .....	19
1.1.2 Delimitação de Escopo .....	21
1.1.3 Justificativa.....	22
<b>1.2 OBJETIVOS .....</b>	<b>23</b>
1.2.1 Objetivo Geral .....	23
1.2.2 Objetivos Específicos .....	23
<b>1.3 METODOLOGIA.....</b>	<b>23</b>
1.3.1 Metodologia da Pesquisa .....	23
1.3.2 Procedimentos Metodológicos.....	24
<b>1.4 ESTRUTURA DA DISSERTAÇÃO.....</b>	<b>27</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>28</b>
<b>2.1 REDES NEURAIS .....</b>	<b>28</b>
2.1.1 Estrutura da rede .....	29
2.1.2 Aprendizado supervisionado.....	32
2.1.3 Tipos de Redes Neurais.....	33
2.1.4 Rede Perceptron .....	34
2.1.5 Rede Multi layer Perceptron (MLP).....	36
2.1.6 Treinamento a partir de padrões comportamentais.....	38
2.1.7 Conjuntos de treinamentos desbalanceados.....	40
2.1.8 Método de treinamento.....	41
<b>2.2 SISTEMAS BASEADOS EM REGRAS .....</b>	<b>45</b>
<b>2.3 AGENTES INTELIGENTES.....</b>	<b>48</b>
2.3.1 Autonomia e aprendizagem.....	50
2.3.2 Mapeamento de perfil em Agentes .....	51
<b>2.4 FERRAMENTA PARA SIMULAÇÃO .....</b>	<b>53</b>
2.4.1 Requisitos da ferramenta .....	55
2.4.2 Ferramentas Pesquisadas .....	55
2.4.3 Comparativo entre as ferramentas.....	58
2.4.4 ROBOCODE .....	59
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>62</b>
<b>3.1 OBJETIVO PRINCIPAL DA REVISÃO SISTEMÁTICA .....</b>	<b>63</b>
<b>3.2 PROTOCOLO DE REVISÃO SISTEMÁTICA .....</b>	<b>63</b>
3.2.1 Objetivo .....	63
3.2.2 Questões de pesquisa.....	63
3.2.3 Fontes de dados .....	65
3.2.4 Critérios de inclusão e exclusão .....	65

3.2.5	Strings de busca.....	66
3.2.6	Seleção dos estudos.....	68
3.2.7	Extração dos dados .....	69
3.2.8	Síntese dos dados .....	69
3.3	ANÁLISE COMPARATIVA.....	70
3.3.1	Estudos Seleccionados .....	70
3.3.2	Foco dos estudos .....	74
3.3.3	Abrangência.....	77
3.3.4	Metodologia Utilizada.....	78
3.3.5	Classificação dos resultados obtidos.....	80
3.4	CONSIDERAÇÕES .....	82
4	DESENVOLVIMENTO.....	85
4.1	DEFINIÇÕES DAS VARIÁVEIS PARA MAPEAMENTO.....	85
4.1.1	Robocode como ferramenta de simulação .....	85
4.1.2	Estrutura do Agente.....	87
4.1.3	Método de extração de dados.....	89
4.1.4	Definição das variáveis de Movimentação.....	91
4.1.5	Definição das variáveis de Controle da Torre.....	95
4.2	MODELO DE TREINAMENTO SUBDIVIDIDO EM TAREFAS .....	98
4.2.1	Movimentação .....	100
4.2.2	Controle da Torre .....	101
4.3	CICLO DE OBTENÇÃO DOS DADOS DE TREINAMENTO .....	102
4.4	TRATAMENTO DOS DADOS PARA TREINAMENTO.....	104
4.5	PROCESSO DE TREINAMENTO .....	106
4.5.1	Estrutura da Rede e Algoritmos de treinamento para Controle da Torre 108	
4.5.2	Resultados dos treinamentos.....	109
4.6	UTILIZAÇÃO DA REDE TREINADA .....	111
4.7	ANÁLISE DO AGENTE TREINADO.....	112
4.8	AVALIAÇÃO DO COMPORTAMENTO MAPEADO.....	114
4.8.1	Instrumento de avaliação .....	115
4.8.2	Design do experimento.....	117
4.9	CONSIDERAÇÕES .....	120
5	RESULTADOS .....	121
5.1	ANÁLISE DAS RESPOSTAS .....	122
5.2	COMPARAÇÃO ENTRE AS ABORDAGENS .....	127
5.3	RESPOSTAS COMPLEMENTARES.....	129
5.4	CONSIDERAÇÕES .....	141
6	CONCLUSÕES .....	144
6.1	SUBJETIVIDADE, VIESES E OUTRAS AMEAÇAS.....	146

<b>6.2 CONTRIBUIÇÕES DA DISSERTAÇÃO .....</b>	<b>148</b>
<b>6.3 TRABALHOS FUTUROS .....</b>	<b>149</b>
<b>REFERÊNCIAS .....</b>	<b>150</b>
<b>APÊNDICE A – INSTRUMENTO DE AVALIAÇÃO .....</b>	<b>156</b>

# 1 INTRODUÇÃO

A cada dia pesquisadores e desenvolvedores da área de jogos trabalham para melhorar a qualidade do comportamento de personagens controlados pelo computador. A busca desta melhoria não está ligada somente ao entretenimento, mas sim a qualidade das ações dos personagens de jogos quando comparados aos próprios jogadores (MOZGOVOY; UMAROV, 2010).

É comum na área de jogos pesquisadores apontarem que os usuários estão sempre buscando serem desafiados nos jogos, mas não apenas pela dificuldade em derrotar seus oponentes, mas também pelo desafio intelectual que atrai os jogadores em partidas entre si, ou seja, a qualidade da inteligência por traz do personagem (SCHWAB, 2004).

Esta é uma das principais razões que levam pesquisadores a tentarem elevar a qualidade dos personagens controlados por computador a um nível de controle mais próximo possível das ações que um jogador humano realizaria.

Apesar da área de jogos investir neste tipo de pesquisa, outras áreas como a robótica são pioneiras e também buscam resultados através de pesquisas deste tipo (THURAU; BAUCKHAGE, 2004). A grande empolgação por parte dos pesquisadores se deve ao fato de criar uma inteligência artificial a um nível alto de complexidade permitindo que as ações sejam as mais próximas possíveis as do ser humano. Neste caso, a área de jogos por seus cenários elaborados e milhares de componentes que fazem parte do mundo do jogo, proporciona um ambiente adequado para testes (LAIRD; Van LENT, 2000).

Um dos métodos mais utilizados para a atribuição deste nível de IA em personagens é o mapeamento do comportamento humano por imitação, nele são utilizadas técnicas para se obter os dados do jogador humano e transmiti-los aos personagens através de técnicas de IA (GORMAN *et al*, 2006b).

O objetivo da imitação de comportamento em jogos é utilizar as sessões de partidas realizadas por jogadores humanos e realizar uma “engenharia reversa” com as ações e decisões do jogador para atingir seus objetivos (GORMAN *et al*, 2006a).

Apesar do alto nível de inteligência que pesquisas na área de imitação do comportamento buscam mapear, nem todos os estudos abordam todas as ações dos personagens. Muitas tratam de apenas um aspecto no cenário do jogo como, por exemplo, a movimentação do personagem.

Isto ocorre, pois experimentos com a aplicação de IA em agentes de jogos sugerem que programar o comportamento dos jogadores não é uma tarefa fácil. Por exemplo, as forças armadas americanas gastam milhões de dólares para criar agentes autônomos com IA em seu simulador militar (BRYANT; MIIKKULAINEN, 2007).

Devido a esta dificuldade, sistemas baseados em IA na área de jogos utilizam tipicamente esquemas de reconhecimento de padrões e aprendizado de máquina, priorizando abordagens de Maquinas de Estados Finitos (MEF) e algoritmos de *Graph-Searching* (GORMAN *et al*, 2006a).

Em geral, ações e comportamentos de agentes nos jogos são baseados em scripts ou se baseiam em MEF's (CUNHA; CHAIMOWICZ, 2002). É ao menos questionável, que abordagens determinísticas como a de MEF levam a agentes que simulem o comportamento humano de maneira convincente (THURAU; BAUCKHAGE, 2004).

Apesar de ser comum encontrar abordagens de aprendizado de máquina utilizando as ações dos jogadores para treinamento, estas abordagens, assim como a maioria dos esforços na área de IA, são tradicionalmente focadas em obter soluções que são voltadas a execução da tarefa de forma otimizada. Sendo que, quando o objetivo é gerar um comportamento visivelmente inteligente em um ambiente mais complexo, esta função se torna extremamente difícil de especificar (BRYANT; MIIKKULAINEN, 2007).

As afirmações citadas sobre abordagens determinísticas para jogos de maneira geral, podem ser atribuídas a jogos de estratégia também, nos quais este trabalho se baseia. Normalmente, MEF's são utilizadas para controle de agentes nos jogos de estratégia. Estes agentes proporcionam desafios baseados em dificuldade, por executarem ações da maneira mais perfeita possível ao invés de proporcionar uma jogabilidade superior, fazendo com que possam ser facilmente diferenciados de jogadores humanos (THURAU; BAUCKHAGE, 2004).

Em jogos e simuladores, personagem com abordagens de IA que querem demonstrar um maior nível de inteligência devem parecer de fato se comportar de maneira inteligente aos olhos de

observadores, mesmo que signifique a perda de uma performance mais otimizada (BRYANT; MIIKKULAINEN, 2007).

Em geral jogadores sabem que o atual estado da IA nos jogos está muito longe da inteligência humana. Quando se planeja um jogo contra um ou vários personagens controlados por computador (NPC - *non-player character*), fica logo evidente que não se está jogando contra outro jogador humano e devido a padrões fáceis de identificar o jogo se torna entediante. Um método para evitar este comportamento óbvio de NPC's é através da utilização de técnicas para enganar o jogador, sendo que isto também se torna um problema, pois o uso de muitas trapaças para derrotar o jogador, também torna o jogo entediante (MUÑOZ; GUTIERREZ; SANCHIS, 2009).

Devido a estes fatores, na área de jogos a similaridade do agente com o jogador humano se torna um componente chave para uma implementação bem sucedida de IA. Esta similaridade é reconhecida como um fator importante tanto para pesquisadores quanto para desenvolvedores de jogos (GORMAN *et al*, 2006a).

Parece haver um consenso entre os autores de que o mapeamento do comportamento humano através da imitação contribui muito para o aumento do nível de inteligência e proporciona a criação de personagens muito superiores a abordagens determinísticas normalmente utilizadas em personagens da área de jogos.

Entretanto, os autores apontam que este é um mapeamento complexo, além da dificuldade em medir de fato o quão boa é uma inteligência em um personagem e se essa melhora se reflete na percepção do jogador de que a máquina é mais inteligente. Sendo assim, esta pesquisa busca contribuir para a área de inteligência artificial em jogos, investigando a qualidade da implementação através do método de mapeamento do comportamento humano por imitação com a utilização da técnica de Redes Neurais Artificiais (RNA).

A utilização de RNA se deve ao fato de que a técnica é usada em processos que tentam simular, de maneira computacional, o funcionamento do cérebro humano de forma simplificada. O principio básico é realizar o processamento de informações tendo como base a organização de neurônios do cérebro. As RNA's são capazes de reconhecer padrões, extrair regularidades e detectar relações subjacentes em um conjunto de dados aparentemente desconexos (LUDWING JR.; COSTA, 2007).

O ambiente para trabalho utilizado nesta pesquisa foi baseado na ferramenta Robocode, que é um simulador de batalha com tanques de guerra baseado em Java e permite o controle dos personagens do jogo através de IA. Os tanques foram programados para se mover, atacar e defender formulando estratégias de vitória no campo de batalha. Através do Robocode foi possível implementar a inteligência nos agentes para que os mesmos possam ser comparados com outras abordagens (detalhes podem ser vistos na Seção 4 ). Na ferramenta foi implementado o agente com a técnica de RNA baseado nas ações de um jogador humano e posteriormente foi observado se as ações do personagem foram refletidas em seus aspectos mais básicos no agente implementado. A etapa de observação foi importante para verificar se o agente de fato obteve os dados do jogador.

Com o treinamento efetuado, foi utilizada uma variação do teste realizado por Gorman *et al* (2006b) e Bauckage *et al* (2007) para avaliação de qualidade do personagem com a abordagem de Inteligência Artificial. Este teste foi realizado através de seções gravadas com confrontos entre o jogador humano, o agente implementado com a abordagem de RNA e agentes determinísticos.

## 1.1 PROBLEMA DE PESQUISA

Algumas pesquisas já foram realizadas abordando o mapeamento do comportamento humano na implementação de agentes inteligentes em jogos.

Bryant e Miikkulainen (2007) realizam o mapeamento do comportamento dos jogadores em um jogo de estratégia focando os testes na taxa de erro na execução das tarefas dos agentes, cruzando estes dados com a pontuação obtida na realização destas tarefas.

No estudo de Schrum, Karpov e Miikkulainen (2011) foram realizados testes para mapear as ações do jogador e aperfeiçoar as características humanas do personagem em um jogo do tipo FPS (First-person shooter), em que foram utilizados gráficos de navegação para observar se os agentes exploram os cenários da mesma maneira que o jogador humano.

Ki, Lyu e Oh (2006) realizaram o mapeamento do comportamento do jogador para movimentação e disparo das armas em um jogo de estratégia similar ao jogo Starcraft, no qual a partir de várias simulações, foram realizadas observações sobre a qualidade no confronto entre dois grupos de agentes, um com abordagem determinística e outro com o mapeamento incorporado.

Em outro trabalho Gorman e Humphys (2006) realizam o mapeamento de aspectos de ação do agente referente à suas estratégias de movimentação para execução de objetivos pré-determinados, avaliando as áreas do cenário graficamente afetadas para conclusão dos mesmos objetivos do jogador, verificando que o padrão de movimentação era similar ao do agente. Entretanto os próprios autores reforçam que os testes de credibilidade baseados nas observações dos jogadores seriam importantes para uma avaliação completa do nível de comportamento mapeado.

Entre as pesquisas citadas, observa-se a utilização de métodos variados para avaliação da qualidade da implementação. Estas variações de métodos são compreensíveis devido a complexidade de se avaliar o resultado do mapeamento do comportamento humano. Hy *et al* (2004) citam que é difícil avaliar a qualidade do mapeamento do comportamento humano em agentes inteligentes, sendo que em métodos de testes similares aos testes de Turing, obtendo o feedback dos jogadores, o resultado para uma avaliação deste tipo pode ser obtido.

Outras pesquisas como Mozgovoy e Umarov (2010) descrevem a realização de um teste híbrido, no qual parte do experimento utilizou o *feedback* de jogadores para avaliação, sendo o teste final focado em uma análise baseada em um método de comparação automática de perfil.

Já Gorman *et al* (2006b) realizaram um teste a partir da avaliação de jogadores em sua pesquisa e apontaram que o comportamento humano foi identificado nos agentes, mas descrevem que somente as estratégias de navegação e movimentação foram mapeadas, o combate não foi considerado no estudo e o foco principal foi apenas testar a credibilidade do agente mapeado para as ações definidas.

Os autores ainda descrevem que em pesquisas correlatas, observa-se que a maior parte destes estudos na área, tem sido limitados, informais e muitas vezes inconclusivos e de fato estas observações ainda se provam verdadeiras em muitos casos.

Estes estudos ainda tem em comum o fato de buscarem a utilização do método de mapeamento do comportamento humano através de técnicas de IA, por avaliarem que na percepção dos usuários esta abordagem é superior a técnicas determinísticas comumente utilizadas.

Analisando os resultados das pesquisas citadas anteriormente observa-se, em geral, que existem indícios de que é possível realizar o mapeamento do comportamento humano em agentes de jogos. Entretanto, algumas destas pesquisas ainda não inferem atenção suficiente na avaliação de

qualidade destas implementações ou necessitam de uma avaliação mais abrangente para garantir estes resultados perante a observação dos usuários.

Sendo assim, o objetivo desta pesquisa é demonstrar que a aplicação do método de imitação do comportamento através da técnica de RNA em um ambiente de estratégia reflete os aspectos comportamentais do jogador.

Esta avaliação do comportamento deve ser apurada principalmente na visão dos jogadores para se obter dados concretos da qualidade desta abordagem. Para uma avaliação concreta deste nível de Inteligência, deve ser considerado que os agentes possam executar todas as ações que o jogador poderia executar durante uma partida. A comparação com agentes determinísticos otimizados para vencer batalhas, é um aspecto que pode interferir na leitura do jogador, principalmente por ser descrita como uma abordagem muito simplificada para atingir os níveis comportamentais de métodos não determinísticos. Por isso, partindo destas considerações, este trabalho também procura demonstrar que a idealização da RNA para se obter um comportamento mais próximo do comportamento esperado por um ser humano é percebida pelos jogadores frente a abordagens determinísticas.

### **1.1.1 Solução Proposta**

Para responder aos problemas abordados anteriormente foi realizado o treinamento de uma rede neural mapeando os padrões de comportamento humano nas tarefas executadas no ambiente Robocode pelo agente (robô) para a movimentação no cenário, ataque, defesa e estratégia de jogo, baseado nos níveis de vida, potência das armas e obstáculos.

Para realização das comparações foi definida a utilização do DrussGt e do DustBunny que são robôs participantes de competições realizadas através do Robocode. As informações e estatísticas dos robôs são disponibilizadas pela RoboWiki (ROBOWIKI, 2011).

O DrussGT é um robô apontado pela RoboWiki (2011) como um dos mais vitoriosos em campeonatos especializados. Já o DustBunny, apesar de também participar de competições, tem uma classificação menor de efetividade em combate frente ao DrussGT. Ambos têm o código fonte aberto, o que permite que possa ser identificada a abordagem determinística utilizada nos robôs.

Utilizando como adversários o DrussGT e DustBunny, foram realizados confrontos entre o agente com o mapeamento do comportamento humano, as técnicas determinísticas utilizadas no DrussGT e DustBunny, além de um terceiro personagem controlado por um jogador humano para comparações de qualidade durante as avaliações.

A partir destes confrontos foi iniciada a segunda etapa do experimento na qual foi verificado se a percepção do comportamento mais humano do agente implementado através da RNA é de fato percebida pelos jogadores.

Para tanto, foram realizados experimentos com 3 grupos de avaliadores, todos integrantes da UNIVALI. O primeiro grupo contém alunos do curso de Jogos Digitais, o segundo grupo contém estudantes dos laboratórios de pesquisa do curso de Ciência da Computação e o terceiro grupo composto por profissionais da área de Tecnologia da Informação. Os grupos foram escolhidos por estarem ligados à área de tecnologia ou diretamente ligados a área de jogos.

Para o experimento foram gravadas sessões das partidas e disponibilizadas para os avaliadores, obtendo sua percepção a cada confronto, baseado em uma variação do questionário de testes utilizado por Bauckage *et al* (2007) e Gorman *et al* (2006b), descrito em detalhes na Subseção 4.8.1 .

Na primeira etapa do questionário foi solicitado aos avaliadores seu nível de experiência em jogos. Na sequência foram apresentados os confrontos realizados entre as diferentes abordagens e solicitada a avaliação dos robôs indicados. Para cada confronto os avaliadores respondiam se consideravam o comportamento do agente humano ou artificial.

Nos casos em que o avaliado indicava que o robô tem uma abordagem artificial, foi apresentada uma escala para avaliar taxa de como este comportamento é percebido, variando de 0 a 10, sendo 10 um pouco mais próximo do humano. Esta taxa representa um dado mais subjetivo e não faz parte da computação direta dos resultados, sendo utilizado apenas para conhecer melhor a opinião dos avaliadores. Na última questão foi disponibilizado aos avaliadores um campo opcional para comentários sobre o comportamento percebido.

A partir da aplicação dos questionários os dados obtidos foram utilizados para computar o grau de percepção dos avaliadores.

Apesar de baseados nos testes utilizados por Bauckage *et al* (2007) e Gorman *et al* (2006b), existe uma diferença significativa entre os experimentos realizados, que serve para situar algumas diferenças sobre o questionário aplicado. Os autores utilizam a apresentação de um personagem para execução de algumas tarefas no cenário do jogo e realizam a avaliação. Entretanto neste trabalho, são demonstradas as batalhas ente dois robôs para a aplicação da avaliação. Devido a esta diferença, sempre são avaliados os resultados de cada abordagem aplicada aos robôs de uma forma comparativa, em uma batalha real pela vitória. Para minimizar o impacto no *framework* utilizado, apenas um dos robôs é avaliado por vez.

Através dos experimentos citados anteriormente pretende-se testar as seguintes hipóteses:

Hipótese 1: O agente com a implementação através da técnica de RNA reflete os aspectos comportamentais do jogador no ambiente de estratégia

Hipótese 2: Na percepção dos jogadores o agente com o uso de RNA gera comportamentos mais realistas do que as técnicas determinísticas

### **1.1.2 Delimitação de Escopo**

Os experimentos descritos têm por objetivo identificar se a percepção dos avaliadores sobre o agente mapeado refletem de fato o comportamento humano frente a abordagens determinísticas, voltadas à execução de tarefas de forma otimizada.

O escopo desta pesquisa foi restrito ao mapeamento das ações do jogador em um ambiente de estratégia, em um cenário de combate, portanto não foram tratados aspectos específicos nos testes, como dados de pontuação na execução de tarefas dentro do ambiente.

Outro fator importante é relativo ao nível de competitividade do agente treinado, já que o mesmo foi criado para executar as ações conforme os exemplos gerados pelo jogador e não para ser o mais efetivo em combate. Por isso, o número de vitórias ou taxas de acerto em relação ao oponente não são levados em consideração.

O mapeamento das ações pode ser realizado de diversas formas, sendo utilizada neste trabalho a obtenção das ações de maneira que permita a captura de dados do jogador tanto em um

nível mais alto de abstração, tentando obter suas estratégias e decisões, quanto na utilização dos dados da forma como são executados pelo jogador.

### 1.1.3 Justificativa

Mesmo com várias pesquisas indicando que é possível realizar o mapeamento do comportamento humano em agentes de jogos, aumentando o nível de complexidade do agente ao aproximar suas decisões do jogador humano, o processo de avaliação destes mapeamentos ainda demonstra metodologias pouco eficientes.

Thureau e Bauckhage (2004) apontam ainda que devido ao fato destes personagens se basearem nas ações do jogador, eles proporcionam níveis de jogabilidade muito superiores na percepção do usuário em comparação as aplicações tradicionais com técnicas determinísticas normalmente utilizadas.

Gorman *et al* (2006b) descrevem que para estes testes serem eficientes, o jogador tem que estar envolvido no processo de avaliação, para que se possa obter a percepção do mesmo sobre a qualidade dos agentes mapeados e muitos estudos na área ainda deixam de realizar os testes de credibilidade dos agentes, tornando os resultados da avaliação inconclusivos.

Em pesquisas que realizam testes comparativos para avaliar a credibilidade dos agentes com comportamento humano, são mapeadas ações específicas dos agentes ou definidos roteiros para que o personagem alcance alguns objetivos e possa ser avaliado neste processo. Sendo que mapear todas as ações que o jogador pode executar no ambiente do jogo se torna uma questão relevante para que o nível do agente possa ser comparado ao jogador como um todo.

Por isso, a relevância deste trabalho está na melhor compreensão sobre como uma implementação de RNA, baseada na imitação do comportamento do jogador humano é capaz de melhorar a qualidade do comportamento de personagens do jogo. Possibilitando com isso a comparação deste mapeamento com outras abordagens, avaliando a percepção dos jogadores nestas comparações. Para isso é utilizada uma metodologia experimental, já que publicações relacionadas a este tema nem sempre se baseiam em fatos com uma metodologia clara de testes para suas afirmações, com poucas pesquisas com um indicador direto sobre a qualidade e benefícios do mapeamento do comportamento humano.

## **1.2 OBJETIVOS**

Esta seção formaliza os objetivos do trabalho, conforme descrito a seguir.

### **1.2.1 Objetivo Geral**

Comparar o método determinístico normalmente aplicado a jogos com uma técnica de RNA através da imitação do comportamento do jogador, demonstrando a relação entre o desempenho e percepção dos jogadores resultantes das comparações e interações.

### **1.2.2 Objetivos Específicos**

1. Identificar as características do agente com abordagem determinística a ser utilizada nos testes;
2. Definir a abordagem da técnica de RNA a ser aplicada aos experimentos;
3. Realizar a simulação do jogo com o agente com abordagem de RNA, o agente com abordagem determinística e o jogador humano no Robocode, gravando os resultados das partidas;
4. Elaborar o instrumento de avaliação a ser utilizado nos testes de percepção dos jogadores;
5. Realizar os experimentos com jogadores, aplicando o instrumento de avaliação a partir das partidas gravadas com os confrontos utilizando diferentes abordagens no Robocode, coletando os resultados; e
6. Definir e aplicar os métodos estatísticos para análise dos resultados.

## **1.3 METODOLOGIA**

### **1.3.1 Metodologia da Pesquisa**

A metodologia de pesquisa utilizada caracteriza um método hipotético-dedutivo por se tratar de uma abordagem que visa testar as hipóteses definidas e a partir das verificações e comparações

entre as diferentes abordagens aplicadas aos agentes do jogo testar sua validade através de experimentos.

### **Sob o ponto de vista de sua natureza**

A pesquisa pode ser classificada como aplicada, pois visa demonstrar a melhoria na qualidade de agentes através do mapeamento do comportamento humano, gerando conhecimentos para soluções práticas que tem como objetivo a criação de agentes inteligentes mais próximos das ações do jogador humano.

### **Sob o ponto de vista da forma de abordagem do problema**

O problema de pesquisa tem uma abordagem qualitativa por ter como objetivo avaliar a percepção dos jogadores através de questionamentos sobre sua percepção, medindo entre diferentes abordagens para jogos utilizando valores estatísticos dos resultados das comparações entre agentes no jogo para avaliação.

### **Sob o ponto de vista de seus objetivos**

O trabalho se classifica como uma pesquisa exploratória por realizar estudos para comparar diferentes abordagens aplicadas a agentes permitindo identificar a proporção da melhora da qualidade da IA com o agente de RNA, através da imitação do comportamento humano.

## **1.3.2 Procedimentos Metodológicos**

Nos procedimentos metodológicos necessários para descrição das abordagens utilizadas no projeto tem-se a caracterização de uma abordagem experimental, portanto, foi realizado um experimento para avaliação do comportamento do agente implementado com base nas ações do jogador humano. Sendo assim, foram disponibilizados vídeos com as batalhas realizadas no Robocode e nestas batalhas cada robô representava uma abordagem diferente.

### **1.3.2.1 Construção do experimento**

Na etapa de construção do experimento foram gravadas as sessões das batalhas no ambiente do Robocode. Nestas gravações foram utilizadas diferentes combinações com base nas diferentes abordagens:

- Robô mapeado com o comportamento humano;
- Robôs com abordagem determinística, otimizados para vitória em batalhas; e
- Robô controlado por um jogador humano.

Em cada sessão foram selecionados 2 robôs em confrontos diretos na arena até a destruição de um deles. Foram atribuídos sistemas de cores diferenciados para identificação das diferentes abordagens.

Após a definição dos confrontos, foram gravadas 3 batalhas com base no agente mapeado com o comportamento humano frente às outras abordagens, na qual os robôs se enfrentaram diretamente, totalizando 6 partidas a serem assistidas pelos avaliadores.

Para adequação do experimento ao *framework* utilizado por Bauckage *et al* (2007) e Gorman *et al* (2006b), apenas um dos robôs é avaliado a cada partida.

Realizadas as gravações de batalhas e definidos os detalhes da avaliação foi determinado o experimento a ser aplicado, em que cada avaliador visualizou 6 partidas e respondeu aos questionamentos a cada batalha apresentada.

### **1.3.2.2 Aplicação do experimento**

Na aplicação do experimento os avaliadores responderam a uma questão inicial relativa a seu nível de experiência como jogador e após este questionamento inicial, foram realizadas as avaliações a partir das batalhas gravadas.

A elaboração dos questionários foi baseada no *framework* de Bauckage *et al* (2007) e Gorman *et al* (2006b), que avalia também o nível de percepção de jogadores. A avaliação em si consiste em questionários de múltipla escolha, em que a cada partida assistida pelo avaliador o mesmo atribuí sua percepção sobre o comportamento do agente que está sendo demonstrado naquela batalha, o instrumento utilizado para avaliação da percepção dos jogadores é apresentado na Subseção 4.8.1 4.8.1 .

Para realizar esta avaliação foram disponibilizados os questionários através de páginas Web para cada um dos avaliadores, junto aos questionamentos é apresentado o vídeo da batalha entre os

robôs. O avaliador assiste ao vídeo e então responde as questões solicitadas e assim que finalizada a avaliação de um dos robôs, outra batalha é apresentada com os mesmos questionamentos.

O questionário foi idealizado para funcionar com escalas dos níveis de comportamento para que estes dados numéricos possam ser utilizados na avaliação estatística. Também são solicitadas duas informações adicionais a cada avaliação, uma referente ao nível de humanidade quando o personagem é considerado artificial e outra solicitando a justificativa da resposta, sendo que estas informações não fazem parte do processo estatístico e são utilizadas apenas para conhecer melhor a opinião dos avaliadores.

### **1.3.2.3 Avaliação do comportamento aplicado ao agente**

Para cada batalha foi realizada a avaliação do nível de percepção atribuído ao agente com comportamento humano. Além disso, foram comparados os resultados do agente com comportamento humano frente ao agente determinístico e ao próprio jogador humano.

Os níveis atribuídos a cada uma das abordagens são utilizados como indicadores do “nível de comportamento humano”.

As avaliações a partir das comparações entre as diferentes abordagens são importantes para avaliar se o fato dos robôs determinísticos serem otimizados para vencer as batalhas, não confunde a percepção dos jogadores.

Já o jogador humano utilizou uma interface de interatividade para controlar o robô, sendo que a utilização deste robô foi incorporada na validação do modelo de avaliação, pois o jogador humano tem que obter o maior nível de comportamento humano ou ao menos o mesmo nível, quando comparado ao agente mapeado com base nas ações do jogador.

Todos os testes de hipóteses foram então executados e as avaliações estatísticas realizadas com base nos resultados obtidos.

As avaliações foram divididas entre as 3 diferentes abordagens: agente com mapeamento do comportamento humano, agentes determinísticos e jogador humano. As proporções do nível de comportamento foram comparadas de acordo com os níveis de percepção disponibilizados aos avaliadores para identificação do valor obtido por cada uma das abordagens.

A partir desta classificação foi realizada uma análise para identificar se o comportamento do jogador humano de fato reflete no agente implementado, considerando os diferentes níveis de percepção.

## 1.4 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho está dividido em 6 seções. A primeira seção trata da introdução do trabalho, o problema que se visa resolver e a solução proposta para o mesmo. Além disso, são apresentados nesta primeira seção os objetivos e a metodologia de pesquisa adotada no trabalho.

Na segunda seção é apresentada a fundamentação teórica, na qual são tratados os conceitos sobre a técnica de inteligência aplicada para mapeamento do comportamento humano em agentes do Robocode, detalhes sobre abordagens determinísticas, os conceitos de agentes inteligentes e as ferramentas e técnicas utilizadas no mapeamento das ações do jogador.

A terceira seção apresenta uma revisão sistemática sobre trabalhos que realizam o mapeamento do comportamento humano, destacando os trabalhos mais similares a esta pesquisa. Estes trabalhos foram destacados conforme as técnicas, metodologia e testes realizados.

Na quarta seção são apresentados os detalhes sobre a implementação do personagem mapeado com o comportamento do jogador. No decorrer da seção é demonstrado o método para obtenção dos dados do jogador, o tratamento destes dados e a definição do treinamento do agente. É descrito também um *framework* para realização dos testes comparativos entre o agente implementado, abordagens determinísticas e um personagem controlado pelo jogador para os testes de percepção.

Os resultados obtidos são apresentados na Seção 5 e na Seção 6 são apresentadas as conclusões sobre a pesquisa realizada.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Esta seção trata os principais temas de interesse desta pesquisa, que foram divididos entre os seguintes tópicos: os fundamentos relativos a Redes Neurais Artificiais utilizados no trabalho; os fundamentos e o tratamento determinístico da implementação tradicional em jogos baseada em regras; a definição e perfis de Agentes Inteligentes; e a definição das ferramentas utilizadas no experimento.

### **2.1 REDES NEURAIAS**

As Redes Neurais Artificiais são utilizadas para auxílio aos mais diversos problemas, devido a sua semelhança estrutural em relação à mente humana que permite que pesquisadores da área de inteligência artificial busquem nestes modelos soluções para os problemas encontrados em outras áreas (FERNANDES, 2003).

As soluções disponibilizadas pelas RNA's abordam o aprendizado de máquina, o que permite que soluções que as utilizem possam gerar aplicações com comportamentos não determinísticos e mais próximos a padrões de complexidade próximos do ser humano em determinadas tarefas (LUDWING JR.; COSTA, 2007).

Fernandes (2003) descreve que os próprios psicólogos tem grande interesse nas RNA's devido a sua semelhança com a estrutura do cérebro humano. As RNA's utilizam características de adaptação do cérebro humano o que permite considerá-las máquinas que modelam a maneira como o cérebro aprende determinadas tarefas e funções.

Todos estes aspectos demonstram que as RNA's podem mapear padrões de comportamento do ser humano para um determinado contexto ou habilidade. Estes princípios indicam a qualidade de abordagens de RNA, mas é importante compreender quais são as características que tornam as RNA's tão benéficas.

As RNAs' são sistemas paralelos e distribuídos compostos por unidades de processamento que calculam determinadas funções matemáticas, normalmente não lineares, dispostas em uma ou várias camadas que estão interligadas por diversas conexões, normalmente unidirecionais (BRAGA; CARVALHO; LUDEMIR, 2000).

A maneira como as redes neurais solucionam problemas supera o desempenho de modelos convencionais, principalmente pela forma como é representada. A capacidade de aprendizado e generalização destas redes está associada à capacidade de aprender através de um conjunto de exemplos, fornecendo respostas coerentes para dados desconhecidos, o que a torna uma ferramenta computacional poderosa e eficaz na solução de problemas complexos (HAYKIN, 2001).

Segundo Fernandes (2003), existem várias definições do que seja uma RNA, sendo que os neurônios, a arquitetura e a aprendizagem, a priori, fazem parte de qualquer definição de RNA.

Nas próximas seções serão abordadas cada uma destas definições com mais detalhes, iniciando com uma visão geral sobre sua estrutura, sendo cada abordagem realizada de acordo com os fundamentos utilizados na execução do experimento ao que esta dissertação se propõe.

### 2.1.1 Estrutura da rede

Assim como o cérebro, as redes neurais alcançam um bom desempenho através da interligação de neurônios artificiais, aplicando um processo de aprendizagem a estes neurônios.

Os neurônios artificiais são unidades de processamento e foram idealizados a partir do neurônio biológico presente no cérebro humano a fim de seguir os mesmos princípios de funcionamento (HAYKIN, 2001). A Figura 1 apresenta o modelo de um neurônio artificial em uma estrutura de rede.

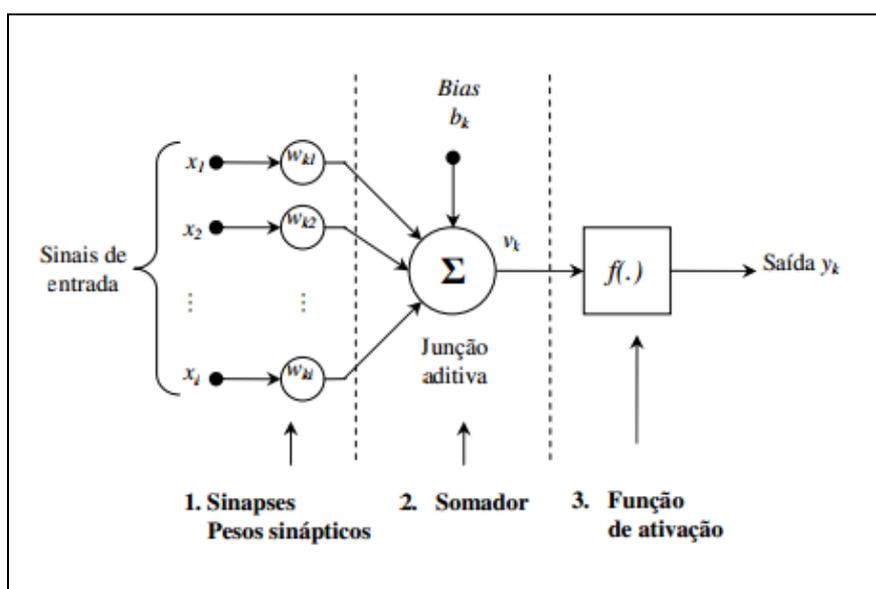


Figura 1. Modelo não-linear de um neurônio.

Fonte: Adaptado de Haykin (2001).

Na Figura 1 é possível perceber que a Rede Neural Artificial é composta por diversas unidades de processamento interligadas através de elos de conexão (sinapses), que assim como no cérebro recebem sinais de entrada que podem ser ativados ou não a partir de funções de ativação que geram os valores de saída da rede. Cada conexão está associada a um determinado peso e as unidades de processamento realizam as operações sobre os dados recebidos através das entradas, as interações entre estas unidades são responsáveis pelo comportamento inteligente destas redes (GURNEY, 1997).

McCullock e Pits propõem um modelo de operação para as unidades de processamento, que segundo Fernandes (2003) pode ser resumido da seguinte maneira:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um peso que indica sua influência na saída;
- É realizada uma soma ponderada dos sinais que produz um nível de atividade; e
- Se este nível de atividade exceder certo limite a unidade de processamento produz uma determinada resposta.

A grande maioria dos modelos de Redes Neurais Artificiais possui alguma regra de treinamento que ajusta os pesos de suas conexões de acordo com padrões detectados, aprendendo através de exemplos.

As unidades de processamento das RNA's podem ser interligadas em uma ou diversas camadas formando diferentes arquiteturas. Um exemplo pode ser visto na Figura 2.

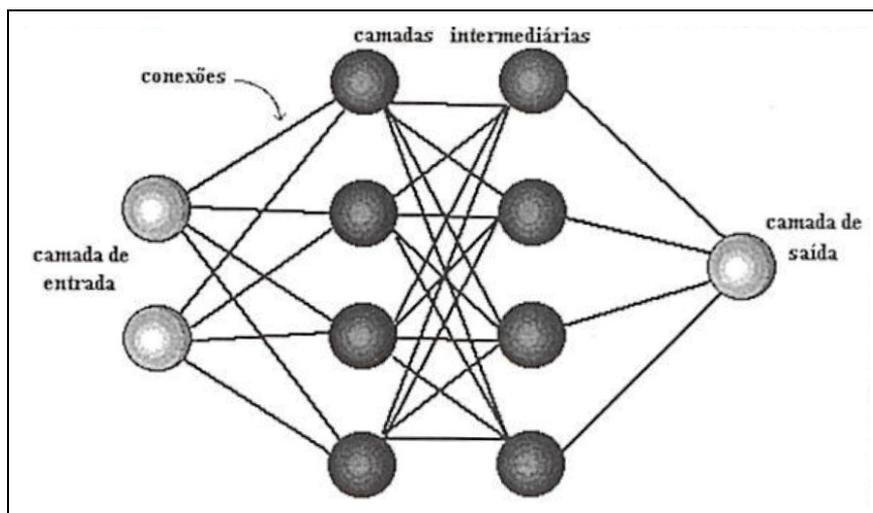


Figura 2. Camadas de um Rede Neural Artificial.

Fonte: Adaptado de Fernandes (2003).

As conexões entre as camadas podem gerar inúmeras arquiteturas diferentes e a disposição destas estruturas está diretamente relacionada com o algoritmo de aprendizagem utilizado para treinamento da rede (LUDWING JR.; COSTA, 2007).

Segundo Ludwig Jr. e Costa (2007) as camadas são normalmente classificadas da seguinte maneira:

- Camada de entrada: onde os padrões de entrada são apresentados à rede;
- Camadas intermediárias: onde é realizada a maior parte do processamento, através das conexões ponderadas; e
- Camada de saída: onde é concluído o processamento e apresentado o resultado final.

A propriedade primordial de uma RNA é sua capacidade de aprendizado a partir de seu ambiente, melhorando seu desempenho através desta aprendizagem. Esta aprendizagem ocorre a partir de um processo iterativo onde são realizados ajustes nos pesos com o intuito de que a cada iteração a rede se torne mais instruída (HAYKIN, 2001).

Estas iterações caracterizam o processo de treinamento da rede. Segundo Valiati (2000) estes métodos estão divididos em duas abordagens:

- Treinamento Supervisionado: Neste tipo de treinamento a entrada e saída desejadas para a rede são disponibilizadas por um "supervisor" que fica responsável por indicar explicitamente se a rede está se comportando da maneira desejada e realizar os ajustes em seus parâmetros para encontrar uma ligação entre as entradas recebidas e a saída fornecida; e
- Treinamento não Supervisionado: Neste tipo de treinamento não existe "supervisor" e não são usadas informações para identificar se a resposta da rede foi correta ou não para realização dos ajustes. No treinamento não supervisionado a rede tenta estabelecer uma harmonia com regularidades estatísticas a partir da entrada para se tornar capaz de identificar as características da entrada e produzir novas classes automaticamente.

O método de treinamento utilizado na primeira etapa deste projeto caracteriza um aprendizado supervisionado, em que a partir das definições do modelo da rede e atributos a rede tenta realizar o reconhecimento das ações do jogador. O único teste realizado é baseado nas ações do personagem que utiliza a Rede Neural, que deve simular os mesmos padrões de ação do jogador de onde foram mapeadas as ações para o treinamento.

Nas próximas seções são explorados conceitos do processo de aprendizado e treinamento supervisionado utilizados no treinamento da rede.

### **2.1.2 Aprendizado supervisionado**

A etapa de aprendizado é primordial em um RNA, que a partir de observações do ambiente e através de um processo interativo os parâmetros da rede são ajustados de forma contínua melhorando seu desempenho através da aprendizagem. Durante o processo de aprendizado é possível observar que a rede é estimulada por um ambiente e sofre modificações em seus parâmetros como resultado destes estímulos, apresentando os dados de saída de uma nova maneira, devido às modificações que ocorreram em sua estrutura interna (HAYKIN, 2001).

No processo de aprendizado supervisionado os dados a serem processados pela rede são representados por conjuntos de exemplos com entradas e saídas e este conjunto é fornecido por um "supervisor" que é responsável por indicar os padrões de ligação entre as entradas e a saída

fornecida. Na Figura 3 é possível visualizar um diagrama em blocos da aprendizagem supervisionada (VALIATI, 2000).

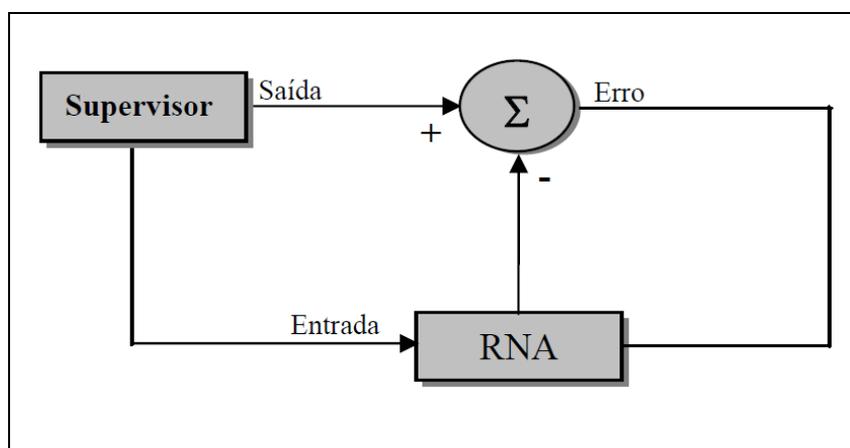


Figura 3. Diagrama em blocos do modelo de aprendizagem supervisionada.  
Fonte: Adaptado de Valiati (2000).

Valiati (2000) exemplifica o processo descrevendo que quando um padrão é apresentado às entradas da rede, ocorre uma comparação entre a saída desejada e a resposta calculada, os pesos então sofrem um ajuste para minimizar a medida do erro.

No caso da rede construída neste projeto o próprio conjunto de dados apresenta as respostas desejadas, desta maneira o conhecimento sobre as ações do ambiente é transferido para a rede a cada iteração até que a condição de treinamento seja completamente alcançada.

A aprendizagem supervisionada faz parte do processo de aprendizado dos tipos de redes neurais utilizadas nesta pesquisa, que são abordadas na próxima subseção.

### 2.1.3 Tipos de Redes Neurais

A escolha dos tipos de rede e métodos de treinamento utilizados nesta pesquisa foram baseados em trabalhos relacionados que realizaram estudos sobre o mapeamento do comportamento humano em personagens de jogos.

Com base nestes trabalhos foi definida a utilização de uma rede do tipo Perceptron para as tarefas de movimentação do robô e uma rede MLP para as tarefas relacionadas ao controle da torre.

Detalhes das definições realizadas podem ser vistos na Seção 4.5.

Os conceitos referentes às redes utilizadas são descritos nas próximas seções.

## 2.1.4 Rede Perceptron

Conforme descrevem Ludwig Jr. e Costa (2007) a rede neural do tipo Perceptron possui apenas uma camada de entrada e uma camada de saída. Os elementos que fazem parte da camada de entrada realizam a distribuição dos sinais recebidos para todos os neurônios que compõem a camada de saída. Estes neurônios são constituídos por uma função de ativação e uma função de transferência. A função de ativação realiza a soma ponderada dos sinais recebidos na entrada, enquanto a função de transferência determina a saída do neurônio.

Braga, Carvalho e Ludemir (2000) explicam que embora o Perceptron seja conhecido como uma rede de única camada a topologia original possui três níveis, composta pelas unidades de entrada ou de retina, por um nível intermediário formado pelas unidades de associação e por um nível de saída. Porém somente o nível de saída possui propriedades que permitam adaptação, o que justifica a maneira como a rede é conhecida.

O Perceptron é construído com base em um neurônio não-linear, conforme o modelo de McCulloch-Pitts. Este neurônio consiste na utilização de um combinador linear, seguido de um limitador que realiza a função de sinal (HAYKIN, 2001).

Haykin (2001) também descreve que o modelo neural utiliza um “nó aditivo”, que realiza o cálculo a partir de uma combinação linear das entradas aplicadas a suas sinapses, além de incorporar um *bias* que é utilizado externamente pela rede.

Os cálculos realizados pela rede Perceptron utilizam um algoritmo próprio de aprendizado para realização do treinamento, os detalhes de funcionamento deste algoritmo são descritos na próxima subseção.

### 2.1.4.1 Algoritmo de aprendizado do modelo Perceptron

O algoritmo de aprendizado do Perceptron de uma camada é representado pela regra delta e o elemento de processamento que utiliza esta regra é chamado de *Adaptive Linear Neuron*. Neste algoritmo atribui-se inicialmente pesos com valores aleatórios e a partir deles é definido um conjunto de sinais de entrada, utilizados para calcular a resposta da rede (LUDWING JR.; COSTA, 2007).

No caso do treinamento supervisionado, após as respostas da rede serem calculadas os valores gerados são comparados com os valores desejados e caso o erro não seja aceitável, é realizado o ajuste dos pesos de maneira proporcional ao erro e ao valor do sinal de entrada correspondentes.

Ludwig Jr. e Costa (2007) explicam que quanto maior é o valor da entrada correspondente a um peso, maior é a participação deste peso na formação dos valores de saída e no erro global gerado. Por isso, este peso precisa receber um valor maior de correção. A expressão que demonstra o cálculo realizado neste processo pode ser visualizada em (1).

$$w(i, j)_{T+1} = w(i, j)_T + \eta E(j)_T x(i) \quad (1)$$

Sendo em (1):

$W(i, j)_{T+1}$  = Valor do peso corrigido.

$W(i, j)_T$  = Valor do peso na iteração anterior.

$E(j)_T$  = Valor do erro para o neurônio  $j$ .

$i$  = Índice do sinal de entrada.

$T$  = Iteração.

$j$  = Índice do neurônio.

$\eta$  = Taxa de aprendizado.

$x(i)$  = Sinal de entrada.

O erro médio para todo o conjunto de treinamento é definido a partir do erro médio de todos os neurônios da camada de saída na iteração  $T$  conforme demonstrado em (2):

$$\varepsilon_{med} = \frac{\sum_{T=1}^n \varepsilon(T)}{n} \quad (2)$$

Sendo em (2):

$\varepsilon_{med}$  = Erro global médio.

$\varepsilon(T)$  = Erro médio para os neurônios da camada de saída na iteração  $T$ .

$n$  = Número de neurônios da camada de saída.

A partir da avaliação do nível de precisão da rede calculado através do erro médio global pode ser realizado o encerramento da seção de treinamento do Perceptron.

### **2.1.5 Rede Multi layer Perceptron (MLP)**

As RNA's do tipo MLP são redes que contém pelo menos uma camada intermediária (BRAGA; CARVALHO; LUDEMIR, 2000).

Nas redes MLP as camadas intermediárias propagam sinais de uma para outra até a camada de saída que recebe os sinais da camada intermediária e fornece o padrão de resposta. As camadas intermediárias funcionam como detectores de características que geram uma codificação dos padrões de entrada, permitindo que a mesma crie sua própria representação do problema, com mais riqueza e complexidade (FERNANDES, 2003).

Braga, Carvalho e Ludemir (2000) descrevem que para uma rede com pelo menos 2 camadas intermediárias são atribuídos os seguintes processamentos a cada camada:

- Primeira camada intermediária: Cada nó contribui aos valores dos padrões de treinamento
- Segunda camada intermediária: Cada nó combina os valores traçados pelos neurônios da camada anterior ao qual estão conectados, formando regiões espaciais conexas, em que o número de lados é definido pelo número de unidades que estão conectadas a ele.
- Camada de saída: Cada um dos nós forma regiões que são combinadas das regiões convexas definidas pelos nós ao qual estão conectados na camada anterior. Desta maneira os nós definem as regiões com formatos abstratos.

#### **2.1.5.1 Treinamentos de redes MLP**

No treinamento os sinais são propagados para as entradas, como um vetor de sinais, é calculado então o campo local induzido e os sinais de saída para os neurônios desta camada. Cada sinal é então distribuído para todos os neurônios da camada seguinte, até a camada de saída em que são obtidos os sinais funcionais da rede (LUDWING JR.; COSTA, 2007).

Para treinamento em redes MLP, assim como em outros tipos de rede, dependendo do algoritmo implementado no treinamento, diferentes operações são realizadas e diferentes resultados

podem ser observados. O conjunto de dados utilizados para o treinamento e o tipo de problema a ser respondido devem ser levados em consideração para aplicação do algoritmo de treinamento adequado.

Na próxima subseção é abordado o algoritmo selecionado para o treinamento da rede MLP que se mostrou mais adequado ao trabalho (detalhes da seleção do algoritmo podem ser vistos na seção 4.5)

### Algoritmo Backpropagation

O algoritmo de Backpropagation tem todo o seu procedimento baseado no sinal de erro. Os sinais são apresentados à entrada e os dados fluem através da rede, passando por todas as camadas, até que uma resposta seja emitida na camada de saída. A saída é então comparada à saída desejada e se não estiver correta, o erro é calculado. Este erro é então propagado desde a camada de saída até a camada de entrada alterando no caminho os pesos das conexões das unidades internas conforme o erro vai sendo retro propagado.

O erro global médio da rede, considerando todo o conjunto de treinamento, para  $N$  iterações, até que todo o ciclo de treinamento esteja completo (a cada ciclo realizado denominamos uma época), é definida pela média aritmética dos erros globais instantâneos, demonstrado por Ludwig Jr. e Costa (2007) em (3):

$$\bar{E} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (3)$$

Sendo em (3):

$\bar{E}$  = Erro global médio.

$E$  = Erro global instantâneo na iteração  $n$ .

$n$  = Número da iteração.

$N$  = Número de iterações para apresentar todo o ciclo de treinamento.

As redes com BP utilizam uma variação da regra delta generalizada que tem por objetivo obter pesos que lhe permitam realizar mapeamentos de forma não linear entre os padrões de entrada e saída, tendo como objetivo principal o ajuste dos pesos para minimizar o erro médio quadrático total (VALIATI, 2000). A regra delta pode ser visualizada em (4).

$$\Delta w_{ij}(n) = -\eta \frac{\partial E(n)}{\partial w_{ij}(n)} \quad (4)$$

Sendo em (4):

$\Delta w_{ij}(n)$  = Incremento do peso sináptico  $w_{ij}$  na iteração  $n$ .

$\frac{\partial E(n)}{\partial w_{ij}(n)}$  = Derivada da função de erro global em relação ao peso sináptico  $W_{ij}$  na

iteração  $n$ .

$n$  = Número da iteração.

$\eta$  = Taxa de aprendizado.

Quanto menor o parâmetro da taxa de aprendizado  $\eta$ , menor serão as variações de pesos sinápticos da rede a cada iteração. No entanto esta melhoria é obtida a custo de uma taxa de aprendizagem lenta. Por outro lado, tentar acelerar a taxa de aprendizado com um valor de  $\eta$  muito alto pode fazer com que a rede oscile demais e não alcance o resultado desejado. A regra delta generalizada apresentada em (4) é utilizada para evitar esta oscilação com a utilização de um termo de momento.

Depois que a rede estiver treinada com o algoritmo de BP, com o erro estando em um nível satisfatório ela pode ser utilizada para classificação de novos dados. Neste caso, novos dados de entrada são recebidos pela rede e processados pelas camadas intermediárias até a saída, mas sem a retro propagação de erro.

### **2.1.6 Treinamento a partir de padrões comportamentais**

O reconhecimento de padrões aproxima muito as redes neurais de seres humanos (SCHOPF, 2007). Assim como citado nas seções sobre os métodos de treinamento das redes MLP os padrões de entrada reconhecidos pelas redes são recebidos repetidamente, a partir de então a rede consegue classificar os padrões reconhecidos através dos dados de treinamento.

Neste trabalho a tarefa da RNA é a de detectar padrões comportamentais de um jogador humano, em que as ações do jogador estão mapeadas no arquivo de treinamento criado e são utilizadas para reconhecimento de suas ações.

As RNAs são utilizadas em diversos trabalhos para a tarefa de reconhecimento de padrões nas mais diversas áreas, sendo em muitos deles para o mapeamento de padrões comportamentais.

Na Tabela 1 são citados alguns destes trabalhos com um breve resumo da tarefa de análise comportamental realizada, nesta tabela são demonstrados desde trabalhos que realizam a classificação de padrões comportamentais em animais até uma amostra de trabalhos similares que realizam a implementação de padrões comportamentais de jogadores humanos (a lista completa por ser visualizada com mais detalhes na Subseção 3.3.1 ).

Tabela 1. Amostra de trabalhos envolvendo análise comportamental.

<b>Autor(es)</b>	<b>Descrição da análise realizada</b>
JHUANG <i>et al</i> , 2010	Classificação de características comportamentais de ratos em gaiola a partir de descritores de movimento
ROUSSEAU <i>et al</i> , 2000	Análise comportamental de uma rato em uma arena para classificação de suas atividades
BAUCKHAGE <i>et al</i> , 2007	Mapeamento do comportamento do jogador para movimentação e estratégia em um personagem do jogo Quake 2
THURAU; BAUCKHAGE, 2004	Reprodução da característica do comportamento humano na movimentação em um jogo FPS
BRYANT; MIKKULAINEN, 2007	Reprodução do comportamento humano em um jogo de estratégia para todas as ações dos personagens

Em todos estes trabalhos, mesmo para diferentes finalidades foram realizadas tentativas de mapeamento dos padrões comportamentais. Uma dificuldade observada em todos eles foi a maneira utilizada para mapear estes padrões, em sua maioria, como em Jhuang *et al* (2010) e Rousseau *et al* (2000) foram utilizados equipamentos para mapeamento das ações dos animais e a partir das leituras realizadas os dados foram utilizados para treinamento nas redes.

Em outros, mais similares a esta pesquisa como em Bauckhage *et al* (2007), Thureau e Bauckhage (2004) e Bryant e Miikkulainen (2007), foram utilizadas ferramentas complementares ao ambiente do jogo para obter as ações do jogador humano e criação do arquivo de treinamento, sendo que as ferramentas utilizadas estavam diretamente ligadas ao ambiente em que o personagem estava sendo simulado. Com isso cada autor definiu sua própria técnica para obtenção destes dados, baseados em suas necessidades.

Devido a este fator foi realizada uma pesquisa para definir a melhor ferramenta para esta tarefa, assim como nos trabalhos similares citados. A escolha foi realizada baseada nos requisitos

para criação do arquivo de treinamento e para realização de testes. A Seção 2.4 demonstra a pesquisa realizada entre as ferramentas mais populares para esta finalidade.

No caso deste trabalho o Robocode foi escolhido para servir como a ferramenta de leitura para as ações do usuário. A partir das ações do jogador ao controlar um robô do jogo em batalhas na arena Robocode os dados são lidos para que possam formar o arquivo de treinamento com as ações do jogador humano.

### 2.1.7 Conjuntos de treinamentos desbalanceados

Na etapa de treinamento da rede os exemplos contidos no conjunto de dados pertencentes a uma determinada classe, permitem que a rede realize o reconhecimento de padrões. Quando existe muita diferença entre o número de exemplos pertencentes a cada classe a aprendizagem da rede pode ficar prejudicada.

Nestes casos, quando os conjuntos de dados de treinamentos estão demasiadamente desbalanceados, os problemas podem ocorrer na generalização de classes menos representadas no conjunto. Frequentemente os modelos utilizados para treinamento tem uma boa precisão para as classes mais representativas, mas a precisão nas classes minoritárias acaba sendo inaceitável. O autor ainda afirma que esta é a regra e não exceção para a maioria das aplicações com conjuntos de dados desbalanceados, pois dos casos as classes minoritárias são de maior interesse (PRATI, 2006).

Devido a estes problemas pesquisadores como Chawla (2005), Han, Wang e Mao (2005) realizam estudos para tratar estes dados. Entre estas pesquisas alguns métodos se destacam, conforme cita Prati (2006):

- Remoção de exemplos das classes majoritárias: Esta é uma abordagem bastante direta para resolução do problema de classes desbalanceadas, em que é realizado o balanceamento artificial a partir da distribuição das classes no conjunto de exemplos. Estes métodos são conhecidos como *under-sampling* e visam balancear o conjunto de dados removendo exemplos das classes majoritárias; e
- Inclusão de exemplos das classes minoritárias: Métodos desta categoria são conhecidos como *over-sampling*, que são muito similares aos métodos *under-sampling*, cujo objetivo é

balancear a distribuição das classes por meio de replicação de exemplos da classe minoritária.

Para qualquer uma das técnicas utilizadas, os tratamentos destes conjuntos desbalanceados visam melhorar o desempenho das classes minoritárias a partir do balanceamento das classes do conjunto de dados, pois ao removerem exemplos da classe majoritária ou incrementarem as classes minoritárias o procedimento torna esta última mais custosa, garantindo que ela será melhor classificada (HAN; WANG; MAO, 2005)

Em contrapartida, estas alterações podem fazer com que a rede se baseie demais em exemplos incorretos e o balanceamento acabe não sendo efetivo, portanto estas alterações devem ser avaliadas cuidadosamente.

### **2.1.8 Método de treinamento**

Para realização de testes durante a etapa de treinamento, diferentes modelos de rede e algoritmos devem ser testados. Para cada configuração uma diferente implementação teria que ser realizada se nenhuma ferramenta de treinamento para esta finalidade fosse utilizada.

Por isso, a utilização de uma ferramenta que pudesse executar a modelagem e treinamento das diferentes configurações da rede foi importante neste trabalho. A ferramenta definida para utilização traz agilidade por disponibilizar a opção de executar testes com diferentes algoritmos de maneira mais rápida, utilizar o mesmo arquivo de treinamento para executar experimentações, exibir estatísticas do processo, disponibilizar resultados matemáticos da rede, além de disponibilizar relatórios que auxiliam na leitura dos dados de treinamento.

Neste trabalho a ferramenta escolhida foi o Neuroph Studio, da Oracle que já vem integrada com outra solução também utilizada na implementação dos personagens do Robocode, o Netbeans.

A escolha específica do Neuroph Studio veio principalmente através da preferência em utilizar uma ferramenta que pudesse gerar os arquivos da rede diretamente em Java, linguagem do Robocode. Também foram analisadas as ferramentas utilizadas para treinamento em trabalhos similares que realizaram a utilização da técnica de RNA no mapeamento das ações do jogador humano. Entre estes trabalhos não foram encontradas soluções para a geração diretamente na linguagem.

Com a utilização do Neuroph Studio o ambiente de treinamento foi executado dentro de uma só ferramenta, trazendo praticidade a execução de treinamento e implementação dos personagens. Outras alternativas foram consideradas, como o JavaNNS, utilizada em diversos trabalhos na área, que não foi utilizado principalmente devido a linguagem de programação.

Na próxima subseção, finalizando os fundamentos sobre RNA utilizados neste trabalho, são apresentadas as principais características do Neuroph.

### 2.1.8.1 Neuroph Studio

O Neuroph Studio é um *framework* Java que permite o desenvolvimento das principais arquiteturas de Redes Neurais Artificiais. É uma ferramenta de código aberto com um pequeno número de classes básicas, o que corresponde aos conceitos de redes neurais simplificados. A ferramenta também dispõe de uma (IDE) gráfica de edição para o usuário, incluindo assistentes para agilizar e facilitar o processo de criação de redes neurais com componentes baseados em código Java.

Graças a IDE disponibilizada o Neuroph simplifica o trabalho de desenvolvedores com a manipulação de redes neurais, além de disponibilizar bibliotecas com as principais arquiteturas de rede implementadas e prontas para uso através dos assistentes para criação, treinamento e implementação das redes.

O Neuroph tem um funcionamento similar ao JavaNNS e outras IDE's disponíveis para auxílio na criação e treinamento de redes neurais, sendo que diferente do JavaNNS o Neuroph tem a capacidade de gerar bibliotecas em Java para importação em aplicações.

De acordo com Neuroph (2012) a ferramenta visa diminuir a complexidade do processo de criação da rede, deixando o desenvolvedor se focar somente na tarefa sem precisar perder muito tempo para aprender a manipular a ferramenta. A ferramenta apresenta recursos importantes como a boa documentação, facilidade de uso e a flexibilidade do framework para uso com outras ferramentas.

Uma lista com as principais características da ferramenta é apresentada na sequência (NEUROPH, 2012):

- Pequeno número de classes essenciais no núcleo da aplicação (apenas 10);

- Suporte para regras de aprendizado supervisionado e não supervisionado;
- Uma estrutura e lógica simples para seguir;
- Uma IDE java voltada para a manipulação de redes neurais baseada na ferramenta Netbeans, permitindo total integração;
- Suporte ao reconhecimento de imagens;
- Exemplos visuais de aprendizado da rede;
- Normalização de dados; e
- Um *framework* para *benchmark* integrado.

A ferramenta também dispõe de suporte as principais arquiteturas de redes:

- Adaline;
- Perceptron;
- Multi Layer Perceptron com Backpropagation e termo Momentum;
- Rede Hopfield;
- Memória associativa bidirecional;
- Rede Kohonen;
- Rede Hebbian;
- Maxnet;
- Rede Competitiva;
- Instar;
- Outstar;
- Rede (RBF);
- Raciocínio Neuro Fuzzy; e
- (NEAT) (Abordagem com Algoritmos Genéticos).

Na sequencia podem ser vistas imagens das principais interfaces do sistema. A Figura 4 demonstra a interface inicial do framework, na Figura 5 o gráfico de visualização da rede, na Figura 6 a interface inicial do assistente e na Figura 7 tela do assistente para seleção da regra de aprendizado da rede MLP.

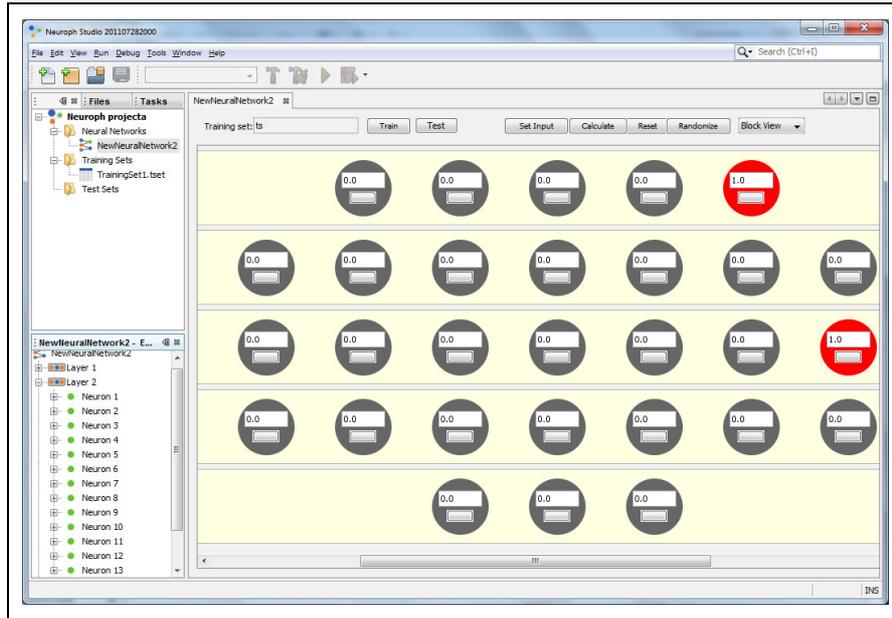


Figura 4. Interface de configuração das entradas.

Fonte: Neuroph (2012).

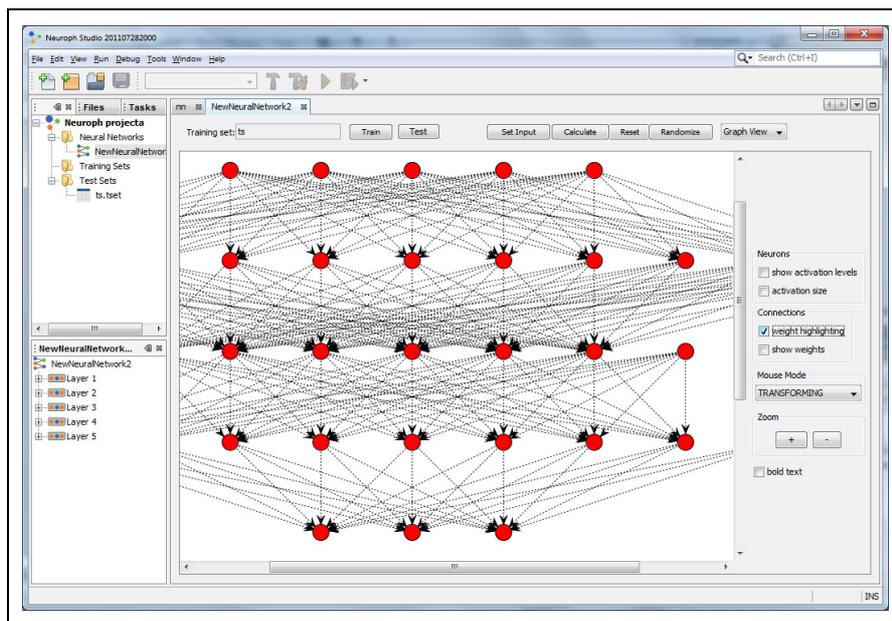


Figura 5. Interface demonstrativa da Rede Neural.

Fonte: Neuroph (2012).

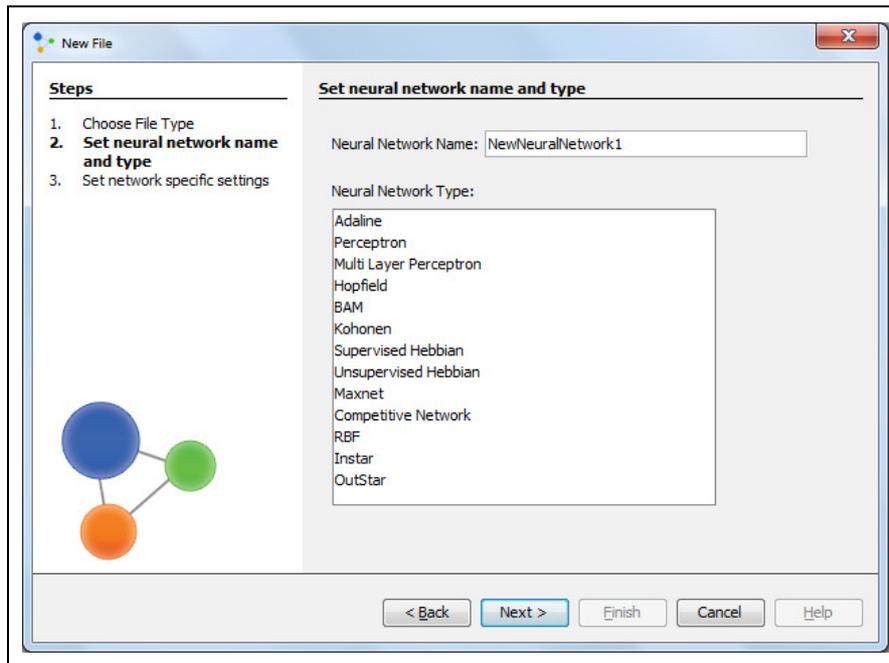


Figura 6. Interface inicial do assistente.

Fonte: Neuroph (2012).

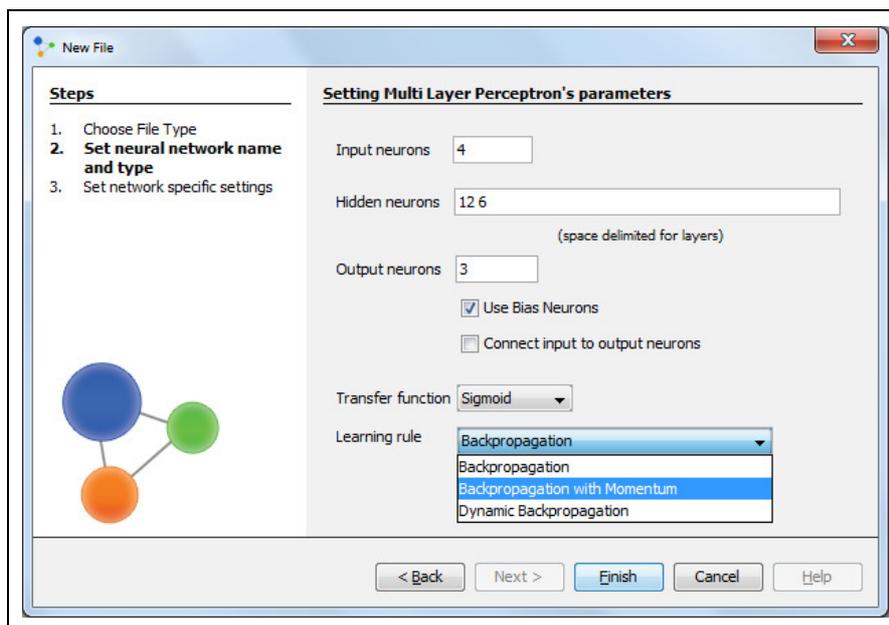


Figura 7. Interface para seleção da regra de aprendizado da rede MLP.

Fonte: Neuroph (2012).

## 2.2 SISTEMAS BASEADOS EM REGRAS

Os sistemas baseados em regras (*Rule Based Systems* - RBS) são muito utilizados para aplicação em personagens de jogos. Um sistema baseado em regras tem a forma simples de

“Condição - ação” se comportando de acordo com seu conjunto de prioridades e regras (COSTA, 2010).

A forma de representação “condição – ação” de um sistema baseado em regras caracteriza um método de baixa complexidade quando comparado a uma técnica de IA mais complexa como a de RNA.

Em alguns casos o uso da programação tradicional acaba tornando maior a dificuldade de gerenciar os recursos da aplicação e no caso dos jogos este gerenciamento consiste em trabalhar cada situação que faz parte do ambiente de maneira organizada e coerente (MALFATTI; FRAGA 2006).

Malfatti e Fraga (2006) também descrevem que para realizar esta tarefa da maneira correta o desenvolvedor precisa definir mecanismos que possam auxiliá-lo durante o desenvolvimento para que ele possa controlar adequadamente as regras aplicadas, além de todo o ambiente do jogo e essa pode ser uma etapa muito custosa se o jogo não estiver bem estruturado.

Isto prova que até mesmo a utilização de uma abordagem simples como a de RBS requer atenção por parte dos desenvolvedores para que se possa utilizá-la adequadamente, pois em caso de um sistema com muitas regras o gerenciamento de todos os recursos pode acabar se tornando muito complexo.

David (2008) cita que existem críticas muito comuns a sistemas baseados em regras, principalmente devido ao que é considerado como uma regra correta, pois dependendo do contexto a mesma regra pode não se aplicar. Outra questão citada pelo autor se refere ao fato de que existe a necessidade regular de adaptação de tais sistemas para que possam trabalhar melhor com as variações do ambiente.

Devido as dificuldades em implementar jogos mais complexos com sistemas de regras, algumas alternativas mais sofisticadas foram sendo adotadas para melhorar estes processos.

Perucia *et al* (2005) afirmam que uma das técnicas mais utilizadas por desenvolvedores para controle dos recursos de um jogo se baseia na utilização da técnica de MEF (Máquinas de Estados Finitos), principalmente pela dificuldade em trabalhar com a complexidade elevada de alguns jogos. As MEF ainda são abordadas de maneira determinística, mas facilitam o mapeamento das regras.

As MEF podem ser descritas de uma maneira geral como um grafo, em que os nós representam suas ações e a cada novo estímulo um novo estado é ativado.

Machado, Moraes e Nunes (2009) descrevem que uma das desvantagens das MEF é que como os estados são finitos, o comportamento executado acaba sendo repetitivo e conseqüentemente, previsível. Os autores ainda afirmam que apesar disso a técnica de MEF é muito utilizada em jogos de computador.

A técnica de MEF tem como principal característica definir o fluxo de funcionamento do jogo. Malfatti e Fraga (2006) descrevem as principais tarefas que caracterizam as MEF's. A principal tarefa consiste na definição dos estados que podem ser assumidos para realização de uma determinada ação no jogo, esta etapa é demonstrada através da Figura 8.

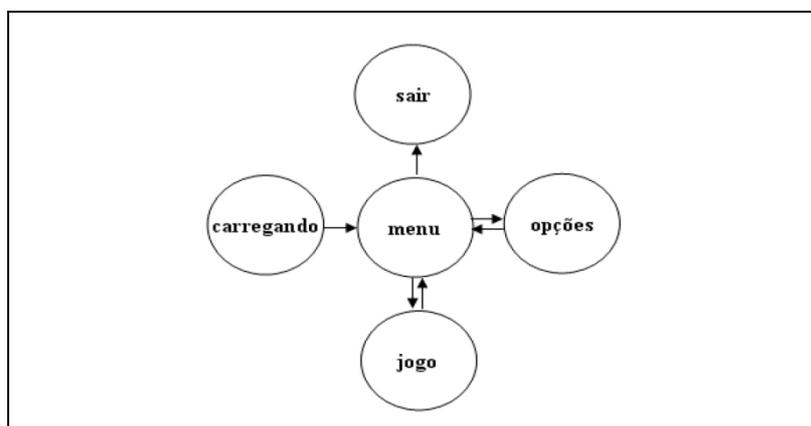


Figura 8. Exemplo de um diagrama de estados finitos.

Fonte: Malfatti e Fraga (2006).

Como o jogo pode contemplar diversas tarefas, a máquina de estados principal demonstrada na Figura 8 pode ser quebrada em várias máquinas de estados para realização de cada tarefa. A mudança entre os estados depende do evento disparado no jogo.

As MEF's também representam bem o método de programação tradicional, muito utilizado em jogos até hoje, que não apresentam os mesmos níveis de qualidade de técnicas de IA. Apesar de melhorarem o processo de mapeamento de regras as MEF's ainda apresentam problemas clássicos em jogos, como as ações repetitivas e *bugs* como no caso de jogos do tipo FPS (*First-Person Shooter*) que ficam pulando contra paredes ou andando em círculos.

As características de personagens construídos com abordagens determinísticas são importantes para que durante os experimentos deste trabalho possam ser avaliadas as características em que o personagem com abordagem não determinística mais se sobressai. Este personagem com abordagem não determinística caracteriza um agente inteligente, termo que será fundamentado na próxima seção.

## 2.3 AGENTES INTELIGENTES

Agentes são sistemas que tem seu comportamento guiado por um processo de raciocínio que representa suas ações.

Okuyama (2006) descreve que existem diversas definições de agentes, uma delas trata de agentes como entidades cognitivas, que assim como os humanos são munidos de percepções, motivações e sentimentos.

O autor também define que um agente pode ser visto como um programa que é capaz de se comunicar, cooperar e aprender, possuindo habilidades como a de realizar algumas tarefas humanas e interagir como se fossem pessoas.

Já O'Connor (1996 *apud* Almeida, 2004) define agentes como entidades de software que realizam operações em favor do usuário ou em favor de outro software com algum grau de autonomia.

Para Gomes (2005) não existe um conceito unânime para o termo “agente” no âmbito da IA, o que existe é uma convergência para as características principais que um software deve possuir para ser considerado um agente.

Segundo Costa (1999 *apud* Fernandes, 2003) agentes podem possuir as seguintes propriedades:

- Autonomia: Poder funcionar sem a intervenção humana e basear suas ações em seu próprio conhecimento armazenado sobre o ambiente;
- Habilidade Social: Interação com outros agentes através de uma linguagem comum;
- Reatividade: Ser capaz de perceber mudanças em seu ambiente e responder de acordo com

estas mudanças; e

- Pró-atividade: Não atuar apenas por percepção, o agente deve procurar alcançar a sua meta, apresentando iniciativa.

Okuyama (2006) ainda apresenta um modelo geral de um agente, conforme pode ser visto na Figura 9, demonstrando as percepções e ações do agente.

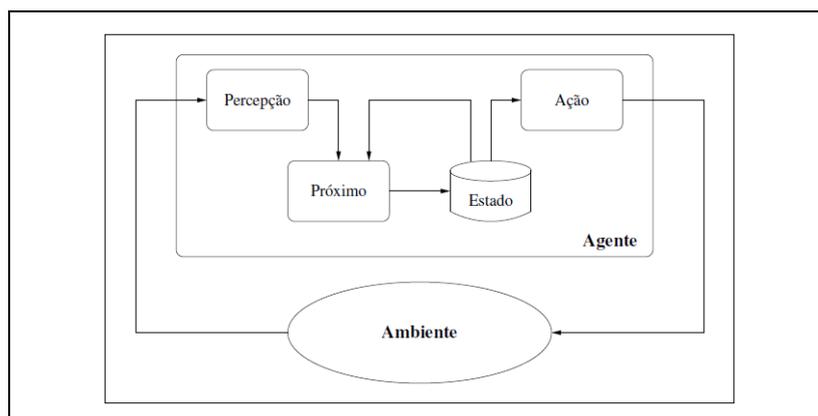


Figura 9. Modelo geral de agente.

Fonte: Okuyama (2006).

Wooldridge e Jennings (1995 *apud* Rodrigues, 2002) descrevem algumas representações de agentes inteligentes para as seguintes entidades ou processos:

- Agente humano: Qualquer pessoa que tenha interação com um sistema de computador de maneira direta ou indireta, mais conhecido como usuário;
- Agente Inteligente: Uma entidade que apresenta o comportamento de um agente;
- Ferramenta de software: Qualquer software que não seja de fato um agente;
- Comunicação ou interação: Indica a passagem de controle ou informações que podem ser transmitidas entre as entidades;
- Observação: Indica uma entidade que recebe informações de outras entidades; e
- Interface: O ponto pelo qual duas ou mais entidades se comunicam e interagem.

Entre as representações citadas o agente humano e inteligente representam os tipo de agentes

contidos neste trabalho.

O agente caracterizado pelos experimentos realizados neste trabalho trata de um agente autônomo, que interage de maneira independente por meio de seus próprios sensores e atuadores para realização de determinada tarefa. As tarefas deste agente podem ser divididas em 2 momentos:

- A primeira tarefa é a de utilizar o jogador para que ele possa interagir a partir do agente e utilizá-lo para mapear suas ações, nesta primeira etapa o agente é controlado pelo jogador e serve apenas como leitor das ações do jogador. Ainda nesta primeira etapa, também é definido o perfil deste jogador, por observação de seu nível de habilidade dentro do jogo, perante outros jogadores e *bots*, que devem refletir no perfil do agente após o treinamento; e
- A segunda tarefa é a implementação das ações do jogador no agente para que ele possa realizar as ações de forma autônoma, baseado no perfil do jogador humano, respondendo as ações executadas no cenário de maneira similar ao jogador.

Nestas tarefas fica claro que o agente tem papel fundamental para o sucesso da primeira etapa do experimento, pois o mesmo serve como mecanismo de leitura e comparação das ações do jogador humano.

Frezza (1997) reforça os argumentos do experimento, descrevendo que características de agentes humanos podem ser aplicadas a agentes artificiais, no que se referem a estados mentais, tais como conhecimento, crença, intenção, compromisso, escolha e desejo.

Conforme aponta Almeida (2004), para que um sistema seja considerado agente, o mesmo não tem necessariamente que apresentar todas as propriedades vistas nas definições de agentes, mas algumas delas são importantes e devem estar presentes.

Na próxima subseção serão abordadas a autonomia e aprendizagem de agentes, para posteriormente serem tratados os perfis, definições importantes para a finalidade que se busca com a utilização dos conceitos de agentes.

### **2.3.1 Autonomia e aprendizagem**

Entre as características abordadas sobre agentes foram citadas a autonomia e aprendizagem, sendo que estes tópicos tratam das principais características dos agentes que se procura construir

neste trabalho.

Autonomia se refere ao princípio básico de que agentes inteligentes possam realizar suas ações de maneira independente, baseados em suas próprias iniciativas, sem a necessidade de serem controlados.

Almeida (2004) cita que agentes podem ser considerados autônomos quando executam suas ações sem a presença do usuário, apenas se baseando em um perfil previamente construído de preferências.

Wooldrige e Jennings (2002) descrevem que a autonomia existe quando os agentes estão aptos a operar sem a intervenção direta de humanos e tem algum controle próprio sobre suas ações.

Outro fator importante é a aprendizagem, pois para que um grau adequado de autonomia seja alcançado o agente precisa aprender suas tarefas para que possa realizá-las de maneira autônoma e coerente.

Costa (1999 *apud* Fernandes, 2003) reforça este argumento, descrevendo que um dos atributos que melhor caracterizam agentes inteligentes é sua capacidade de aprendizado. O autor elabora explicando que uma autonomia real só pode estar realmente presente quando um agente consegue escolher suas ações baseado nas variações de seu ambiente externo.

O aprendizado também pode funcionar de uma maneira interativa, em que o treinador fornece o conhecimento ao agente em forma de instruções (FERNANDES, 2003).

### **2.3.2 Mapeamento de perfil em Agentes**

Para Merrick e Shafi (2011) o mapeamento de perfis em agentes se baseia principalmente na seleção de objetivo ou objetivos que o agente visa alcançar. Considerar múltiplos motivos permite observar diferentes domínios do problema a serem levados em consideração para mapeamento de um comportamento.

O autor ainda cita que estes perfis podem ser atribuídos em agentes artificiais de diversas maneiras, dependendo apenas do modelo de agente que está sendo implementado.

O mapeamento de ações de um jogador humano tem um grande objetivo principal, que se

divide em diversos pequenos objetivos entre as diversas tarefas que o mesmo deve executar durante a partida. Este conjunto de objetivos menores deve ser observado no mapeamento do perfil do jogador para que após o treinamento estas características possam ser observadas.

No trabalho de Mitaim e Kosko (1998) o perfil do usuário é mapeado através de suas escolhas, que descrevem suas preferências e permitem que sejam incorporadas ao agente. No caso do mapeamento do perfil do jogador neste trabalho, um processo similar ocorre, pois suas escolhas são mapeadas pela ferramenta e suas ações são capturadas para utilização no treinamento.

A grande diferença é que o autor precisa utilizar funções matemáticas para transformar as escolhas do usuário em dados computacionais, já neste trabalho a obtenção dos dados é feita pela ferramenta de simulação e a definição da leitura definida na própria rede neural.

Mitaim e Kosko (1998) também descreve que o mapeamento do perfil de um agente pode funcionar como um filtro de dados inteligente, em que o agente recebe os dados que o usuário quer que ele aprenda. O agente funciona como um filtro, porque ele mapeia apenas os dados relativos ao que o usuário deseja que ele aprenda traduzindo-os de forma computacional.

Para Pazzani e Billsus (1997) a implementação de perfis de usuários em agentes se faz da mesma maneira que um aprendizado supervisionado, requerendo uma gama de exemplos positivos e negativos para demonstrar as preferências destes usuários.

No caso de Pazzani e Billsus (1997) é utilizada uma interface Web para obter as preferências do usuário. O mapeamento é feito pelo número de visitas as interfaces preferenciais e *tags* com os termos mais utilizados. Estes dados geram parte dos valores numéricos necessários para mapeamento do agente.

É possível perceber que um dos principais desafios no mapeamento do perfil do usuário no agente fica a cargo da tradução de suas ações, preferências e escolhas de maneira que possam ser mapeadas nestes agentes. Também é possível observar que para cada problema diferentes maneiras de se obter os dados destes usuários são necessárias e dependem de seu contexto.

Portanto, a próxima subseção aborda a definição do perfil do jogador em si, perfil este que reflete as ações do jogador a serem mapeadas e é importante principalmente no momento de avaliar o agente treinado.

### **2.3.2.1 Perfil do jogador**

Conforme citado por Bryant e Miikkulainen (2007) o termo mapeamento do comportamento humano causa muita discussão. No trabalho Mozgovoy e Umarov (2010) são considerados aspectos relativos ao perfil do jogador a ser utilizado no treinamento para que possam ser reconhecidos estes aspectos no agente treinado.

Em contrapartida nos trabalhos de Thureau, Paczian e Bauckhage (2005) e Bauckhage, Thureau e Sagerer (2003), foi realizada a gravação das seções do jogador humano e depois comparadas as áreas graficamente afetadas para realização da mesma tarefa entre o agente e o jogador. Nestes testes os autores comprovaram que para execução da mesma tarefa o agente percorria o mesmo caminho com pequenas diferenças desviando dos obstáculos corretamente e atingindo o objetivo de maneira similar ao jogador.

Isto demonstra que a definição do perfil depende muito do método de testes definido, pois em comparação direta com sessões gravadas do jogador e a repetição destas seções para os mesmos objetivos não é necessário o reconhecimento do perfil, mas apenas das comparações diretas entre as duas abordagens.

Em trabalhos em que não é realizada uma comparação direta, o mapeamento depende das definições do perfil do jogador para avaliar o próprio nível de habilidades do mesmo.

A definição do perfil do jogador neste trabalho tem por objetivo identificar o nível de habilidades dos jogadores para que os dados possam ser avaliados corretamente nas comparações realizadas durante os experimentos.

Como citado, neste trabalho a Rede Neural recebe os dados sobre o jogador, estes dados são obtidos através da ferramenta de simulação que mapeia todas as ações do jogador, criando o conjunto de dados para o treinamento. Entre outros atributos considerados na escolha da ferramenta de simulação, este foi um dos mais importantes e o processo de definição da mesma é abordado com mais detalhes na próxima seção.

## **2.4 FERRAMENTA PARA SIMULAÇÃO**

Esta seção tem como objetivo avaliar a melhor ferramenta para utilização na primeira etapa do experimento deste trabalho, realizando os estudos necessários para a escolha da mesma.

Para a leitura dos dados comportamentais que são utilizados no treinamento da rede é necessária uma ferramenta que permita a leitura das ações realizadas pelo jogador. Neste caso a ferramenta escolhida tem a função de ler os dados do jogador humano, gerar dados para o treinamento e posteriormente servir de interface para os testes.

Para a definição da melhor ferramenta os seguintes tópicos foram abordados:

- Requisitos da ferramenta;
- Ferramentas pesquisadas;
- Comparação entre ferramentas; e
- Ferramenta escolhida.

No decorrer desta seção será feita a descrição do processo de escolha da ferramenta de testes e simulação.

Entre estas definições dos tópicos abordados, alguns detalhes importantes devem ser descritos:

- Requisitos da ferramenta: Nesta primeira etapa foram definidos os requisitos necessários para a realização do experimento, ou seja, quais os atributos que a ferramenta precisa dispor para que o experimento possa ser realizado;
- Ferramentas pesquisadas: Baseado nos requisitos definidos foram pesquisadas ferramentas que se adéquem melhor ao projeto;
- Comparação entre ferramentas: Foi realizado um comparativo entre as ferramentas selecionadas durante a pesquisa com o intuito de verificar as propriedades de cada uma e averiguar qual delas atende melhor aos requisitos do projeto.
- Ferramenta escolhida: Baseado no comparativo realizado foi escolhida a ferramenta que mais se aproxima dos requisitos definidos e a que mais apresenta vantagens em relação às outras ferramentas pesquisadas.

Nas próximas seções cada um dos tópicos definidos será tratado com detalhes, além de uma descrição de cada uma das ferramentas selecionadas durante a pesquisa.

### **2.4.1 Requisitos da ferramenta**

Nesta seção são abordados os requisitos necessários para que a ferramenta de simulação possa ser utilizada.

Primeiramente será realizado o treinamento de uma Rede Neural mapeando os padrões de comportamento humano nas tarefas executadas na ferramenta pelo agente para a movimentação no cenário, ataque, defesa e estratégia de jogo, baseado nos níveis de vida, potência das armas e obstáculos. Portanto, a ferramenta precisa disponibilizar um cenário em que os agentes do jogo possam se movimentar livremente, podendo ou não conter obstáculos. Além disso, a ferramenta precisa proporcionar um arquivo de log para que as ações dos agentes possam ser mapeadas para o treinamento. Outro requisito importante é que seja disponibilizado o estado do agente durante a partida.

A ferramenta tem ainda que permitir que ao menos um agente seja controlado por um usuário a fim de mapear o comportamento humano a ser utilizado para treinamento da rede e o cenário deve atender as definições de um jogo de estratégia.

A partir da implementação do agente baseado na técnica de RNA, serão realizados confrontos entre as diferentes abordagens implementadas nos agentes também na ferramenta, em disputas diretas para se obter os resultados destes confrontos.

Colhidos os resultados dos confrontos para comprovação da qualidade da RNA, será iniciada a segunda etapa do experimento, em que será verificado se a percepção do comportamento mais humano do agente implementado através da RNA é de fato percebida pelos jogadores.

### **2.4.2 Ferramentas Pesquisadas**

A pesquisa foi realizada através de trabalhos da área de IA que abordam assuntos similares e que necessitam de ferramentas com requisitos similares para realizarem seus testes.

O processo de seleção utilizou os seguintes critérios: Todas as ferramentas deveriam ao menos atender as definições do experimento, principalmente em relação ao cenário, arquivos de log para feedback, a opção de inserir novas implementações ao jogo e permitir que o jogador possa interagir com os personagens.

Durante a pesquisa foram selecionadas as ferramentas encontradas que atendiam aos critérios descritos. A partir destas imposições foram selecionadas as seguintes ferramentas:

- Gamebots;
- Jarts;
- Robocode; e
- ORTS.

No caso do ORTS não foram encontradas informações sobre o controle dos personagens nas pesquisas e a ferramenta foi selecionada para testes e identificação da validade do critério.

Nas próximas seções são descritas cada uma das ferramentas selecionadas e após a descrição de cada uma será realizado um comparativo entre elas.

### **2.4.2.1 Gamebots**

O Gamebots é um *framework* que se comunicava com o jogo “Unreal Tournament”, recebe mensagens e percepções como posição e velocidade, vida, nível, munição, visão de oponentes e objetos, entre outros, retornando como resultado as ações do agente (HY, 2004). A ferramenta também gera logs das ações dos jogadores e permite novas implementações de IA aos personagens, além disso, o *framework* provém um ambiente tridimensional no qual os jogadores podem se enfrentar (GAL, 2002).

O *framework* se mostra uma maneira muito efetiva de condução do experimento, pois se baseia em um ambiente de jogo real facilitando a verificação dos resultados de maneira gráfica, proporcionada pelo ambiente.

É importante citar que o Gamebots se comunica com o jogo “Unreal Tournament” por um protocolo de texto utilizando um socket Unix (HY, 2004).

O Gamebots foi selecionado, pois além das citações referenciadas nesta seção também foi citado em outros trabalhos como Iorio (2002), Tong (2011).

### **2.4.2.2 Jarts**

O Jarts é um simulador focado em jogos de estratégia, no qual os utilizadores podem implementar seus personagens se preocupando apenas com seu comportamento. A *engine* da ferramenta se encarrega de fazer a simulação do ambiente (VICENTE FILHO, 2010). O autor ainda cita que o funcionamento do Jarts se assemelha muito ao Robocode e foi inspirado nele, principalmente em relação às funções como movimentação, tiro, mineração, funções estas que permitem ao programador criar comportamentos mais complexos utilizando estas classes de funções.

O Jarts contempla os principais requisitos definidos no projeto, melhorando o aspecto referente ao trabalho com a IA, isso ocorre porque o objetivo do Jarts é justamente implementar melhorias para ferramentas da área (GLEIM, 2002).

O Jarts tem requisitos muito similares aos definidos para seleção da ferramenta, principalmente porque foi criada para beneficiar pesquisados que precisam de uma ferramenta de simulação que facilite a utilização de técnicas de IA em um ambiente de estratégia. Além disso, o Jarts conta com um comparativo entre ferramentas de simulação similares as definidas no projeto e define os benefícios do Jarts sobre as mesmas (ARMBRUSTER, 2007). As ferramentas utilizadas para comparação no trabalho sobre o Jarts são o Robocode, ORTS, Stratagus, Boson e Glest, duas delas também abordadas neste estudo.

### **2.4.2.3 Robocode**

Robocode é um jogo virtual de estratégia baseado em Java idealizado para ensinar técnicas de programação, assim como conceitos de orientação a objeto. A ferramenta consiste em um ambiente de desenvolvimento para robôs em uma arena virtual, em que cada robô batalha por sua vitória (NELSON, 2012).

No ambiente podem ser criados robôs, colocados na arena e realizadas batalhas entre oponentes desenvolvidos com diferentes técnicas. O Robocode é muito utilizado hoje em dia para que desenvolvedores realizem testes em suas implementações, podendo realizar simulações contra as melhores implementações em várias ligas formadas pelo mundo (SILVA, 2007).

O Robocode ainda vem com alguns robôs implementados e também permite construção de personagens através da extensão de classes, além de possibilitar implementações complexas de IA. Hartness (2004) exemplifica alguns algoritmos em seu trabalho demonstrando a utilidade da ferramenta também no uso e ensino de IA.

Os objetos contêm um método principal que fica em *loop* e descreve o comportamento padrão do robô. Além disso, o robô tem outros métodos para invocar em resposta aos eventos que ocorrem durante a simulação da batalha. Estes eventos permitem respostas imediatas ou permitem a gravação dos mesmos para processamento posterior, sendo que as respostas podem incluir comando para verificação da distância em relação ao oponente, ângulo de movimentação, ângulo das armas, potência das armas e etc. Cada robô ainda pode conter informações que possibilitam a criação de uma memória de eventos passados (KENSLE, 2009).

A seleção do Robocode foi feita devido a referências encontradas em vários trabalhos como Woolley e Peterson (2009), Harper (2011), O'Kelly e Gibson (2006), Liu (2008), Bonakdarian e White (2004), além de trabalhos sobre outras ferramentas afins como GameBots, Jars ou Orts que em sua maioria referenciam o Robocode como alternativa.

#### **2.4.2.4 ORTS**

O ORTS (Real-Time Strategy) é uma ferramenta de simulação voltada para o estudo de IA em tempo real e faz parte do gênero de estratégia. O usuário pode definir o cenário e características do ambiente utilizando scripts que descrevem as unidades, estruturas e suas interações. O sistema ainda conta com um sistema cliente servidor para controle dos personagens no ambiente de jogo construído (VICENTE FILHO, 2010).

O ORTS foi escolhido por ser citado em diversos trabalhos como Huang (2009), Navarro e Corruble (2009), Hagelbäck e Johansson (2008). Inclusive uma pesquisa sobre a ferramenta também foi realizada no trabalho sobre a ferramenta Jarts.

### **2.4.3 Comparativo entre as ferramentas**

Esta seção apresenta um comparativo entre as ferramentas definidas. Para uma melhor abordagem dos dados colhidos sobre este comparativo foi criada a Tabela 2 com dados de cada uma das ferramentas. Os principais requisitos definidos para análise foram descritos na Subseção 2.4.1 .

Tabela 2. Comparativo entre ferramentas de simulação.

	<b>Gamebots</b>	<b>Jarts</b>	<b>Robocode</b>	<b>ORPS</b>
CENÁRIO DE ESTRATÉGIA	SIM	SIM	SIM	SIM
LOG COM DADOS DO JOGO E PERSONAGEM	SIM	SIM	SIM	SIM
PERMITIR CONTROLE DO AGENTE	SIM (NÃO UTILIZA FRAMEWORK)	SIM	SIM	NÃO
PERMITIR IMPLEMENTAÇÃO DE AGENTES NO JOGO	SIM (APENAS PELO PROTOCOLO DE COMUNICAÇÃO)	SIM	SIM	SIM

No comparativo é possível perceber que o Gamebots, Jarts e Robocode tem mais pontos positivos do que o ORPS.

Em comparação direta com o Gamebots, o Robocode e Jarts se mostram melhores para o desenvolvimento da IA, pois no Gamebots o personagem se comunica por um protocolo de rede, ou seja, a IA desenvolvida para os personagens no Gamebots não atua diretamente na ferramenta, apenas executa os comandos pelo protocolo, tornando o nível de detalhes menor. Além disso, no Gamebots não é possível verificar como os comandos são trabalhados dentro do jogo, pois o nível de visualização se limita ao disponibilizado pela ferramenta.

Com isso a escolha foi definida entre o Robocode e Jarts. Analisando os detalhes sobre as ferramentas, o Robocode apresenta um conjunto melhor de benefícios para construção do experimento, um deles é a integração com outras IDE's Java de desenvolvimento, como o NetBeans e o Eclipse (NELSON, 2011). Além disso, durante as pesquisas foi identificado um framework Java ligado a IDE Netbeans para treinamento de redes neurais. Com isso o desenvolvimento pôde ser centralizado melhorando o processo de criação do agente a partir do Robocode.

GADE *et al* (2003) ainda descrevem outro ponto forte do Robocode relativo a grande comunidade de desenvolvedores ativa, devido a difusão da ferramenta disponibilizada desde 2001.

Outro fator levado em consideração, foi que segundo Vicente Filho *et al* (2010) o trabalho sobre a ferramenta Jarts utilizou o Robocode como referência no projeto, demonstrando a qualidade da ferramenta frente a uma análise científica de trabalhos similares.

#### **2.4.4 ROBOCODE**

De acordo com o comparativo realizado na seção anterior foi possível perceber que o Robocode apresenta valores que correspondem ao que se busca na ferramenta de simulação. Os requisitos citados na Subseção 2.4.1 foram todos atendidos e o Robocode demonstrou ser uma ferramenta prestigiada e consolidada.

Além de atender aos requisitos citados, outro ponto forte do Robocode fica a cargo da integração com outras IDE's de desenvolvimento, o que permite a diminuição de trabalho durante as implementações. O Robocode é mantido pela IBM e no site da empresa podem ser encontrados desde tutoriais com passos para iniciantes, até guias avançados de implementações utilizando técnicas mais sofisticadas de IA. A própria difusão da ferramenta citada anteriormente, faz com que diversos artigos e trabalhos na área de jogos e IA disponibilizem documentos e procedimentos sobre a ferramenta.

Outro fator importante é em relação ao foco do próprio Robocode, que é o ensino da linguagem de programação Java e justamente por ser voltando ao ensino, a implementação apresenta muitas facilidades para construção dos personagens.

Na Figura 10 pode ser visualizada a imagem do ambiente padrão do Robocode, no qual também podem ser vistos diversos personagens.

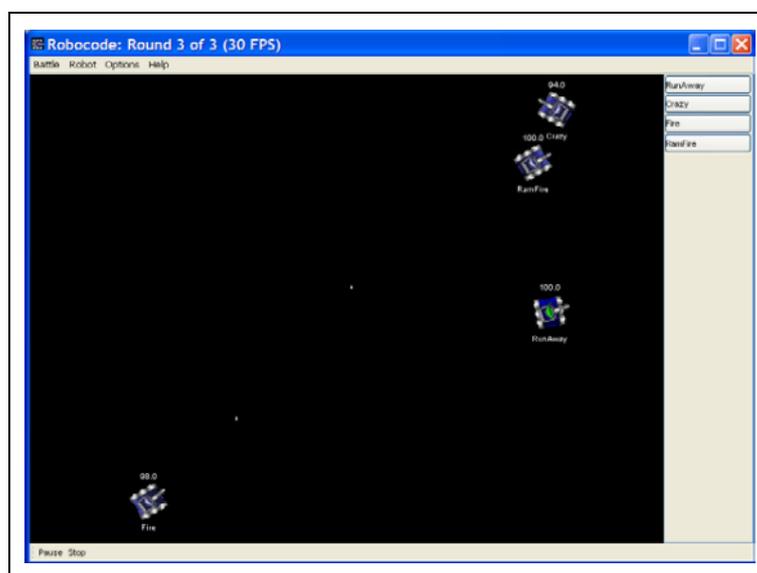


Figura 10. Ambiente do Robocode e personagens.

Fonte: Hartness (2004).

A Figura 11 apresenta a IDE de implementação do Robocode, esta IDE permite que o código possa ser trabalhado dentro da própria ferramenta.

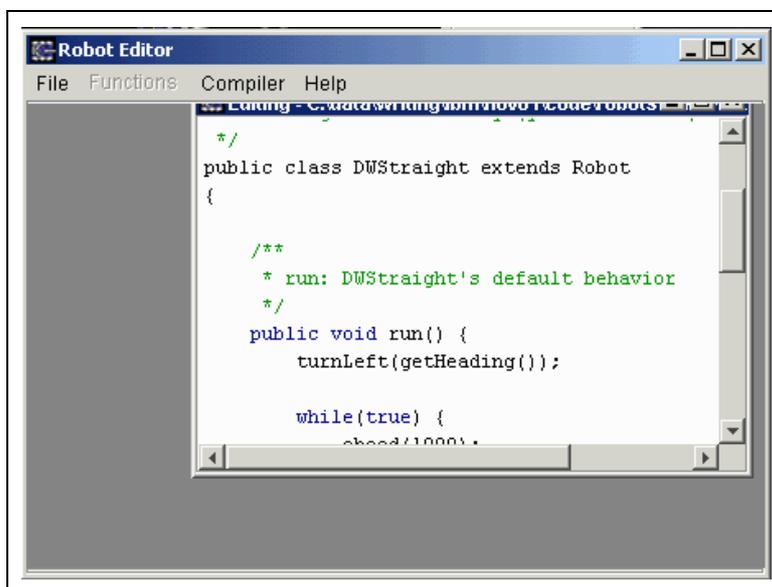


Figura 11. IDE de implementação do Robocode.

Fonte: Nelson (2011)

As Figura 10 e Figura 11 apresentam os dois principais ambientes do Robocode, sendo que, além disso, a ferramenta dispõe de menus para criação do jogo, ajustes das partidas e permite a exibição de detalhes das batalhas para auxiliar na construção dos personagens.

### 3 TRABALHOS RELACIONADOS

Para uma pesquisa mais apurada de trabalhos relacionados foi realizada uma revisão sistemática da literatura.

Revisões sistemáticas da literatura constituem uma prática que se baseia em evidências e combinam os resultados de vários estudos afins. Este processo garante maior qualidade e credibilidade aos estudos, devido ao fato de que a quantidade de pesquisas desenvolvidas que devem ser consideradas está em constante expansão (DYBÅ; DINGSØYR, 2008).

A estrutura do processo de revisão é formada pela descrição do protocolo de revisão sistemática na Seção 3.2, seguido da análise dos trabalhos retornados para cada pergunta definida no protocolo de revisão na Seção 3.3 e das considerações na Seção 3.4.

Os termos para seleção dos trabalhos definidos no critério de inclusão tiveram como objetivo identificar os estudos que realizaram o mapeamento do comportamento humano. Os termos mais abrangentes foram utilizados com o intuito de obter o máximo de pesquisas como o mesmo tema.

Apesar disso existem diferenças entre as técnicas utilizadas, categorias de jogos e até mesmo as ações que se deseja mapear dos jogadores. Por exemplo, foram identificados ambientes de estratégia, FPS, Luta e corrida, em que diferentes ações foram mapeadas nestes trabalhos e o ambiente tem muita influência nos detalhes do mapeamento do comportamento do jogador.

Por isso foram extraídos os detalhes de cada pesquisa comparando ambientes e técnicas utilizadas a fim de identificar as pesquisas com conceitos e metodologias com mais detalhes em comum com este trabalho.

Também foram considerados todos os trabalhos que realizam o mapeamento de comportamento humano na revisão sistemática e nas considerações foram destacados os trabalhos com técnicas e ambientes mais similares.

As próximas seções apresentam as etapas da revisão sistemática citadas com mais detalhes, iniciando pelo protocolo de revisão.

### **3.1 OBJETIVO PRINCIPAL DA REVISÃO SISTEMÁTICA**

Esta revisão teve por objetivo identificar pesquisas focadas no mapeamento do comportamento humano em agentes, para área de jogos, através do uso de técnicas de Inteligência Artificial que apresentem a metodologia e testes realizados.

A busca por estas pesquisas seguem as definições apresentadas por Kitchenham e Charters (2007) no guia de revisão sistemática da literatura. Com isso é possível identificar pesquisas afins, além dos métodos, técnicas e medições utilizadas pelos pesquisadores em cada uma delas.

### **3.2 PROTOCOLO DE REVISÃO SISTEMÁTICA**

Conforme o guia de Kitchenham e Charters (2007), foi definido o protocolo de revisão sistemática se baseando em 5 princípios básicos conforme o método (PICOC): população, intervenção, comparação, resultados e contexto. Estes princípios são adotados para estruturar as questões de pesquisa.

Detalhes dos tópicos definidos no protocolo de busca são apresentados nas próximas seções, assim como seus resultados.

#### **3.2.1 Objetivo**

Identificar pesquisas focadas no mapeamento do comportamento humano em agentes, para área de jogos, através do uso de técnicas de Inteligência Artificial.

#### **3.2.2 Questões de pesquisa**

Conforme os princípios do método PICOC utilizados, foram definidos os seguintes critérios para contextualização das questões de pesquisa:

- População: pesquisadores da área de jogos (jogos)
- Intervenção: mapeamento do comportamento através da interação do jogador humano (de forma similar ao ser humano, nível humano, jogador humano, pensamento humano, interação, comportamento)

- Comparação: aprendizado por imitação de comportamento (imitação, aprendizado)
- Resultados: principais pesquisas
- Contexto: técnicas de IA (inteligência artificial, agentes inteligentes)

Os resultados das buscas tem como foco pesquisas que apresentem o mapeamento do comportamento humano e a descrição do que se busca nos trabalhos está compreendida na intervenção e comparação, portanto os resultados não foram adicionados as *tags* de busca.

Esta revisão sistemática tem uma questão principal de pesquisa definida, além de outras questões secundárias, todas voltadas a identificar detalhes para comparação com a pesquisa realizada neste trabalho. A questão primária definida neste estudo, Q1 e as demais questões secundárias são demonstradas a seguir:

Q1: Quais os principais trabalhos ligados ao mapeamento do comportamento humano através de técnicas de Inteligência Artificial na área de jogos?

Q2: O mapeamento ocorre de forma geral (para todas as ações do personagem) ou apenas para um aspecto específico de ação (movimentação, seleção de objetos) do personagem?

Q3: Quais as métodos (classificação de padrões, análise de imagens, ferramentas de log de ações do personagem) utilizados para o mapeamento dos padrões comportamentais do jogador nestas pesquisas?

Q4: Quais as principais técnicas de IA (Redes Neurais, Redes Bayesianas, Lógica Fuzzy) utilizadas para o mapeamento de padrões comportamentais?

Q5: Quais os jogos e suas respectivas categorias utilizados nestas pesquisas.

Q6: Qual a taxa de sucesso (atingiram resultados esperados) dos experimentos pesquisados e método de avaliação dos resultados (pesquisas com a percepção do usuário, análise de imagens, observação)?

### 3.2.3 Fontes de dados

Foram utilizadas fontes de bases de dados eletrônicas bem conceituadas na comunidade científica.

As fontes selecionadas oferecem facilidade na recuperação dos textos completos dos artigos e o fato destas fontes oferecerem acesso a publicações significativas na área de IA para jogos também contribuiu para suas escolhas. Na Tabela 3 podem ser visualizadas as fontes escolhidas.

Tabela 3. Fontes de Dados.

<b>Nome da Fonte</b>	<b>Link de Acesso</b>
IEEEExplore	<a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a>
ACM Digital library	<a href="http://portal.acm.org">http://portal.acm.org</a>
Google Scholar	<a href="http://scholar.google.com">http://scholar.google.com</a>
Springer	<a href="http://www.springerlink.com">http://www.springerlink.com</a>
ScienceDirect	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>
CiteSeerX	<a href="http://citeseer.ist.psu.edu/">http://citeseer.ist.psu.edu/</a>
Scopus	<a href="http://www.scopus.com">http://www.scopus.com</a>

### 3.2.4 Critérios de inclusão e exclusão

Os critérios de inclusão e exclusão definidos são os seguintes:

- Estudos que apresentem pesquisa sobre o mapeamento do comportamento humano em personagens para a área de jogos;
- Estudos que apresentem o processo utilizado no mapeamento das ações do jogador humano nos personagens do jogo;
- Documentos publicados a partir de 01/2000; e
- Artigos não podem ser curtos (resumos expandidos) .

No critério de inclusão foi definida a necessidade de um segundo avaliador para verificar se os critérios foram abordados de maneira coerente.

### 3.2.5 Strings de busca

Utilizando os conceitos do método PICOC foi definida a seguinte expressão base para busca:

- *(games) AND (human like OR human level OR human players OR human thinking OR interactive OR behavior) AND (imitation learning OR imitation OR imitative OR learning) AND (AI OR artificial intelligence OR intelligent agent OR artificial agent)*

A *string* de busca definida é utilizada como base para todas as pesquisas, sendo que devido a máquina de busca de cada base são necessárias algumas modificações para que o objetivo da pesquisa seja alcançado, evitando a perda de documentos relevantes.

Assim como em Abrantes e Travassos (2007), foram levados em consideração os seguintes critérios para modificação das strings nas diferentes bases de busca:

- Manter a *string* logicamente equivalente a string original, ou
- Quando não for possível de manter a equivalência, a string derivada deve ser mais abrangente para evitar a perda de documentos relevantes.

As *strings* definidas em cada base foram comparadas a string inicial para garantir que nenhum trabalho relevante fosse deixado de fora e foram ajustadas até reproduzir um resultado satisfatório

Além disso, os trabalhos mais relevantes encontrados nas leituras iniciais foram utilizados como validação para ajustes das *strings* em cada uma das bases

Também foram adicionadas as bases a limitação por artigos acima do ano 2000, que faz parte de um dos critérios de exclusão.

As *strings* definidas em cada base, são apresentadas no decorrer desta seção.

**Google Scholar:** A *string* foi adaptada, pois a maneira como o Google realiza a busca não respeita totalmente os operadores lógicos, trazendo muitas inconsistências. No Google Scholar termos como “human” já trazem todas as alternativas para o termo, o que em outras bases devem ser especificadas. Além disso, foi identificado que os termos deveriam ser mais abrangentes para

que não fossem deixados de fora trabalhos relevantes. A *string* final utilizada para o Google Scholar pode ser vista na sequência:

- *(games) AND (humanlike OR interactive) AND ('imitation learning' OR behavior OR imitative OR imitation OR learning) AND (AI or 'artificial intelligence' OR agents)*

**IEEE:** Foram realizados ajustes apenas para descartar termos muito abrangentes. A *string* final utilizada para a IEEE pode ser vista na sequência:

- *("Abstract":games) AND (p\_Abstract:behavior OR humanlike OR "human-like" OR "human-level" OR "human players" OR "human thinking") AND (p\_Abstract:"imitation learning" OR imitative OR imitation OR learning) AND (p\_Abstract:AI or "artificial intelligence" OR "intelligent agent" OR "artificial agent")*

**ACM:** Assim como na base da IEEE, foi ajustada apenas para descartar termos muito abrangentes. A *string* final utilizada para a ACM pode ser vista na sequência:

- *(Abstract:(games) AND Abstract:(behavior OR humanlike OR "human-like" OR "human-level" OR "human players" OR "human thinking") AND Abstract:("imitation learning" OR imitative OR imitation OR learning) AND Abstract:(AI or "artificial intelligence" OR "artificial agent" OR "intelligent agent")) and (not Abstract:emotional) and (PublishedAs:journal OR PublishedAs:proceeding OR PublishedAs:transaction OR PublishedAs:magazine OR PublishedAs:newsletter) and (AbstractFlag:yes)*

**Springer:** Foi ajustada para termos mais genéricos a fim de não deixar trabalhos relevantes de fora. A *string* final utilizada para a Springer pode ser vista na sequência:

- *ab:((games) AND (human OR behavior) AND ("imitation learning" OR imitative OR imitation OR learning) AND (AI or "artificial intelligence" OR agents)) 'published between '1 Jan 2000' and '4 Jul 2012'*

**ScienceDirect:** Assim como como na IEEE e ACM, a *string* foi ajustada apenas para descartar termos muito abrangentes. Um dos operadores foi modificado, para que não fossem deixados de fora trabalhos relevantes. A *string* final utilizada para a ScienceDirect pode ser vista na sequência:

- *pub-date > 1999 and tak(games) AND (humanlike OR "human-like" OR "human-level" OR "human players" OR "human thinking") AND tak("imitation learning" OR behavior OR imitative OR learning) AND (AI OR "artificial intelligence" OR agents)*

**CiteSeer:** Assim como na Springer a *string* foi ajustada para termos mais genéricos a fim de não deixar trabalhos relevantes de fora. A *string* final utilizada para a CiteSeer pode ser vista na sequência:

- *(games) AND (humanlike OR interactive) AND ('imitation learning' OR behavior OR imitative OR learning) AND (AI or 'artificial intelligence')*

**Scopus:** Assim como a IEEE e ACM a *string* foi ajustada apenas para descartar termos muito abrangentes. A *string* final utilizada para a Scopus pode ser vista na sequência:

- *ABS((games) AND (human-like OR human-level OR "human players" OR "human thinking" OR behavior) AND ("imitation learning" OR imitative OR learning) AND ("AI" OR "artificial intelligence" OR "intelligent agent" OR "artificial agent")) PUBYEAR > 1999 AND (LIMIT-TO(SUBJAREA, "COMP"))*

### 3.2.6 Seleção dos estudos

O processo de seleção foi realizado inicialmente pela análise do resumo, em que foram descartados os artigos não relevantes (de acordo com os critérios de inclusão e exclusão). Após este primeiro filtro foi realizada a leitura completa da pesquisa e a seleção dos estudos mais relevantes foi finalizada.

Todo o processo de seleção levou em conta os critérios de inclusão e exclusão definidos para determinar a seleção ou não dos estudos.

Na revisão sistemática a avaliação por um segundo revisor é importante para garantir que os critérios estão definidos de forma coerente com os estudos procurados, sendo que não é possível que um segundo revisor realize a avaliação de cada um dos artigos selecionados. Por isso um segundo avaliador realizou uma avaliação de confiança destes critérios.

O segundo avaliador avaliou 5 dos estudos selecionados de maneira aleatória e executou sua avaliação baseada nos critérios de inclusão e exclusão do protocolo de revisão sistemática. Os

resultados do primeiro e deste segundo avaliador foram comparados e não foram encontradas diferenças.

A análise dos resultados obtidos na etapa de seleção é tratada na Seção 3.3.

### 3.2.7 Extração dos dados

O processo de extração dos dados foi realizado através da avaliação de cada um dos artigos selecionados. Estes dados foram tratados através de uma planilha do Microsoft Excel, em que os aspectos relativos as perguntas de pesquisa foram avaliados e os dados foram classificados de acordo com o que foi definido no protocolo de revisão sistemática.

Os dados definidos para extração das informações e sua relação com questões de pesquisa são apresentados na Tabela 4.

Tabela 4. Extração dos dados para cada questão.

Dados extraídos	Descrição	Questões
Dados Bibliográficos	Autores, ano de publicação, título, fonte	Q1
Foco do estudo	Descrição de objetivos dos estudos	Q1
Abrangência	Quais as ações dos personagens do jogo são mapeadas	Q2
Metodologia Utilizada	Método de mapeamento das ações do jogador, técnica de IA utilizada, nome e categoria do jogo	Q3, Q4, Q5
Resultados	Resultados obtidos e métodos utilizados para análise	Q6

### 3.2.8 Síntese dos dados

Para que os dados extraídos permitam identificar as informações buscadas através das perguntas de pesquisa os mesmos devem ser mapeados e a apresentação destas informações deve ser realizada de uma maneira que seus resultados possam ser claramente identificados.

Os critérios para análise dos dados relativos às questões da avaliação qualitativa definidos no protocolo de revisão sistemática são os seguintes:

- Referências dos estudos selecionados;
- Total de artigos selecionados por base;
- Total de artigos publicados por ano;

- Objetivo dos estudos selecionados;
- Verificar se o mapeamento ocorre para todas as ações do personagem;
- Verificar os métodos utilizados para mapeamento das ações do jogador;
- Dividir entre as técnicas de IA abordadas;
- Dividir entre os jogos utilizados e respectivas categorias; e
- Verificar se os resultados esperados foram alcançados e os métodos utilizados para avaliação dos mesmos.

Os dados extraídos foram apresentados de acordo com os critérios definidos para a análise de dados no protocolo de revisão sistemática.

Os dados extraídos a partir destes critérios são demonstrados na Seção 3.3.

### **3.3 ANÁLISE COMPARATIVA**

Nesta seção são apresentados os resultados das seleções e análise dos resultados da revisão sistemática. Esta seção utiliza os dados extraídos dos estudos, conforme definido no protocolo de revisão, para responder as perguntas definidas na revisão sistemática sobre os trabalhos similares.

#### **3.3.1 Estudos Selecionados**

Após a execução das busca nas bases definidas, foram obtidos um total de 661 estudos potencialmente relevantes. As bases que apresentaram maior número de estudos e as únicas que retornaram resultados maiores que 100 registros, foram a Google Scholar com 214 estudos, a Springer com 125 estudos e a Scopus com 102 estudos.

A seleção realizada no Google Scholar trouxe mais de 3000 resultados, sabendo que o motor de busca do Google operara de uma forma diferenciada das outras bases, foi utilizada uma técnica para definir somente os arquivos relevantes.

A técnica consiste em ordenar os artigos por relevância, realizando a análise do abstract e conclusão dos trabalhos retornados. Foi realizada então a leitura de todos os estudos e marcados os

mais relevantes. Quando o último artigo relevante era encontrado era iniciada uma contagem dos artigos subsequentes. Quando a busca nos 50 artigos subsequentes não apresentou resultados relevantes, baseados nos critérios definidos para inclusão, a região de corte foi obtida e foram definidos 214 artigos para seleção na base do Google Scholar.

Analisando os resultados da seleção de forma geral, algumas questões importantes foram observadas (Q1).

Um dos fatos que chamou a atenção foi o retorno de artigos repetidos entre as bases (considerando somente selecionados) em um número elevado, 17 artigos, representando 50% do total selecionado, que apesar de comprovar que a configuração da *string* de busca entre as bases apresentou o funcionamento esperado, também demonstrou que algumas bases que trouxeram estudos repetidos não fariam diferença se descartadas.

O total de artigos retornados e selecionados considerando as repetições entre bases é apresentado na Tabela 5.

Tabela 5. Seleção por base.

Nome da Fonte	Quantidade retornada	Selecionados
IEEEExplore	27	1
ACM Digital library	48	6
Google Scholar	214	12
Springer	125	3
ScienceDirect	84	1
CiteSeerX	61	4
Scopus	102	7
Totais	661	34 (5,14%)*

\*Entre os selecionados, foram identificados 17 artigos repetidos entre as bases.

Com estes resultados, dos 661 estudos analisados foram selecionados 17 estudos baseados nos critérios de inclusão e exclusão definidos, totalizando 5,14% do total retornado. Na Tabela 6 são apresentados os 17 estudos selecionados.

Tabela 6. Referências dos estudos selecionados por base.

Autores	Ano	Nome do trabalho	Bases
Gorman <i>et al</i>	2006	Bayesian Imitation of Human Behavior in Interactive Computer Games	IEEE, ACM, Scopus, Google Scholar
Bryant e Miikkulainen	2007	Acquiring Visibly Intelligent Behavior with Example-Guided Neuroevolution	ACM, Scopus

Gorman <i>et al</i>	2006	Believability Testing and Bayesian Imitation in Interactive Computer Games	ACM, Springer, CiteSeer, Scopus, Google Scholar
Zanetti e Rhalibi	2004	Machine Learning Techniques for FPS in Q3	ACM, Scopus
Ki <i>et al</i>	2006	Real-Time neuroevolution to imitate a game player	ACM, Springer
van Hoorn <i>et al</i>	2009	Robust player imitation using multiobjective evolution	ACM, Scopus, Google Scholar
Gorman e Humphrys	2006	Towards integrated imitation of strategic planning and motion modeling in interactive computer games	ACM, CiteSeer, Scopus, Google Scholar
Bauckhage Thureau e Sagerer	2003	Learning Human-like Opponent Behavior for Interactive Computer Games	Springer, CiteSeer
Hy <i>et al</i>	2004	Teaching Bayesian behaviours to video game characters	ScienceDirect, Google Scholar
Thureau, Paczian e Bauckhage	2005	Is Bayesian Imitation Learning the Route to Believable Gamebots	CiteSeer, Google Scholar
Munoz <i>et al</i>	2009	Controller for TORCS created by imitation	Google Scholar
Gorman e Humphrys	2007	Imitative Learning Of Combat Behaviours In First-Person Computer Games	Google Scholar
Bauckhage <i>et al</i>	2007	Learning Human Behavior from Analyzing Activities in Virtual Environments	Google Scholar
Thureau e Bauckhage	2004	Learning human-like Movement Behavior for Computer Games	Google Scholar
Schrum Karpov e Miikkulainen	2011	UT2 Human-like Behavior via Neuroevolution of Combat Behavior and Replay of Human Traces	Google Scholar
Mozgovoy e Umarov	2010	Building a Believable and Effective Agent for a 3D Boxing Simulation Game	Scopus, Google Scholar
Obradovic e Stahl	2008	Learning by Observing Case-Based Decision Making in Complex Strategy Games	Google Scholar

Entre as bases utilizadas a ACM teve a maior taxa de estudos relevantes com 12,5% dos 84 estudos retornados selecionados, conforme pode ser visto na Tabela 7.

Tabela 7. Estudos selecionados por base.

Base	Totais retornados	Estudos selecionados
IEEEExplore	27	3,70%
ACM Digital library	48	12,5%
Google Scholar	214	5,61%
Springer	125	2,4%
ScienceDirect	84	1,19%

CiteSeerX	61	6,55%
Scopus	102	6,86%
Totais	661	5,14%

Algumas bases que apresentaram muitos estudos potenciais como a Springer e ScienceDirect, apresentaram poucos artigos na seleção final, pois mesmo contendo artigos na área de imitação do comportamento humano em agentes, os detalhes relativos ao que era mapeado e como o processo era realizado acabavam distanciando os estudos dos critérios de inclusão e exclusão definido para seleção dos estudos.

Observando a proporção entre os estudos retornados e efetivamente selecionados, a Springer apresentou uma taxa 2,4% do total de resultados retornados e a ScienceDirect uma taxa 1,19% do seu total retornado.

Todas as bases tiveram estudos selecionados, sendo que devido aos resultados duplicados as bases com poucos resultados como a Springer (3 estudos) e ScienceDirect (1 estudo) se tornaram menos relevantes na projeção final.

Mais informações sobre os dados coletados podem ser vistos nas Figura 12 e Figura 13, respectivamente a quantidade de trabalhos obtidos e selecionados por base e a proporção de artigos entre o total de selecionados por base.

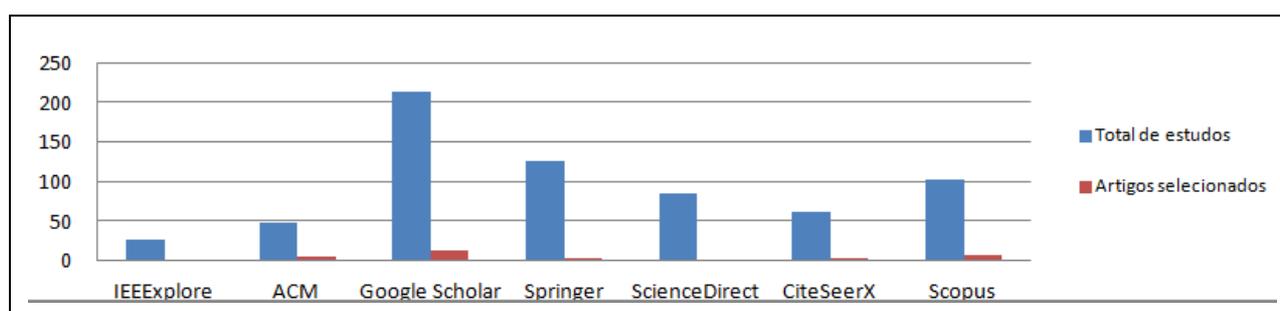


Figura 12. Quantidade de trabalhos obtidos por base de dados.

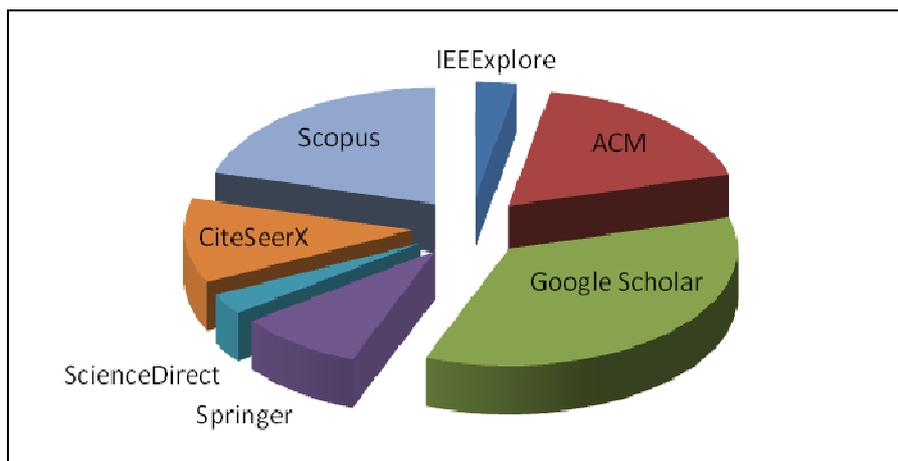


Figura 13. Proporção de artigos entre o total de selecionados por base.

Além da descrição de artigos selecionados por base, a revisão visa identificar a proporção de publicação por ano, a fim de determinar a frequência de publicação dos estudos mais relevantes selecionados(Q1). Os dados de publicação por ano demonstram que 70,59% dos estudos selecionados foram realizados nos últimos 6 anos. Apesar de as buscas terem considerado artigos a partir do ano 2000 somente 5 estudos estão com mais de 6 anos de publicação, demonstrando que estudos de qualidade na área do mapeamento de comportamento humano estão em crescimento. Os dados com a quantidade de artigos retornados, divididos por ano, são apresentados na Tabela 8.

Tabela 8. Estudos selecionados por ano.

Ano de Publicação	Ocorrências
2003	1
2004	3
2005	1
2006	4
2007	3
2008	1
2009	2
2010	1
2011	1
Total	17

### 3.3.2 Foco dos estudos

O foco dos estudos selecionados, baseados nas descrições de seus objetivos, demonstra que os critérios definidos nesta revisão sistemática estão sendo atendidos (Q1). Na sequência são descritos os trabalhos e seus respectivos objetivos:

- **Bayesian Imitation of Human Behavior in Interactive Computer Games (2006):** Este trabalho é situado no campo da imitação do comportamento, mais especificamente apresentando uma abordagem em Redes Bayesianas para extrair os padrões de movimento e estratégias de um jogador humano, fazendo com que um agente possa realizar suas operações de maneira similar ao jogador.
- **Acquiring Visibly Intelligent Behavior with Example-Guided Neuroevolution (2007):** Este trabalho apresenta um mecanismo para utilizar exemplos gerados por um controlador humano para utilização em uma Rede Neural evolutiva, visando gerar políticas de comportamento inteligente. O método é avaliado em um jogo de estratégia no qual a abordagem pode ser medida quantitativamente.
- **Believability Testing and Bayesian Imitation in Interactive Computer Games (2006):** É realizado um estudo que visa mapear o comportamento do jogador através de Redes Bayesianas.
- **Machine Learning Techniques for FPS in Q3 (2004):** Este trabalho apresenta um jogo de tiro em primeira pessoa (FPS) baseado em Inteligência Artificial que faz uso de técnicas de aprendizado para obter o comportamento e estratégias similares ao do um jogador humano.
- **Real-Time neuroevolution to imitate a game player (2006):** Neste trabalho é apresentado um algoritmo baseado em redes neurais para imitar a maneira de jogar de um jogador humano.
- **Robust player imitation using multiobjective evolution (2009):** O objetivo do trabalho é utilizar a técnica de algoritmos evolucionários multiobjetivos para imitar o comportamento do jogador humano no ambiente de jogo.
- **Towards integrated imitation of strategic planning and motion modeling in interactive computer games (2006):** Neste artigo é descrito o trabalho de imitação do comportamento de jogadores humanos orientados a seus objetivos.
- **Learning Human-like Opponent Behavior for Interactive Computer Games (2003):** Neste trabalho é investigado se as ações do oponente podem ser tratadas como parâmetros para o aprendizado de seu comportamento no jogo.

- **Teaching Bayesian behaviours to video game characters (2004):** Este trabalho explora a aplicação de Redes Bayesianas para programar o comportamento de personagens de jogos através da imitação do jogador humano.
- **Is Bayesian Imitation Learning the Route to Believable Gamebots (2005):** Neste trabalho é investigado se o uso de Redes Bayesianas no aprendizado por imitação torna o comportamento dos personagens do jogo mais realistas.
- **Controller for TORCS created by imitation (2009):** Um dos objetivos deste trabalho é a criação de um agente controlador para o jogo TORCS imitando os jogadores humanos.
- **Imitative Learning Of Combat Behaviours In First-Person Computer Games (2007):** Neste trabalho, na área do aprendizado por imitação, o objetivo foi investigar a melhor maneira de utilizar dados obtidos das sessões de jogo, realizadas por jogadores humanos, para criar agentes inteligentes convincentes.
- **Learning Human Behavior from Analyzing Activities in Virtual Environments (2007):** Este trabalho realiza o aprendizado por imitação das ações dos jogadores humanos e realiza o teste de percepção da credibilidade dos agentes treinados.
- **Learning human-like Movement Behavior for Computer Games (2004):** Este trabalho propõe um método de aprendizado das estratégias dos jogadores humanos observando suas ações para mapeamento em agentes inteligentes, visto como uma questão de reconhecimento de padrões.
- **UT2 Human-like Behavior via Neuroevolution of Combat Behavior and Replay of Human Traces (2011):** A partir do bot com 27,2727% de características humanas (de acordo com o resultado do BotPrize 2010), este trabalho visa aperfeiçoar, através da imitação do comportamento dos jogadores, estas características humanas
- **Building a Believable and Effective Agent for a 3D Boxing Simulation Game (2010):** Este trabalho descreve uma abordagem usada para construir e aperfeiçoar uma solução de AI para um simulador de Boxe 3D, tentando mapear um comportamento mais próximo do humano no agente do jogo.

- **Learning by Observing Case-Based Decision (2008):** Neste trabalho é apresentada uma abordagem para desenvolver agentes inteligentes em um jogo de estratégia através da observação do jogador humano, aprendendo com suas decisões.

Apesar dos trabalhos terem suas especificidades, em geral todos tratam do mapeamento do comportamento humano em agentes do jogo, sendo que a categoria dos jogos, os métodos e a análise de resultados seguem abordagens diferenciadas para cada situação.

Também são percebidas pequenas diferenças na descrição dos objetivos em relação ao mapeamento do comportamento por imitação que em outros trabalhos são descritos como mapeamento por observação, sendo que para ambas as definições são realizados os mapeamentos das ações dos jogadores e utilizadas posteriormente para treinamento e implementação nos agentes.

### 3.3.3 Abrangência

Em cada estudo são tratados diferentes aspectos dos personagens e nem sempre são mapeadas todas as ações destes agentes no jogo. Isto ocorre devido ao crescimento da complexidade de acordo com o número de ações que o personagem do jogo executa.

Entre os trabalhos selecionados 52,94% tratam da imitação do comportamento relativo a todos os aspectos dos personagens, conforme demonstra a Tabela 9 (Q2).

Tabela 9. Ações Mapeadas.

<b>Ação Mapeada</b>	<b>Totais retornados</b>
Movimentação	2 (11,76%)
Movimentação e estratégia	2 (11,76%)
Movimentação e Mira	2 (11,76%)
Mira	1 (5,88%)
Ações de decisão (Mapeamento criado pelos autores)	1 (5,88%)
Todas as ações dos personagens	9 (52,94%)

Entre estes trabalhos pode-se observar que apenas 3 trabalhos tratam de aspectos únicos do ambiente do jogo ao que foram mapeados, 2 mapeando apenas a movimentação e 1 apenas a mira do jogador.

Estes dados se tornam mais representativos quando a categoria dos jogos é conhecida, pois as diferenças se tornam importantes para que se possa observar as ações mapeadas de acordo com sua complexidade. Na Tabela 10 podem ser vistas as ações mapeadas divididas por categoria.

Tabela 10. Ações Mapeadas por categoria de jogos.

<b>Ação Mapeada</b>	<b>Categoria</b>	<b>Totais retornados</b>
Movimentação	FPS	2
Movimentação e estratégia	FPS	2
Movimentação e Mira	FPS	1
Movimentação e Mira	Estratégia	1
Mira	FPS	1
Ações de decisão (Mapeamento criado pelos autores)	Estratégia	1
Todas as ações dos personagens	FPS	5
Todas as ações dos personagens	Corrida	2
Todas as ações dos personagens	Estratégia	1
Todas as ações dos personagens	Luta	1

Como os jogos do tipo FPS são mais representativos, sendo utilizados em 11 dos 17 estudos selecionados, os dados das ações mapeadas nesta categoria de jogo nos permitem uma análise mais apurada. Neste caso é possível observar que dos 11 jogos, cerca de 45,45% mapearam todas as ações dos personagens. Isto demonstra que mesmo que a complexidade em mapear todas as ações dos personagens seja maior, quase metade dos estudos utiliza esta abordagem. Neste caso o foco de cada pesquisa e as escolhas dos pesquisadores parece determinar o tipo de abordagem.

### 3.3.4 Metodologia Utilizada

O método de mapeamento utilizado é importante para avaliar a qualidade da massa de dados utilizada na construção do personagem mapeado com características do jogador humano (Q3).

Como pode ser visto na Tabela 11 em todas as pesquisas são utilizados logs com ações do jogador, sendo que apenas a ferramenta utilizada varia, pois depende do jogo utilizado.

Tabela 11. Método de Mapeamento das ações do jogador.

<b>Ferramenta de mapeamento</b>	<b>Método de mapeamento</b>	<b>Totais</b>
Através do próprio jogo	Através de logs com ações do jogador	4
Ferramenta cliente servidor	Através de logs com ações do jogador, obtidas pela rede	11*
Ferramenta Middleware para o jogo	Através de logs com ações do jogador	1

Ferramenta para filtrar somente decisões	Através de logs com ações do jogador (realizada programação para filtrar 'decisões')	1
--	--	---

*\*Apesar de se tratarem dos mesmos jogos, a ferramenta exata de mapeamento não foi especificada nos artigos, apenas os logs foram citados.*

Os logs de ação permitem que todas as respostas dos jogadores possam ser lidas e depois tratadas para aplicação aos personagens através da utilização de técnicas de IA.

Outro dado importante em relação aos métodos utilizados em cada pesquisa são as técnicas de Inteligência Artificial utilizadas em cada um dos estudos para mapear as ações obtidas pelo jogador nos personagens do jogo (Q4).

Na Tabela 12 podem ser observadas as técnicas utilizadas em cada estudo selecionado. A técnica identificada como a mais utilizada com 35,29% dos estudos selecionados é a de Redes Bayesianas, sendo que os trabalhos utilizando esta técnica de IA foram identificados como sendo pesquisas do mesmo núcleo de pesquisadores.

Tabela 12. Técnicas de IA utilizadas.

<b>Técnica Utilizada</b>	<b>Totais retornados</b>
Redes Bayesianas	6 (35,29%)
Redes Neurais e Algoritmos Genéticos	5 (29,41%)
Lógica Fuzzy	1 (5,88%)
Redes Neurais	3 (17,65%)
Raciocínio baseado em casos	2 (11,76%)

As pesquisas que utilizam redes neurais de forma híbrida ou isoladamente, somam 47,06% das pesquisas selecionadas, demonstrando que a técnica ganha destaque neste tipo de estudo.

Outro fator de escolha importante extraído das pesquisas selecionadas é relativo ao jogo e respectiva categoria, utilizados para realizar o mapeamento e aplicação do comportamento humano aos personagens (Q5).

Entre os estudos selecionados, os jogos do tipo FPS são os mais utilizados no mapeamento do comportamento humano somando 64,7%, conforme pode ser observado na Tabela 13.

Tabela 13. Jogos e respectivas categorias utilizadas nos estudos.

<b>Nome</b>	<b>Categoria</b>	<b>totais</b>
Quake II	FPS	8 (47,05)
Quake III	FPS	1 (5,88%)
Unreal Tournament	FPS	2 (11,76%)
TORCS	Corrida	2 (11,76%)

Boxe 3D	Luta	1 (5,88%)
COSAIR	Estratégia	1 (5,88%)
Jogo similar a Starcraft	Estratégia	1 (5,88%)
The Legion II	Estratégia	1 (5,88%)

Jogos de estratégia representam 17,65% do total de estudos, enquanto outras categorias de jogo somadas chegam a este mesmo valor .

Outro ponto que chama a atenção é a proporção de utilização do jogo Quake II, que representa 72,72% ente todos os 11 jogos do tipo FPS utilizados nos estudos, indicando que o jogo contém funcionalidades que facilitam os testes do mapeamento do comportamento humano nos agentes do jogo.

### 3.3.5 Classificação dos resultados obtidos

Os resultados alcançados em cada pesquisa foram categorizados de acordo com as conclusões dos autores. Após extrair os resultados de cada um dos estudos selecionados foi possível observar que 64,7% dos resultados foram alcançados plenamente, ou seja, os autores afirmaram ter sucesso no mapeamento do comportamento humano (Q6). Estes dados podem ser observados na Tabela 14.

Tabela 14. Resultados Alcançados.

<b>Descrição do resultado</b>	<b>Totais retornados</b>
Resultados alcançados plenamente	11 (64,7%)
Resultados alcançados parcialmente	5 (29,41)
Resultados não alcançados	1 (5,88%)

Apenas uma das pesquisas não conseguiu fazer com que as ações fossem refletidas nos agentes e as 5 pesquisas que apresentaram resultados parciais, representando 29,41% dos estudos selecionados, observaram que alguns comportamentos inesperados foram encontrados nos testes, mas que para determinadas ações o comportamento foi obtido conforme o esperado.

Além de analisar os resultados alcançados em cada pesquisa se observou que os métodos utilizados para avaliar estes resultados também são importantes para identificar a qualidade das avaliações realizadas. Os dados sobre os métodos utilizados podem auxiliar pesquisadores a realizar suas escolhas e por isso uma descrição, mesmo que sucinta do método, representa um dado importante.

Para descrição básica dos métodos utilizados os mesmos foram generalizados para fins comparativos. A intenção desta generalização é observar os métodos de avaliação mais utilizados.

Pra que fiquem claras as convenções utilizadas para generalização dos métodos, são realizadas as seguintes descrições:

- **Métricas de pontuação para atingir determinada tarefa:** Nestes trabalhos foram utilizadas as pontuações do jogo e número de vitórias para definir o sucesso do mapeamento para uma determinada tarefa executada pelos personagens.
- **Dados sobre o sucesso do algoritmo de treinamento e observação:** Nestes trabalhos os valores obtidos através das técnicas de IA no treinamento e observações sobre o comportamento dos personagens criados foram utilizados para avaliação dos personagens.
- **Comparação das áreas graficamente para execução de determinada tarefa:** Nestes trabalhos foram comparadas as zonas de atuação do personagem e jogador orientado ao mesmo objetivo, se áreas similares nos mapas fossem ativadas os gráficos resultantes demonstravam as similaridades das comparações.
- **Comparação das áreas por clusterização para execução de determinada tarefa:** Nestes trabalhos foi utilizada a clusterização para avaliação e comparação dos dados do jogador e personagem criado na execução de tarefas similares
- **Testes com jogadores (Survey):** Nestes trabalhos foram realizadas pesquisas através de questionários de avaliação para se obter a percepção de jogadores ao avaliar se era possível distinguir entre os personagens treinados com comportamento humano e os jogadores humanos.
- **Comparação das áreas por taxa de similaridade para execução de determinada tarefa:** Nestes trabalhos foram utilizadas taxas de similaridade previamente definidas para avaliação e comparação dos dados do jogador e personagem criado na execução de tarefas similares.

A partir da avaliação realizada conforme os métodos descritos, foi possível verificar que apenas 2 trabalhos realizaram a pesquisa de credibilidade dos personagens criados com

comportamento humano (Testes com jogadores (Survey)), representando 11,76%, conforme pode ser visto na Tabela 15.

Tabela 15. Métodos utilizados.

<b>Método de avaliação</b>	<b>Totais</b>
Métricas de pontuação para atingir determinada tarefa	5 (29,41%)
Dados sobre o sucesso do algoritmo de treinamento e observação	3 (17,65%)
Comparação das áreas graficamente para execução de determinada tarefa	4 (17,65%)
Comparação das áreas por clusterização para execução de determinada tarefa	2 (11,76%)
Testes com jogadores (Survey)	2 (11,76%)
Comparação das áreas por taxa de similaridade para execução de determinada tarefa	1 (5,88%)

Estudos que realizaram a comparação por áreas gráficas (Comparação das áreas graficamente para execução de determinada tarefa), representando 23,52% dos estudos selecionados, se destacaram por demonstrar as características de movimentação mapeadas de uma maneira mais clara, tornando a comparação mais objetiva.

Mas os métodos mais utilizados foram os orientados a comparação da pontuação entre os agentes treinados e os jogadores (métricas de pontuação para atingir determinada tarefa), representando 29,41% dos estudos selecionados.

### 3.4 CONSIDERAÇÕES

Esta revisão sistemática teve como principal objetivo a resposta a questão primária da revisão (Q1), relativo aos principais trabalhos ligados ao mapeamento do comportamento humano através de técnicas de Inteligência Artificial na área de jogos.

Entre os trabalhos retornados foram selecionados os que se enquadravam nos critérios de inclusão e exclusão definidos e com isso pesquisas com o mesmo tema foram descartadas por detalhes da elaboração dos trabalhos que não atendiam aos critérios.

Entre os trabalhos selecionados foi verificado que muitos estudos foram repetidos entre as bases, o que acabou demonstrando que algumas bases não foram relevantes.

Pela lista de estudos selecionados, apresentados na Tabela 6, foi possível observar que um grupo de autores esteve envolvido em vários estudos selecionados em pesquisas com temas

similares, sendo que este fator não interfere na qualidade dos estudos retornados, pois os trabalhos apresentarem diferentes contribuições e objetivos.

Pelo ano dos estudos também foi possível observar que apesar do critério definido trazer trabalhos desde o ano 2000 a maioria dos estudos tem menos de 6 anos.

Na extração dos dados foram realizadas observações sobre os principais pontos dos trabalhos relativos ao processo de mapeamento, sendo que estes dados foram extraídos apenas para descrever detalhes dos estudos de forma mais clara. Todos estes dados foram obtidos para responder as questões de pesquisa definidas no protocolo de revisão sistemática.

Entre os dados extraídos foram analisadas quais ações executadas pelos jogadores foram mapeadas nos agentes em cada estudo (Q2), o método utilizado para leitura das informações dos jogadores (Q3), as técnicas utilizadas (Q4), os jogos e respectivas categorias (Q5) além dos resultados e métodos de avaliação da qualidade dos agentes implementados (Q6).

Apesar de considerados todos os principais trabalhos selecionados a partir da revisão sistemática com foco no mapeamento do comportamento humano, nem todos os trabalhos apresentam a mesma metodologia para mapeamento das ações do jogador, outros são de categorias bem distintas como no caso dos jogos de corrida. Ainda assim, todos os trabalhos apresentam informações importantes para comparação e auxílio das escolhas realizadas.

O fato de alguns trabalhos focarem no mapeamento de ações específicas, como no caso da movimentação dos personagens, deixaram os testes mais simples e de certa forma com um bom nível de precisão, mas também deve ser ponderado que a complexidade é menor em relação a trabalhos que consideraram outras ações no mapeamento do comportamento do jogador.

Um dado interessante extraído destes trabalhos foi em relação ao método de mapeamento das ações do jogador, em que todos utilizaram o próprio jogo ou ferramentas que analisam o tráfego de ações enviadas ao jogo para criação de *logs* com as ações dos jogadores. Já o tratamento dos dados e a utilização variaram em cada um dos trabalhos de acordo com os objetivos e ambientes.

Outros dados que se destacam nos trabalhos similares são as técnicas de IA aplicadas, em que foi possível perceber que a técnica de RNA, isolada ou em conjunto com a técnica de

Algoritmos Genéticos, foi a mais utilizada para o treinamento dos agentes com o comportamento do jogador humano.

A categoria dos jogos utilizados também chamou a atenção ao apontar que os jogos de FPS são os mais utilizados, mais especificamente o ambiente do jogo Quake II utilizado na maioria dos jogos de FPS. Apesar disso, o fato de um dos grupos de pesquisa envolver os mesmo autores em alguns destes trabalhos justifica a repetição do ambiente em alguns casos.

A categoria também demonstrou que as ações dos jogos de FPS em relação ao Robocode são muito similares, pois ambos compartilham ações de movimentação e estratégia de batalha como mirar nos inimigos e desviar de ataques.

Quando avaliada a metodologia e resultados das pesquisas, dois trabalhos de Bauckage *et al* (2007) e Gorman *et al* (2006b) merecem destaque devido ao método de avaliação de qualidade dos personagens, pois ambos os trabalhos apresentam uma variação do teste de Turing para leitura da percepção dos avaliadores. Apesar das diferenças no objetivo do teste em relação à ação dos agentes, o modelo de avaliação foi idealizado e descrito em detalhes, proporcionando um *framework* de avaliação coerente para aplicação neste trabalho.

Enfim, a partir de uma revisão sistemática da literatura que dispõe de um processo sistemático e mais confiável de revisão foram realizadas as seleções de estudos sobre o mapeamento de comportamento humano, similares a este trabalho, através de técnicas de Inteligência Artificial para a área de jogos.

## **4 DESENVOLVIMENTO**

Nesta seção são apresentados detalhes do desenvolvimento do agente inteligente baseado nas ações do jogador humano utilizado nos experimentos realizados nesta pesquisa.

Primeiramente é apresentado o processo de definição das variáveis que foram mapeadas para representar adequadamente as ações do jogador. Em seguida são descritos os métodos utilizados para leitura, obtenção e tratamento dos dados gerados pelo jogador em partidas realizadas no Robocode. Na sequência é abordado o processo de treinamento da Rede Neural com base nos dados gerados. Após o treinamento são descritos os detalhes da utilização da rede treinada, além de uma etapa definida para análise inicial das ações do agente treinado antes da realização do experimento. Por fim, é descrito o instrumento de avaliação que foi utilizado no experimento realizado.

### **4.1 DEFINIÇÕES DAS VARIÁVEIS PARA MAPEAMENTO**

Esta etapa tem como objetivo principal definir quais os dados devem ser mapeados a partir da estrutura dos agentes do Robocode de forma a melhor representar as ações do jogador.

Antes de definir as variáveis que devem ser mapeadas e o método utilizado para o mapeamento, é importante conhecer alguns aspectos do ambiente que interferem nestas definições, além de características do agente que servem como interface de leitura e reprodução das ações do jogador. Estes aspectos são descritos em detalhes nas próximas seções.

#### **4.1.1 Robocode como ferramenta de simulação**

A ferramenta Robocode foi escolhida como instrumento para leitura das ações do jogador e também para testes com robô treinado a partir destas ações mapeadas. A ferramenta proporciona um ambiente para batalha entre dois ou mais robôs, em que os mesmos se enfrentam até a destruição de um deles. O robô pode realizar certas ações, como se movimentar pelo cenário, observar obstáculos e inimigos no ambiente a sua volta, mirar e atirar.

O Robocode também retorna informações relativas aos eventos que ocorrem durante a partida de acordo com o disparo destes eventos, conforme mudanças no ambiente. Entre estes eventos obtidos através do método de scanner do Robô, pode-se identificar quando o robô atinge

uma parede, quando é atingido por um projétil ou quando atinge um oponente (ROBOCODE, 2012).

O Robocode executa as batalhas em um modo que de acordo com Czajkowski e Patan (2009) simula um ambiente de execução em tempo real. Isto ocorre, pois durante a execução da batalha todos os cálculos são realizados em turnos alternados. Cada um destes eventos é lido de acordo com este método de atualização do Robocode, que funciona como um método de *loop*, em que a cada nova iteração podem ser verificadas as informações que norteiam os robôs e o ambiente.

Para qualquer leitura que precise ser realizada no ambiente, são utilizadas estas frações de tempo da execução do *loop*, sendo que os eventos de disparos individuais também podem ser utilizados para obter informações específicas de outros eventos. Este método de leitura caracteriza o primeiro desafio na obtenção de dados do jogador durante o mapeamento de suas ações, pois a cada iteração são recebidos dados dos sensores definidos para coleta das ações do jogador.

Gade *et al* (2003) descrevem a execução em turnos do Robocode de acordo com os seguintes aspectos:

- Cada robô executa seus códigos até que ações sejam tomadas;
- O tempo é atualizado;
- Todos os disparos se movem e são verificadas colisões;
- Todos os robôs se movem, sendo executadas as ações na seguinte ordem:
  - Direção;
  - Aceleração;
  - Velocidade;
  - Distância;
- Todos os robôs executam suas leituras; e
- O campo de batalha é atualizado.

O principal método para criação dos robôs utilizado neste trabalho, foi definido a partir da extensão da classe `AdvancedRobot`, que é uma classe Java que trabalha de maneira assíncrona, permitindo ao robô executar ações simultaneamente.

Por ser uma ferramenta que preza pela facilidade de implementação o Robocode apresenta uma documentação rica em informações, na qual estão disponíveis centenas de métodos para realizar o mapeamento de dados de movimentação, leitura do oponente, armas e potência dos disparos.

Por exemplo, somente para leitura da movimentação, podem ser utilizadas as próprias coordenadas cartesianas do robô, ou até mesmo ângulos em relação as paredes e oponentes, com diferentes escalas. A quantidade de métodos disponíveis faz com que seja possível obter muitos detalhes necessários sobre as ações que o jogador esta realizando.

Entendidos estes aspectos, a próxima seção apresenta detalhes sobre a estrutura do Robô e sua arquitetura para demonstrar claramente as definições utilizadas para seleção dos sensores relativos as ações do jogador.

#### **4.1.2 Estrutura do Agente**

Os agentes da ferramenta Robocode são caracterizados pelo personagem de um robô no estilo tanque de guerra. As características destes robôs permitem que a leitura de suas ações sejam obtidas através de métodos que podem ser utilizados como sensores de leitura.

Cada robô inicia a batalha com 100 pontos de energia e são destruídos assim que estes pontos chegarem a zero. A arma utilizada pelos robôs é composta por uma torre que pode ser movimentada em ângulos de 360° (Gade *et al*, 2003).

A movimentação do robô pelo cenário é independente da direção da torre que comporta a arma, com isso o robô pode estar correndo em direção oposta ao inimigo e ainda assim estar realizando um ataque ao mesmo tempo.

Uma das características mais importantes do robô é o seu radar, pois a partir dele são realizadas as leituras do ambiente para detectar as ações dos oponentes. O radar permite que sejam verificadas a distância em relação ao oponente e as paredes do cenário, além da velocidade do

oponente, níveis de energia e posição do inimigo. O radar também possibilita uma leitura em 360° que é independente da movimentação do robô e da direção da torre.

A base do tanque é responsável pela movimentação e as ações que podem ser realizadas pela torre são subdivididas em mira e disparo.

A potência do canhão tem diferentes níveis, um com uma potência maior que é disparada em uma velocidade menor e outro disparo com uma potência menor, mas com maior velocidade de disparo (Gade *et al*, 2003).

Os disparos ainda exercem outro efeito sobre os agentes, pois os projéteis trabalham em um sistema de recompensa em que os acertos atribuem vida ao robô que efetuou o disparo e os erros o fazem perder pontos de vida. A Tabela 16 apresenta a proporção entre a potência dos disparos, o dano causado, a recompensa pelo acerto e a velocidade do disparo (Gade *et al*, 2003).

Tabela 16. Relação de potência, dano causado, dano recebido e velocidade do projétil.

<b>Potência do disparo</b>	<b>Dano causado</b>	<b>Energia recuperada</b>	<b>Velocidade da bala (pixels/turno)</b>
0.1	0.4	0.3	19.7
1.0	4.0	3.0	17
2.0	10.0	6.0	14
3.0	16.0	9.0	11

Outro aspecto que resulta na perda de vida dos robôs são as colisões, pois a cada colisão com um inimigo o robô perde pontos de vida, sendo que estas perdas valem para ambos os robôs envolvidos na colisão, mas se o ataque é percebido pelo Robocode como uma corrida em velocidade contra o oponente este causa mais dano do que recebe. A colisão contra paredes também resulta na perda de vida do robô.

A Figura 14 demonstra a imagem do robô, que tem como tamanho padrão 36 x 45 pixels, com suas principais características.

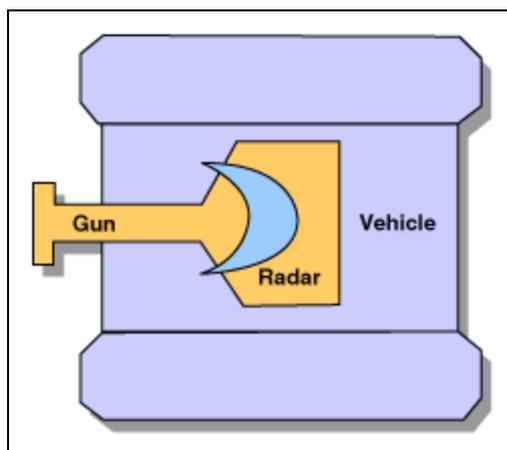


Figura 14. Arquitetura do robô

Fonte: Adaptado de Gade *et al* (2003).

Estes aspectos citados permitem ao robô se movimentar pelo cenário para frente com um atributo de velocidade positivo ou para trás com um atributo de velocidade negativo, e a velocidade máxima em qualquer uma das direções é de 8 pixels por turno (Gade *et al*, 2003).

Estas informações são importantes para que os tratamentos realizados na leitura de dados dos robôs possam ser compreendidos de forma mais clara nas próximas seções.

### 4.1.3 Método de extração de dados

A tarefa de mapeamento do comportamento humano é de difícil abordagem e conseqüentemente torna difícil a definição de uma metodologia concreta para este fim (GORMAN; HUMPHRYS, 2007).

O método utilizado para extração de dados das ações do jogador se baseia principalmente na definição de informações que permitam ao agente perceber o ambiente a sua volta, realizando a leitura do campo de batalha e do oponente. As informações obtidas a partir destas leituras são utilizadas para mapear as ações executadas pelo jogador.

Como complemento ao método de extração de dados, foram utilizados conceitos do método COCOM. Este modelo é utilizado em um trabalho similar no qual foram mapeadas as ações do jogador, realizado por Gorman e Humphrys (2007) e foi utilizado para auxiliar na definição do tipo das variáveis de leitura das ações do jogador.

O COCOM subdivide aspectos de aprendizado que devem ser considerados no mapeamento de maneira que possam ser obtidos de forma literal (o que foi lido é utilizado no treinamento) e os dados que devem ser considerados em um nível de organização estrutural (tratados para se obter uma estratégia ou objetivo mais abstrato). Um exemplo do COCOM pode ser visualizado na Figura 15.

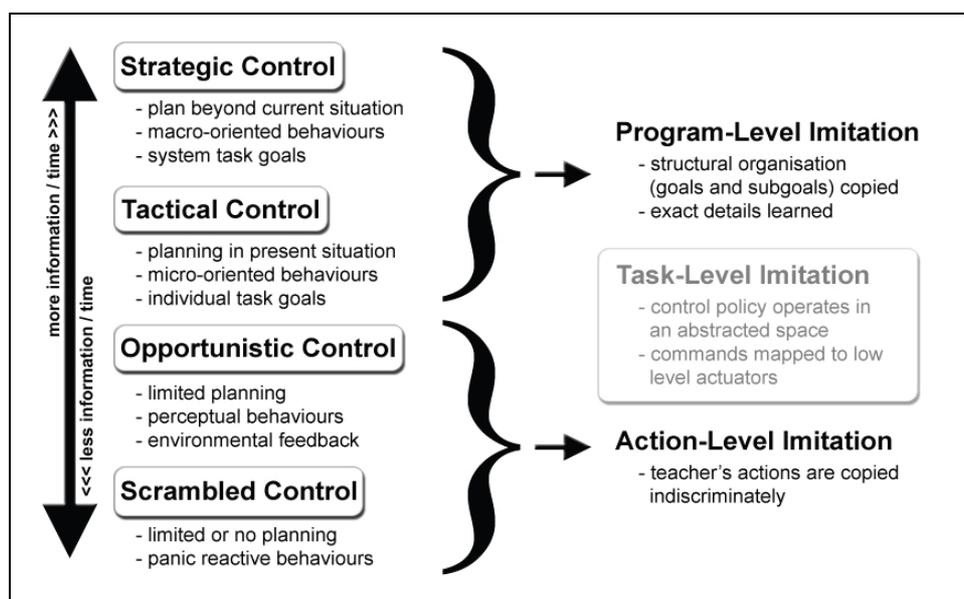


Figura 15. Modelo do método COCOM

Fonte: Adaptado de Gorman e Humphrys (2007).

Como pode ser visto no modelo, controles ligados a estratégia e as táticas são abordados em um nível de programa, considerando a estrutura organizacional a partir dos objetivos.

A relevância do método COCOM para este trabalho está na definição dos tipos de dados utilizados no mapeamento baseado nas ‘tarefas’ executadas pelo jogador humano, em que certos comandos são obtidos de maneira abstrata de acordo com as operações realizadas no ambiente e outros comandos são mapeados com base em atuadores de baixo nível.

Considerando a estrutura do robô a definição das variáveis foi dividida entre a movimentação e o controle da torre.

A partir do método de extração de dados descrito, foram definidas as variáveis para o mapeamento das ações do agente na movimentação do tanque e controle da torre, que são tratados em detalhes na próxima seção.

#### 4.1.4 Definição das variáveis de Movimentação

Para a movimentação, foram definidos sensores para realizar a leitura de dados de maneira que fosse possível obter as ações do jogador em relação ao ambiente e oponente.

A definição dos sensores relativos ao cenário e ao oponente foi baseada na abordagem de Nielsen e Jensen (2010), em que os sensores de leitura para movimentação foram criados de forma a abranger o campo de visão do robô, além de sensores relativos ao oponente. Os autores obtiveram ângulos e distâncias em relação as paredes e ao oponente.

A partir do modelo de Nielsen e Jensen (2010) e análises do ambiente, foram realizados ajustes em relação à posição dos sensores, as alterações se basearam na observação de que com menos sensores já é possível se obter a percepção do ambiente. Ao invés de utilizar sensores para abranger o campo de visão frontal do robô, foram utilizados dois sensores laterais e um traseiro, com a finalidade de orientar o robô em relação ao cenário.

Também foram removidas as informações de ângulo em relação às paredes do cenário, considerando que a distância das paredes é suficiente para que o agente possa se orientar e evitar trancamento quando encurralado pelo oponente.

Outra modificação em relação às distâncias das paredes foi à criação de um ponto de intersecção definido em 100 pixels. O ponto de intersecção foi definido para que a leitura das distâncias fosse mantida constantemente em 100 pixels e somente após o ponto de intersecção ser ativado os valores em relação às paredes eram considerados.

Com as mudanças realizadas a orientação principal do robô ficou baseada nas ações do oponente, com os sensores em relação às paredes responsáveis apenas por situar o robô no cenário. Para leitura das informações relativas ao oponente foram definidos dois sensores, para obtenção do ângulo e distância.

Além dos sensores relativos ao oponente e cenário, foi definido um sensor para obtenção de um dado estratégico com a finalidade de auxiliar na identificação das manobras de aproximação do oponente. Nielsen e Jensen (2010) utilizam uma informação similar chamada de “sensor de perigo”, sendo que neste trabalho o sensor leva em consideração somente o fator de aproximação em relação ao oponente.

Na sequência são resumidos os 6 sensores definidos para movimentação:

- Sensor 1: Sensor que recebe o ângulo em relação ao oponente;
- Sensor 2: Sensor que recebe a distância em relação ao oponente;
- Sensor 3: Sensor de perigo utilizado para detectar a proximidade em relação ao oponente;
- Sensor 4: Sensor que recebe a distância da parte traseira do robô até o ponto final do cenário;
- Sensor 5: Sensor que recebe a distância da lateral esquerda do robô até o ponto final do cenário; e
- Sensor 6: Sensor que recebe a distância da lateral direita do robô até o ponto final do cenário;

A Figura 16 apresenta um exemplo a partir das linhas de orientação geradas no Robocode para alguns dos sensores definidos, em que o sensor 6 é representado pela linha amarela, o sensor 5 pela linha azul e o sensor 4 pela linha verde. O semicírculo na ponta das linhas marca o ponto final para medição da distância. Os círculos menores observados nas linhas são referentes ao ponto de intersecção. A linha vermelha representa os sensores 1 e 2 que marcam a distância e ângulo em relação ao oponente.

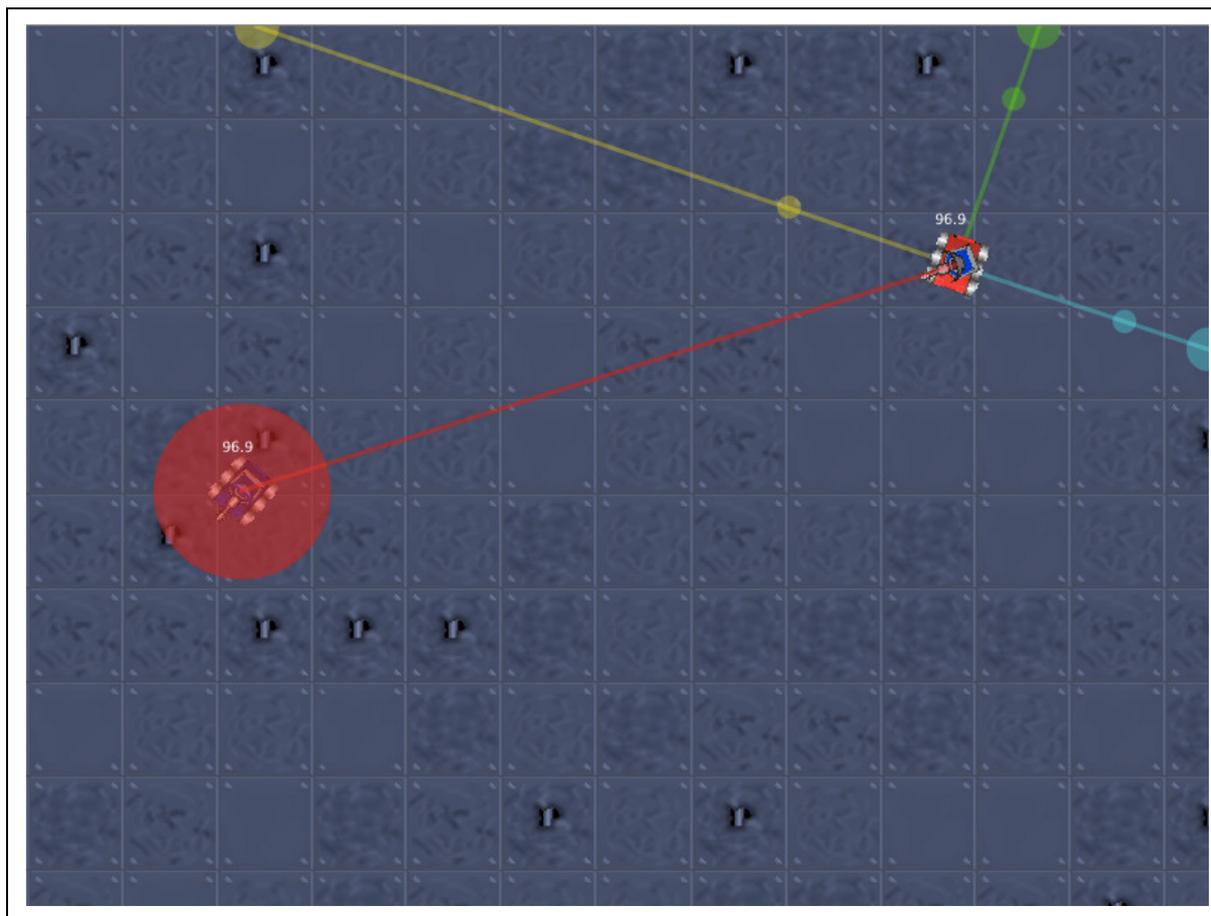


Figura 16. Exemplo de sensores de movimentação

Alguns sensores definidos para leitura das ações do jogador puderam ser lidos diretamente através de métodos disponíveis pela biblioteca do Robocode, sendo que outros tiveram que ser tratados para que os valores pudessem ser recebidos da maneira desejada.

Os métodos utilizados para leitura de cada um dos sensores são apresentados na sequência:

- Sensores relativos ao cenário (sensores 4, 5 e 6): Para estes sensores foram calculadas as distâncias a partir das coordenadas do robô em relação às paredes do cenário, conforme apresentado na Figura 16 o cálculo retorna a distância em pixels destas retas. A origem destas retas considera ângulo relativo à orientação do próprio robô, com a frente do robô considerada  $0^\circ$ . Em (5) é apresentado o cálculo principal utilizado para obtenção da distância em relação as paredes;

$$D = \sqrt{(FX - IX)^2 + (FY - IY)^2} \quad (5)$$

Sendo em (5):

$D$  = Distância em relação a parede.

$FX$  = Coordenada X da posição final no cenário.

$IX$  = Coordenada X inicial baseada na posição do robô.

$FY$  = Coordenada Y da posição final no cenário.

$IY$  = Coordenada Y inicial baseada na posição do robô

- Ângulo em relação ao oponente (sensor 1): O método `getBearing()` do Robocode já traz a posição em relação ao oponente através de ângulos positivos até 180° (para leituras realizadas quando o oponente se localiza a direita do robô) e negativos até -180° (quando localizado a esquerda);
- Distância em relação ao oponente (sensor 2): A distância é calculada a partir das coordenadas em relação ao oponente. O cálculo consiste na obtenção das coordenadas do inimigo e na obtenção das coordenadas do próprio robô, traçando uma reta entre os dois pontos e calculando a distância em pixels desta reta. A fórmula utilizada para este cálculo é apresentada em (6);

$$Denemy = \sqrt{(RY - SY)^2 + (RX - SX)^2} \quad (6)$$

Sendo em (6):

$Denemy$  = Distância em relação ao inimigo.

$RX$  = Coordenada X da posição do robô.

$SX$  = Coordenada X da posição do robô inimigo.

$RY$  = Coordenada Y da posição do robô.

$SY$  = Coordenada Y da posição do robô inimigo.

- Sensor de perigo (sensor 3): O sensor de perigo é ativado em 1 e desativado em 0 para verificação de distâncias pequenas entre o robô e o oponente.

Além dos sensores definidos para leitura das ações executadas pelo jogador, também precisam ser lidas suas escolhas através da definição de atuadores para este fim. Para a movimentação são lidas as direções executadas pelo jogador (para cima, baixo, direita e esquerda). Estas direções são lidas a partir de eventos contidos na interface utilizada pelo jogador para interagir com o Robocode, demonstrada na Seção 4.3.

Em resumo, as variáveis foram divididas em 6 sensores e 4 atuadores utilizados para leitura das ações do jogador em relação a movimentação do robô.

#### **4.1.5 Definição das variáveis de Controle da Torre**

Inicialmente para obtenção de dados de mira e disparo foi realizada a análise dos trabalhos de Czajkowski e Patan (2009) e Nielsen e Jensen (2010) que realizam tarefas de controle da torre em agentes do Robocode. Em ambos os trabalhos citados é realizada a tentativa de prever a posição do inimigo, através de suas últimas posições, para realizar o disparo. Porém, deve-se levar em consideração que o objetivo não é criar o robô que mais acerta e sim aquele que utiliza os padrões identificados através do jogador para mirar e disparar.

Portanto foi definida uma abordagem diferenciada que visa identificar o padrão de disparos do jogador e foram descartados os métodos de previsão das posições futuras do inimigo para executar o maior número de acertos. O método definido se baseia na leitura de informações relativas ao oponente que permitam a identificação de suas ações em momentos em que o jogador executa as ações para mirar e disparar.

Para tanto foram definidos 5 sensores. Conforme descritos a seguir:

- Sensor 1: Sensor que recebe o ângulo da arma em relação ao oponente;
- Sensor 2: Sensor que recebe a direção do oponente, considerando valores positivos e negativos em relação a torre através da leitura das duas últimas posições do oponente;
- Sensor 3: Sensor que recebe a distância em relação ao oponente;
- Sensor 4: Sensor que recebe a velocidade do oponente; e
- Sensor 5: Sensor utilizado para detectar a proximidade em relação ao oponente;

Através destes sensores foi possível obter os dados que refletem os momentos em que são executadas as ações de mira e disparo, pois é possível saber a localização do oponente em relação a torre, a distância e velocidade, além do reforço através do sensor de proximidade para identificar momentos propícios para o aumento da quantidade e intensidade dos disparos.

A Figura 17 apresenta um exemplo dos pontos de leitura para alguns dos sensores definidos, onde área em amarelo representa o sensor 1 que marca o ângulo da arma em relação ao oponente. Os círculos verdes representam a posição atual e anterior utilizadas pelo sensor 2. O sensor 3 é representado pela linha vermelha, utilizado para medição da distância em relação ao oponente.

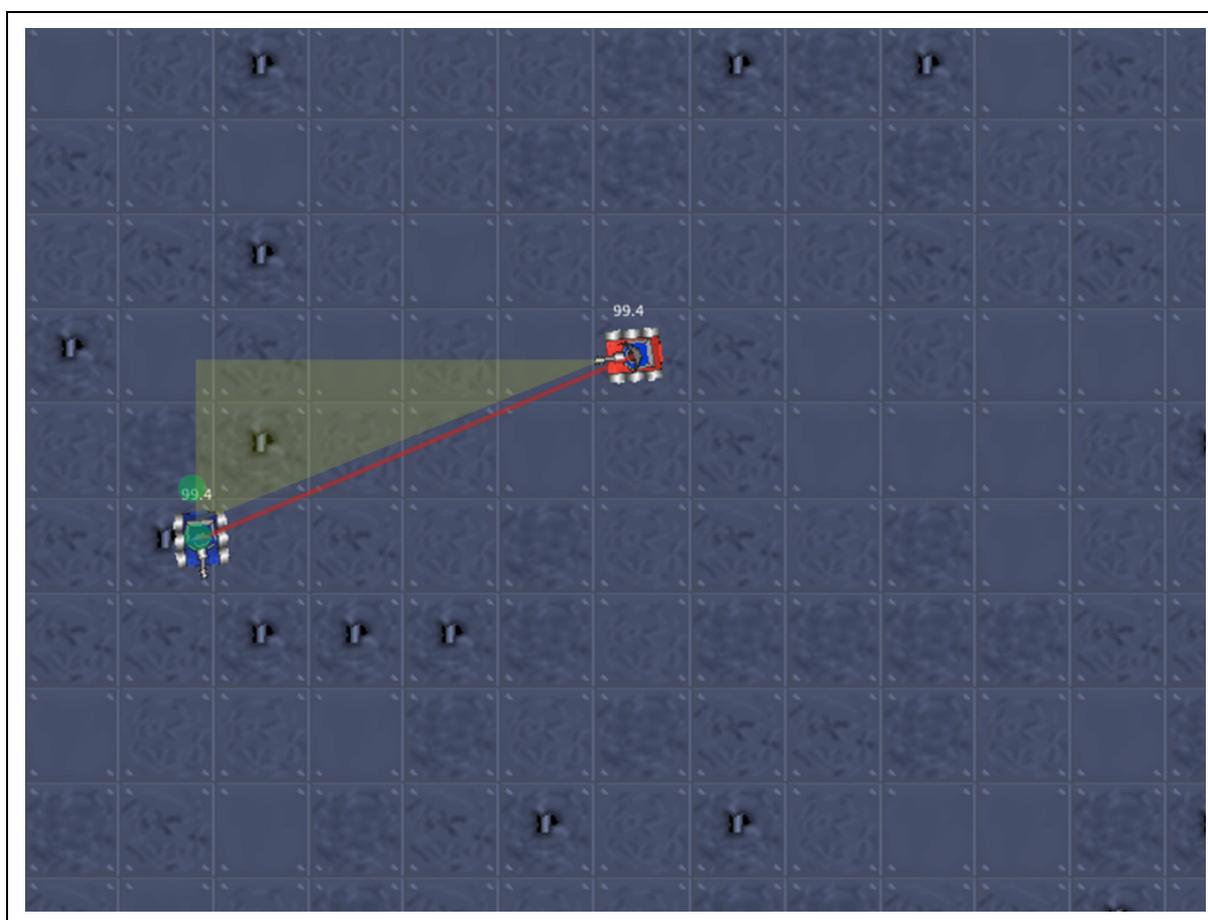


Figura 17. Exemplo de sensores para controle da torre

Os sensores de distância em relação ao oponente (sensor 3) e perigo (sensor 5) são os únicos sensores que também são utilizados na movimentação, pois o ângulo utilizado em relação ao oponente é baseado na torre e não no tanque.

Assim como na movimentação, alguns sensores definidos para leitura das ações do jogador no controle da torre puderam ser lidos diretamente através de métodos disponíveis pela biblioteca do Robocode, sendo que outros tiveram que ser tratados para que os valores pudessem ser recebidos da maneira desejada.

Os métodos utilizados para leitura de cada um dos sensores são apresentados na sequência:

- **Ângulo da arma em relação ao oponente (sensor 1):** Para obtenção do ângulo da arma em relação ao oponente, foi criado o método `getGunAngle()`. Para a criação do método foi utilizada a diferença entre o ângulo frontal do robô em relação ao cenário e a própria torre. Conhecida a posição da torre em relação ao cenário, foi utilizada a diferença entre a posição do oponente através do método `getBearing()` e a posição da torre, para que fosse obtido o ângulo da torre em relação ao oponente. Para esta definição foram utilizadas funções para o tratamento dos ângulos obtidos, pois as leituras dos ângulos da arma e do oponente podem ficar próximas da barreira entre  $360^\circ$  e  $0^\circ$  do cenário. O ângulo de saída é similar ao disponibilizado pelo método `getBearing()`, sendo neste caso em relação a torre e não ao tanque, que retornam ângulos positivos até  $180^\circ$  (para leituras realizadas quando o oponente se localiza a direita da torre) e negativos até  $-180^\circ$  (quando localizado a esquerda da torre);
- **Direção do oponente (sensor 2):** O método `getDirecao()` foi criado para obtenção da direção do oponente. Este método utiliza a leitura das duas últimas posições do oponente em relação a torre e identifica a partir dos ângulos, se o inimigo se move em sentido horário ou anti horário. Se as últimas posições caracterizam um movimento da torre em sentido horário é atribuído um valor positivo definido em 1, do contrário é atribuído um valor negativo em -1;
- **Distância em relação ao oponente (sensor 3):** A distância é calculada a partir das coordenadas em relação ao oponente. O cálculo consiste na obtenção das coordenadas do inimigo e na obtenção das coordenadas do próprio robô, traçando uma reta entre os dois pontos e calculando a distância em pixels desta reta. A fórmula utilizada para este cálculo é apresentada em (6);
- **Velocidade do oponente (sensor 4):** O método `getVelocity()` do Robocode já traz a velocidade de movimentação do oponente, sendo que o mesmo traz valores positivos quando o robô se movimenta para frente e negativos quando se movimenta para traz. Por

isso a informação foi tratada para obtenção do valor absoluto da velocidade. Os valores vão de 0 até 8; e

- Sensor de perigo (sensor 3): O sensor de perigo é ativado em 1 e desativado em 0 para verificação de distâncias pequenas entre o robô e o oponente.

Além dos sensores, também precisam ser lidas as ações de mira e disparo do jogador através da definição de atuadores para este fim. O direcionamento da torre retorna um valor no sentido horário e anti-horário que são separados em 2 atuadores distintos, no qual é atribuído o valor 1 para o atuador ativado e 0 para não ativado.

Já o disparo foi definido em 3 níveis de intensidade, 1, 2 e 3, quando o valor é zero não foi executado nenhum disparo. Quando é efetuado o disparo o atuador recebe a informação com a intensidade atribuída pelo jogador entre os 3 níveis disponibilizados.

As direções e a ação de disparo no controle da torre são lidas a partir de eventos contidos na interface utilizada pelo jogador para interagir com o Robocode, demonstrada na Seção 4.3.

Em resumo, as variáveis foram divididas em 5 sensores e 3 atuadores utilizados para leitura das ações do jogador em relação ao controle da torre.

## **4.2 MODELO DE TREINAMENTO SUBDIVIDIDO EM TAREFAS**

A divisão em tarefas foi uma solução encontrada durante as primeiras tentativas de treinamento do agente a partir das ações do jogador. Esta solução se baseou na divisão de atividades do jogador em tarefas para que o comportamento fosse identificado e reproduzido, permitindo que a cada etapa fosse possível avaliar a evolução do treinamento realizado para uma determinada tarefa.

As primeiras tentativas de leitura das ações do jogador foram realizadas utilizando todos os sensores e atuadores definidos para movimentação e controle da torre em conjunto. Nestas primeiras etapas o jogador controlava o robô no cenário em partidas frente a oponentes como o MyFirstJuniorRobot, do próprio Robocode e o DrussGT uma abordagem mais competitiva definida para o experimento na Subseção 4.8.2 .

O primeiro conjunto de exemplos criado foi baseado em 100 batalhas realizadas no Robocode, gerando aproximadamente 38000 exemplos. A RNA não conseguiu identificar as ações

do jogador durante o treinamento, resultando em um agente que se movimentava de maneira muito diferente das atividades treinadas. O resultado apresentou um robô que corria em direção a paredes, ou girava constantemente em torno do seu próprio eixo.

A primeira alternativa na busca de alcançar o treinamento baseado nos exemplos gerados pelo jogador foi através da execução de novas batalhas, em que a cada 10 batalhas executadas eram realizadas novas tentativas de treinamento com os novos exemplos.

Em alguns momentos durante a execução de novas batalhas o robô se comportava de maneira mais aceitável, sendo que os problemas de trancamentos, movimentos repetidos e giros em torno do próprio eixo ainda ocorriam.

Foram realizadas até 500 batalhas para geração de novos exemplos, sendo que os resultados continuavam similares, em que o robô se trancava em paredes já na primeira movimentação, ou executava 2 ou 3 manobras antes de parar em algum dos cantos do cenário.

Na tentativa de solucionar os problemas, foram divididas as ações de movimentação e controle da torre utilizando novamente os exemplos gerados, divididos de acordo com seus sensores e atuadores.

Os resultados iniciais, tanto para movimentação quanto controle da torre foram similares aos problemas obtidos na tentativa de treinamento com os dados em conjunto.

Com base nestas dificuldades, foi realizada a tentativa de dividir as ações de treinamento em tarefas menores para movimentação. Em um dos testes realizados, em que foi executada uma tarefa de perseguição ao inimigo, foi possível obter resultados com baixo erro no treinamento e foram identificadas ações do robô coerentes com a tarefa ao qual ele foi treinado.

Com o início de funcionamento a partir desta divisão das ações de treinamento do jogador, foi definido o modelo de treinamento dividido em tarefas.

Esta divisão permite que possam ser identificadas previamente se as ações que estão sendo treinadas estão refletindo no agente, possibilitando identificar o número de exemplos gerados para treinamento de uma determinada tarefa. Com isso a geração de exemplos para a movimentação, por exemplo, pôde ser analisada e os exemplos balanceados para utilização no treinamento.

As tarefas de obtenção dos dados realizadas separadamente foram adicionadas ao mesmo arquivo de treinamento, sempre após a análise e tratamento dos dados. Quando todas as tarefas definidas para movimentação foram realizadas o arquivo de treinamento final para movimentação foi gerado, o mesmo foi realizado para o controle da torre.

Devido a esta divisão em tarefas, foi considerada a perda de exemplos mais específicos gerados pelo jogador, que poderiam ser gerados em batalhas jogadas sem a divisão. Em contrapartida foram descartados exemplos muito específicos que prejudicavam o treinamento da rede. Além disso, a base de exemplos, mesmo dividida em tarefas ainda se baseia nas ações do jogador para gerar os dados de treinamento.

As tarefas foram definidas de acordo com a observação de ações realizadas pelo jogador durante uma batalha no Robocode. As tarefas foram realizadas separadamente para as atividades de movimentação e controle da torre. As próximas seções abordam a divisão de tarefas realizadas para cada uma das atividades.

#### **4.2.1 Movimentação**

As tarefas de movimentação foram definidas de acordo com a observação das ações realizadas pelo jogador na batalha, de forma que todas as tarefas em conjunto permitissem a reprodução das atividades realizadas.

Na sequência é apresentada uma descrição das ações realizadas em cada uma das tarefas definidas para treinamento da movimentação:

- Perseguição e defesa: Pode-se dizer que esta tarefa contém duas subtarefas. A tarefa foi definida para que em cada batalha fosse realizada primordialmente a perseguição ao inimigo. A perseguição consiste em se movimentar em direção ao oponente continuamente até o fim da batalha. Sempre que o oponente contra atacava (vinha em direção ao robô) ou o robô estava muito próximo ao oponente eram realizadas manobras de defesa. Nestas manobras são realizados apenas desvios e recuos de acordo com o avanço do oponente. A cada batalha na aproximação do oponente eram realizadas manobras para manter a distância, tentando evitar que o robô se tornasse um alvo fácil.

- **Localização do inimigo:** Foram realizadas manobras para que o inimigo sempre estivesse próximo ao campo de visão do robô. A cada movimentação do inimigo na tentativa de fugir do campo de visão, eram realizadas manobras para localizar este oponente.
- **Reações em relação ao cenário:** A tarefa foi criada para evitar o encurralamento do robô uma vez que o mesmo ignora os limites do cenário nas outras tarefas. O intuito da tarefa não foi evitar a colisão e sim o trancamento, pois a colisão ocorre em partidas realizadas pelo jogador. Esta tarefa apenas realiza a leitura dos sensores relativos às paredes.

Com a obtenção das ações do jogador para cada uma destas tarefas, é possível na junção de todas elas criar um agente baseado em seu comportamento para as situações treinadas. Este agente pode evitar o encurralamento, localizar o inimigo no campo de batalha, atacar e recuar de acordo com a situação da batalha, da mesma forma que o que foi treinado pelo jogador.

#### **4.2.2 Controle da Torre**

As tarefas relativas ao controle da torre foram mais simples de serem definidas, isto porque o jogador consegue controlar apenas dois aspectos relativos a torre. As tarefas que o jogador pode realizar são a de mira e disparo.

Na sequência são descritas as ações realizadas em cada uma das tarefas definidas para treinamento do controle da torre:

- **Mira:** A movimentação da torre foi executada de forma a manter a mira sempre próxima ao oponente. Com isso foi possível obter dados da movimentação da arma para momentos que antecedem o disparo, auxiliando a Rede Neural a identificar durante a movimentação da torre os momentos escolhidos pelo jogador para realizar esta ação. Neste caso a movimentação serve como um “radar” que auxilia na identificação do melhor momento para o disparo.
- **Disparo:** A ação de disparo considera apenas o momento em que projéteis são acionados na tentativa de realizar o acerto ao oponente.

A partir das tarefas de mira e disparo foi possível obter as ações do jogador no controle da torre.

### 4.3 CICLO DE OBTENÇÃO DOS DADOS DE TREINAMENTO

Nesta etapa foram realizadas batalhas com o jogador humano controlando o robô para que pudessem ser lidas suas ações durante as partidas. Os arquivos para treinamento foram gerados de acordo com as ações realizadas pelo jogador para cada uma das tarefas definidas.

Para tanto foi utilizada como base a interface “Interactive\_V2” do próprio Robocode. A Interactive\_V2 é uma interface que permite que um jogador possa controlar um dos robôs do Robocode a partir da utilização do mouse e teclado. A interface permite que as teclas direcionais sejam utilizadas para controle da movimentação e o mouse permite que o jogador possa mirar e disparar contra o inimigo.

Porém foram realizadas modificações na interface para permitir que o robô pudesse se movimentar para trás. Com isso foram ajustados os ângulos e distâncias utilizados na movimentação. A partir destas modificações foi gerada a interface chamada de “ANNRobot”, que é utilizada tanto para leitura das ações do jogador e criação dos arquivos de treinamento como para a execução da rede treinada.

Outras modificações realizadas na interface ANNRobot foram relativas aos eventos de leitura das atividades de movimentação e controle da torre. Para a geração do arquivo com as informações de movimentações do robô, foi criado um evento acionado a partir das teclas direcionais utilizadas para controlar o mesmo, em que cada vez que uma tecla era pressionada uma linha contendo os sensores e atuadores relativos aos dados de movimentação era adicionada ao arquivo de treinamento.

Para o controle da torre foram definidos dois eventos, um acionado pela movimentação do mouse para mira e outro relativo a execução do disparo, ambos alimentando um arquivo de treinamento relativo a torre com os sensores e atuadores definidos.

O modelo de leitura baseado apenas nos eventos gerados pelo jogador foi definido devido ao método de atualização utilizado pelo Robocode, que é executado através de turnos, no qual é realizada a leitura das informações que norteiam o ambiente a cada turno.

O problema com este tipo de leitura é que os ciclos são atualizados com tempos de limite muito curtos. Com isso o resultado é um arquivo de treinamento muito grande com diversas

repetições, principalmente depois da execução de múltiplas batalhas. A repetição de valores que não refletem os ciclos de ação do jogador pode afetar o valor das épocas de treinamento negativamente.

Como no caso deste trabalho são lidos tanto os dados de movimentação quanto os dados de controle da torre, um método que pudesse realizar as leituras coerentes a cada conjunto de treinamento e evitasse o problema do tempo limite do Robocode teve que ser definido.

Configurada a interface de leitura através do ANNRobot, foram realizadas diversas batalhas para cada uma das tarefas de movimentação e controle da torre. O número de batalhas realizadas para cada uma das tarefas definidas foram os seguintes:

- Perseguição: 50 batalhas realizadas;
- Localização: 50 batalhas realizadas;
- Reações em relação ao cenário: 100 batalhas realizadas para equilibrar o número de exemplos das outras tarefas de movimentação;
- Mira: 50 batalhas realizadas;
- Disparo: 125 batalhas realizadas devido a menor proporção de exemplos gerados por partida em relação a mira.

Conforme já citado, os inimigos utilizados durante a obtenção de dados nas tarefas definidas, foram o MyFirstJuniorRobot, do próprio Robocode e o DrussGT. Porém o inimigo principal utilizado durante o treinamento foi o MyFirstJuniorRobot, pois as batalhas contra este inimigo eram mais longas, permitindo a geração de mais exemplos. Para adicionar exemplos mais complexos, mesmo com batalhas mais curtas, foi utilizado o DrussGT, que permitiu a diversificação dos exemplos gerados com um padrão mais elaborado de movimentação e disparo.

Ao final da obtenção de dados em cada uma das tarefas, todos os exemplos gerados a partir das ações do jogador foram analisados e tratados na Seção 4.4 para então serem atribuídos aos arquivos de movimentação e controle da torre.

A obtenção dos dados com base nas ações do jogador se deu exatamente como descrito na Seção 4.2, sendo realizadas as tarefas de perseguição/defesa, localização do inimigo, mira e disparo. Já a tarefa de reações em relação ao cenário, foi obtida de maneira diferenciada.

A tarefa foi tratada separadamente, pois ainda eram identificados muitos trancamentos no cenário pelo robô. Isto ocorreu devido ao numero de exemplos obtidos que eram menores em proporção aos exemplos gerados em outras tarefas. O problema ocorria, pois a intersecção era ativada poucas vezes durante a batalha.

Para resolver este problema foram ignorados os sensores relativos ao cenário para as tarefas de movimentação e definida uma tarefa especifica para este fim. Com base no numero de exemplos gerados em cada tarefa de movimentação foram obtidos os exemplos relativos ao cenário, sempre na tentativa de equilibrar o número de exemplos gerados com as tarefas anteriores.

Diferente de outras tarefas, as reações em relação ao cenário utilizaram apenas o MyFirstJuniorRobot, já que apenas a movimentação próxima as paredes e cantos era considerada e o MyFirstJuniorRobot costuma se manter na área onde a batalha é iniciada. Neste caso foram consideradas apenas as batalhas iniciadas próximas as paredes e cantos.

A adição destes sensores ao arquivo de treinamento da movimentação é demonstrado na próxima seção.

#### **4.4 TRATAMENTO DOS DADOS PARA TREINAMENTO**

Antes de realizar o treinamento da rede a partir dos arquivos gerados, conforme descrito na Seção 4.3, foram realizados ajustes nos dados a fim de auxiliar o treinamento da Rede Neural.

A quantidade de exemplos gerados por tarefa permitiu uma análise mais clara dos dados, o que possibilitou a identificação de exemplos que poderiam afetar negativamente o treinamento. Este foi um dos benefícios gerados pela divisão em tarefas.

Conforme as análises e testes iniciais foram definidas 2 etapas para tratamento dos dados gerados:

A primeira etapa foi realizada para balancear os exemplos referentes aos ângulos em relação ao inimigo, tanto para movimentação como controle da torre. A técnica utilizada nesta etapa foi a de *under-sampling*.

Os exemplos foram balanceados devido aos problemas identificados nas primeiras etapas do treinamento, em que o robô se comportava de maneira correta para ângulos negativos, por exemplo, mas se perdia com ângulos positivos, ou não apresentava reação a distâncias mais altas em relação ao oponente.

A segunda etapa, também identificada nas primeiras tentativas de treinamento realizadas, demonstrou que exemplos discrepantes gerados entre uma batalha e outra, mesmo que em menor proporção no arquivo, afetavam o treinamento negativamente.

Para evitar o problema, os ajustes foram realizados com base na identificação da faixa em que estavam concentrados os valores obtidos em cada uma das tarefas. A partir da identificação da concentração dos valores foram definidos valores mínimos e máximos para o ângulo ou distância na tarefa. Os exemplos fora da faixa de valores foram analisados para se obter sua representatividade em relação ao total de exemplos gerados. Se a faixa ficava abaixo de 0,1% o descarte era considerado válido.

Para o descarte foi utilizada uma regra de três simples para validar as regiões de corte definidas.

Com a definição das faixas foi possível descartar ângulos e distâncias muito altos ou muito baixos que poderiam prejudicar o treinamento.

Finalizados os tratamentos nos dados gerados para as tarefas de movimentação e controle da torre, foi realizada a última etapa de ajustes relativa a tarefa de reações em relação ao cenário.

Nesta etapa foram atribuídos os exemplos relativos as paredes do cenário ao arquivo contendo as tarefas de movimentação. Para tanto foram duplicados os exemplos gerados no arquivo e adicionados os sensores e atuadores relativos as paredes do cenário. Com isso o número de exemplo para cada tarefa de movimentação passou a conter os exemplos para movimentação sem considerar as paredes e a mesma proporção de exemplos para ações próximas as paredes. Com isso os sensores passaram a ganhar um peso maior quando ativados e evitaram os trancamentos.

A atribuição dos exemplos relativos as paredes do cenário no arquivo de dados de movimentação, foi realizado com base nos ângulos positivos e negativos em relação ao oponente para cada uma das tarefas.

Como a interface ANNRobot foi definida para corrigir o problema da geração de exemplos obtidos dos ciclos de atualização do Robocode, nenhum outro tratamento se mostrou necessário nos arquivos de dados.

## **4.5 PROCESSO DE TREINAMENTO**

O processo de treinamento demonstra os passos para treinamento da Rede Neural Artificial a partir dos dados obtidos nas partidas executadas pelo jogador humano.

O processo de treinamento foi realizado com base nas redes definidas para o treinamento da movimentação e controle da torre. A escolha dos tipos de rede e métodos de treinamento utilizados nesta etapa foram baseados inicialmente em trabalhos relacionados que realizaram estudos sobre o mapeamento do comportamento humano em personagens de jogos.

Com base nestes trabalhos foi definida a utilização de uma rede do tipo Perceptron para as tarefas de movimentação do e uma rede MLP para as tarefas relacionadas ao controle da torre do robô.

A abordagem Perceptron foi considerada inicialmente para a movimentação, pois na análise de trabalhos similares, o trabalho de Ki, Lil e Oh (2006) abordou a utilização da rede para realizar o treinamento da movimentação de personagens em um jogo de estratégia. Os resultados gerados durante o processo de treinamento geraram resultados que apoiaram a escolha realizada, como pode ser visto no decorrer desta seção.

Já para o treinamento do controle da torre, a rede MLP foi inicialmente definida.

Nos trabalhos de Zanetti e Rhalibi (2004) e Bauckhage, Thureau e Sagerer (2003), ambos trabalhos similares com foco no mapeamento do comportamento humano, também são utilizadas redes do tipo MLP com bons resultados.

Em um trabalho realizado no ambiente Robocode, Czajkowski e Patan (2009) realizaram a tentativa de prever a posição do inimigo no Robocode para realização de disparos. No trabalho também foi utilizada uma rede MLP.

No caso do treinamento do controle da torre, além do tipo da rede, diferentes algoritmos foram considerados para que se pudesse chegar até o modelo mais adequado. Assim como na escolha do tipo de rede utilizada, a primeira etapa de escolha destes algoritmos foi baseada em trabalhos similares e a segunda etapa baseada em testes preliminares abordados na próxima seção.

No trabalho de Bryant e Miikkulainen (2007) que realizaram o mapeamento do comportamento humano em agentes de um jogo de estratégia, foi utilizado o algoritmo de Backpropagation para treinamento, no qual todas as ações do personagem foram mapeadas para refletir o comportamento do jogador.

Muñoz, Gutierrez, Sanchis (2009) e van Hoorn *et al* (2009) também realizaram estudos na área de mapeamento do comportamento humano e utilizaram o algoritmo de BP para mapear determinadas ações dos personagens.

No trabalho de Czajkowski e Patan (2009) citado anteriormente, os testes foram focados na identificação do melhor algoritmo para treinamento, em que foram testados 3 diferentes algoritmos, BackPropagation (BP), Conjugate Gradient (CG) and Levenberg-Marquardt (LM) e foi verificado pelos autores que o algoritmo de BP foi o mais eficiente, tanto nos testes das ações do robô quanto na generalização durante o treinamento.

Os autores ainda afirmam que devido a dificuldade em se trabalhar com os limites de tempo de atualização do ambiente no Robocode, o aprendizado com algoritmos muito sofisticados não é recomendado.

Baseado nos trabalhos similares descritos o algoritmo BP foi definido inicialmente para utilização nos treinamentos da rede MLP.

Com base na rede definida para o treinamento do controle da torre, esta seção foi dividida em duas etapas. A primeira etapa demonstra as simulações realizadas para identificação da melhor estrutura a ser utilizada no treinamento do controle da torre. A última etapa apresenta os resultados dos treinamentos (para movimentação e controle da torre).

### **4.5.1 Estrutura da Rede e Algoritmos de treinamento para Controle da Torre**

Esta etapa apresenta o processo de seleção da melhor estrutura definida para as ações relativas ao controle da torre do robô para que a Rede Neural pudesse processar os dados corretamente durante o treinamento. Para tanto foram necessários diversos ajustes durante os treinamentos a fim de identificar a melhor topologia e a taxa de aprendizado adequada a partir dos resultados obtidos.

Com base nos dados gerados a partir das ações do jogador foram realizadas diferentes simulações com diferentes configurações para identificação da melhor estrutura. Cada rede foi treinada ao menos uma vez durante as simulações.

O erro padrão do Neuroph de 0.01 foi mantido durante os treinamentos, sendo que nenhuma rede chegou a alcançar este valor. Porém, as redes treinadas estabilizavam o erro médio após um determinado tempo de treinamento, que variava dependendo da configuração utilizada. Para estabelecer um critério na leitura dos erros foi definida a seguinte regra.

Durante o treinamento foi analisado o erro médio constantemente, uma vez que a rede não apresentasse variação por mais de 250 iterações, o treinamento era interrompido e o total de iterações lido. Com base neste critério foram lidos os valores das redes com melhores desempenhos e armazenadas as configurações utilizadas para alcance destes valores.

O número de neurônios utilizados na camada de entrada e saída da rede foram baseados nos sensores e atuadores definidos na Subseção 4.1.2. Para o controle da torre foram definidos 5 entradas conforme os sensores e 3 saídas com base nos atuadores.

Para início das simulações foram importados os dados de controle da torre na ferramenta Neuroph Studio. Após a importação foram realizadas diferentes simulações na tentativa de identificar o número de camadas intermediárias, utilizando variações de uma a três camadas e a quantidade de neurônios contidos em cada camada com variações de 8 a 64 neurônios.

A taxa de aprendizado aplicada foi variada entre 0.1 e 0.2, com taxas maiores os erros médios (MSE) também aumentaram. O termo de momentum também foi utilizado variado entre 0.3 e 0.7 para testes, sendo que alguns testes foram realizados sem o termo.

As redes com erros mais baixos entre as configurações simuladas foram armazenadas para testes com o robô e as outras foram descartadas. Na Tabela 17 são apresentadas as redes que obtiveram os melhores resultados durante as simulações. Na tabela é apresentada a topologia utilizada, a taxa de aprendizado, o termo de Momentum, o erro médio obtido e o número de iterações realizadas até a finalização do treinamento.

Tabela 17. Redes com melhores resultados para controle da torre

Nome	Estrutura	Taxa de aprendizado	Momentum	MSE	Iterações
RT1	R <sup>3</sup> 5,8,3	0,1	0,3	0,318	254
RT2	R <sup>4</sup> 5,15,15,3	0,1	-	0,312	383
RT3	R <sup>4</sup> 5,8,8,3	0,2	-	0,302	287
RT4	R <sup>4</sup> 5,10,6,3	0,1	0,3	0,311	484
RT5	R <sup>4</sup> 5,25,9,3	0,1	-	0,31	260
RT6	R <sup>5</sup> 5,10,16,6,3	0,1	-	0,31	857
RT7	R <sup>5</sup> 5,25,64,9,3	0,1	0,3	0,318	567

Em grande parte dos testes realizados o erro variou entre 0.31 e 0.49, sendo os menores valores de MSE obtidos próximos a 0.31 conforme demonstra a Tabela 17. Os erros similares para as diferentes combinações demonstraram o limite mínimo do erro médio alcançado pelo modelo MLP BP com base no arquivo de dados criado para o controle da torre.

Para o controle da torre foi realizada ao menos uma simulação utilizando as redes com erros médios mais baixos para observação das ações do robô. As redes RT3, RT5 e RT6 foram utilizadas para os testes, sendo a rede RT3 que teve o MSE mais baixo obteve o melhor resultado.

Durante os testes a rede RT5 apresentava disparos contínuos mesmo sem mira e travamentos a distâncias maiores. A rede RT6 apesar de um bom desempenho apresentou o travamento da torre em determinados momentos da batalha. Já a rede RT3 desempenhou a tarefa de mira e disparos durante todo o teste e baseados nestes testes a rede foi definida para utilização no controle da torre. Os resultados do treinamento são apresentados na Subseção 4.5.2.

## 4.5.2 Resultados dos treinamentos

Nesta Subseção são apresentados os resultados e configurações das redes utilizadas para o treinamento da movimentação com base em um rede Perceptron de uma camada e da rede responsável pelo controle da torre com base em uma rede MLP utilizando o algoritmo BackPropagation.

#### 4.5.2.1 Movimentação

Desde o início do treinamento dividido em tarefas para movimentação a rede Perceptron desempenhou satisfatoriamente as ações treinadas. Os resultados obtidos a partir de todas as tarefas de movimentação demonstraram a efetividade da rede.

O treinamento final realizado com a rede Perceptron para todas as tarefas de movimentação apresentou um MSE de 0.19 além de ações correspondentes as esperadas na movimentação do robô. Para o treinamento foi utilizada uma taxa de aprendizado de 0.2.

Conforme as características da rede Perceptron a topologia utilizou 6 neurônios de entrada e 4 neurônios na camada de saída. Figura 18 apresenta a estrutura da RNA utilizada no treinamento da movimentação, caracterizando a rede Perceptron de uma única camada.

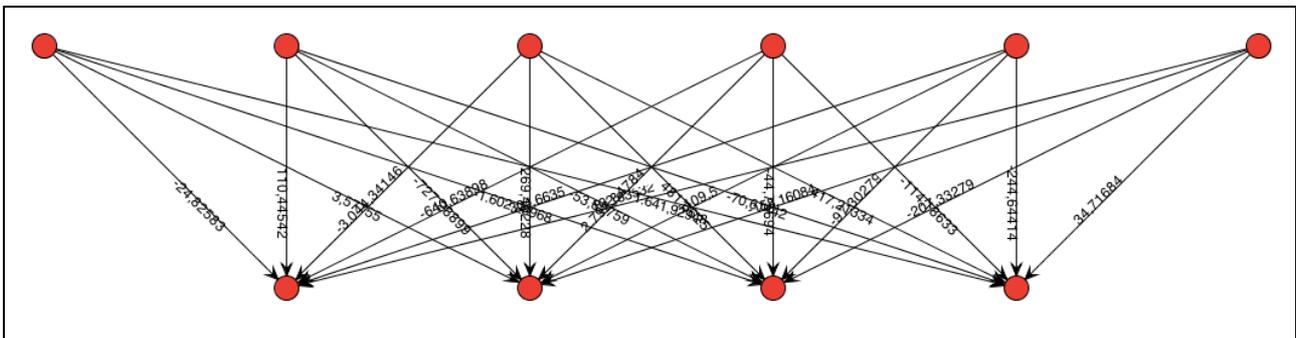


Figura 18. Topologia da rede responsável pela movimentação

Durante o treinamento foram executadas 76 iterações . O processo total de treinamento levou menos de 1 minuto para ser finalizado. A interrupção do treinamento foi realizada pelo próprio Neuroph baseada no número de exemplos disponibilizados.

Após o treinamento, a rede foi aplicada ao robô para observação das ações de movimentação executadas pelo mesmo. A aplicação da rede utilizando a interface ANNRobot é demonstrada na Seção 4.6, já os resultados das observações relativas as ações de movimentação são descritas na Seção 4.7.

#### 4.5.2.2 Controle da torre

A partir dos testes realizados na Subseção 4.5.1 foi identificada a Rede Neural a ser utilizada no treinamento final do controle da torre. Para o treinamento a partir da rede RT3 foi utilizada uma topologia com 5 neurônios de entrada, 2 camadas ocultas com 8 neurônios cada e uma camada de

saída com 3 neurônios (5-8-8-3). A taxa de aprendizado utilizada foi de 0.2 sem a utilização do termo de Momentum e o algoritmo utilizado foi o Backpropagation.

A Figura 19 apresenta a estrutura da RNA utilizada no treinamento do controle da torre, caracterizando a rede Multi-Layer-Perceptron.

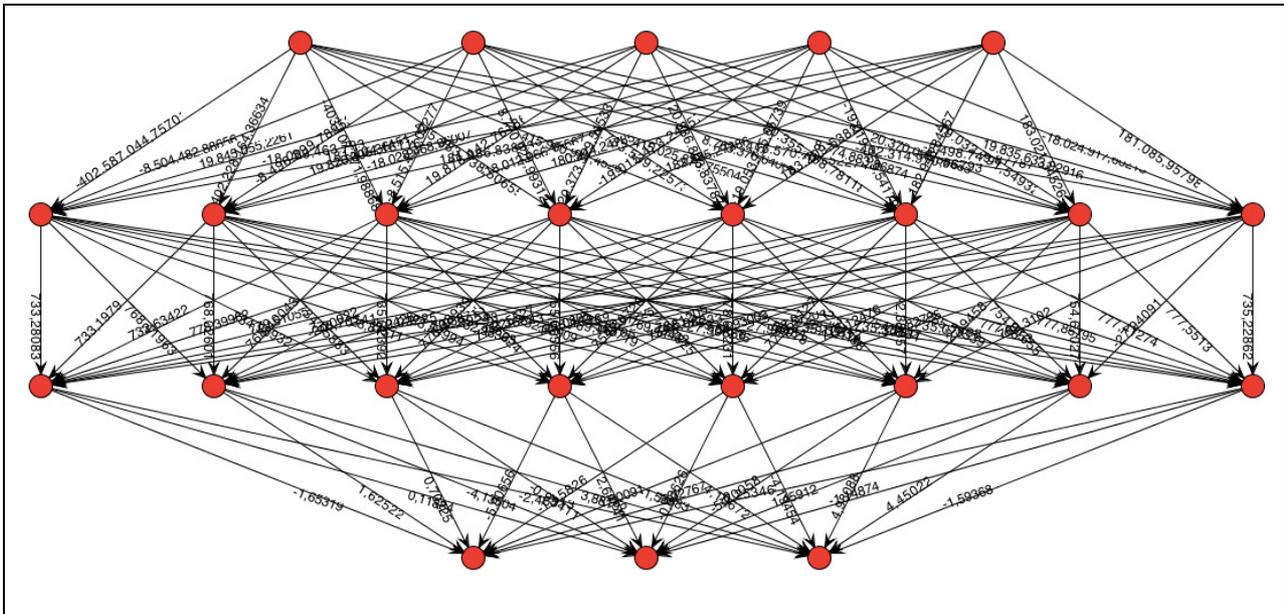


Figura 19. Topologia da rede responsável pelo controle da torre

Durante o treinamento da rede responsável pelo controle da torre foram realizadas 287 iterações que levaram cerca de 2 minutos para serem executadas, resultando em um MSE de 0.3.

Após o treinamento, a rede foi aplicada ao robô para observação das ações de controle da torre executadas pelo mesmo. A aplicação da rede utilizando a interface ANNRobot é demonstrada na Seção 4.6, já os resultados das observações relativas as ações de controle da torre são descritas na Seção 4.7.

## 4.6 Utilização da rede treinada

A interface ANNRobot foi criada tanto para a leitura dos dados do jogador quanto para a execução da rede treinada. A interface foi definida para que na utilização da Rede Neural fossem acionadas as mesmas tarefas disparadas pelo jogador. Para que isso fosse possível os ângulos e distâncias utilizadas na execução das tarefas de leitura foram utilizadas também na execução.

No entanto dois ajustes foram necessários para execução da rede de maneira correta. Um dos ajustes foi realizado devido aos ciclos de atualização do Robocode. Os ciclos realizavam a leitura dos dados gerados pela rede de acordo com a atualização do ambiente e com isso a próxima tarefa era lida antes do final da execução da tarefa que estava sendo executada. Este problema não ocorre na leitura, pois o disparo da tarefa é realizado a partir do evento gerado pelo jogador com os direcionais do teclado. Para resolver o problema foi gerado um atraso baseado no número de FPS (*Frames* por segundo) do Robocode para que a tarefa fosse executada até o fim. No entanto, este foi um problema ocorrido somente na tarefa de movimentação.

Na tarefa de controle da torre apesar da mira e disparos serem executadas em tempo real, o ângulo de giro da torre é baseado no controle do mouse e não tem um ângulo definido. Porém, o valor utilizado para a movimentação da torre foi baseado no número de FPS (*Frames* por segundo) para que a cada leitura do Robocode a movimentação da arma fosse executada com fluidez, pois na execução de ângulos baixos que não respeitem os FPS a arma se movimentava em câmera lenta.

## 4.7 ANÁLISE DO AGENTE TREINADO

Esta seção tem por objetivo avaliar o robô treinado com os dados gerados pelo jogador, observando se as ações executadas pelo robô refletem aspectos de seu comportamento.

A primeira etapa foi realizada para detectar erros primários, relativos a movimentos sem objetividade, trancamento em paredes ou quaisquer outros comportamentos que deixem óbvios problemas nas ações do personagem. Os problemas comumente identificados durante a etapa de treinamento variavam desde corridas em uma mesma direção, giros contínuos do robô e da torre, trancamentos sucessivos e movimentações repetidas nas mesmas direções.

Em 50 batalhas realizadas não foram identificados nenhum dos problemas comumente identificados durante a etapa de treinamento. O agente utilizou a mira e realizou disparos durante todas as batalhas e se movimentou constantemente pelo cenário.

Posteriormente foi avaliado se o agente realiza todas as tarefas treinadas pelo jogador. A Tabela 18 apresenta os resultados obtidos para cada uma das tarefas.

Tabela 18. Resultados das tarefas treinadas para 50 batalhas

<b>Tarefa</b>	<b>Resultado</b>
Perseguição e defesa	Tarefa realizada em todas as batalhas
Localização do inimigo	Tarefa realizada em todas as batalhas
Reações em relação ao cenário	Nenhum trancamento em todas as batalhas
Mira	Tarefa realizada em todas as batalhas
Disparo	Tarefa realizada em todas as batalhas

Outra observação importante está ligada a avaliação do comportamento, em que determinadas características observadas no agente treinado podem refletir nos resultados. Portanto, antes de realizar qualquer comparação é importante situar o nível de habilidades do jogador com base na avaliação de comportamento. Neste caso o jogador humano se auto classificou com o nível 2 de experiência “Pouca familiaridade com jogos (Já jogou / Joga esporadicamente)”.

Este nível de experiência e o questionário utilizado para a avaliação do comportamento podem ser visualizados na Subseção 4.8.1 . O nível é importante para justificar que a qualidade do agente, mesmo que mais baixa em níveis competitivos, são reflexos da baixa experiência do treinador.

Com base nas observações realizadas entre as batalhas, também foi possível identificar aspectos em comum entre o agente treinado e o jogador, como os padrões de movimentação e disparo, padrões estes que também foram identificados pelos jogadores na avaliação de comportamento.

Porém em determinadas situações entre as batalhas observadas o agente apresentou erros nos disparo mesmo quando o oponente se encontrava parado o que não ocorreu com o jogador. Além disso, quando o oponente passava a repetir suas ações de movimentação, reações similares na movimentação do agente treinado também foram observadas.

Apesar de não serem identificados trancamentos, em algumas batalhas o robô parecia repetir as manobras na tentativa de desviar do oponente quando era encurralado, esta característica foi identificada em duas partidas, apesar disso o robô conseguiu sair da situação de encurralamento em ambos os casos.

O problema de trancamentos é comum em trabalhos similares, em alguns casos tratados com algoritmos específicos para evitar estes trancamentos e em outros apenas tratados como problemas comuns na implementação da movimentação dos agentes. Um dos méritos do agente treinado é

justamente o de se movimentar livremente pelo cenário contra qualquer um dos oponentes utilizados. Em nenhuma das 50 simulações realizadas o robô ficou sem ação ou trancado em alguma das paredes do cenário.

## 4.8 AVALIAÇÃO DO COMPORTAMENTO MAPEADO

Após a implementação do robô é necessário avaliar se o agente reflete o comportamento humano ao qual foi mapeado.

A identificação do melhor método de avaliação foi realizada a partir de trabalhos similares abordados na Seção 3 . Entre os melhores modelos de avaliação identificados está o método aplicado por Bauckage *et al* (2007) e Gorman *et al* (2006b), em que é avaliado o nível de percepção de jogadores. Segundo os autores a maneira mais óbvia de determinar o grau de percepção dos jogadores é através da condução de uma pesquisa.

Como identificado na revisão citada, não há um método padrão para realização desta avaliação do comportamento aplicado ao agente, a não ser pela condução da pesquisa de percepção que tenta justamente definir um *framework* adequado para esta avaliação. Por isso o método definido para avaliação do comportamento utilizado neste trabalho é baseado em uma variação do *framework* de testes utilizado por Bauckage *et al* (2007) e Gorman *et al* (2006b).

Na montagem do *framework* foram apontados fatores importantes para definição de um modelo consistente, pois a avaliação do comportamento humano através de questionários aplicados para obter esta percepção, pode levantar algumas questões sobre a subjetividade e o nível de experiência do avaliador.

Por isso a montagem do *framework* foi realizada de forma a tentar minimizar estes problemas. Em contrapartida estas questões são importantes e devem ser ponderadas na realização da análise dos resultados obtidos.

Através deste questionário é possível verificar se o comportamento aplicado ao agente através da técnica de RNA é de fato percebido pelos jogadores, para que as hipóteses desta pesquisa possam ser confirmadas.

Nas próximas seções será descrito o instrumento de avaliação utilizado na pesquisa e a definição do experimento aplicado.

#### **4.8.1 Instrumento de avaliação**

O instrumento de avaliação foi idealizado para ser disponibilizado através de um navegador Web e consiste em dois formulários que compõem as duas etapas do teste. Como já citado, ambos os formulários são baseados no framework abordado por Bauckage *et al* (2007) e Gorman *et al* (2006b).

Na primeira etapa é disponibilizado um questionário que solicita informações sobre o nível de experiência do avaliador em jogos. Para isto são disponibilizados 5 diferentes níveis para seleção. As questões apresentadas são demonstradas a seguir:

1. Nunca jogou, joga raramente, ou nunca conheceu;
2. Pouca familiaridade com jogos (Já jogou / Joga esporadicamente);
3. Joga ocasionalmente (Mensalmente / A cada par de meses);
4. Joga regularmente (Semanalmente); e
5. Joga frequentemente (Diariamente).

Respondida a primeira etapa, os jogadores passam para uma série de páginas, em que cada página apresenta um vídeo com a demonstração de uma batalha realizada no Robocode. O avaliador então deve responder aos questionamentos apresentados em cada página a partir dos vídeos assistidos.

Em cada uma das seções apresentadas são apresentados dois robôs em batalha até a destruição de um deles. Em cada batalha são utilizadas diferentes combinações entre o agente treinado com o comportamento humano, o agente determinístico e o próprio jogador para avaliação. Ao avaliador é feito o questionamento sobre o comportamento de um dos robôs em cada uma das batalhas apresentadas para indicar se ele acredita que o robô avaliado é um jogador humano ou um personagem artificial.

Nas páginas de avaliação do comportamento disponibilizadas, não existe limite de tempo para conclusão das respostas e os vídeos podem ser assistidos mais de uma vez pelo avaliador. Também é indicado qual dos robôs está sendo avaliado no momento da pergunta sobre o comportamento observado, onde são apresentados os itens para resposta. As questões disponibilizadas em conjunto com cada uma das batalhas gravadas são as seguintes:

1. Humano;
2. Provavelmente Humano;
3. Não sei;
4. Provavelmente Artificial; e
5. Artificial.

A seleção destas repostas representa a questão principal da pesquisa e deverá ser utilizada para que seja possível formar o índice de credibilidade do comportamento humano. A formulação deste índice deve servir tanto para avaliar se o comportamento humano foi reconhecido no agente mapeado com as ações do jogador, como para realizar a comparação entre a abordagem determinística e não determinística.

Em cada uma das páginas em que são apresentados os vídeos e questionamentos sobre a avaliação do comportamento, também é disponibilizado ao avaliador um campo adicional para comentários para que ele possa explicar sua escolha.

Em casos em que a resposta selecionada indicar que se acredita que o robô seja artificial, também será perguntado o quão 'humano' o avaliador considera o robô em uma escala de 1 a 10. Como explica Bauckage *et al* (2007) os valores obtidos a partir desta resposta são subjetivos e não devem ser utilizados no índice criado para avaliação do comportamento. Por isso, este valor foi obtido apenas como informações adicionais na análise dos resultados para verificar as diferentes opiniões dos avaliadores sobre os diferentes agentes.

A partir das questões demonstradas que devem ser aplicadas em conjunto com os vídeos das batalhas, fica caracterizado o instrumento utilizado para a avaliação do comportamento.

## 4.8.2 Design do experimento

A primeira etapa do experimento de avaliação do comportamento consiste na realização de batalhas utilizando 3 diferentes abordagens. O Robô mapeado com o comportamento humano, dois robôs utilizando abordagens determinísticas e um robô controlado pelo jogador humano.

Para realização das comparações com o agente treinado a partir do comportamento do jogador, foram selecionados dois robôs utilizados em campeonatos especializados, entre as mais de 3000 implementações disponíveis no site RoboWiki. Os robôs, conhecidos como DrussGT e DustBunny, tem o código fonte disponibilizado para avaliação e utilizam uma implementação baseada em regras determinísticas como base para seu funcionamento.

A intenção de utilização do DrussGt é fazer com que o avaliador observe uma abordagem determinística efetiva em combate que realiza suas ações de acordo com a movimentação do oponente, tornando mais difícil a identificação de comportamentos repetitivos.

Além do DrussGT foi selecionado outro robô competidor chamado DustBunny, considerado menos efetivo em combate do que o DrussGT segundo a classificação do site RoboWiki. O DustBunny também realiza suas ações de acordo com a movimentação do oponente. A diferença entre os dois está apenas na efetividade em destruir o oponente.

Para a execução das batalhas foram definidas as seguintes regras:

- Somente dois robôs podem se enfrentar em cada batalha; e
- A batalha só é finalizada quando um dos robôs é destruído.

Foram utilizadas 3 batalhas com cada combinação, em que os robôs se enfrentaram diretamente, totalizando 6 partidas a serem assistidas pelos avaliadores. As combinações utilizadas foram as seguintes:

- Robô com abordagem de RNA contra o determinístico DrussGT;
- Robô com abordagem de RNA contra o jogador humano; e

- Robô com abordagem de RNA contra o determinístico DustBunny.

Para realização da segunda etapa da avaliação, foram gravadas as sessões das batalhas realizadas. Cada robô recebeu uma cor diferente em cada batalha, o sistema de cores serve somente para controle do experimento e são modificados em cada sessão.

Realizadas as gravações das batalhas e definidos os detalhes da avaliação foi finalizada a primeira etapa do experimento. Na segunda etapa cada avaliador visualizou as 3 partidas gravadas e respondeu aos questionamentos a cada batalha apresentada.

Na aplicação do questionário foi solicitada a avaliação de um robô por vez para adequar o experimento ao framework utilizado, em que apenas um dos agentes é avaliado a cada partida. Nesta segunda etapa os avaliadores também responderam a uma questão inicial relativa ao seu nível de experiência em jogos e após este questionamento foram realizadas as avaliações de percepção do comportamento para cada abordagem.

A avaliação em si consiste na aplicação do framework adaptado de Bauckage *et al* (2007) e Gorman *et al* (2006b) em que a cada partida assistida pelo avaliador o mesmo realiza sua avaliação sobre o comportamento do agente que está sendo demonstrado naquela batalha.

Para realizar esta avaliação foram disponibilizados os questionários através de páginas Web, criados a partir da ferramenta Google Forms. Junto aos questionamentos é disponibilizado o link contendo o vídeo da batalha entre os robôs que devem ser avaliados. A Figura 20 demonstra um exemplo de uma das telas que são apresentadas ao avaliador:

### Avaliação de comportamento

**\*Obrigatório**

Para responder as questões apresentadas, você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=LqMNgtHLsJA>

**Com foco no robô de cor "branca", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

1. Humano  
 2. Provavelmente Humano  
 3. Não sei  
 4. Provavelmente Artificial  
 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô branco \***

Ações previsíveis  
 Ações imprevisíveis  
 Em alguns momentos parece imprevisível em outros não  
 Busca desviar de ataques  
 Não apresenta resposta efetiva aos ataques  
 Realiza ações de forma dinâmica  
 Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô branco**

**Caso tenha considerado o robô branco como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**  
(10 mais próximo do humano, 1 menos próximo)

1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10

Figura 20. Tela de exemplo do questionário de avaliação

O avaliador então assiste ao vídeo e responde as questões solicitadas. Assim que finalizada a avaliação de um dos robôs, outro vídeo é apresentado com os mesmos questionamentos até que todos os robôs sejam avaliados.

Os questionários foram idealizados para funcionar com escalas dos níveis de comportamento para que estes dados numéricos possam ser utilizados na avaliação estatística. Também são solicitadas duas informações adicionais a cada avaliação, sendo que estas informações não fazem parte do processo estatístico e são utilizadas apenas para conhecer melhor a opinião dos avaliadores.

O questionário completo utilizado na avaliação pode ser visualizado no Apêndice A.

## 4.9 CONSIDERAÇÕES

Na etapa de desenvolvimento, foram descritas inicialmente todas as variáveis necessárias para definição dos aspectos de comportamento do jogador humano que deveriam ser mapeados. A definição destas variáveis foi baseada em trabalhos similares e na análise do ambiente Robocode.

Na sequência foi realizada a descrição do modelo de treinamento utilizado a partir da divisão das ações do jogador em tarefas, no qual também foram abordadas questões que levaram a criação deste modelo.

Com base nas ações do jogador foram gerados os arquivos de treinamento a partir de cada uma das tarefas definidas. Nesta etapa foi descrita a interface utilizada pelo jogador para obtenção dos dados, além da descrição do número de batalhas realizadas e oponentes enfrentados. Durante a geração de dados pelo jogador também foram definidos os critérios para formação dos arquivos de movimentação e controle da torre.

Na última etapa antes do treinamento foi descrito o tratamento realizado nos dados obtidos para balanceamento dos exemplos e junção das tarefas relativas a movimentação.

Na etapa de treinamento foram demonstrados os testes realizados com diferentes combinações para identificação da melhor rede a ser utilizada no treinamento final do agente com base nas ações do jogador. A partir dos testes realizados foram demonstrados os resultados obtidos.

Após o treinamento ainda foram descritos os ajustes necessários na interface ANNRobot para utilização da rede treinada no Robocode. Depois da aplicação da rede foi realizada a análise do agente treinado para identificar problemas na execução das tarefas de movimentação e controle da torre. Foram realizadas 50 batalhas de testes para avaliar se todas as tarefas treinadas foram executadas pelo agente. Também foram realizadas observações sobre o comportamento do agente nas batalhas.

Enfim foi descrito o método de avaliação a ser utilizado para identificar o comportamento humano no agente treinado perante as abordagens determinísticas e o próprio jogador.

## 5 RESULTADOS

Para responder aos questionamentos desta pesquisa foram realizados experimentos com 3 grupos de pessoas da área de tecnologia. Os experimentos foram realizados em uma única etapa, onde foi aplicado o questionário para avaliação do comportamento com cada um dos grupos.

Participaram dos experimentos alunos do curso de Jogos Digitais, participantes dos laboratórios de pesquisa do curso de Ciência da Computação e profissionais da área de tecnologia da informação, todos da UNIVALI.

Os grupos foram definidos apenas pela relação com a área de tecnologia ou diretamente com a área de jogos a fim de se obter amostras com participantes de diferentes níveis de experiência com jogos. Os diferentes grupos foram tratados na avaliação como uma mesma população, sendo divididos de acordo com sua experiência.

O nível é solicitado na primeira parte do questionário, a partir disso a avaliação de comportamento é realizada. Conforme definido no questionário o avaliador pode se situar em 5 diferentes níveis de experiência:

- Nunca jogou, joga raramente, ou nunca conheceu;
- Pouca familiaridade com jogos (Já jogou / Joga esporadicamente);
- Joga ocasionalmente (Mensalmente / A cada par de meses);
- Joga regularmente (Semanalmente); e
- Joga frequentemente (Diariamente).

Esta divisão é necessária para identificar se existe diferença entre as avaliações realizadas de acordo com as habilidades do jogador, ou seja, demonstrar como avaliadores com diferentes níveis de experiência identificaram as diferentes abordagens.

O questionário foi disponibilizado digitalmente pela internet através de um formulário Web, apesar disso, os testes foram aplicados de forma presencial a fim de esclarecer possíveis dúvidas durante as respostas.

No decorrer desta seção são apresentados os resultados obtidos na avaliação. Considerando os níveis de experiência são analisadas as classificações (entre mais humano ou mais artificial) de cada um dos personagens com cada uma das abordagens definidas para o teste (abordagem com Redes Neurais Artificiais, determinística e controlada pelo próprio jogador)

No total foram executadas 3 batalhas, e a partir destas batalhas foram realizadas 6 avaliações, sendo uma abordagem avaliada de cada vez. As batalhas realizadas e abordagens envolvidas foram as seguintes:

- Robô com abordagem de Redes Neurais contra o determinístico DrussGT;
- Robô com abordagem de Redes Neurais contra o jogador humano; e
- Robô com abordagem de Redes Neurais contra o determinístico DustBunny.

Para facilitar a leitura, durante toda esta seção a seguinte nomenclatura foi adotada. Para referenciar a batalha que está sendo analisada são exibidos os nomes dos robôs envolvidos na batalha, em que o primeiro robô é o que está sendo avaliado no momento. Por exemplo, na batalha entre o robô com abordagem de Redes Neurais Artificiais (RNA) contra o determinístico DrussGT, se o robô com RNA estiver sendo avaliado a referência a batalha seria realizada da seguinte maneira (RNA X DrussGT), do contrário se o avaliado na batalha fosse o DrussGT a batalha seria referenciada da seguinte maneira (DrussGT x RNA)

Na avaliação dos resultados foram aplicados alguns testes estatísticos para que fosse possível obter a medida de proximidade do comportamento humano de cada abordagem e determinar a aceitação ou rejeição das hipóteses de pesquisa.

## **5.1 ANÁLISE DAS RESPOSTAS**

Na Tabela 19, são apresentados os dados relativos às questões respondidas sobre a avaliação do comportamento com os 3 grupos em que o teste foi aplicado, totalizando 70 avaliações realizadas. O instrumento de avaliação possui 3 questões para avaliar cada abordagem, além de uma questão inicial para situar o nível de experiência do jogador. As questões podem ser visualizadas no apêndice A.

É importante ressaltar que a Tabela 19 mostra apenas os dados relativos ao nível do jogador e as questões de avaliação do comportamento, outras questões complementares não entram na análise estatística não são consideradas nesta primeira etapa.

Tabela 19. Respostas relativas a avaliação do comportamento.

Avaliador	Nível de exp. em jogos	RNA x DrussGT	DrussGT x RNA	Humano x RNA	RNA x Humano	RNA x DustBunny	DustBunny x RNA
1	4	2	4	3	3	1	5
2	2	5	1	3	3	4	3
3	3	4	2	5	2	2	5
4	5	2	4	4	2	4	2
5	4	4	4	4	4	4	4
6	3	2	4	1	1	5	5
7	5	4	5	2	4	5	3
8	5	5	1	4	4	1	5
9	4	5	2	5	5	1	4
10	5	3	4	4	3	2	5
11	4	3	3	3	3	4	3
12	5	2	4	5	5	5	5
13	5	5	5	5	5	1	3
14	5	5	5	2	5	2	5
15	5	4	2	2	3	4	1
16	4	5	2	5	5	2	5
17	4	4	2	4	3	3	2
18	2	3	3	2	4	4	2
19	5	5	4	2	3	2	3
20	5	5	5	1	4	5	5
21	4	5	5	2	5	5	5
22	5	5	2	2	5	5	2
23	4	4	1	1	4	1	5
24	2	1	4	1	5	1	5
25	3	2	5	5	1	4	1
26	5	5	5	4	4	5	2
27	4	2	4	3	2	4	3
28	4	5	5	4	5	1	4
29	5	5	5	5	4	2	4
30	3	2	4	4	3	4	4
31	4	2	4	5	5	1	5
32	2	4	2	4	4	5	2
33	5	4	5	2	5	1	1
34	3	3	2	2	2	4	1
35	3	5	4	2	4	4	4
36	3	5	4	2	3	5	4
37	3	5	5	2	1	5	2
38	3	5	4	2	4	4	4
39	5	5	5	4	5	5	5
40	5	4	2	3	1	4	5
41	5	5	1	4	4	1	1
42	5	5	2	2	4	2	4

43	3	2	4	2	2	3	4
44	4	4	3	4	2	3	4
45	3	1	5	1	3	5	2
46	2	1	4	2	1	1	1
47	3	4	2	4	2	1	4
48	4	4	2	1	4	5	2
49	4	5	2	4	4	1	4
50	2	3	5	2	3	4	1
51	4	4	2	4	3	4	4
52	3	2	4	2	4	2	3
53	4	5	2	2	2	1	4
54	3	4	2	2	4	1	5
55	3	2	5	1	4	1	1
56	2	4	5	4	4	4	4
57	2	2	2	4	3	4	2
58	2	4	1	2	1	1	4
59	3	4	2	4	2	4	2
60	1	4	5	2	5	2	4
61	4	1	5	4	2	4	4
62	3	4	4	4	4	5	5
63	2	3	1	3	4	1	3
64	2	2	4	4	2	2	2
65	3	4	4	2	2	4	4
66	1	5	4	4	4	4	2
67	4	4	5	2	4	4	4
68	5	2	5	2	1	1	5
69	1	2	5	5	2	5	1
70	4	2	4	4	3	1	4

Para se obter os resultados da avaliação do comportamento para cada abordagem foi preciso levar em conta a maneira como o experimento foi desenvolvido, a fim de identificar os tipos de testes estatísticos que deveriam ser aplicados.

Conforme descrevem Siegel e Castellan Jr. (2006) é importante analisar a maneira como os dados são mesurados a fim de identificar adequadamente sua implicação para interpretação dos testes estatísticos.

Primeiramente, foi analisada a maneira como é mensurada a questão principal da pesquisa que solicita a avaliação do comportamento observado pelo avaliador. Esta questão está distribuídas em forma de uma escala ordinal com 5 níveis. O primeiro nível identifica um comportamento avaliado como humano e o último nível um comportamento artificial. Assim como a questão principal, a questão sobre os níveis de experiência dos jogadores também são apresentados em um escala ordinal com 5 opções.

Siegel e Castellan Jr. (2006) explicam que a estatística mais apropriada para descrever a tendência central dos valores em uma escala ordinal é através da mediana, pois relativamente a distribuição dos valores, a mediana não é afetada por mudanças em valores acima ou abaixo dela, desde que os valores de limites máximos e mínimos permaneçam os mesmos. Para testes em escalas ordinais devem ser considerados testes não paramétricos, algumas vezes chamados de estatísticas de postos.

Além disso, a principal pergunta do questionário de avaliação determina se o agente foi avaliado como mais humano ou mais artificial, o que classifica estes dados inerentemente em postos. Isto permite que os dados sejam categorizados como “mais” ou “menos” e conforme afirmam Siegel e Castellan Jr. (2006) eles não podem ser tratados por métodos paramétricos.

Uma vez que os dados foram mensurados em uma escala ordinal, foi realizado um teste inicial para verificar se existia relação entre a experiência do jogador e a avaliação de comportamento observado pelo mesmo, o que permitiu identificar se jogadores com diferentes níveis de experiência apresentam avaliações similares para cada abordagem (em cada cruzamento apresentado) e podem ser considerados como um único grupo para avaliação, ou se existem diferenças significativas que obriguem a divisão da avaliação de acordo com estes níveis. Esta avaliação inicial serve apenas para identificar se o nível de experiência dos jogadores influenciou na avaliação ou não.

Para tanto foi utilizado o teste de correlação de postos de Spearman. O teste de Spearman analisa a relação entre duas variáveis, desde que ambas sejam medidas em uma escala no mínimo ordinal, de modo que os objetos ou indivíduos do teste possam ser dispostos em postos em duas séries ordenadas (SIEGEL; CASTELLAN JR, 2006).

O coeficiente de correlação do teste de Spearman mede a intensidade da relação entre as variáveis ordinais, sendo que ao invés do valor, é considerada apenas a ordem das observações. Este coeficiente pode variar de -1 a 1 e quanto mais próximo aos extremos, maior é a correlação entre as variáveis. Os valores negativos da correlação apontam que as categorias mais elevadas de uma variável são associadas a categorias mais baixas da variável comparada.

Se nenhuma das variáveis apresentarem correlação, o teste Spearman assume que não existe diferença significativa entre os níveis de experiência dos avaliadores em jogos e os níveis de

comportamento atribuídos pelos mesmos para as diferentes abordagens (isto é, são independentes e a avaliação não precisa considerar os níveis).

A Tabela 20 demonstra o resultado do teste de Spearman para cada cruzamento. Na tabela os valores  $p$  definem se as duas variáveis do teste são ou não correlacionadas do ponto de vista estatístico. Em geral, se esse valor for menor que 0,05, considera-se que a correlação entre as variáveis trabalhadas é significativa. O teste de Spearman também retorna um valor  $R$  do coeficiente de correlação, que conforme citado é utilizado para medir a força e o sentido da relação entre as variáveis.

Tabela 20. Resultados do teste de Spearman

<b>Cruzamento</b>	<b>Spearman <math>R</math></b>	<b>Valor <math>p</math></b>
RNA1 x DET1	0,332123	0,004971
DET1 x RNA1	0,076102	0,531209
HMN x RNA	0,096459	0,426987
RNA2 x HMN	0,243600	0,042140
RNA3 x DET2	-0,051829	0,670029
DET2 x RNA3	0,250891	0,036176

Nos resultados apresentados na Tabela 20 pode ser observado que 3 cruzamentos apresentaram correlações significativas, com valor  $p$  menor do que 0,05 (nível de significância). É importante observar que não são correlações fortes, que seriam mais próximas de 1, mas são correlações positivas, no sentido de que quanto maior o nível de experiência do jogador, maior foi a tendência de atribuir escores de avaliação similares.

Com este resultado foi possível identificar que existe uma diferença significativa entre os níveis de experiência para alguns cruzamentos (o que compreende todas as batalhas). A partir disso, é possível afirmar que o nível influenciou na avaliação do comportamento realizada, portanto os testes foram realizados considerando os diferentes níveis de experiência.

Considerando as correlações identificadas no teste de Spearman, resolveu-se agrupar os jogadores em dois grupos, de acordo com sua experiência em jogos. Pela escala do questionário, os escores 1, 2, 3 ficaram no grupo dos “menos experientes”, com 4 e 5 no grupo dos “mais experientes”.

Analisando como os jogadores foram distribuídos na Tabela 21, observou-se que a proporção de jogadores em cada grupo ficou homogênea, 32 e 38.

Tabela 21. Distribuição dos jogadores por nível

<b>Nível</b>	<b>Descrição</b>	<b>Quantidade</b>
1	Nunca jogou, joga raramente, ou nunca conheceu	3
2	Pouca familiaridade com jogos (Já jogou / Joga esporadicamente)	11
3	Joga ocasionalmente (Mensalmente / A cada par de meses)	18
4	Joga regularmente (Semanalmente)	19
5	Joga frequentemente (Diariamente)	19

O formato final dos grupos definidos de acordo com os níveis dos jogadores é apresentado na Tabela 22.

Tabela 22. Grupos definidos

<b>Níveis</b>	<b>Descrição</b>	<b>Quantidade</b>
1,2,3	Menos Experientes	32
4,5	Mais Experientes	38

## 5.2 COMPARAÇÃO ENTRE AS ABORDAGENS

Os testes para identificar as abordagens avaliadas como mais humanas foram realizadas com base nos grupos (menos experientes e mais experientes) definidos a partir do teste de Spearman.

Para determinar quais das abordagens foram avaliadas como mais humanas, foi preciso comparar qual robô poderia ser considerado mais humano em relação aos outros, a partir da escala de avaliação dos níveis de humanidade. Com isso é possível identificar se há diferença nos escores de 1 a 5 atribuídos pelos jogadores.

Como o mesmo jogador atribuiu o nível de humanidade para diferentes abordagens em cada batalha apresentada (ora avaliando uma abordagem e ora avaliando outra), foi preciso aplicar um teste estatístico pareado para observar a opinião do jogador para cada abordagem avaliada na batalha. Por exemplo, na primeira batalha o jogador realizou a avaliação da abordagem de RNA frente ao determinístico DrussGT (RNA X DrussGT). Em seguida o mesmo jogador realizou a avaliação da abordagem determinística DrussGT frente a abordagem de RNA (DrussGT X RNA), por isso a aplicação do teste pareado, para comparar duas amostras a cada batalha. Além disso, a pergunta principal de pesquisa visa identificar como a abordagem de RNA se sai frente as abordagens determinísticas e em sua comparação com o jogador humano, o que também justifica a realização do teste pareado.

O teste estatístico indicado para comparar duas amostras pareadas em um nível ordinal (teste não paramétrico) é o teste de Wilcoxon pareado. O teste de Wilcoxon avalia a diferença no valor central de duas amostras a partir da mediana, o que permite identificar o nível de humanidade inferido para cada abordagem.

O teste de Wilcoxon é muito útil para análises de dados comportamentais e permite ao avaliador identificar a diferença entre quaisquer dois scores surgindo de dois pares quaisquer (SIEGEL; CASTELLAN JR, 2006).

A partir da aplicação do teste de Wilcoxon foram obtidos os valores das medianas para cada abordagem, sempre avaliadas par a par, a fim de identificar se a diferença entre os níveis de humanidade obtidos entre as abordagens foi significativa com 95% de confiança ( $\alpha = 0.05$ ). A Tabela 23 demonstra os resultados obtidos para cada abordagem entre os jogadores menos experientes e a Tabela 24 demonstra os resultados entre os jogadores mais experientes.

Muito embora o teste se baseie na mediana, optou-se por apresentar também a média e o desvio padrão na tabela dos resultados do teste, por serem duas medidas amplamente utilizadas, uma para se avaliar a tendência central e a outra a variabilidade presente nas respostas. Estas informações possibilitam um pouco mais de compreensão na análise dos resultados, como por exemplo, quando ocorrem empates nos valores da mediana e quando se quer discutir também se houve muita variabilidade nas respostas.

Tabela 23. Resultado das comparações entre as diferentes abordagens para jogadores menos experientes

<b>Abordagens</b>	<b>Média</b>	<b>Mediana</b>	<b>Desvio padrão</b>	<b>Valor p</b>
RNA	3,22	3,50	1,31	0,348772
DrussGT	3,50	4,00	1,34	
Humano	2,78	2,00	1,26	0,648229
RNA	1,26	3,00	1,23	
RNA	3,28	4,00	1,51	0,526682
DustBunny	3,00	3,00	1,41	

Para o grupo de jogadores menos experientes, em todos os 6 testes realizados para avaliar cada abordagem, os valores  $p$  foram bem maiores do que o nível de significância adotado (0.05) o que não permite afirmar que existe uma diferença significativa entre as abordagens avaliadas. Como pode ser observado a partir dos valores da mediana, os jogadores com menos experiência em jogos

tenderam a ficar indecisos, não atribuindo valores significantes para o nível mais humano ou artificial para nenhuma abordagem.

Tabela 24. Resultado das comparações entre as diferentes abordagens para jogadores mais experientes

Abordagens	Média	Mediana	Desvio padrão	Valor p
RNA	3,95	4,00	1,23	0,118798
DrussGT	3,47	4,00	1,43	
Humano	3,24	4,00	1,26	0,103554
RNA	3,66	4,00	1,19	
RNA	2,82	2,50	1,66	0,017331
DustBunny	3,71	4,00	1,29	

Já para o grupo de jogadores mais experientes a batalha entre a abordagem de RNA e o determinístico DustBunny apresentou um valor  $p = 0.017$ , menor do que o nível de significância adotado, o que demonstra que os valores resultantes da comparação entre estas abordagens foi estatisticamente significativo. Com isso é possível afirmar através do valor das medianas que a abordagem de RNA foi avaliada como mais próxima ao comportamento humano, enquanto o determinístico DustBunny foi avaliado como mais próximo ao artificial.

Porém, para as outras batalhas analisadas, os resultados foram similares aos obtidos pelos jogadores menos experientes, em que os 4 testes realizados para avaliar as demais abordagens, retornaram valores  $p$  maiores do que o nível de significância adotado (0.05). Estes resultados também não permitiram afirmar que existe uma diferença significativa entre as demais abordagens avaliadas.

Assim como para os jogadores menos experientes, pode ser observado a partir dos valores da mediana, que os jogadores mais experientes também tenderam a ficar indecisos, não atribuindo valores significantes para o nível mais humano ou artificial para nenhuma abordagem. Em contrapartida, os valores de significância resultantes para os jogadores mais experientes foram mais próximos do nível de significância adotado.

### 5.3 RESPOSTAS COMPLEMENTARES

Além da análise principal do experimento relativa à avaliação de comportamento, foram respondidas questões complementares que possibilitaram uma melhor compreensão das avaliações

realizadas. Estas questões não foram tratadas na análise estatística principal e nessa seção são descritas para utilização na discussão sobre os resultados obtidos.

A análise dos dados complementares foi realizada levando em consideração o nível de experiência dos jogadores, já que na primeira etapa da análise dos resultados foi identificado que o nível impactou em suas respostas. A divisão pelos níveis de experiência também permite relacionar as respostas com a análise inicial realizada permitindo maior clareza nas considerações sobre os resultados obtidos.

A primeira questão relativa às informações complementares abordou as respostas obtidas quando o agente avaliado foi considerado como artificial. Neste caso, foi solicitado ao jogador que atribui-se uma nota (de 1 a 10), do quão semelhante ao humano o agente avaliado estava.

A Tabela 25 demonstra as médias das notas obtidas para cada abordagem, porém, somente são contabilizados os resultados quando o robô foi avaliado como artificial. Uma informação que deve ser apontada em relação a este campo fica a cargo do entendimento da questão. Como alguns avaliadores responderam a questão mesmo quando o comportamento não era artificial, duas conclusões podem ser tiradas. A nota foi atribuída involuntariamente, ou a interpretação da questão foi incorreta. Nestes casos em que a nota foi atribuída para um agente não considerado artificial a nota foi desconsiderada.

Tabela 25. Média das notas para atribuição artificial de comportamento

	Média das notas					
	RNA x DrussGT	DrussGT x RNA	Humano x RNA	RNA x Humano	RNA x DustBunny	DustBunny x RNA
Média Menos Experientes	5	5,37	4	4,27	3,86	4,2
Média Mais Experientes	4,12	5,67	4	4,15	2,75	3,42

Esta avaliação permitiu identificar que os jogadores consideraram que o determinístico DrussGT aparentava mais traços humanos do que outras abordagens quando considerado artificial, sendo que o RNA teve uma média próxima quando enfrentou o mesmo DrussGT. Neste caso o nível de experiência alterou os valores mas manteve a ordem das médias.

O DustBunny também apresentou uma média um pouco maior do que o RNA quando considerados artificiais na batalha em que se enfrentaram.

O jogador humano e a abordagem com RNA tiveram médias próximas mesmo quando considerados artificiais. O resultado era esperado devido ao fato da abordagem com RNA estar diretamente ligada ao jogador, afinal o treinamento da RNA foi baseada nas ações deste jogador. O resultado pode ser observado tanto para avaliadores menos experientes quanto mais experientes.

Com exceção das médias próximas entre humano e RNA a avaliação desta questão não trouxe grandes contribuições para a avaliação de comportamento. Porém é possível perceber que houve poucas variações com base no nível dos avaliadores para este quesito.

Outro dado complementar utilizado para discussão sobre os resultados foi baseado no campo de comentários. O campo contou com descrições pré-determinadas, em que o jogador podia selecionar algumas *tags* para justificar sua escolha, além da descrição livre, na qual o jogador poderia elaborar seu comentário sobre a abordagem analisada.

As descrições pré-determinadas foram utilizadas para impedir que a fadiga fizesse com que os jogadores ignorassem os comentários. A decisão se mostrou acertada já que apenas 20 jogadores realizaram descrições adicionais nos comentários.

Os comentários realizados desta maneira foram agrupados pela quantidade em que foram realizados para cada cruzamento e também foram divididos de acordo com o nível de experiência dos jogadores. Tabela 26 são apresentados os resultados obtidos para o grupo dos menos experientes em jogos e na Tabela 27 os resultados do grupo dos mais experientes.

Tabela 26. Totais de comentários similares para cada abordagem entre jogadores menos experientes

Comentário	Totais de comentários similares (Menos Experientes)					
	RNA x DrussGT	DrussGT x RNA	Humano x RNA	RNA x Humano	RNA x DustBunny	DustBunny x RNA
Ações previsíveis	4	12	3	8	13	7
Ações imprevisíveis	1	4	10	7	4	6
Em alguns momentos parece imprevisível em outros não	13	3	12	7	4	6
Busca desviar de ataques	11	15	5	6	7	9
Não apresenta resposta efetiva aos ataques	5	0	5	5	2	9
Realiza ações de forma dinâmica	4	9	11	5	10	3

Em alguns momentos parece se perder	8	0	5	7	6	8
-------------------------------------	---	---	---	---	---	---

A partir de observações nos dados obtidos dos avaliadores com menor experiência em jogos, é possível destacar a identificação de “ações previsíveis” percebidas pelos avaliadores quando o robô DrussGT enfrentou a abordagem com RNA. Como o DrussGT utiliza algumas técnicas de movimentação e disparo determinísticas para garantir a efetividade dos ataques, era esperado que os avaliadores percebessem algumas repetições destas ações.

Porém a abordagem de RNA também obteve muitas marcações de ações previsíveis quando enfrentou o determinístico DustBunny. Como a abordagem de RNA também teve mais comentários como imprevisível quando enfrentou o jogador humano, é possível perceber que devido ao fato das respostas do agente com abordagem de RNA serem afetadas primariamente pelas ações do oponente, um oponente mais complexo como o jogador humano refletiu no seu comportamento atribuindo-lhe respostas mais complexas.

Outra resposta sobre ações consideradas “ora previsíveis, ora imprevisíveis”, demonstrou que os avaliadores identificaram mudanças no comportamento dos agentes avaliados durante as batalhas. Estas dúvidas foram mais identificadas na abordagem de RNA quando enfrentou o determinístico DrussGT. Por um lado isto demonstra a diversificação das ações do robô com abordagem de RNA durante a batalha, porém também demonstra que em muitos momentos a abordagem se demonstrou previsível. Como o jogador humano foi o que recebeu o segundo maior número de comentários sobre as ações ora previsíveis, ora imprevisíveis, esta informação não pode ser tratada como afirmação de que estas marcações caracterizam um jogador como artificial, a não ser que este fosse um comportamento constante durante toda a batalha.

Os apontamentos sobre “desvio de ataques” também apresentaram uma informação relevante, pois a batalha entre o agente com abordagem de RNA e o determinístico DrussGT foi a que recebeu o maior número destes comentários. O DrussGT recebeu mais comentários devido a sua efetividade em combate (o desvio é uma característica já esperada). Como o RNA recebeu o segundo maior número de comentários é possível perceber que a abordagem se adaptou ao DrussGT já que em nenhuma outra batalha os desvios foram tão destacados para o RNA.

As *tags* sobre “repostas não efetivas aos ataques” reforçam o maior nível de competitividade do DrussGT, já que o mesmo não recebeu nenhuma marcação deste tipo. Esta informação também

aponta outra questão relevante nas avaliações realizadas, onde na batalha entre RNA e DustBunny o RNA recebeu apenas duas macacões de respostas não efetivas, enquanto o oponente derrotado (DustBunny) recebeu 9. Em outras batalhas a média do RNA foi mais alta e a do DustBunny mais baixa, demonstrando que esta informação tende a beneficiar o robô mais efetivo em batalha.

As marcações para “ações realizadas de forma dinâmica” também tiveram mais ocorrências entre os robôs vitoriosos, mais uma vez demonstrando que os considerados mais dinâmicos também foram os mais efetivos.

Os avaliadores também identificaram que a abordagem de RNA pareceu se perder em alguns momentos da batalha, sendo que todos apresentaram uma média similar de respostas neste quesito, com exceção do DrussGT, que nunca recebeu esta observação. Isto demonstra que suas ações contínuas de ataques mais uma vez refletiram nas avaliações. Por exemplo, o jogador humano em alguns momentos variou sua estratégia de ataque, o que pode ter sido interpretado como “se perder durante a batalha”, já o DrussGT sempre focava a destruição do inimigo.

Em geral, para avaliadores menos experientes é possível perceber que o robô com RNA e o jogador humano foram os que mais apresentaram ações diversificadas durante as batalhas. Sendo que a efetividade de ações durante a batalha foi considerado um fator preponderante na avaliação das abordagens. Talvez a principal informação extraída desta etapa da avaliação seja relativa a maneira como os eventos das batalhas alteraram a percepção dos avaliadores, onde algumas abordagens obtiveram marcações completamente opostas quando enfrentaram diferentes oponentes.

Tabela 27. Totais de comentários similares para cada abordagem entre jogadores mais experientes

Comentário	Totais de comentários similares (Mais Experientes)					
	RNA x DrussGT	DrussGT x RNA	Humano x RNA	RNA x Humano	RNA x DustBunny	DustBunny x RNA
Ações previsíveis	11	14	13	12	14	12
Ações imprevisíveis	1	5	10	8	3	6
Em alguns momentos parece imprevisível em outros não	13	10	8	10	11	8
Busca desviar de ataques	12	13	8	4	9	7
Não apresenta resposta efetiva aos ataques	9	2	5	12	9	9
Realiza ações de forma	8	16	8	6	10	5

dinâmica						
Em alguns momentos parece se perder	8	2	11	11	3	9

Um dos dados que chamou a atenção entre os avaliadores mais experientes foi em relação as “ações previsíveis”, pois o comentário foi realizado em uma quantidade relativamente alta para grande parte das abordagens. Além disso, uma abordagem determinística e uma com base em RNA foram as que mais obtiveram comentários deste tipo. Este resultado dificulta a afirmação de que uma abordagem foi considerada mais previsível do que outra. Esta informação apenas indica que a exigência dos avaliadores mais experientes é muito mais alta.

Porém, os comentários que identificaram as abordagens como “as vezes imprevisível e outras vezes não” demonstra que as dúvidas dos avaliadores durante a análise das batalhas levou ao equilíbrio de alguns comentários como o de ações previsíveis.

Já em relação à “ações imprevisíveis” o jogador humano e a RNA que o confrontou obtiveram mais comentários sobre a imprevisibilidade das ações executadas por ambas as abordagens. Estes comentários deixam claro que o adversário influenciou consideravelmente nas escolhas dos avaliadores, uma vez que o RNA frente ao DrussGT foi considerado imprevisível apenas por 2 avaliadores. Esta informação demonstra novamente, desta vez para os avaliadores mais experientes, a similaridade das ações entre o jogador humano e a abordagem com RNA.

Em relação aos comentários sobre o “desvio de ataques”, o nível dos avaliadores não pareceu interferir nas respostas, pois assim como para os avaliadores menos experientes, se destacou a batalha entre a RNA e o DrussGT, onde a abordagem de RNA se adaptou durante a batalha apresentando mais desvios do que em outros cruzamentos, devido a competitividade do DrussGT.

A *tag* sobre “repostas não efetivas aos ataques” também demonstrou pouca variação entre os diferentes níveis dos avaliadores, onde o DrussGT foi o que recebeu menos marcações deste tipo. Em contrapartida, os avaliadores mais experientes apresentaram uma maior média de marcações deste tipo para as outras abordagens. Devido à média relativamente alta de repostas para praticamente todas as abordagens é possível afirmar que a efetividade foi relacionada mais uma vez ao agente mais competitivo, ou seja, o que menos era atingido devido ao seu padrão de

movimentos. Sendo que a efetividade não apresenta indícios de que aquele agente se movimentava de forma mais natural.

As marcações efetuadas para “ações realizadas de forma dinâmica” tiveram mais ocorrências entre os robôs vitoriosos, assim como para os avaliadores menos experientes. Porém a abordagem de RNA contra o DrussGT também apresentou uma média alta de respostas mesmo este não tendo sido o agente vencedor. Como o DrussGT recebeu a maior média de avaliações neste quesito e o RNA demonstrou se adaptar as ações do DrussGT durante esta batalha, este fator pode ter elevado as avaliações da abordagem com RNA.

Os avaliadores mais experientes também identificaram que grande parte dos robôs pareciam “se perder durante alguns momentos da batalha”. Com exceção do DrussGT e DustBunny, todos apresentaram uma média similar de respostas neste quesito. Isto reforça a avaliação dos jogadores menos experientes, sendo que neste caso as duas abordagens determinísticas receberam menos observações deste tipo. Assim como para os avaliadores menos experientes as ações contínuas de ataques das abordagens determinísticas parecem ter refletido neste quesito. Isto reforça a observação inicial de que até mesmo uma mudança de estratégia que não fosse diretamente ligada ao ataque ou desvio pode ter sido considerada como fator para identificação da abordagem como “perdida” durante alguns momentos da batalha.

Em geral, avaliadores mais experientes tiveram observações similares aos menos experientes, onde foi possível perceber que o robô com RNA e o jogador humano foram os que mais apresentaram ações diversificadas durante as batalhas. Também foi possível reforçar que a efetividade de ações durante a batalha foi considerado um fator importante na avaliação das abordagens. Os eventos das batalhas também demonstraram alterações na percepção dos avaliadores, onde algumas abordagens obtiveram marcações completamente opostas quando enfrentaram diferentes oponentes.

Independente do nível dos avaliadores foi possível perceber a similaridade na avaliação do jogador humano e da abordagem de RNA, porém a efetividade do DrussGT foi aparentemente um fator levado em consideração pelos jogadores para atribuir seu nível de comportamento. Isto indica que o nível de experiência do jogador de onde foram extraídos os dados para treinamento da abordagem de RNA deve ser considerado na análise dos resultados, pois os jogadores parecem naturalmente esperar que o jogador humano seja mais competitivo.

É importante reforçar que as respostas obtidas a partir das *tags* pré-definidas trazem detalhes que podem auxiliam no entendimento das respostas dos avaliadores, porém não podem ser usadas como parâmetros para afirmar que o robô foi considerado mais humano ou artificial devido a quantidade de respostas de uma determinada *tag*.

Como apenas 20 avaliadores realizaram comentários livres, na sequência todos são descritos, levando em consideração a avaliação realizada e o nível do jogador para um melhor entendimento. Os comentários são exibidos de acordo com cada abordagem avaliada.

### Avaliação do robô com abordagem de RNA frente ao robô determinístico DrussGT

Tabela 28. Comentários atribuídos a abordagem de RNA frente ao determinístico DrussGT

Nível	Avaliação	Comentários
2	5	Ele não apresenta uma resposta efetiva para os ataques, não toma uma decisão eficiente em relação a maioria dos ataques
2	4	Parece somente responder as ações de ataque do amarelo e sempre se baseando nas coordenadas de onde recebeu o tiro disparado pelo amarelo.
3	4	Pouca amplitude de movimentos.
3	5	Notei que o robô branco passa maior parte do tempo tentando aproximar-se do seu inimigo, e que sua manobra de afastamento aparenta ser previsível. Talvez seja apenas uma estratégia de ataque, mas a um ser humano isso é muito variável, sendo que dependendo do momento em que se encontra no jogo a estratégia muda, talvez mais defensivo e "fugindo"(afastando-se com maior distância e frequência) a medida que vai sendo atingido pelo inimigo.
3	4	Não se defende de forma efetiva, parece artificial ou um humano com mínima experiência em jogos.
3	2	Padrão nos movimentos de defesa.
4	2	O robô se movimenta de forma aparentemente não previsível, buscando desviar os ataques.
4	4	Acredito que seja Inteligência Artificial, porém, uma inteligência muito bem desenvolvida, pois, parece um Humano controlando.
4	2	Parece mais humano, pois é familiar quando a minha namorada joga comigo, por não possuir destreza.
5	2	A impressão que eu tive é que o robô branco era 'ruim de pilotagem', não de forma aleatória como uma IA seria, mas no sentido efetuar as curvas e mirar no adversário.
5	5	O Robô branco tem uma taxa de resposta ao comportamento do outro robô em menos de um segundo.
5	5	Atira na posição onde o inimigo estava e não onde vai estar
5	4	O robô branco apresenta movimento programado através de ondas senoidais
5	5	Minha opinião para melhorar o funcionamento dele, deveria ser gerado aleatoriamente (ou através de uma política de escolha), para estado dele, como por exemplo: Ofensivo, Defensivo. Ele parece ter um propósito básico de funcionamento, o que leva a uma fraca inteligência artificial.
5	4	Gostei o modo que o robô desvia das balas inimigas, em conjunto a sua movimentação.  Não gostei a forma que ele atira, ou seja, a estratégia usada sempre mira o tiro na posição em que esta o robô amarelo facilitando a sua esquiva, enquanto que o robô amarelo atira no lado do robô branco forçando a robô branco ter um bom nível de esquiva.
5	5	Todas as ações do branco parecem ser reações, não consegui ver o branco tomando a iniciativa no jogo.

Entre os comentários sobre o agente com abordagem de RNA frente ao determinístico DrussGT apresentados na Tabela 28, alguns chamaram a atenção por apontar uma questão já esperada, relativa a influência do nível de competitividade do robô treinado.

Em um dos comentários é citado que o agente parece um “humano com mínima experiência em jogos” o que é verdade já que o jogador que realizou o treinamento do agente com RNA tem

pouca experiência em jogos (nível 2). Neste caso o robô mesmo sendo descrito como humano, foi avaliado como artificial, demonstrando a influência do nível de habilidades do jogador na avaliação.

Outros dois comentários reforçam a identificação do aparente baixo nível de experiência da abordagem com RNA, onde é citado que o robô parecia “ruim de pilotagem” e no outro comentário é citado que o robô parece ser controlado por alguém que “não possui destreza”.

### **Avaliação do robô com abordagem determinística (DrussGT) frente ao robô com abordagem de RNA**

Tabela 29. Comentários atribuídos ao determinístico DrussGT frente a abordagem de RNA

Nível	Avaliação	Comentários
2	1	O comportamento dinâmico e adaptável ao comportamento do adversário
2	1	Ele parece ser humano, e usa mão de uma estratégia de disparar tiros para atrair a atenção do robô branco, enquanto espera ele responder aos tiros disparados para distrair e abre fogo efetivamente acertando o robô branco nos lado aonde ele fica desguarnecido.
3	2	Movimentação estratégica
3	4	Faz a manobra como se estivesse cercando o inimigo (anda em círculo), visando mais precisão no ataque em relação ao branco porém ainda sim desviando dos disparos.
3	2	Ataques efetivos e também se desvia melhor que o branco
4	4	Movimentos aparentemente previsíveis (anda lateralmente de um lado para o outro), porém é muito efetivo nos disparos, o que também não é comum para um humano.
4	5	O robô amarelo se comporta de uma forma padrão, sempre se movimentando em círculo em referência ao oponente.
5	4	O robô amarelo 'detectou' seu objetivo muito mais rápido do que o branco. Ou ele era um jogador experiente e que conhecia esse ambiente de jogo, ou ele era uma IA. E o comportamento dele, extremamente ofensivo o tempo todo, sugere uma IA.
5	5	O robô amarelo, como o branco, move sua base constantemente, o que difere muito de um comportamento advindo de um input humano, que tende a ter pequenos momentos de movimentação linear.
5	5	Anda em círculos a uma certa distância de um certo ângulo a um certo ângulo.
5	5	Muito ofensivo, em poucos momentos decide alguma escolha mais estratégica, porem tem bons cálculos para executar os tiros.
5	2	Apenas não gostei que não varia a sua forma de atirar.
5	2	o robô branco não dificultou muito as coisas Toma a iniciativa do jogo.

Nos comentários sobre o DrussGT frente a abordagem de RNA apresentados na Tabela 29, conforme esperado, destacam sua efetividade em combate. Em alguns comentários os jogadores perceberam suas técnicas de movimentação e disparo. Em um comentário o avaliador afirma que o robô “faz a manobra como se estivesse cercando o inimigo (anda em círculo)”. Outro comentário aponta que o robô realiza “movimentos aparentemente previsíveis (anda lateralmente de um lado para o outro), porém é muito efetivo nos disparos, o que também não é comum para um humano”. A técnica de movimentação também é percebida por outro avaliador quando aponta que “o robô amarelo se comporta de uma forma padrão, sempre se movimentando em círculo em referência ao oponente”.

Entre outros comentários ainda se destacam a afirmação de que o robô “anda em círculos a uma certa distância de um certo ângulo a um certo ângulo” e que o robô é “muito ofensivo, em poucos momentos decide alguma escolha mais estratégica”

### **Avaliação do robô controlado pelo jogador humano frente ao robô com abordagem de RNA**

Tabela 30. Comentários atribuídos ao jogador humano frente a abordagem de RNA

Nível	Avaliação	Comentários
2	3	Difícil responder em alguns momentos ele apresenta um comportamento dinâmico e em outros não
3	5	os movimentos me parecem sem estratégia.
3	2	Tem certa similaridade aos movimentos do robô branco do vídeo anterior porém com mais precisão nos disparos. E desviando-se melhor!
3	2	Tem ações para direita e esquerda, frente e para traz de maneira consciente e ataca bem.
3	1	Bateu na parede.
4	3	Difícil identificar, comportamentos similares dos robôs
4	4	Parece ter sempre o mesmo comportamento.
4	5	Percebi que várias vezes, os 2 robôs faziam os mesmo movimentos.
5	4	O robô preto parecia reagir em função do que o verde fazia, e ele executava as ações de forma melhor.
5	2	Ações suicidas como ir para frente atacando.
5	2	O robô apresenta um comportamento agressivo, tentando esmurrar o adversário sempre que o mesmo se aproxima.
5	4	Bem programado, sendo imprevisível aparenta ser mais humano do que os outros robôs.
5	2	Parece ter comportamento humano.

O agente controlado pelo jogador humano traz comentários importantes para comparação entre o jogador e o agente com abordagem de RNA treinado a partir de suas ações, conforme apresentado na Tabela 30.

Um dos comentários aponta que é “difícil identificar, comportamentos similares dos robôs”. Mais um comentário descreve, “Tem certa similaridade aos movimentos do robô branco do vídeo anterior, porém com mais precisão nos disparos. E desviando-se melhor!”. Outro avaliador afirma que “várias vezes, os 2 robôs faziam os mesmo movimentos”. Estes fatores demonstram que um dos objetivos de mapear o perfil do jogador no agente com abordagem de RNA foi identificado por alguns avaliadores.

Outros comentários que chamaram a atenção foram relativos à aparente baixa experiência do jogador. Um dos avaliadores cita que “Ações suicidas como ir para frente atacando”, que também é citado em outros comentários similares sobre a abordagem de RNA. Isto demonstra que o menor nível de habilidades do jogador foi percebido e que este comportamento refletiu no agente treinado.

## Avaliação do robô com abordagem de RNA frente ao robô controlado pelo jogador humano

Tabela 31. Comentários atribuídos a abordagem de RNA frente ao jogador humano

Nível	Avaliação	Comentários
2	3	Também não consegui identificar devido ao comportamento que em dado momento esta dinâmico e adaptado e outros momentos não
2	1	Ao tentar desviar dos ataques, algumas vezes erra trajetória e é alvejado pelos tiros.
3	2	Reage se afastando do adversário, mira ruim, mas sempre buscando alvo.
3	4	Visa o ataque porém a defesa parece não ser muito relevante!
3	4	Parece estar buscando apenas a direção do oponente, não demonstra nenhuma técnica ou tática de jogo.
4	3	Difícil identificar comportamentos similares dos robôs
4	3	Em alguns momentos ele parece ter o comportamento de IA, e em outros ele muda o comportamento para algo mais natural, como se um humano estivesse controlando.
4	5	Os 2 robôs fazem movimentos muito parecidos.
5	2	Não parecia um bom piloto, atropelava os tiros do inimigo e errava vários tiros em linha reta. Ou era uma IA de um piloto ruim, ou era um real piloto ruim.

Alguns dos comentários sobre o agente com abordagem de RNA apresentados na Tabela 31, reforçam as similaridades identificadas com o jogador humano. Um dos comentários aponta que “Os 2 robôs fazem movimentos muito parecidos”.

Outros comentários que chamaram a atenção foram relativos a aparente falta de experiência do robô. Um dos avaliadores cita que o agente “reage se afastando do adversário, mira ruim, mas sempre buscando alvo”. Outro comentário aponta que o agente “Visa o ataque, porém a defesa parece não ser muito relevante!”. Inclusive, este último comentário é muito similar a um dos comentários recebidos pelo agente controlado pelo jogador humano. Isto novamente demonstra que o menor nível de habilidades do jogador foi percebido e que este comportamento refletiu no agente treinado.

Outro comentário deixa estas afirmações ainda mais claras, em que o avaliador cita que “não parecia um bom piloto, atropelava os tiros do inimigo e errava vários tiros em linha reta. Ou era uma IA de um piloto ruim, ou era um real piloto ruim.”

## Avaliação do robô com abordagem de RNA frente ao robô determinístico DustBunny

Tabela 32. Comentários atribuídos a abordagem de RNA frente ao determinístico DustBunny

Nível	Avaliação	Comentários
2	4	Apesar da variedade de jogadas ainda se perde em alguns momentos sem saber o que fazer
2	1	Desvia dos tiros e contra ataca efetivamente
3	2	boa estratégia.
3	1	Busca constantemente atacar, semelhante ao que eu faria!
4	1	Ações de perseguição e desvio de tiros muito efetivos e sem padrão único ou previsível, o que caracteriza ação de humano.
4	1	Humano porque sempre quer atacar

5	4	O robô vermelho era ofensivo o tempo todo e procurava se desviar caso um tiro fosse em sua direção, vez ou outra. Me pareceu um comportamento padrão.
5	4	Iniciou a reação apenas quando um ataque foi iniciado.
5	2	Parece ser humano, pois tem muitas colisões nas bordas.

Na avaliação do robô com abordagem de RNA frente ao determinístico DustBunny apresentada na Tabela 32, uma informação que contrasta com comentários anteriores foi detectada em relação aos comentários sobre sua “boa estratégia”. Esta informação chama a atenção, pois em outras batalhas o mesmo RNA recebeu comentários afirmando que foi observada uma “estratégia ruim, ou nenhuma estratégia”. Esta informação mais uma vez aponta que os eventos das batalhas modificam muito a maneira como o agente é avaliado e o RNA pareceu se adaptar de forma diferenciada de acordo com o oponente enfrentado.

Outros comentários reforçam que o robô mais efetivo em combate interferiu na opinião dos jogadores (nesta batalha o RNA foi vitorioso), como pode ser visto no comentário que cita “Humano porque sempre quer atacar”, ou “busca constantemente atacar, semelhante ao que eu faria!”.

### **Avaliação do robô com abordagem determinística (DustBunny) frente ao robô com abordagem de RNA**

Tabela 33. Comentários atribuídos ao determinístico DustBunny frente a abordagem de RNA

Nível	Avaliação	Comentários
2	3	Tem uma variabilidade de jogadas porem em alguns momentos se perde
3	5	roda em torno se seu próprio eixo muitas vezes,
3	5	Movimenta-se pelos cantos e não demonstra ações de ataque efetivo.
4	5	Robô se movimenta sempre pelas laterais da arena em movimento ondulados, parecendo um comportamento pré definido.
4	5	Teve momentos que, ficou meio trancando sem saber o que fazer.
4	4	Ele gira bastante mas a mira está sempre certinha em cima do robô vermelho, parece muito correto para um humano, mas também pode ser um humano muito habilidoso quanto a mira...
5	2	Era um piloto ruim, colidiu várias vezes nas paredes e se sentiu acuado com o robô vermelho.
5	5	Movimenta muito o corpo sem movimentar o cano (ou melhor dizendo, corrigido automaticamente a movimentação do cano com muita precisão)
5	5	Calculo de onde o tiro deve ser efetuado, tanto quanto a mira para o robô adversário são feitos de forma errada, esta mal programado tendo um grande índice de erros, parece só acertar quando o inimigo esta no quarto quadrante.
5	4	Sai das paredes do mesmo jeito.

Entre os comentários relativos ao determinístico DustBunny frente a abordagem de RNA, apresentados na Tabela 33, se destacam aqueles que deixam claras as técnicas determinísticas utilizadas durante a batalha. Neste caso a grande vantagem do DrussGT está em utilizar técnicas mais discretas e efetivas de combate, sendo que, como foi percebido nos comentários sobre o DrussGT, também detectáveis para alguns avaliadores

Entre estes comentários se destacam a citação de que o robô “se movimenta sempre pelas laterais da arena em movimento ondulados, parecendo um comportamento pré-definido”, ou a afirmação de que “movimenta-se pelos cantos”.

Avaliando todos os comentários de forma geral entre todas as abordagens, uma informação identificada tanto para os mais experientes quanto os menos experientes, foi em relação aos critérios utilizados pelos jogadores. Entre as avaliações realizadas foi possível perceber que o mesmo critério serviu muitas vezes para considerar um robô como artificial e outras vezes como humano. Um exemplo disso foi o comentário sobre “ações dinâmicas e sem repetição”, em que para alguns avaliadores um jogador humano tende a se repetir e para outros a repetição é executada quando um agente é artificial.

É importante lembrar que os dados complementares abordados nesta seção são utilizados apenas para discussão dos resultados em conjunto com informações obtidas através da análise estatística que identificou como foram avaliados os comportamentos entre as diferentes abordagens.

## **5.4 CONSIDERAÇÕES**

Participaram do experimento 3 grupos de pessoas da Univali. Os experimentos foram realizados com diferentes grupos da área de tecnologia ou diretamente ligado a área de jogos a fim de diversificar o nível de experiência dos avaliadores. Os diferentes grupos foram tratados na avaliação como uma mesma população, sendo avaliados de acordo com sua experiência em jogos.

Na primeira etapa de avaliação dos resultados, foi aplicado um teste estatístico para identificar se jogadores com diferentes níveis de experiência apresentavam avaliações similares para cada abordagem e poderiam ser considerados como um único grupo para avaliação, ou se existiam diferenças significativas que obrigassem a divisão dos jogadores de acordo com seus níveis de experiência.

Como foram identificadas diferenças nas avaliações com base nos níveis de experiência em jogos, os jogadores foram divididos em 2 grupos, um grupo de “menos experientes” e um grupo de “mais experientes”. Os grupos foram formados de maneira que as amostras fossem balanceadas e não prejudicassem os testes.

A partir dos grupos definidos com base na experiência dos jogadores, foi realizado um teste para determinar qual abordagem poderia ser considerada mais humana em relação as outras, a partir da escala de avaliação dos níveis de humanidade do experimento. Este teste foi realizado para responder a pergunta principal de pesquisa que visa identificar como a abordagem de RNA se saiu frente as abordagens determinísticas, além de compará-la com o próprio jogador humano

Analisando os resultados da avaliação, verificou-se que o grupo de jogadores menos experientes tendeu a ficar indeciso, não atribuindo um valor estatisticamente significativo para o nível de comportamento humano nem artificial para nenhuma das abordagens.

Já para os jogadores mais experientes foram atribuídos níveis de comportamento estatisticamente significativos na batalha entre a abordagem de RNA e o determinístico DustBunny. Os valores obtidos na avaliação de comportamento demonstraram que a abordagem de RNA recebeu uma avaliação próxima do comportamento humano, quanto o determinístico DrussGT foi avaliado com um comportamento próximo ao artificial. Porém na avaliação de outras abordagens os resultados foram similares aos obtidos pelos jogadores menos experientes, em que o grupo de jogadores mais experientes também tendeu a ficar indeciso, não atribuindo um valor estatisticamente significativo para o nível de comportamento humano nem artificial.

Estes resultados são coerentes com os comentários realizados pelos jogadores, em que foi possível observar que nível de detalhes observados pelos jogadores menos experientes foi menor, em diversos casos apontando dúvidas nas avaliações realizadas. Os jogadores mais experientes também demonstraram algumas dúvidas nas avaliações, sendo que seus comentários apresentaram um maior nível de detalhes observados.

Também foram identificadas outras observações dos jogadores que auxiliaram no entendimento dos resultados. Uma destas observações diz respeito a identificação de alguns padrões nas ações dos robôs determinísticos. No caso do DrussGT, apesar de identificado alguns padrões, estas características foram sutis, quando comparadas ao DustBunny, em que os jogadores identificaram mais características “padronizadas”, principalmente no caso de jogadores mais experientes. Estas informações são coerentes com os resultados obtidos, pois no cruzamento entre o determinístico DrussGT e a abordagem de RNA as avaliações do nível de comportamento foram muito mais próximas. Já para o DustBunny em que foram identificados mais padrões no comportamento, o RNA foi considerado mais humano na batalha.

Outro fator que colaborou com os resultados foi relativo a efetividade de determinados robôs nas batalhas, em que foi identificado que essa era uma característica que influenciava na opinião de alguns avaliadores, pois alguns comentários apontaram que era esperado um maior nível de competitividade do jogador humano. Como a batalha em que o RNA foi mais efetivo (venceu a batalha) foi contra o DustBunny este fator pode ter auxiliado na obtenção de uma avaliação mais próxima do nível de comportamento humano. Em contrapartida, como a abordagem de RNA foi derrotada em outras batalhas, este mesmo fator pode ter impedido a abordagem de ser avaliada como mais humana frente ao determinístico DrussGT e o próprio jogador.

Entre outras observações realizadas pelos avaliadores, uma das mais importantes foi obtida na comparação entre o jogador humano e a abordagem de RNA. Nesta batalha muitos dos comentários realizados apontaram a semelhança entre os dois robôs durante a batalha. Estas observações indicaram que um dos objetivos, que foi mapear as ações do jogador no agente com abordagem de RNA foi identificado durante as avaliações.

## 6 CONCLUSÕES

Esta pesquisa avaliou quantitativamente os resultados de comparações entre uma abordagem com Redes Neurais Artificiais, baseada no comportamento de um jogador humano, frente a abordagens determinísticas comumente aplicadas a jogos, além de realizar comparações com o próprio jogador.

Estas comparações foram realizadas com o objetivo de identificar quais abordagens mais se aproximavam do comportamento humano e conseqüentemente comprovar as hipóteses de pesquisa levantadas, que foram:

Hipótese 1: O agente com a implementação através da técnica de RNA reflete os aspectos comportamentais do jogador em um ambiente de estratégia; e

Hipótese 2: Na percepção dos jogadores o agente com o uso de RNA gera comportamentos mais realistas do que as técnicas determinísticas

Para realizar estas comparações, todos os objetivos específicos foram cumpridos. Inicialmente foram definidos os agentes com abordagem determinística que foram utilizados nos testes. Posteriormente foi realizado o treinamento de uma rede neural mapeando os padrões de comportamento humano para implementação de um agente baseado nas ações do jogador.

A partir da definição das abordagens a serem utilizadas nos testes foram gravadas batalhas em um ambiente de estratégia, entre o agente com abordagem de RNA, os agentes determinísticos e o próprio jogador humano (controlando um dos agentes).

Realizadas as gravações foi aplicado um teste em que jogadores com diferentes níveis de experiência em jogos avaliaram cada uma das abordagens, a fim de se obter a percepção destes jogadores, possibilitando identificar quais foram consideradas mais próximas do comportamento humano. Com base nos resultados obtidos nestes testes, foram aplicados métodos estatísticos para analisar as respostas obtidas.

A primeira etapa de análises realizada nesta pesquisa identificou a existência de correlações significativas entre os níveis de experiência dos avaliadores em jogos e os níveis de comportamento atribuídos pelos mesmos para as diferentes abordagens. Estes resultados forçaram a divisão das

avaliações com base nos níveis de experiência dos jogadores, que foram divididos entre “menos experientes” e “mais experientes”.

Analisando os resultados das comparações, verificou-se que o grupo de avaliadores menos experientes tendeu a ficar indeciso, não atribuindo um valor estatisticamente significativo para o nível de comportamento humano nem artificial para nenhuma das abordagens.

Já para o grupo de avaliadores mais experientes os resultados alcançados permitiram que fossem confirmadas as hipóteses de pesquisa levantadas inicialmente. Para confirmar a hipótese 1 e a hipótese 2, foram encontradas evidências significativas de que o agente com abordagem de RNA foi identificado com comportamento mais humano do que a abordagem determinística ao qual foi comparado. Os valores utilizados para identificar este resultado foram obtidos através da mediana, que apontou um valor de 2,50 (mais próximo ao nível comportamento humano) para o agente com abordagem de RNA e um valor de 4,00 (mais próximo ao nível de comportamento artificial) para o determinístico.

Como citado, estas evidências confirmaram a hipótese 1, que também foi respondida através de dois testes qualitativos, um baseado na análise do agente com abordagem de RNA após seu treinamento e outro a partir das observações dos jogadores durante as comparações.

Na análise do agente treinado, foram identificadas que todas as tarefas realizadas pelo jogador durante o treinamento da rede foram executadas pelo agente. Esta análise foi realizada em 50 batalhas. Como a execução destas tarefas está diretamente ligada aos dados de treinamentos gerados pelo jogador, o fato de todas as tarefas serem identificadas, apresentou um forte indício de que o agente estava reproduzindo as ações do jogador humano. Nas observações dos avaliadores durante a batalha entre o jogador humano e a abordagem de RNA, foram apontadas diversas semelhanças identificadas entre os dois agentes, que possibilitaram identificar que na visão dos avaliadores o perfil do jogador foi reconhecido no agente com abordagem de RNA.

Como as evidências encontradas nas comparações indicaram que a abordagem de RNA foi a única que apresentou um nível significativamente próximo do comportamento humano, é possível afirmar que a abordagem foi identificada como mais humana quando mapeada através das ações do jogador.

Um aspecto que não se pode deixar de considerar está ligado ao fato da abordagem de RNA ser identificada somente frente a um dos agentes determinísticos (2 foram testados), indicando que o nível de competitividade dos agentes na batalha influenciou nas avaliações dos jogadores, já que o determinístico mais competitivo não apresentou diferenças significativas frente ao RNA. Além disso, o personagem controlado pelo jogador humano também não apresentou diferenças significativas frente ao RNA (os jogadores tenderam a ficar em dúvida), sendo que este era um resultado esperado, já que o agente com abordagem de RNA foi baseado no comportamento deste jogador.

Os resultados obtidos certamente são passíveis de discussão, devido a características do próprio teste aplicado que buscou obter as percepções de avaliadores para gerar as conclusões realizadas. Por isso a próxima seção discute algumas das ameaças do experimento realizado. As ameaças são discutidas conforme as definições de Bauckage *et al* (2007) e Gorman *et al* (2006b) de onde o experimento para avaliação do comportamento humano foi baseado.

## 6.1 SUBJETIVIDADE, VIESES E OUTRAS AMEAÇAS

A primeira ameaça que pode ser identificada nesta pesquisa é relativa ao efeito do observador e o nível de subjetividade que pode estar envolvido em sua avaliação. Este não é um problema exclusivo desta pesquisa e conforme citam Bauckage *et al* (2007) este é um viés comum em trabalhos deste tipo.

Um dos objetivos do experimento realizado foi mensurar a maneira como os avaliadores identificaram os agentes com diferentes abordagens a fim de identificar o quão próximas do comportamento humano foram percebidas. Para avaliar a percepção dos jogadores os vieses detectados tiveram que ser minimizados ou eliminados.

A escolha do framework de testes utilizado na pesquisa já foi baseada nas preocupações em reduzir estes vieses.

Uma das armadilhas identificadas já nas primeiras etapas de definição do experimento foi relativa a seleção dos agentes com abordagens determinísticas, pois a seleção dos agentes deveria garantir que não fossem realizadas escolhas que pudessem beneficiar determinadas características que pudessem influenciar nas respostas dos avaliadores. Na tentativa de minimizar este problemas

foram escolhidos robôs participantes de competições disponibilizadas pela RoboWiki (2011) de forma aleatória, apenas levando em consideração seu nível de competitividade, em que foi definida a utilização de um agente com nível de competitividade alto e outro com nível intermediário. Os níveis foram considerados com a finalidade de identificar durante as comparações a sua influência nos resultados. Para ser elegível o agente também deveria ter código fonte aberto para que pudesse ser observada a abordagem determinística utilizada nos robôs.

Um dos principais fatores considerados na idealização do teste foi relativo à seleção dos vídeos das batalhas de maneira a evitar a escolha de um vídeo que pudesse beneficiar algum comportamento de uma determinada abordagem. Para evitar este problema foram primeiramente geradas batalhas entre todos os agentes que foram avaliados a um total definido em 10 vídeos de cada batalha. Posteriormente foram identificados que a média de duração das batalhas foi de 15 segundos. A fim de garantir batalhas com maior duração, para permitir ao avaliador mais tempo para análise e garantir a aleatoriedade da escolha dos vídeos, foram escolhidas as batalhas de maior duração.

Outro fator que poderia influenciar nas respostas seria a avaliação realizada para um robô em uma batalha prévia, que pudesse influenciar em suas respostas nas batalhas subsequentes. Para evitar este problema foram modificadas as cores dos robôs em todas as batalhas, não possibilitando identificar qual robô foi avaliado anteriormente.

Outra precaução foi realizada para evitar qualquer confusão na avaliação, já que dois robôs batalhavam enquanto era realizada a observação, em todas as perguntas era referenciada a cor do robô que estava sendo avaliado naquele momento para guiar o avaliador. Além disso, foram utilizadas cores contrastantes para garantir que não houvesse confusão na identificação de qual robô estava sendo avaliado.

Uma questão relacionada a esta preocupação foi relativa ao cuidado na idealização das perguntas apresentadas para os avaliadores, a fim de evitar comentários que pudessem influenciar em suas respostas. Para isso foram seguidas as definições do framework de avaliação para elaborar estas perguntas.

Uma questão também considerada na criação do experimento, foi em relação a duração do teste, que poderia fazer com que os avaliadores perdessem o interesse no decorrer do mesmo. A

maior parte do tempo utilizado no teste é correspondente a visualização dos vídeos das batalhas, sendo que o tempo total para assistir a todos os vídeos não é maior do que 3 minutos e 18 segundos e a maior parte dos questionamentos é disponibilizado em questões de múltipla escolha. Apenas o campo referente a comentários necessitou de uma descrição do avaliador e para minimizar a fadiga foram adicionadas *tags* com descrições pré-determinadas que podiam ser selecionadas para compor estes comentários.

Apesar da preocupação em identificar se o perfil do jogador foi mapeado no agente treinado para realização dos testes, algumas características mais complexas do jogador podem não ter refletido no agente, principalmente devido ao método de treinamento subdividido em tarefas. Este é um fator que deve ser levado em consideração, pois pode ter minimizado as chances de identificação do agente como humano. Em contrapartida os resultados dos testes não apontaram indícios deste problema.

## **6.2 CONTRIBUIÇÕES DA DISSERTAÇÃO**

A metodologia utilizada para mapeamento das ações do jogador humano é uma das principais contribuições do trabalho. Devido a dificuldade em reproduzir as ações comportamentais do jogador, muitas pesquisas ainda necessitam de métodos concretos para tentar reproduzir estas ações.

As características do Robocode ainda permitem que a abordagem utilizada possa ser adaptada para aplicação no mundo real em um robô físico. Com isso os resultados podem ser aproveitados pela área da robótica que é uma das que mais contribuiu ao longo dos anos para as pesquisas sobre o mapeamento do comportamento humano.

O método de obtenção dos dados do jogador humano, que evita o tratamento externo dos arquivos para treinamento gerados, também é por si só uma contribuição. A elaboração do método depende de funções aplicadas na leitura dos dados para que os eventos disparados a partir dos controles do jogador possam ser tratados e lidos já na forma ideal. Além disso, o problema nas fatias de tempo do Robocode também é tratado no método de obtenção dos dados orientado aos eventos disparados pelo jogador.

A contribuição mais significativa do trabalho deve ficar a cargo dos aspectos ligados aos procedimentos e resultados dos experimentos. Os procedimentos metodológicos definidos podem ser utilizados como base para outras pesquisas em que o objetivo seja mapear as ações do jogador através de técnicas fundamentadas na área do mapeamento do comportamento humano, principalmente devido à área ainda necessitar de métodos bem definidos.

### **6.3 TRABALHOS FUTUROS**

Um dos principais trabalhos futuros desta pesquisa é a replicação dos experimentos com um número maior de avaliadores, pois com amostras maiores é possível que as pequenas diferenças na comparação de algumas abordagens tornem-se estatisticamente mais significativas.

Outro trabalho futuro também está ligado a replicação do experimento, neste caso para um nível de perfil específico de jogadores, sendo para isso necessário um pré-teste para realização deste filtro.

O treinamento do agente também pode ser realizado a partir de diferentes jogadores com diferentes níveis, alterando o modelo para identificar se o comportamento do agente é capaz de se adaptar ao nível de habilidades do adversário.

Um novo método de avaliação também pode ser definido para que ao invés das gravações das batalhas, os jogadores possam interagir diretamente contra as diferentes abordagens no próprio Robocode.

A metodologia também pode ser aplicada a um robô físico, desde que as condições da arena sejam replicadas e com isso diferentes tarefas possam ser mapeadas a partir das ações do jogador.

Outro trabalho futuro seria aplicar parte da metodologia em um ambiente de jogo mais complexo para um treinamento completo no agente baseado nas ações do jogador.

## REFERÊNCIAS

- ABRANTES, J. F.; TRAVASSOS, G. H. Caracterização de Métodos Ágeis de Desenvolvimento de Software. *In: Workshop de Desenvolvimento Rápido de Aplicações - SBQS 2007, Porto de Galinhas – PE, 2007. Anais...* Porto de Galinhas, 2007.
- ALMEIDA, V. N. **Geração de parâmetros de busca baseada em perfis de usuário.** 2004. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós Graduação em Computação, Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, 2004.
- ARMBRUSTER, A. *et al.* A real-time Java virtual machine with applications in avionics. **ACM Transactions on Embedded Computing Systems (TECS)**, v.7, n.1, p.1-49, 2007.
- BAUCKHAGE, C. *et al.* Learning Human Behavior from Analyzing Activities in Virtual Environments. *In: MMI Interaktiv, Israel, 2007. Proceedings...* Israel, v.1, n.12, p.3–17, 2007.
- BAUCKHAGE, C.; THURAU, C.; SAGERER, G. Learning Humanlike Opponent Behaviour for Interactive Computer Games. *In: 25th DAGM Symposium, Magdeburg, Germany. Proceedings...* Springer-Verlag, v.2781, p.148-155, 2003
- BONAKDARIAN, E.; WHITE, L. Robocode throughout the curriculum. **Journal of Computing Sciences in Colleges**, v.19, n.3, p.311-313, 2004.
- BRAGA, A. P.; CARVALHO, A. P.; LUDEMIR, T. B. **Redes neurais artificiais: teoria e aplicações.** Rio de Janeiro: LTC, 2000.
- BRYANT, B. D.; MIIKKULAINEN, R. Acquiring visibly intelligent behavior with example-guided Neuroevolution. *In: Twenty-Second National Conference on Artificial Intelligence, Austin, USA. Proceedings...* ACM, p.801-808, 2007.
- CASS, S. Mind Games. **IEEE SPECTRUM**, v.39, n.12, p.40-44, 2002.
- CHAWLA, N. V. Data Mining for Imbalanced Datasets: An Overview. **Data Mining and Knowledge Discovery Handbook**, Springer US, p.853-867, 2005.
- COSTA, A. F. **Jogos e Sistemas Especialistas.** 2010. 71p. TCC (Tecnólogo em Sistemas de Computação) – Curso de Tecnologia em Sistemas de Computação, Universidade Federal Fluminense, Niterói, 2010.
- CUNHA, R. L. de F.; CHAIMOWICZ, L. An Artificial Intelligence System to Help the Player of Real-Time Strategy Games. *In: Brazilian Symposium on Games and Digital Entertainment (SBGAMES '10), Washington, USA. Proceedings...* IEEE, 2010, p.71-81.
- CZAJKOWSKI, A.; PATAN, K. **Real-Time Learning of Neural Networks and its Application to the Prediction of Opponent Movement in the Robocode Environment.** XI International PhD Workshop (OWD 2009). Conference Archives, 2009. Disponível em: <<http://mechatronika.polsl.pl/owd/pdf2009/384.pdf>>. Acesso em: 20 mai. 2012.

DAVID, A. B. Rule effectiveness in rule-based systems: A credit scoring case study. **Expert Systems with Applications**, v. 4, n. 34, p. 2783–2788, 2008.

DYBÅ, T.; DINGSØYR, T. Strength of evidence in systematic reviews in software engineering. *In: Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany. Proceedings...* ACM, 2008, p.178-187.

FERNANDES, A. M. R. **Inteligência artificial: noções gerais**. Florianópolis: Visual Books, 2003.

FROZZA, R. **SIMULA - Ambiente Para Desenvolvimento de Sistemas Multiagentes Reativos**. 1997. 116p. Dissertação (Pós-Graduação em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 1997.

GADE, M. *et al.* Applying Machine Learning to Robocode. **JSK**, Aalborg University, Aalborg, 2003.

GAL, A. *et al.* GameBots: a flexible test bed for multiagent team research. **Communications of the ACM**, v.45, n.1, 2002.

GLEIM, Urs. JaRTS: A Portable Implementation of Real-Time Core Extensions for Java. *In: 2nd Java™ Virtual Machine Research and Technology Symposium, Berkeley, USA. Proceedings...* USENIX, 2002, p.139-149.

GORMAN, B.; HUMPHRYS, M. Imitative learning of combat behaviours in first-person computer games. *In: 11th International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games (CGAMES 2007), Louisville, USA. Proceedings...* 2007.

GORMAN, B.; HUMPHRYS, M. Towards Integrated Imitation of Strategic Planning and Motion Modelling in Interactive Computer Games. *In: 3rd ACM International Conference in Computer Game Design and Technology (GDTW 2005), New York, USA. Proceedings...* ACM, p.92-99, 2006.

GORMAN, B. *et al.* Bayesian Imitation of Human Behavior in Interactive Computer Games. *In: International Conference on Pattern Recognition (ICPR'06), Dublin, Ireland. Proceedings...* IEEE, 2006a, p.1244–1247.

GORMAN, B. *et al.* Believability Testing and Bayesian Imitation in Interactive Computer Games. *In: 9th International Conference on the Simulation of Adaptive Behavior (SAB'06), Berlin, Germany. Proceedings...* Springer-Verlag, 2006b, p. 655-666.

GURNEY, K. **An Introduction to Neural Networks**. London: Routledge, 1997.

HAYKIN, S. **Redes neurais: Princípios e Prática**. 2. ed. Porto Alegre: Bookman, 2001.

HAGELBÄCK, J.; JOHANSSON, S. J. A multiagent potential field-based bot for real-time strategy games. **International Journal of Computer Games Technology**, p.1-10, 2009.

\_\_\_\_\_. Using multi-agent potential fields in real-time strategy games. *In: 7th international joint conference on Autonomous agents and multiagent systems, Richland, USA. Proceedings... IFAAMAS, 2008, p. 12-16.*

HAN, H.; WANG, W. Y.; MAO, B. H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *In: International Conference on Intelligent Computing (ICIC'05), Hefei, China. Proceedings... Springer-Verlag, 2005, p. 878-887.*

HARPER, R. Co-evolving Robocode tanks. *In: 13th annual conference on Genetic and evolutionary computation (GECCO '11), New York, USA. Proceedings... ACM, 2011, p.1443-1450.*

HARTNESS, K. Robocode: using games to teach artificial intelligence. **Journal of Computing Sciences in Colleges**, v.19, n.4, p.287-291,2004.

HUANG, S. *et al.* ORTS: a tool for optimized regression testing selection. *In: 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA '09), New York, USA. Proceedings... ACM, 2009, p. 803-804.*

HY, R. L. *et al.* Teaching Bayesian behaviours to video game characters. **Robotics and Autonomous Systems**, v.47, p. 177-185, 2004.

IORIO, V. DI. *et al.* What's the Name of the Game? Formal Specification of Artificial Intelligence Games. **Original Research Article Electronic Notes in Theoretical Computer Science**, v.130, p.129-150, 2002.

JHUANG, H. *et al.* Automated home-cage behavioural phenotyping of mice. **Nature Communications**, v.1, n.68, 2010.

KI, H.; Lyu, J.; OH, K.; Real-Time Neuroevolution to Imitate a Game Player. *In: First international conference on Technologies for E-Learning and Digital Entertainment (Edutainment'06), Berlin, Germany. Proceedings... Springer-Verlag, 2006, p.658-668.*

KENSLER, J. A.; Agah, A. Neural networks-based adaptive bidding with the contract net protocol in multi-robot systems. **Applied Intelligence**, v. 31, n.3, 2009.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering (version 2.3). **Technical report**, Keele University and University of Durham, 2007.

LAIRD J. E.; van LENT, M. Interactive Computer Games: Human-Level AI's Killer Application. *In: Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, Austin, USA. Proceedings... AAAI Press, 2000, p.1171-1178.*

LIU, P. L. Using open-source Robocode as a Java programming assignment. **SIGCSE Bull**, n.40, v.4, p.63-67, 2008.

LUDWING Jr., O.; COSTA, E. M. M. **Redes Neurais: fundamentos e aplicações com programas em C**. Editora Ciência Moderna Ltda. Rio de Janeiro, 2007.

MACHADO, L. S.; MORAES, R. M.; NUNES, F. (2009) **Serious Games para Saúde e Treinamento Imersivo**. Book Chapter. *In: Abordagens Práticas de Realidade Virtual e Aumentada*. Porto Alegre: SBC, p. 31-60.

MALFATTI, S. M.; FRAGA, L. M. Utilizando Behaviors Para o Gerenciamento da Máquina de Estados em Jogos Desenvolvidos com Java 3D. *In: V Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES 2006)*, Recife, Brasil. **Proceedings...** SBGAMES, 2006.

MERRICK, K. E.; SHAFI, K. Achievement, affiliation, and power: Motive profiles for artificial agents. **Adaptive Systems Journal**, Thousand Oaks, California, USA, v. 19, n. 1, p. 40-62, 2011.

MITAIM, S.; KOSKO, B. Neural Fuzzy Agents for Profile Learning and Adaptive Object Matching. **Teleoperators and Virtual Environments Journal**, Cambridge, Massachusetts, USA, v. 7, n. 6, p. 617-637, 1998.

MOZGOVOY, M.; UMAROV, I. Building a Believable and Effective Agent for a 3D Boxing Simulation Game. *In: 3rd IEEE International Conference on Computer Science and Information Technology*, Aizu-Wakamatsu, Japan. **Proceedings...** IEEE, 2010, p.658-668 2010, p.14-18.

MUÑOZ, J.; GUTIERREZ, G.; SANCHIS, A. Controller for torcs created by imitation. **Symposium on Computational Intelligence and Games**, pp. 271–278, 2009.

NAVARRO, L.; CORRUBLE, V. Extending the Strada Framework to Design an AI for ORTS. *In: 8th International Conference on Entertainment Computing (ICEC '09)*, Berlin, Germany; **Proceedings...** Springer-Verlag, 2009, p. 270-275.

NIELSEN, J. L.; JENSEN, B. F. **Modern AI for Games: Robocode**. Disponível em: <<http://www.jonnielson.net/RoboReportOfficial.pdf>>. Acesso em: 07 set. 2012.

NELSON, M. **Robocode-IBM**. Disponível em: <http://www.ibm.com/developerworks/java/library/j-robocode>. Acesso em: 01 set. 2011.

NEUROPH. **Java Neural Network Framework**. Disponível em: <<http://neuroph.sourceforge.net/documentation.html>>. Acesso em 01 mar. 2012.

OBRADOVIC, D.; STAHL, A. Learning by Observing: Case-Based Decision Making in Complex Strategy Games. *In: 31th Annual German Conference on Artificial Intelligence*, Berlin, Germany. **Proceedings...** Springer-Verlag, 2008, p.284-291.

O'KELLY, J.; GIBSON, J. P. RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. **SIGCSE Bull**, n.38, v.3, p.217-221, 2006.

OKUYAMA, F. Y. **Descrição e Geração de Ambientes para Simulações com Sistemas Multiagentes**. 2003. 119p. Dissertação (Pós-Graduação em Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

PAZZANI, M.; BILLSUS, D. Learning and Revising User Profiles: The Identification of Interesting Web Sites. **Special issue on Multistrategy Learning Journal**, Hingham, Massachusetts, USA, v. 27, n. 3, p. 313-331, 1997.

PERUCIA, A. S. *et al.* **Desenvolvimento de Jogos Eletrônicos, Teoria e Prática**. São Paulo: Novatec, 2005.

PRATI, R. C. **Novas abordagens em Aprendizado de Máquina para a Geração de Regras, Classes Desbalanceadas e Ordenação de Casos**. 2006. Tese (Doutorado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, ICMC/USP, São Paulo, 2006.

ROBOWIKI. **Robocode Community**. Disponível em: <<http://robowiki.net>>. Acesso em: 01 set. 2011.

RODRIGUES, A. P. **E-Avalia - Um Agente para Avaliação de Ensino-Aprendizagem em Educação a Distância**. 2002. 112p. Dissertação (Pós-Graduação em Ciência da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

ROUSSEAU, J. B. I. *et al.* Classification of rat behavior with an image-processing method and a neural network. **Behavior research methods, instruments & computers: a journal of Psychonomic Society**, v.32, n.1, p.63-71, 2000.

SCHOPF, E. C. **Método Neuro-estatístico para Predição de Séries Temporais Ruidosas**. 2007. 105p. Dissertação (Pós-Graduação em Ciência da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

SCHRUM, J.; KARPOV, I. V.; MIIKKULAINEN, R. Ut2: Human-like behavior via neuroevolution of combat behavior and replay of human traces. *In: IEEE Conference on Computational Intelligence and Games (CIG'11)*, Seoul, South Korea. **Proceedings... IEEE**, 2011, p.329–336.

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

SIEGEL, S.; CASTELLAN Jr., J. **Estatística não-paramétrica para ciências do comportamento**. Porto Alegre: Artmed, 2006.

SILVA, F. C. L. **Uma ferramenta para o ensino de inteligência artificial usando jogos de computador**. 2007. Dissertação (Mestrado em Ciências) - Instituto de Matemática e Estatística da Universidade de São Paulo, Universidade de São Paulo, São Paulo, 2007.

SUGIMOTO, E. S.; DIAS, L. C. **Solução de Estratégia Para o Futebol De Robôs Baseada Em Redes Neurais**. Disponível em: <[http://si-emerson-lucas.googlecode.com/files/Emerson\\_Lucas\\_ArtigoSistemasInteligentesProjetoFinal.pdf](http://si-emerson-lucas.googlecode.com/files/Emerson_Lucas_ArtigoSistemasInteligentesProjetoFinal.pdf)>. Acesso em: 25 out. 2011.

THURAU, C.; BAUCKHAGE C. Learning Human-Like Movement Behavior for Computer Games. *In: International Conference on the Simulation of Adaptive Behavior, Odense, Denmark. Proceedings...* MIT Press, 2004, p.315–323.

THURAU, C.; PACZIAN, T.; BAUCKHAGE, C. Is bayesian imitation learning the route to believable gamebots? **International Journal of Intelligent Systems Technologies and Applications**, v. 2, p. 284–295, 2005.

TONG, C. K. *et al.* The Evolution of Gamebots for 3D First Person Shooter (FPS). *In: Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA '11)*, Washington, USA. **Proceedings...** IEEE, 2011, p.21-26.

VALIATI, J. F. **Reconhecimento de Voz para Comandos de Direcionamento por Meio de Redes Neurais**. 2000. 128p. Dissertação (Pós-Graduação em Ciência da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

van HOORN, N. *et al.* Robust player imitation using multiobjective evolution. *In: Congress on Evolutionary Computation, Trondheim, Norway. Proceedings...* IEEE, 2009, p.652–659.

VICENTE FILHO, *et al.* JaRTS: Java RTS Simulator. *In: V Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES 'V)*, Recife, Brasil. **Proceedings...** SBGAMES, 2010.

WOOLLEY, B. G.; PETERSON, G. L. Unified Behavior Framework for Reactive Robot Control. **Journal of Intelligent and Robotic Systems**, v.55 n.2-3, p.155-176, 2009.

ZANETTI, S.; RHALIBI, A. E. Machine learning techniques for FPS in Q3. *In: Advances in Computer Entertainment Technology*, New York, USA. **Proceedings...** ACM, p.239–244, 2004.

# APÊNDICE A – INSTRUMENTO DE AVALIAÇÃO

## Questões



\*Obrigatório

Nome \*

Qual sua experiência em jogos? \*

- 1. Nunca jogou, joga raramente, ou nunca conheceu
- 2. Pouca familiaridade com jogos (já jogou / Joga esporadicamente)
- 3. Joga ocasionalmente (Mensalmente / A cada par de meses)
- 4. Joga regularmente (Semanalmente)
- 5. Joga frequentemente (Diariamente)

[Continuar »](#)

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

## Avaliação de comportamento

\*Obrigatório

Para responder as questões apresentadas, você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=LqMNgTHLsjA>

**Com foco no robô de cor "branca", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô branco \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô branco**

**Caso tenha considerado o robô branco como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## Avaliação de comportamento

### \*Obrigatório

Para responder as questões apresentadas você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=LqMNgTHLsjA>

**Com foco no robô de cor "amarela", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô amarelo \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô amarelo**

**Caso tenha considerado o robô amarelo como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## Avaliação de comportamento

\*Obrigatório

Para responder as questões apresentadas você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=Y87vISGgFPs>

**Com foco no robô de cor "preta", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô preto \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô preto**

**Caso tenha considerado o robô preto como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## Avaliação de comportamento

\*Obrigatório

Para responder as questões apresentadas você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=Y87vISGgFPs>

**Com foco no robô de cor "verde", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô verde \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô verde**

**Caso tenha considerado o robô verde como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## Avaliação de comportamento

\*Obrigatório

Para responder as questões apresentadas você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=Q-nndwjMJCw>

**Com foco no robô de cor "vermelha", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô vermelho \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô vermelho**

**Caso tenha considerado o robô vermelho como Artificial, responda em uma escala de 1 a 10 o quanto próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## Avaliação de comportamento

\*Obrigatório

Para responder as questões apresentadas você deve acessar o vídeo da batalha em:

<https://www.youtube.com/watch?v=Q-nndwJMJCw>

**Com foco no robô de cor "azul", assistido no vídeo acima, atribua um dos níveis de comportamento de acordo com sua observação para este robô: \***

- 1. Humano
- 2. Provavelmente Humano
- 3. Não sei
- 4. Provavelmente Artificial
- 5. Artificial

**Justifique o porque você atribuiu este nível de comportamento ao robô azul \***

- Ações previsíveis
- Ações imprevisíveis
- Em alguns momentos parece imprevisível em outros não
- Busca desviar de ataques
- Não apresenta resposta efetiva aos ataques
- Realiza ações de forma dinâmica
- Em alguns momentos parece se perder

**Adicione seu comentário se deseja complementar sua atribuição sobre o nível de comportamento atribuído ao robô azul**

**Caso tenha considerado o robô azul como Artificial, responda em uma escala de 1 a 10 o quão próximo do nível do jogador humano você o considera?**

(10 mais próximo do humano, 1 menos próximo)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

## **Considerações**

Como pode ser observado o questionário apresenta a avaliação de um robô por vez a partir da cor em que o mesmo é representado no vídeo da batalha. A primeira questão é relativa ao nível de comportamento identificado e as demais questões servem apenas como informações complementares as escolhas realizadas.

## **Robôs e Cores**

As cores utilizadas no questionário para avaliação dos robôs representam as seguintes abordagens:

- Branco: Robô com abordagem que utiliza RNA (baseado no comportamento do jogador humano);
- Amarelo: Robô com abordagem determinística (DrussGT);
- Preto: Robô controlado pelo jogador humano;
- Verde: Robô com abordagem que utiliza RNA (baseado no comportamento do jogador humano);
- Vermelho: Robô com abordagem que utiliza RNA (baseado no comportamento do jogador humano); e
- Amarelo: Robô com abordagem determinística (DustBunny).

## **Questões de comentários**

A segunda e terceira questão apresentadas na avaliação de cada robô, foram tratadas como uma única questão de comentários sobre a avaliação realizada, sendo que a primeira parte apresenta textos pré-determinados para evitar que a fadiga impeça o avaliador de deixar suas considerações.