



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
INSTITUTO METRÓPOLE DIGITAL  
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA DA INFORMAÇÃO  
MESTRADO PROFISSIONAL EM TECNOLOGIA DA INFORMAÇÃO

# SISTEMA PARA AUXÍLIO DE ANÁLISES PSICOFISIOLÓGICAS COM O USO DE JOGOS DIGITAIS

Jaime Bruno Cirne de Oliveira

Natal-RN  
Setembro 2019

Jaime Bruno Cirne de Oliveira

# SISTEMA PARA AUXÍLIO DE ANÁLISES PSICOFISIOLOGICAS COM O USO DE JOGOS DIGITAIS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Tecnologia da Informação da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Mestre em Tecnologia da Informação.

Universidade Federal do Rio Grande do Norte – UFRN  
Instituto Metr pole Digital – IMD  
Programa de P s-Gradua o em Tecnologia da Informa o

Orientador: Prof. Dr. Charles Andry  Galv o Madeira

Natal-RN  
Setembro de 2019

Universidade Federal do Rio Grande do Norte - UFRN  
Sistema de Bibliotecas - SISBI  
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Oliveira, Jaime Bruno Cirne de.

Sistema para auxílio de análises psicofisiológicas com o uso de jogos digitais / Jaime Bruno Cirne de Oliveira. - 2019.  
122 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Instituto Metr pole Digital, Programa de P s-Gradua o em Tecnologia da Informa o, Natal, RN, 2019.

Orientador: Prof. Dr. Charles Andrye Galvao Madeira.

1. Neurociencias - Disserta o. 2. An lises psicofisiol gicas - Disserta o. 3. Jogos digitais - Disserta o. 4. EEG - Disserta o. 5. FPS - Disserta o. I. Madeira, Charles Andrye Galvao. II. T tulo.

RN/UF/BCZM

CDU 612.8(043.3)

JAIME BRUNO CIRNE DE OLIVEIRA

Sistema para auxílio de análises psicofisiológicas com o uso de jogos digitais

Dissertação de Mestrado sob o título, Sistema para auxílio de análises psicofisiológicas com o uso de jogos digitais, apresentada por Jaime Bruno Cirne de Oliveira e aceita pelo Programa de Pós-Graduação em Tecnologia da Informação da Universidade Federal do Rio Grande do Norte, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

---

Prof. Dr. Charles Andryê Galvão Madeira  
Orientador  
IMD – Instituto Metr pole Digital  
UFRN – Universidade Federal do Rio Grande do Norte

---

Prof. Dr. Selan Rodrigues dos Santos  
Examinador  
DIMAp – Departamento de Inform tica e Matem tica Aplicada  
UFRN – Universidade Federal do Rio Grande do Norte

---

Prof. Dr. Sidarta Tollendal Gomes Ribeiro  
Examinador  
ICe – Instituto do C rebro  
UFRN – Universidade Federal do Rio Grande do Norte

---

Prof. Dr. Alberto Signoretti  
Examinador  
Dep. Ci ncia da Computa o, Campus de Natal  
UERN – Universidade do Estado do Rio Grande do Norte

Natal-RN

Setembro de 2019

# Sistema para auxílio de análises psicofisiológicas com o uso de jogos digitais

Autor: Jaime Bruno Cirne de Oliveira

Orientador: Prof. Dr. Charles Andryê Galvão Madeira

## Resumo

Análises psicofisiológicas são poderosas ferramentas que revolucionaram a área de neurociências a partir de 1929 com a criação da eletroencefalografia (EEG). Essas ferramentas possibilitaram perceber o comportamento do cérebro sob diferentes estímulos, gerando uma melhor compreensão dos processos psicológicos por avaliação das reações fisiológicas, sendo amplamente utilizadas em estudos de comportamento. Os jogos digitais se inserem neste contexto por serem utilizados com o intuito de fornecer estímulos sensório motores, exigindo um processamento cognitivo em diferentes níveis de interpretação. No entanto, essa área de pesquisa emergente enfrenta grandes desafios, uma vez que os eventos dos jogos necessitam ser computados com alta resolução temporal para alcançar uma maior compreensão da experiência. Para contribuir neste sentido, este trabalho propõe e implementa uma ferramenta para facilitar as pesquisas que utilizam análise psicofisiológica, auxiliando no processo de análise do comportamento com o uso de jogos digitais pelo registro automático de eventos com a frequência e a precisão necessárias para acompanhar sistemas de registros psicofisiológicos. A ferramenta foi validada em experimentos que provam o conceito sobre sua capacidade de analisar de forma bem mais eficaz os dados psicofisiológicos quando pesquisadores usarem jogos digitais nas suas pesquisas.

**Palavras chave:** neurociências, análises psicofisiológicas, jogos digitais, EEG, FPS.

# System of psychophysiological support with the use of digital games

Author: Jaime Bruno Cirne de Oliveira

Orienter: Prof. Dr. Charles Andryê Galvão Madeira

## Abstract

Psychophysiological analyzes are powerful tools that have revolutionized neurosciences since 1929 with the creation of electroencephalography (EEG). These tools made it possible to perceive the behavior of the brain under different stimuli, providing a better understanding of psychological processes by evaluating the physiological reactions, being widely used in behavioral studies. Digital games are included in this context because they are used to provide sensorial stimuli, requiring a cognitive processing at different levels of interpretation. However, this emerging research area faces major challenges since game events need to be computed with high temporal resolution to achieve a greater understanding of the experience. To contribute to this context, this work proposes and implements a tool to make the research that uses psychophysiological analysis easier by using digital games in the process of behavior analysis, through the automatic registration of events with the frequency and precision required to follow systems of psychophysiological records. The tool has been validate by bulletproof experiment about it ables to analyze psychophysiological data more effectively when researchers using digital games in their research.

Keywords: Neuroscience, Psychophysiological analyzes, digital games, EEG, FPS.

# Lista de Figuras

Figura 1 - Exemplo de tela de um jogo de FPS no qual é mostrado o cenário do jogo visto a partir do personagem controlado pelo jogador	11
Figura 2 - Tipos de ondas referentes às frequências alfa, beta, teta e delta de um eletroencefalograma	21
Figura 3 – Tela de gameplay do jogo Doom	29
Figura 4 – DOOM Builder editor de mapas do jogo DOOM	30
Figura 5 - Potenciais de Eventos Relacionados (ERP)	31
Figura 6 - Equação do cálculo da média de ERP para N testes	32
Figura 7 - James Bond 007: NightFire Gameplay	33
Figura 8 - Half-Life gameplay	35
Figura 9 - Rastreador de olhos	36
Figura 10 – Unreal Level Editor: o editor de níveis do Unreal Tournament	43
Figura 11 – Processamento visual VizDOOM	44
Figura 12 – Comparação DOOM e UT2004	45
Figura 13 – Componentes Pogamut	48
Figura 14 - Camadas da biblioteca GaviaLib	51
Figura 15 - Herança sensomotric	51
Figura 16 - Arquitetura Neuro Game Logger	61
Figura 17 - Pogamut 3 – Virtual Humans Made Simple	62
Figura 18 - Fragmento principal diagrama de Classe	63
Figura 19 - Pinagem da porta paralela	65
Figura 20 - UT2004 e Neuro Game Logger em execução	66
Figura 21 - Registro no log	67
Figura 22 - Fotosensor	69
Figura 23 - Monitor dentro da sala escura	70
Figura 24 - EEG	71
Figura 25 - Voluntário no setup de experimento e sistema de monitoramento do experimentador	73
Figura 26 - Touca EEG no padrão Brain Products de 64 canais com eletrodos vistas de cima	74
Figura 27 - Pontos do mapa DM-UnrealVille	75
Figura 28 - Sinais do trails sobrepostos do evento ESC	77
Figura 29 - Sinais do trails sobrepostos do evento tiro	78
Figura 30 - Histograma do evento ESC	79
Figura 31 - Histograma do evento de tiro	81
Figura 32 - Linha do tempo de eventos e transmissão	83
Figura 33 - Diagrama de Classes do Gamebots	83
Figura 34 - Mapa de calor potência de teta (2-6 Hz) no eletrodo T7 - Primeiro piso	87
Figura 35 - Mapa de calor potência de teta (2-6 Hz) no eletrodo T7 - Segundo piso	87
Figura 36 - Sala com alto valor teta	88
Figura 37 - Visão de fora da sala	88
Figura 38 - ERP após o tiro todos os eletrodos	90
Figura 39 - ERP após o tiro Topografia	91
Figura 40 - ERP após o tiro C5 e C6	91

## Lista de Tabelas

Tabela 1 - Técnicas psicofisiológicas, objetividade e restrição	23
Tabela 2 - DOOM+ VizDOOM vs UT2004 + Pogamut	46

# Sumário

<b>1 Introdução</b>	<b>10</b>
<b>1.1 Problemática e motivação</b>	<b>12</b>
<b>1.2 Objetivos</b>	<b>13</b>
<b>1.2.1 Validação</b>	<b>15</b>
<b>1.2.2 Expectativas sobre os resultados</b>	<b>15</b>
<b>1.3 Estrutura do documento</b>	<b>16</b>
<b>2 Pesquisas cognitivas com jogos digitais</b>	<b>17</b>
<b>2.1 Psicofisiologia e Neurociência cognitiva</b>	<b>18</b>
<b>2.2 Análises de estudos cognitivos: métodos, técnicas e estímulos.</b>	<b>19</b>
<b>2.3 Motivação para utilização de jogos em estudos cognitivos</b>	<b>24</b>
<b>2.4 Limitações para utilização de jogos em estudos cognitivos</b>	<b>26</b>
<b>2.5 Escolha de estilo de jogo adequado à pesquisa</b>	<b>27</b>
<b>2.6 Jogos de tiro em primeira pessoa no contexto de estudos cognitivos</b>	<b>29</b>
<b>2.6.1 Jogos de tiro em primeira pessoa e análise de potenciais de eventos relacionados</b>	<b>31</b>
<b>2.6.2 Jogos de tiro em primeira pessoa e Registro de Eventos</b>	<b>34</b>
<b>2.7 Conclusões sobre pesquisas cognitivas e jogos de tiro em primeira pessoa</b>	<b>37</b>
<b>2.7.1 Problemática das Soluções atuais</b>	<b>38</b>
<b>2.7.2 Similaridade entre estudos cognitivos e personagens inteligentes</b>	<b>39</b>
<b>3 Soluções de IA aplicadas a estudos cognitivos</b>	<b>40</b>
<b>3.1 Personagens Inteligentes em Jogos de FPS</b>	<b>40</b>
<b>3.2 Aplicação de uma plataforma de desenvolvimento de personagens inteligentes para uma ferramenta de análises psicofisiológicas</b>	<b>44</b>
<b>3.2.1 Detalhes do Pogamut</b>	<b>47</b>
<b>3.2.2 Gamebots</b>	<b>48</b>
<b>3.2.3 API para o Mundo Virtual</b>	<b>50</b>
<b>3.3 Conclusões sobre a aplicação de soluções em estudos cognitivos</b>	<b>52</b>
<b>4 Neuro Game Logger: uma proposta de ferramenta para análise cognitiva com jogos</b>	<b>53</b>
<b>4.1 Metodologia</b>	<b>54</b>
<b>4.2 Análise de Requisitos</b>	<b>54</b>
<b>4.2.1 Reflexões sobre a Análise para uso do Unreal Tournament 2004</b>	<b>57</b>
<b>4.3 Descrição do projeto arquitetural e desenvolvimento</b>	<b>60</b>
<b>4.3.1 Visão da Estrutura</b>	<b>60</b>
<b>4.4 Neuro Game Logger</b>	<b>62</b>
<b>4.4.1 MouseLogger e KeyLogger</b>	<b>64</b>

<b>4.4.2 TriggerParallelPort</b>	<b>65</b>
<b>4.5 Visão de ação do usuário</b>	<b>66</b>
<b>4.6 Logs gerados</b>	<b>67</b>
<b>4.7 Validação</b>	<b>68</b>
<b>4.7.1 Metodologia experimental para determinar a precisão temporal</b>	<b>69</b>
<b>4.7.2 Metodologia experimental usada na prova de conceito</b>	<b>73</b>
<b>4.7.3 Resultados sobre a precisão temporal</b>	<b>77</b>
<b>4.7.4 Investigação da maior variância detectada em eventos utilizando o Pogamut</b>	<b>82</b>
<b>4.7.5 Reflexão quanto ao resultado da validação de precisão temporal</b>	<b>84</b>
<b>4.8 Resultados da Prova de conceito</b>	<b>85</b>
<b>4.8.1 Análise da potência de teta</b>	<b>86</b>
<b>4.8.2 ERP após disparo de tiro</b>	<b>89</b>
<b>4.9 Conclusão da validação</b>	<b>92</b>
<b>5 Conclusão e trabalhos futuros</b>	<b>94</b>
<b>Referências bibliográficas</b>	<b>99</b>
<b>Anexo 2</b>	<b>119</b>
<b>Casos de Uso</b>	<b>119</b>
<b>CSU01 – Iniciar experimento</b>	<b>119</b>
<b>CSU02 – Interromper experimento</b>	<b>120</b>
<b>CSU03 – Finalizar experimento</b>	<b>121</b>
<b>CSU04 – Monitorar Jogo</b>	<b>122</b>
<b>CSU05 – Gerar Log</b>	<b>122</b>
<b>CSU06 – Responder a evento</b>	<b>123</b>
<b>CSU07 – Enviar Sinal a Porta Paralela</b>	<b>124</b>

# Capítulo 1

## Introdução

Análises psicofisiológicas são métodos poderosos que revolucionaram a área de neurociências, se mostrando muito relevantes para o estudo com humanos pela utilização da eletroencefalografia (EEG) a partir de 1924, segundo Haas (2003). A EEG possibilita perceber o comportamento elétrico do cérebro de maneira não invasiva, gerando uma melhor compreensão dos processos psicológicos por avaliação das reações fisiológicas, sendo amplamente utilizadas em estudos de comportamento.

Por meio dessa técnica é possível estudar fenômenos psicológicos, medindo a atividade fisiológica humana, como descrito por Cacioppo (2007). Fenômenos fisiológicos se relacionam com processos psicológicos e vice versa, entretanto não necessariamente possuem uma relação direta ou unitária. Dessa forma, é necessário que o pesquisador interprete essa relação. A fim de permitir uma interpretação mais objetiva possível, é necessário expor o experimentado a uma série de eventos que gerem o processo estudado em questão.

Para entender processos cognitivos complexos, como o aprendizado por exemplo, são necessários estímulos elaborados. Estes são gerados pela exposição a um ambiente de aprendizado interativo, com atividades tanto cognitivas como motoras.

Neste contexto, jogos eletrônicos se oferecem como ótimos ambientes interativos para tarefas sensoriais, ao mesmo tempo que exigem um processamento cognitivo em diferentes níveis de interpretação e operam em escalas de tempo muito fragmentadas que podem levar de segundos a até algumas horas (KLIMMT, 2003).

Especialmente jogos de FPS (*First Person Shooter* – ou Tiro em Primeira Pessoa) têm sido bastante usados como estímulos experimentais uma vez que nesse gênero de jogo o jogador tem uma perspectiva que simula a visão através dos olhos da sua representação no jogo, ou seja, seu avatar. Isso dá a sensação ao jogador de estar agindo direta e explicitamente no mundo virtual tridimensional (3D) hostil com gráficos e sons altamente realistas (RAVAJA, 2006).

Na visão em primeira pessoa, parte da exibição do cenário geralmente mostra as armas e as mãos do protagonista, além de alguns dados essenciais, tais como munições disponíveis, nível de saúde, armadura e tipo de arma que estão sendo usados, como mostrado na Figura 1.



**Figura 1 - Exemplo de tela de um jogo de FPS no qual é mostrado o cenário do jogo visto a partir do personagem controlado pelo jogador.**

Várias pesquisas utilizam esse tipo de jogo como estímulos nos seus experimentos como Klimmt (2003); Field (2009); Cacioppo (2007) e Kallio (2011), nos quais sessões de jogos são utilizadas como forma de gerar processos psicofisiológicos que se querem ser estudados. A resposta fisiológica que a experiência oferece ao jogador pode ser quantificada por métodos de análises psicofisiológicas, que consistem na utilização de sinais fisiológicos para estudar fenômenos psicológicos (CACIOPPO, 2007). Entretanto a dificuldade de experimentos desse tipo é registrar de maneira idêntica e controlada a atividade fisiológica no momento exato do evento.

O registro automático de eventos é a melhor maneira para entender o comportamento do usuário dentro de sistemas interativos conforme demonstra a psicologia experimental (HILBERT, 2000). Soluções de registro automatizado, como "câmara de condicionamento operante de Skinner", mantém o controle das interações de

animais para estudar o seu comportamento, entretanto em jogos digitais isso é especialmente mais complexo.

Apesar dos avanços da área, ainda existem muitas limitações do uso de métodos psicofisiológicos em estudos com jogos digitais, sendo estas principalmente de natureza técnica. Por exemplo, é difícil estimar, com a precisão dos SCRs (*Silicon-Controlled Rectifier* ou Retificador Controlado de Silício), eventos que ocorrem concorrentemente ou em períodos muito próximos, uma vez que os efeitos se sobrepõem nas atividades complexas simuladas pelos jogos digitais.

A análise das métricas e dados dos jogos digitais pode tornar-se complexa e pouco precisa, devido a falta de acesso direto a essas informações. Isso pode prejudicar a análise do tipo ERP (*Event Related Potential* ou Potencial de Eventos Relacionados), por exemplo, visto que nesse tipo de análise a resposta do cérebro é medida como o resultado direto de um evento sensorial ou motor (SUR, 2009). A resposta eletrofisiológica estereotípica a um estímulo deve poder ser sincronizada com uma alta precisão temporal.

Essas questões foram exploradas em algumas pesquisas pioneiras tais como as de Mandryk e Inkpen (2004) e Hazlett (2006), que apresentaram estudos de apoio à utilização de medidas psicofisiológicas com jogos, embora com uma dimensão pequena de amostras.

O desafio na investigação dessa rica fonte de informações, tanto referente aos dados fisiológicos do jogador quanto aos dados de métricas e eventos do jogo, motiva o desenvolvimento de uma solução de software para viabilizar uma melhor análise da experiência de jogo.

## 1.1 Problemática e motivação

O cenário exposto apresenta jogos de FPS como sendo uma alternativa interessante para ambiente interativo para pesquisas que realizam experimentos psicofisiológicos com estímulos complexos. Entretanto há dificuldades de utilizar jogos nessa condição quanto a registrar eficientemente os eventos, com a precisão temporal necessária para viabilizar a pesquisa e acessar as informações a fim de remontar a experiência de jogo da sessão.

Uma solução de registro automatizado para jogos de FPS que pudesse ser utilizada para vários tipos de experimentos e pesquisas diferentes, seria algo de grande valor. Na evolução da psicologia, há uma extensa utilização de registros automáticos de eventos para entender o comportamento dos usuários no contexto de um sistema interativo analógico.

Ao se analisar softwares (jogos digitais inclusos), geralmente utilizam-se sistemas que geram *log* de dados (*log file*), ou seja, registros de eventos que ocorrem em um sistema. Dessa forma, uma pesquisa de como os jogos de FPS trabalham com esses dados permitiria desenvolver uma ferramenta de registro automatizado para esses jogos digitais.

Além de organizar os *logs* de uma maneira que poderiam ser analisados por um pesquisador, essa ferramenta poderia ainda, caso tenha desempenho suficiente, notificar um equipamento de EEG por meio de envio de marcadores na porta paralela. O marcador é um valor inteiro de 8 bit que um computador envia pela porta paralela para um equipamento de EEG. No momento que recebimento ocorre, há uma marcação daquele ponto do registro que orienta o pesquisador sobre o momento em que um determinado evento ocorreu no experimento (RUGG, 1996). Uma ferramenta com a capacidade de registrar automaticamente os eventos e enviar marcadores reduziria o tempo de marcação manual do pesquisador no registro do EEG após o experimento, evitando erros humanos de registro e garantindo uma precisão temporal muito maior para permitir utilizar técnicas avançadas de análise.

## 1.2 Objetivos

O presente trabalho tem como objetivo geral desenvolver um software para *logging* de jogos de FPS e registro automático de eventos que ocorrem nesse gênero de jogo, viabilizando a utilização desses como ambientes de estímulos psicofisiológicos para auxiliar nas pesquisas em Neurociência.

Esta ferramenta deve garantir suporte aos experimentos que utilizam jogos de tiro em primeira pessoa e efetuam algum tipo de registro psicofisiológico, como por exemplo, por meio de um equipamento de eletroencefalografia (EEG). Opções de experimentos com um ou mais jogadores deve ser possível, além de registrar

continuamente o estado (localização no mapa, campo de visão, entre outros) e ações realizadas pelos personagens dos jogadores nas partidas.

Como um dos objetivos específicos, é muito importante a capacidade da ferramenta de reconhecer eventos que ocorrem na partida, tal como um disparo ou um salto. O pesquisador deve ser capaz de avaliar facilmente o que está acontecendo ao ler o *log*, assim o *log file* deve ser estruturado para garantir essa legibilidade.

A geração de *log* feita pela ferramenta deve ocorrer paralelamente com o registro de dados fisiológicos gerado por um equipamento especializado. A ferramenta deve enviar marcadores para o equipamento do registro fisiológico quando houver a ocorrência de eventos predeterminados pelo pesquisador, sendo transmitidos por uma porta paralela disponível no equipamento.

Outro objetivo específico é o envio de marcador para o equipamento de registro, sendo essa uma característica especialmente desejada no contexto de análises temporais e para esse tipo de análise é crucial que seja feito o cruzamento do volume de dados do jogo e dos dados fisiológicos do jogador, sincronizando-os em um mesmo quadro temporal a fim de gerar um panorama completo da experiência de jogo. Esse é um dos grandes desafios, já que a precisão necessária para a comunicação com sistemas de registros psicofisiológicos tem que ser elevada para ser possível sincronizar os dados de diferentes fontes de maneira automatizada.

Outro ponto importante é o tipo de pesquisa para a qual a ferramenta desenvolvida pode ser mais útil. O contexto proposto indica o possível uso da solução em pesquisas sobre aprendizado e memória sensório motor. Isso pode ser facilmente avaliado quantitativamente por meio das métricas do jogo, mas também é possível o uso em pesquisas comportamentais.

Para permitir flexibilização do uso da ferramenta entre diversos tipo de projetos de pesquisa e experimentos é desejável que o jogo seja compatível com a ferramenta, permitindo a configuração de cenários. O pesquisador poderá assim configurar o cenário do jogo para que se adeque ao seu experimento.

Quaisquer outras necessidades mais específicas podem ser adicionadas durante análise de requisitos de software e investigações posteriores.

### 1.2.1 Validação

Com o objetivo de validar a ferramenta foi avaliado seu funcionamento em um *setup* de experimento psicofisiológico, como um EEG. Todas as funcionalidades listadas nos objetivos foram satisfeitas, bem como atendem as quaisquer outras necessidades para os experimentos em neurociências, identificadas na análise de requisitos.

Foram identificados os diversos critérios de precisão temporal possíveis avaliados por pesquisadores neurocientistas, dado que a validação deste trabalho foi realizada em conjunto com o Laboratório de Sono, Sonhos e Memória do Instituto do Cérebro (ICe) da Universidade Federal do Rio Grande do Norte (UFRN), que sedia o Programa de Pós-Graduação em Neurociências em níveis de mestrado e doutorado.

Com a avaliação positiva da precisão temporal na detecção de eventos, é garantida a possibilidade de conseguir realizar o ERP de um dos eventos, como por exemplo o disparo de tiro. A solução proposta foi habilitada a realizar um experimento de estudo de caso no contexto da utilização que a ferramenta se propõe a atender. Para isso, se faz necessário o domínio do conhecimento da área de estudo, mas também de um conjunto de conhecimentos técnicos.

### 1.2.2 Expectativas sobre os resultados

É possível reduzir consideravelmente o tempo necessário para um pesquisador em neurociências realizar uma análise quando comparado ao processo manual, uma vez que o registro manual de eventos do jogo, através da observação de um vídeo gravado durante uma partida, passaria a ser desnecessário.

Além disso, um sistema de registro automatizado permite um ganho em precisão em relação ao registro manual, que tem uma maior probabilidade de introduzir erros, garantindo assim uma integração dos eventos de jogo e das respostas fisiológicas.

A ferramenta fornece dados ricos em detalhes (como velocidade de deslocamento, movimentação do eixo x, y e z) ou dados de eventos não percebidos de forma visual como por exemplo, pisar em um item. Todos esses eventos podem ser mensurados de forma exata.

Espera-se com isso que este trabalho sirva de referência teórica e prática para o desenvolvimento de uma solução capaz de gerar um *log file* para ser integrado com os

registros de sistemas psicofisiológicos, construindo assim registros completos das experiências e eliminando as limitações técnicas das ferramentas utilizadas atualmente pelos neurocientistas.

### 1.3 Estrutura do documento

O restante do documento está organizado conforme descrito a seguir:

- **Capítulo 2 — Pesquisas cognitivas com jogos digitais:** Este capítulo apresenta o levantamento de informações quanto à área de estudos sobre pesquisas cognitivas e jogos digitais, bem como as motivações e contraindicações relacionadas à utilização deste tipo de ferramenta, técnicas e relatos das experiências anteriores;
- **Capítulo 3 — Soluções de IA aplicadas a estudos cognitivos:** Este capítulo apresenta como as ferramentas de Inteligência Artificial oriundas da área de jogos digitais podem ser aplicadas como uma alternativa para facilitar o desenvolvimento da solução proposta neste trabalho;
- **Capítulo 4 — Neuro Game Logger: uma proposta de ferramenta para análise cognitiva com jogos:** Este capítulo apresenta uma proposta de projeto de software como solução para auxiliar os pesquisadores em neurociência nas suas investigações;
- **Capítulo 5 — Conclusões e trabalhos futuros:** Este capítulo apresenta as conclusões obtidas com o desenvolvimento do trabalho e expõe possibilidades de trabalhos futuros que poderão dar continuidade à pesquisa.

## Capítulo 2

### Pesquisas cognitivas com jogos digitais

Este capítulo tem como objetivo construir uma síntese teórica da área de conhecimento principal desse trabalho. Movido pelo desejo de facilitar experiências de estudos cognitivos por meio de análises psicofisiológicas, utilizando como estímulo a experiência com jogos, se faz necessário investigar esse domínio de conhecimento.

Depois da explanação sobre estudos cognitivos, este capítulo dará continuidade à discussão sobre psicofisiologia e associa esses conceitos com o mundo dos jogos digitais, dando assim a base para sustentar os temas essenciais e as ideias apresentadas posteriormente neste trabalho.

É importante entender a motivação de utilizar os jogos digitais como ferramenta para estímulos em estudos cognitivos. Apesar da aparente falta de relação entre essas áreas, uma aproximação das mesmas vem sendo tema de diversos estudos que tentam demonstrar os efeitos psicológicos com jogos digitais por meio de evidências fisiológicas.

Para responder às questões da pesquisa, uma introdução sobre estudos científicos efetuados no contexto dos jogos digitais também se faz necessária, mais especificamente referente ao subgênero de jogos de Tiro em Primeira Pessoa (FPS).

Saber captar e processar informações sobre como um jogador interage com um ambiente de jogo e como ele se sente enquanto está jogando, são apenas alguns exemplos que podem ajudar a melhorar a compreensão sobre como os jogos digitais influenciam seus jogadores.

Com isso, se torna possível examinar como processos psicológicos ocorrem durante a experiência de jogo, sendo crucial compreender a relação entre os componentes fisiológicos envolvidos nesses processos por meio de técnicas de geração de registros, ou seja, com a marcação dos eventos ocorridos nos jogos digitais para proporcionar uma melhor rastreabilidade dos padrões e análise dos mesmos.

## 2.1 Psicofisiologia e Neurociência cognitiva

Boa parte do conhecimento científico sobre o entendimento das nossas capacidades mentais e o funcionamento do cérebro só foi possível devido à utilização da tecnologia como ferramenta de auxílio que permitiu pôr em prática diversos tipos de experimentos e observações. Através da ativação cerebral foi possível identificar uma correlação rica e precisa de variáveis fisiológicas.

Essa área de estudo é pertinente às ciências cognitivas tais como a psicofisiologia e a neurociência cognitiva. A definição de psicofisiologia adotada neste trabalho tem como base Andreassi (2000), que consiste no estudo das relações entre manipulações psicológicas e respostas fisiológicas resultantes, medidas no organismo vivo, para compreensão promovida da relação entre processos mentais e corporais. Sentimentos puros e pensamento se manifestam fisicamente através de respostas corporais, assim como o domínio do conhecimento trata sobre como os processos mentais e comportamentais que reproduzem respostas mecânicas, físicas e bioquímicas nos organismos vivos. A anatomia e fisiologia do corpo estão intrinsecamente relacionadas a essa área.

A psicofisiologia é considerada uma grande área de investigação. Com o desenvolvimento das neurociências, ela se ramificou em áreas bem especializadas, como a neurociência cognitiva. Essa área é um campo acadêmico que trata do estudo científico de substratos biológicos subjacentes da cognição, com um foco específico nos substratos neurais de processos mentais (GAZZANIGA, 2002).

Um exemplo que demonstra a relação mental e física pode ser percebido por meio da observação do aumento da atividade na musculatura orbicular dos olhos (um tipo de musculatura da pálpebra) que está ligada com a expressão de emoções positivas ou negativas (EKMAN, 1990). Estado emocional positivo ou negativo é claramente uma atividade psicológica e pode ser mensurada por uma resposta fisiológica: a atividade muscular orbicular.

Ambas as áreas se interessam em questões sobre a relação entre mente e cérebro e sua capacidade da sensação de bem-estar com os seus pensamentos (CACIOPPO, 2000), bem como questões básicas sobre a mente, os processos cognitivos superiores e o

interesse em relacionar esses processos cognitivos superiores com a integração dos processos centrais e periféricos.

Esses fatores contribuem para a plasticidade, adaptabilidade e variabilidade no comportamento, como a análise dos sistemas simbólicos de representações (por exemplo, linguagem e matemática) utilizados para a comunicação ou reflexão sobre eventos acontecidos e seus interesses, na tentativa de explicar a experiência humana e o comportamento.

O conhecimento científico estabelecido atualmente foi fundado sobre análise rigorosa de experiências, metodologicamente e tecnicamente realizadas a fim de mensurar estímulos psicofisiológicos (CACIOPPO, 2017). O entendimento de como são realizadas essas análises, conhecimento das técnicas e métodos, torna-se indispensável.

## 2.2 Análises de estudos cognitivos: métodos, técnicas e estímulos.

A análise de uma resposta psicofisiológica envolve o estudo de algo desconhecido, portanto, são necessários esforços para minimizar os erros. O estudo dos mecanismos fisiológicos e adoção de algumas metodologias podem refinar hipóteses e reduzir erros conceituais ou de medições. Entre as questões metodológicas importantes para essa formulação e delineamento, tem-se:

- 1) Selecionar as variáveis apropriadas para o estudo;
- 2) Utilizar um detalhamento refinado para ter medição das variáveis selecionadas;
- 3) Verificar a situação de variabilidade do indivíduo em relação à psicofisiologia, estando ligada ao referencial teórico pensado para o experimento;
- 4) Definidas as medidas fisiológicas, utilizar uma moda (valor que detém o maior número de observações) rigorosa para indexar fatores psicológicos.

Essas diretrizes são importantes para se evitar problemas inerentes a esse tipo de pesquisa em virtude da natureza complexa entre os eventos psicológicos e seus componentes fisiológicos. Uma pesquisa psicofisiológica é definida como a utilização de sinais fisiológicos para estudar fenômenos psicológicos (CACIOPPO, 2007). No

entanto, como processos fisiológicos não necessariamente possuem uma relação com fenômenos psicológicos, é preciso que o pesquisador interprete essa relação.

Há cinco categorias diferentes para descrever as relações em eventos psicológicos e fisiológicos (CACIOPPO, 2000). As seguintes relações podem ser vistas como projeções de um ou mais elementos do conjunto de eventos psicológicos e fisiológicos:

- Relação de “um para um” (*one-to-one*), de tal forma que um elemento no conjunto psicológico está associado a um e somente um elemento no conjunto fisiológico;
- Relação de “um para muitos” (*one-to-many*) significa que um elemento no conjunto psicológico está associado a um subconjunto fisiológico de elementos;
- Relação “muitos para um” (*many-to-one*) significa que dois ou mais elementos no conjunto psicológico estão associados a um único elemento no conjunto fisiológico;
- Relação “muitos para muitos” (*many-to-many*) significa que dois ou mais elementos no conjunto psicológico estão associados com dois ou mais elementos (subconjunto) no conjunto fisiológico;
- Relação nula (null) significa que não existe nenhuma associação entre elementos no conjunto psicológico e com elementos no conjunto fisiológico.

A atribuição de significado psicológico para fisiologia é uma das últimas fases da análise, devendo levar em consideração as respostas que residem em fatores, tais como a qualidade do desenho experimental, as propriedades psicométricas das medidas, bem como a adequação da análise de dados e interpretação.

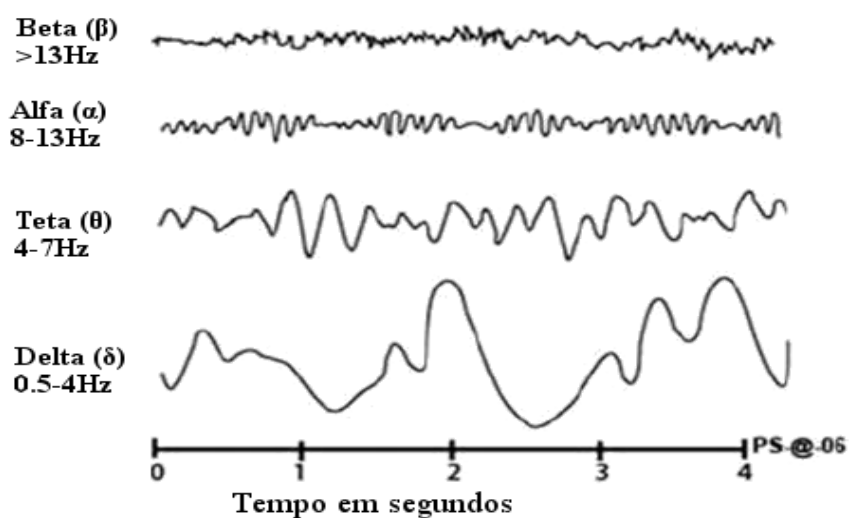
Além dessas questões de fundo metodológico, no contexto técnico existem outras medidas de controle com chance de causar imprecisão ou ambiguidade ao se definir a relação, destacando assim a importância de ferramentas de análise para uma abordagem mais refinada.

Aplicativos de planilhas eletrônicas como o Excel, Spotfire e Tableau permitem a visualização rápida de grandes conjuntos de dados para explorar rapidamente o significado e as relações dos mesmos, e escolher entre visualizações, incluindo gráficos e diagramas de dispersão. Além disso, os dados exibidos podem ser selecionados e

alterados dinamicamente. Noldus e Mangold, como descrito por Bakeman (2008), fornecem softwares para a aquisição, análise e apresentação de vídeo, áudio e dados do sensor (incluindo o olhar e dados psicofisiológicos) a partir de estudos comportamentais.

O arcabouço técnico é importante para a aquisição dos dados. Por meio de um equipamento de análise psicofisiológica que permite monitorar as estruturas ou funções de células ou órgãos, é possível investigar o organismo e suas relações com o ambiente físico ou sociocultural. Considera-se uma coleção elaborada de medidas fisiológicas, descritas por Van Weert (2006), na qual se deve incluir os seguintes instrumentos: medidas cardíacas e respiratórias, medidas de piscar de olhos, rastreamento do olhar, dilatação da pupila, expressões faciais (eletromiografia facial), atividade do cérebro (eletroencefalografia).

A eletroencefalografia (EEG) oferece um método não invasivo para estudar as respostas corticais a partir de diferentes eventos de estímulo com uma resolução temporal alta. Quanto às características eletrofisiológicas, existe uma ampla área de estudo que analisa esse tipo de sinais. Na maior parte do tempo, as ondas cerebrais são irregulares no eletroencefalograma, porém em algumas situações aparecem padrões, sendo alguns deles fisiológicos (frequências do tipo alfa, beta, teta e delta, conforme representadas na Figura 2) e outros característicos de anormalidades específicas (GOMES, 2010).



**Figura 2 - Tipos de ondas referentes às frequências alfa, beta, teta e delta de um eletroencefalograma.**

Desde a sua invenção, essa ferramenta foi amplamente utilizada em pesquisas aplicadas em neurociências pela quantidade de dados apresentáveis nas suas diferentes faixas de frequência: pós-circulação, sincronização beta associada com as funções motoras; o aumento do poder alfa tem sido associado com lentidão cortical ou inatividade; várias respostas oscilatórias foram relatadas durante tarefas que requerem processos de memória; e respostas relacionadas com emoções também têm sido relatadas.

Métodos computacionais de processamento podem oferecer uma maneira melhor para a obtenção e seleção dos sinais interpretáveis, especificamente para o EEG. Com esses métodos é possível fazer uma análise independente que permite a separação dos sinais que se decompõem ou discernem a combinação linear de outros sinais (ONTON, 2006). Sendo usados para recuperar sinais que representam ativações específicas do cérebro e separação de ruído, essas diferentes fontes se misturam durante o processo de registro do EEG como utilizado por Groppe (2009) e McMenamin (2010). McMenamin (2010) ainda utilizou o método de regressão de potência, necessário para ajudar a reduzir o problema comum de ruído elétrico gerado pelo movimento muscular involuntário.

A importância do desenvolvimento de procedimentos mais avançados de registros psicofisiológicos, fenômenos previamente não observáveis, são atualmente possíveis. Contudo, os procedimentos de registros mais avançados por si só, não são suficientes para o progresso da área.

Com formas mais poderosas e completas de análise é possível fornecer estímulos mais ricos para experimentos. O conhecimento sobre os sistemas fisiológicos é logicamente necessário para atribuir significado psicológico a esses eventos. Podemos ainda comparar várias outras técnicas utilizadas nesse tipo de investigação.

A Tabela 1 mostra alguns exemplos de outras técnicas utilizadas em pesquisas cognitivas, descrevendo seus atributos de objetividade e restrição. Objetividade descreve o grau de precisão dos dados, fazendo uma reflexão quanto ao impacto desse atributo em experimentos que utilizam estímulos dinâmicos e interativos como é o caso dos jogos digitais, nos quais observa-se que o uso exclusivo do auto-relato ou de um questionário é problemático (KIVIKANGAS, 2011).

**Tabela 1 - Técnicas psicofisiológicas, objetividade e restrição.**

<b>Técnica</b>	<b>Objetividade</b>	<b>Restrição</b>
Auto-relato	Baixo	Médio
Questionário	Baixo	Médio
Gravação	Baixo	Baixo
Teste de Associação Implícita	Baixo	Baixo
Eletrocardiograma (ECG)	Alto	Alto
Eletromiografia Facial (EMG)	Alto	Alto
Eletroencefalografia (EEG)	Alto	Alto

Interromper a sessão de jogo após a ocorrência do evento estudado, para um responder um relatório por exemplo, interfere na experiência global. Sua intervenção após a sessão de jogo também não é confiável, em consequência da erosão dos sentimentos pela distância do tempo passado desde o acontecimento do evento. Por fim, uma vez que cerca de 80% das ações humanas são subconscientes descritos por Ruch e Zimbardo (1974), os resultados de auto-relatos e questionários são considerados pobres em informação. Os resultados de ambas as técnicas podem ser considerados como tendenciosos, pelo aspecto subjetivo dos mesmos.

Kivikangas (2011) apresenta várias vantagens que as medidas psicofisiológicas possuem em relação ao auto-relato na pesquisa com jogos digitais:

- Medições podem ser realizadas continuamente com alta resolução temporal;
- Processos de interesse podem ser avaliados de forma encoberta;
- Potencial de informações sobre respostas emocionais e de atenção, o que podem não estar disponíveis para a percepção consciente;
- Grande quantidade de dados que permite o uso de técnicas de análise de séries e temporais em conjunto.

A característica de Restrição é definida como o quanto a técnica diminui a liberdade de movimentos ou ações do usuário. Realizando a reflexão quanto ao impacto sobre esse atributo, as medições com EEG, que incluem a fixação de eletrodos sobre a pele humana, por exemplo, podem ser consideradas como muito invasivas. Entretanto essa restrição diminui em experimentos que utilizam jogos digitais como estímulo, não sendo necessariamente algo tão impactante, uma vez que existem muitos jogos digitais que não exigem amplos movimentos físicos.

Entendendo isso, é possível determinar um conjunto bem definido de medidas a serem escolhidas em relação ao contexto da pesquisa. É difícil fazer a associação entre um evento fisiológico específico e um processo psicológico por ser muito propenso a ambiguidades.

Por fim, os estudos relatando experimentos com jogos digitais têm se revelado muito interessantes apesar disso tem sido menos utilizado do que se poderia por questões técnicas. O uso de medidas psicofisiológicas no contexto dos jogos digitais descreve uma tarefa complexa (KLIMMT, 2003). As conclusões apoiam a utilização de jogos digitais e métodos de análise psicofisiológica em pesquisas, mas resta a clara necessidade de um modelo experimental mais adequado.

### 2.3 Motivação para utilização de jogos em estudos cognitivos

A jogatina exige um conjunto de habilidades cognitivas (elaborar uma estratégia, por exemplo) e coordenação psicomotora. Essa associação de habilidades cria um ambiente propício à plasticidade cerebral (WU, 2012).

Como definido por Livingston (1966) plasticidade do cérebro é a capacidade de modificar a sua função, estrutura e química em resposta às novas experiências e os jogos digitais acabam sendo um motor para testar a capacidade de plasticidade do cérebro e a sua interação com o ambiente simulado.

O controle desse ambiente, aliado à capacidade de analisar as variações das reações físicas de um indivíduo, permitem mensurar objetivamente os processos emocionais e cognitivos do jogador durante a experiência de jogo. Além disso, esse recurso habilita medidas sensíveis o suficiente para extrair respostas que o olho humano não consegue detectar.

É possível avaliar vários estados emocionais dos jogadores por meio de sinais fisiológicos, como a atividade elétrica na pele, frequência cardíaca, pressão arterial, respiração, temperatura e até mesmo o EEG (CHANEL, 2011; RANI, 2006; TIJS, 2008), permitindo o acesso a informações praticamente impossíveis por métodos subjetivos (observação ou questionário).

A literatura científica possui diversas experiências nesse contexto. Alguns dos primeiros pesquisadores a utilizarem esse caminho foram Mandryk e Inkpen (2004) assim como Hazlett (2006), que apresentaram estudos, embora com pequena dimensão amostral, apoiando a utilização de medidas psicofisiológicas em pesquisas com jogos digitais. Mais recentemente, Nacke tentou desenvolver uma metodologia de abordagem para orientar projetos desse tipo em seus artigos (NACKE 2009, 2010; DRACHEN, 2009).

Em paralelo, o projeto FUGA (RAVAJA, 2006) teve como objetivo criar novos métodos e melhorar os critérios existentes, a fim de analisar a maneira como os diferentes aspectos da experiência de jogo (emoções e cognições) podem ser avaliados de forma abrangente e em alta resolução temporal. Foram utilizadas análises com equipamentos de EMG (Electromyography), EEG (Electroencephalography), EDA (Electrodermal Activity), FMRI (Functional Magnetic Resonance Imaging) e rastreador dos olhos, para avaliar os indicadores comportamentais (KIVIKANGAS, 2009; SENNERSTEN, 2008; KLASSEN, 2008; VAN DEN HOOGE, 2009).

Os dados dessas primeiras pesquisas geraram resultados interessantes com uma boa concordância entre previsão e observação. No projeto FUGA foram utilizados EDA e EMG para medir a atividade facial (FUGA, 2009). Mandryk e Atkins (2007) usaram modelos de lógica difusa para mapear automaticamente gravações de dispositivos fisiológicos (EDA, EMG facial e medidas cardíacas), reconhecendo posteriormente com sucesso, cinco emoções relevantes aos jogos digitais.

Em estudos mais recentes vários grupos desenvolveram métodos para avaliar o desempenho dos jogadores nos jogos digitais por meio de dispositivos de geração de registros psicofisiológicos (ANGUERA, 2013; MATHEWSON, 2012; SIVANATHAN, 2014; SPULER e NIETHAMMER, 2015).

No entanto, um problema persistente nas soluções da neurociência cognitiva se refere à generalização. Uma vez que interpretações neuronais são feitas em um contexto

específico durante uma tarefa cuidadosamente controlada, é difícil transferir conhecimento de uma tarefa para outra.

Em pesquisas mais recentes (CAVANAGH, 2016), jogos digitais foram aplicados para tentar resolver esses problemas de generalização utilizando uma plataforma de 8-bits, através de uma série de eventos orientados, tais como se chocar em uma parede ou explodir um inimigo com um míssil, tendo como objetivo definir o significado psicológico desses eventos. Para atingir este objetivo, foram utilizados classificadores padrões para derivar pesos preditivos de eventos como detecção do alvo excêntrico (*oddball target detection*) e aposta de vitórias e perdas (*gambling wins and losses*).

O *oddball* é um projeto experimental utilizado na investigação ERP, em que apresentações de sequências de áudio ou estímulos visuais repetidos são interrompidos por um estímulo aberrante. O *gambling* é outro projeto experimental utilizado na investigação ERP, ao colocar o indivíduo em situação na qual ele pode ganhar ou perder, existem a ocorrência de padrões estabelecidos para cada cenário.

Foram comparadas essas medidas como as atividades do EEG durante os eventos do jogo. Todos os eventos estudados se mostraram significantes no intervalo de tempo P3 (a onda P300 é um ERP relacionado ao processo de tomada de decisão). Considera-se que esse potencial é endógeno, a ocorrência determinando a presença de um processo de detecção.

Isso permitiu demonstrar, em estudos experimentais, os benefícios das medições efetuadas a partir de registros gerados automaticamente e continuamente em tempo real, aliados às características dos jogos digitais de serem um poderoso meio de estímulo. Ainda assim, é inegável que existem limitações em relação às experiências no mundo real.

## 2.4 Limitações para utilização de jogos em estudos cognitivos

Os jogos contemporâneos oferecem uma interpretação visual, auditiva e tátil do mundo do jogo, deixando de fora muitas das sutilezas que compõem a nossa percepção do mundo real (como nosso senso de equilíbrio, etc.). A perda mais significativa da

percepção deriva da impossibilidade de transferir para o mundo virtual os estados internos do avatar do jogador (KIVIKANGAS, 2011).

Não existe possibilidade dos jogadores fazerem uso dos mecanismos de percepção normalmente utilizados para estimar o estado de saúde, exaustão, etc. Além disso, os mecanismos internos básicos, tais como sensação de velocidade ou sensação de altura são reduzidos à interpretação da informação visual, auditiva e tátil transmitida (KLEVJER, 2006).

Dados os hardwares atuais permitem uma experiência de jogo é obtida por meios visuais, auditivos e formas limitadas de feedback tátil, o que corresponde à percepção humana. Logo, é necessário explorar os elementos da interface gráfica para maximizar a percepção do jogador sobre seu avatar. Essa "consciência" dentro do mundo do jogo pode se manifestar assemelhando-se a presença humana no mundo real (ERIK, 2009).

Mesmo se as pessoas apresentam grande atenção aos sentidos da visão e da audição para navegar no mundo real, há uma série de sentidos e mecanismos de percepções básicas que nunca são empregadas quando se joga um jogo digital, servindo de obstáculo a um maior grau de envolvimento no mundo do jogo. Além das características que os jogos digitais possuem, ainda existem questões da própria natureza da análise psicofisiológica usando o EEG.

Apesar desses equipamentos possuírem uma requintada fidelidade temporal, a resolução e sensibilidade multidimensional do EEG são barreiras que limitam a inferência inversa. A inferência tradicional em estudos de neurociência é quando se analisa certo processo cognitivo e uma área cerebral específica é ativada. A inferência inversa, quando a ativação de uma área cerebral associada a um um processo cognitivo, essa associação é propensa a falha, especialmente, em EEG (POLDRACK, 2008). Dificuldades com o uso de jogos digitais e EEG são relatadas em experiências anteriores (HAVRANEK, 2012; SALMINEN e RAVAJA, 2008; SPAPÉ, 2013; SUBHANI, 2012), ou ambientes de realidade virtual (CALLAN, 2013).

## 2.5 Escolha de estilo de jogo adequado à pesquisa

Apesar das limitações apresentadas até o momento, jogos digitais oferecem opções viáveis de estimulação com vários estudos que os associam a medições psicofisiológicas em diferentes áreas científicas, com diversas motivações.

Processos psicofisiológicos necessitam de uma precisão temporal, medição contínua e exigem que se extraiam partes dos dados que correspondem a um evento repetitivo (MANDRYK e ATKINS, 2007). Desse modo, é possível testar estatisticamente a tendência de todas as reações do experimento.

Jogos podem ser especialmente interessantes para estudos envolvendo sonhos pois se assemelham aos jogos, no sentido que são práticas para eventos futuros, não estando limitados a simulações de ameaças. Este entendimento dos sonhos recebeu atenção empírica e teórica (BULKELEY, 2004) e tem sido sugerido que os sonhos podem oferecer um melhor modelo do ambiente do que a própria consciência (REVONSUO, 2006).

Revonsuo (2004) indica que ambos, sonhos e ambientes de realidade virtual (VR), são simulações poderosas do mundo real que resultam em modelos auto-referenciáveis. Wamsley (2014) também indica que o cérebro é geneticamente dotado de um gerador de realidade virtual inato que, através de plasticidade dependente da experiência, torna-se um modelo generativo ou assistido do mundo. Em outras palavras, podemos concluir que sonhos e VR possuem equivalência, sendo mundos gerados artificialmente, um deles biologicamente impulsionado e o outro tecnologicamente conduzido.

Modelos auto-referenciáveis (sonhos, VR, vigília) se afetam às vezes de maneira profunda como nos pesadelos onde a vítimas acordam ofegantes, outras vezes o impacto é menos profundo, como quando se joga um jogo digital por muito tempo, no qual a ação de levantar do sofá resulta em tonturas, de forma equivalente ao "acordar" da realidade virtual.

Para a utilização de jogos digitais nesse tipo de análise, é exigida a capacidade de respostas rápidas a eventos. Resultados de pesquisas anteriores indicam que jogos de FPS promovem aumento de excitação (batimento cardíaco, resposta galvânica da pele, EEG) e do afeto positivo (EMG) (MANDRYK e INKPEN, 2004), enquanto que o ato

de alcançar metas promove a diminuição da excitação e o aumento do afeto positivo (SALMINEN e RAVAJA, 2007). Mais interessante, e ao contrário do que se poderia esperar, já foi percebido nos jogos de FPS que a morte do personagem do jogador parece causar um afeto positivo, enquanto que eliminar um personagem do adversário induz a uma reação negativa, no entanto isso pode ser causado pelas limitações do experimento (RAVAJA, 2005, 2006, 2008).

## 2.6 Jogos de tiro em primeira pessoa no contexto de estudos cognitivos

No contexto da psicologia experimental, jogos de FPS apresentam características interessantes, como em ambientes de jogos digitais que permitem experimentar questões éticas ou situações similares da vida real que seriam difíceis ou custosas de se reproduzir (RAVAJA, 2006). Jogar também evoca respostas afetivas, tais como agressão e várias emoções que podem ser relacionadas e quantificadas por meio das medidas de respostas fisiológicas específicas (RAVAJA, 2006). Dado isso, os jogos de FPS correspondem a um gênero que consegue explorar essas características de maneira bastante interessante.

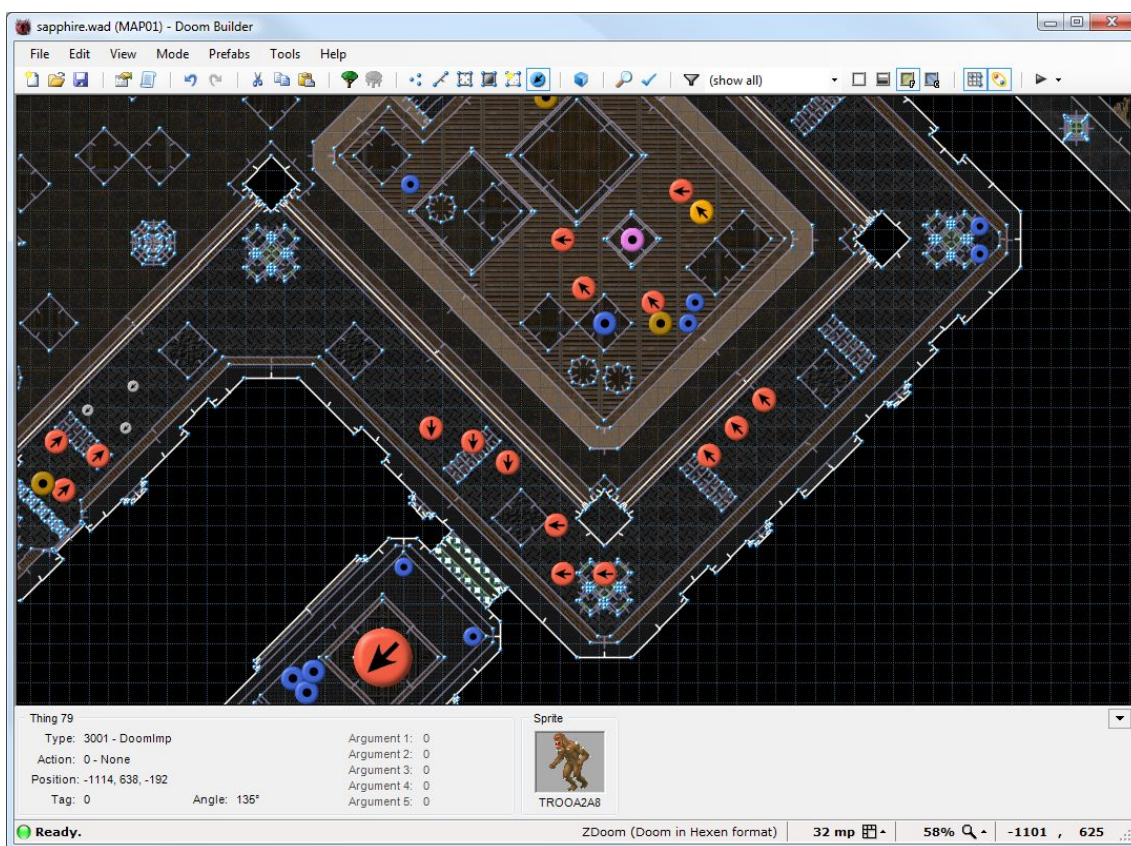
O primeiro título com maior notoriedade desse gênero de jogos digitais, chamado DOOM (ver Figura 3), foi lançado em 1993 pela Id Software LCC.



Figura 3 – Tela de gameplay do jogo DOOM.

Nesta época, a principal preocupação dos desenvolvedores era oferecer uma experiência motora aos jogadores, o que consiste em uma característica importante de muitos jogos de FPS.

Esse jogo era jogado através de um computador pessoal (PC) e acompanhava uma ferramenta de edição de nível (ver Figura 4), permitindo a criação de uma *modificação* do jogo conhecida como *mod*. Isso permitiu aumentar a flexibilidade da experiência de jogo e criar variações de acordo com as necessidades dos jogadores (CIFALDI e FRANK, 2009).

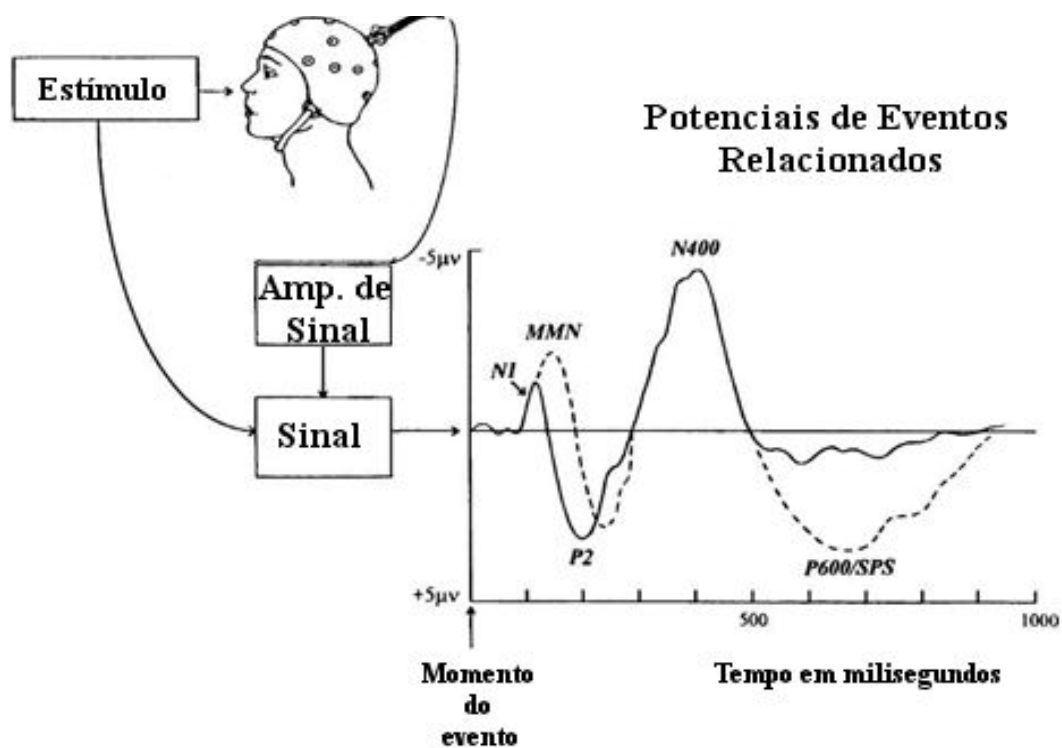


**Figura 4 – DOOM Builder editor de mapas do jogo DOOM.**

De maneira geral, jogos de FPS começaram com essa característica de criar modificações e estenderam a outros estilos de jogos digitais. Este gênero exige que o jogador execute tarefas que fazem uso da visão periférica (inimigos muitas vezes aparecem pela primeira vez na periferia da tela), com alta precisão temporal (a velocidade de jogo geralmente é intensa) e prestando bastante atenção como muitos inimigos, muitas vezes aparecem simultaneamente (KLEVJER, 2006).

O processo de imersão é algo desejado no contexto dos estudos cognitivos. De acordo com Rune Klevjer (2006), jogos de FPS podem ser vistos como uma espécie de simuladores (KLEVJER, 2006), envolvendo assim a percepção, habilidades de raciocínio e até mesmo a emoção dos jogadores durante cada sessão de jogo, de forma bastante similar ao que se obtém no mundo real. Neste contexto, é possível identificar diversas experiências utilizando jogos de FPS em pesquisas na área de psicofisiologia.

### 2.6.1 Jogos de tiro em primeira pessoa e análise de potenciais de eventos relacionados



**Figura 5 - Potenciais de Eventos Relacionados (ERP)**

É possível realizar análises que permitem identificar uma atividade cerebral específica quando um indivíduo é exposto a determinados estímulos (internos ou externos). Potenciais de Eventos Relacionados (ERP), conforme Figura 5, são técnicas que permitem medir a atividade cerebral por meio dos eventos de um EEG, através das flutuações de tensão positivas e negativas (Osterhout, 2004).

Através da análise de ERP é possível descrever como o ato de jogar um jogo de FPS altera diretamente as atividades neurais, buscando possíveis alterações de

neuroplasticidade associadas a esse tipo de jogo. Uma pesquisa realizada neste sentido por Wu (2012) efetuou análises de ERP em alguns participantes.

A atividade do cérebro reflete algum estímulo gerado por um evento, entretanto como outros sinais também ocorrem simultaneamente, diversas atividades se apresentam associadas com o evento de interesse. Isto significa que a resposta do cérebro a um estímulo de interesse normalmente não é visível no registo de EEG logo no primeiro ensaio. Assim para se mensurar de forma confiável é necessário repetir o estímulo a fim de filtrar os sinais gerados por outros milhares de processos cerebrais que não são alvo do estudo, mas são ativados simultaneamente.

Essa filtragem analisa as respostas do cérebro a um estímulo, muitos experimentos, sendo calculado a média dos resultados, fazendo com que a atividade aleatória do cérebro seja desprezada para obter uma forma de onda relevante. O cálculo da média, enquanto que atividade cerebral de interesse (LUCK, 2014), é definido por  $k$  (número do teste) e  $t$  (tempo decorrido após o evento  $k$ -enésimo), cada teste gravado podendo ser modelado pela seguinte equação:  $x(t, k) = s(t) + n(t, k)$ , onde  $s(t)$  é o sinal e  $n(t, k)$  é o ruído. Nota-se que, sob as premissas acima, o sinal não depende do teste específico, enquanto o ruído depende. A função de cálculo da média de ERP de  $N$  testes é apresentada na Figura 6.

$$\bar{x}(t) = \frac{1}{N} \sum_{k=1}^N x(t, k) = s(t) + \frac{1}{N} \sum_{k=1}^N n(t, k)$$

**Figura 6 - Equação do cálculo da média de ERP para N testes,  $k$  (número do teste) e  $t$  (tempo decorrido após o evento  $k$ -enésimo),  $s(t)$  é o sinal e  $n(t, k)$  é o ruído**

Através deste procedimento de cálculo da média de sinal é possível isolar a resposta do cérebro que é especificamente induzida em resposta a algum caso de interesse. O ERP normalmente é nomeado em termos de sua polaridade e latência de pico (em milissegundos).

Devido à complexidade desse tipo de análise, utilizar jogos de FPS como estímulo é considerado uma tarefa desafiadora. Uma pesquisa pioneira realizada por Ravaja (2008) teve como objetivo analisar a jogabilidade no contexto da psicofisiologia.

Essa pesquisa descreveu o uso de sistemas de registros psicofisiológicos nos jogos aplicados às pesquisas em neurociências.

Niklas Ravaja, em sua publicação, analisou respostas fisiológicas por meio do EEG para duas situações violentas durante a experiência de jogadores do jogo *James Bond 007: NightFire* (ver Figura 7).



**Figura 7 - James Bond 007: NightFire Gameplay**

Dois tipos de eventos mapeados nesses experimentos consistiram nas ações de ferir o personagem do jogador e de destruir o personagem do oponente. O EEG foi utilizado para registrar as partidas jogadas.

Os dados da leitura foram segmentados em intervalos com duração de 1 segundo antes das ocorrências dos eventos e em intervalos com duração de 2 segundos durante a ocorrência dos eventos. Estimativas de potência elétrica captadas pelos eletrodos foram

derivadas com a transformada de Fourier (FFT) para cada evento registrado no equipamento de EEG.

Os resultados foram discutidos no contexto das respostas a atividades eletromiográficas e eletrotérmicas faciais evocadas por estes mesmos eventos, indicando que as respostas do EEG relatadas poderiam ser atribuídas a processos afetivos relacionados com estes acontecimentos de jogos violentos.

Os resultados da pesquisa demonstraram uma relação causal direta entre jogar um jogo de FPS e a atividade neural que suporta atenção seletiva espacial. Além da melhoria comportamental, os substratos eletrofisiológicos de atenção seletiva espacial foram modificados no caso dos jogadores que apresentaram alto desempenho. Os efeitos ocorreram numa fase relativamente tardia de processamento de informações visuais (cerca de 200 milissegundos após o início do estímulo) durante a alocação de recursos de atenção (SORTE e VOGEL, 2000).

## 2.6.2 Jogos de tiro em primeira pessoa e Registro de Eventos

Os jogos de FPS se mostram como ferramentas eficazes para o treinamento uma vez que são capazes de melhorar uma variedade de habilidades de atenção e de percepção (SPENCE e FENG, 2010; DYE e VERDE, 2009; FENG, 2007). Contudo é importante definir alguns requisitos necessários para viabilizar essa ferramenta como ambiente de aprendizagem.

Para reunir dados significativos para posterior análise, marcadores são geralmente inseridos manualmente durante o procedimento de aquisição de dados. Contudo isso geralmente consome um tempo considerável da pesquisa, além de propiciar a inserção de erros, portanto uma possível alternativa consiste em um sistema de geração automatizada de registros de eventos com o objetivo de facilitar a análise das marcações dos eventos ocorridos no ambiente dos jogos.

Um sistema de geração de registros que tenha interface eficaz com o jogo digital, em tempo de execução e envie eventos relacionados com o jogo na mesma resolução temporal, seria muito benéfico para os pesquisadores interessados no estudo da experiência dos jogadores.

A geração de registros de tais eventos, com a ajuda do mesmo hardware que é

usado para a geração dos registros de sinais biológicos, poderia permitir aos pesquisadores efetuarem a tarefa de análise de forma mais rápida e eficaz.

O grande número de estímulos gerados pelos jogos de FPS tem como resultado a geração de um volume de dados tão grande quanto, dito isto, no contexto das pesquisas nessa área, é necessário encontrar formas de como conseguir perceber eventos específicos nesse ambiente de proporções grandiosas. Nessa conjuntura, existem dois escopos de ocorrências de eventos quando utilizando jogos como ferramenta de estimulação:

1. Eventos que ocorrem no mundo virtual do jogo, como um personagem sendo destruído, sofrendo danos, recebendo uma recompensa, etc.;
2. Eventos que ocorrem no mundo real com o jogador, por exemplo, mudanças no padrão de ondas perceptíveis pelo EEG, gritos do jogador devido ao seu estado de raiva, movimentos bruscos devido ao nervosismo, etc.

Em uma pesquisa coordenada por Nacke (2008, 2009), foi desenvolvida uma plataforma de geração de registros de eventos para o jogo Half-Life, FPS desenvolvido pela Valve (ver Figura 8), seguindo a ideia de automatização do procedimento de análise.



**Figura 8 - Half-Life gameplay**

Essa ferramenta tinha como objetivo determinar a valência emocional e a excitação provocada por alguns eventos específicos através de um rastreador dos olhos (ver Figura 9), conforme citados abaixo:

- Quando o jogador dispara um tiro com uma arma;
- Quando o jogador se machuca;
- Quando o jogador morre.



**Figura 9 - Rastreador de olhos**

Os eventos ocorridos são enviados via porta paralela do computador durante a execução do jogo, gerando um arquivo de *log* (registro de eventos). No arquivo de *log*, existe uma descrição sobre o tipo do evento e o instante da sua ocorrência. Também é comparado com o registro do rastreador dos olhos, sendo esse um dispositivo que serve para medir as posições dos olhos e seus movimentos. Esses equipamentos são usados em pesquisas sobre o sistema visual, na psicologia, na psicolinguística, marketing, na forma de dispositivo de entrada para interação homem-computador, além do campo de design de produtos.

A capacidade de registrar dados de duas maneiras distintas permite que os pesquisadores desprovidos de dispositivos de geração de registros psicofisiológicos (que recebe dados de uma porta paralela), obtenham os dados automaticamente em formato

de arquivos de log.

## 2.7 Conclusões sobre pesquisas cognitivas e jogos de tiro em primeira pessoa

Existem muitas vantagens relacionadas à utilização de jogos digitais como ferramenta de estímulo para experimentos na área de psicofisiologia, uma vez que eles oferecem estímulos complexos e variados aos jogadores, como sensorio motor que exigem um processamento cognitivo em diferentes níveis de interpretação e operam em escalas de tempo muito fragmentadas que podem levar desde alguns segundos até algumas horas (KLIMMT, 2003).

Apesar dessas vantagens, as análises psicofisiológicas que recorrem a esse tipo de estímulo necessitam de precisão temporal e medição contínua (MANDRYK e ATKINS, 2007), visando monitorar em tempo real e analisar psicofisiologicamente uma sessão de jogo. Essa é uma necessidade tão importante quanto o monitoramento por fontes distintas, podendo assim recolher os dados comportamentais por meio de *logfile* do jogo (DEKKER e CHAMPION, 2007; COWLEY, 2009). Estes dados relacionados aos eventos também podem ser acompanhados por registros de vídeo para permitir prover informação contextual.

A combinação de diferentes metodologias de análise (vídeo do *gameplay*, *logfile*, registros psicofisiológicos) fornece uma visão mais completa da experiência de jogo, proporcionada pela integração de fontes de dados diferentes, mas com o mesmo quadro temporal. Isso permite uma melhor investigação dos padrões estudados e um controle maior do experimento e, eventualmente, resultados mais fidedignos e generalizáveis.

Muitos sistemas operacionais, estruturas de software e programas incluem um sistema de registro. Um padrão de *log* amplamente utilizado é o *syslog*, definido no *Internet Engineering Task Force* (IETF) RFC 5424. Esse padrão também pode ser útil para combinar entradas do arquivo de *log* a partir de múltiplas fontes. Esta abordagem, em combinação com uma análise estatística, pode produzir correlações entre eventos aparentemente não relacionados em diferentes servidores, outra solução consiste em empregar toda a rede de consulta e relato.

A análise contínua gera um grande volume de dados, portanto as reações fisiológicas alvo do estudo podem se confundir com o sinal gerado pela interação de outros fenômenos não relacionados com o objetivo da pesquisa, a não ser que o fenômeno de interesse seja muito forte ou se tenha o tamanho amostral grande o suficiente. O pesquisador deve ponderar durante a execução de um estudo, em adotar certos controles para que não ocorra essa confusão conservando o resultado final (MAXWELL e DELANEY, 2004).

A existência de um cenário de jogo, assim como a capacidade de interação com os seus elementos é importante. Essa característica permite utilizar as respostas fisiológicas registradas durante os experimentos para se observar *insights* gerados pelos participantes durante a jogatina.

Jogos digitais possibilitam uma grande oportunidade de interações com o ambiente e gravação de eventos ocorridos nessa experiência, o apelo cognitivo gerado por essa interação ajuda o jogador a recordar a experiência de jogo no evento pertinente à pesquisa e essa característica é crítica para seu funcionamento.

### 2.7.1 Problemática das Soluções atuais

Pesquisas que relacionam jogos e análises psicofisiológicas ou mesmo estudos do sono, demonstram uma abordagem poderosa, apesar disso muitas não possuem um aproveitamento pleno do potencial de análise que esse tipo de ferramenta pode oferecer. Uma ferramenta para auxiliar a pesquisa poderia evitar o mau aproveitamento de experimentos, possibilitando análises mais profundas e a diminuição do trabalho por meio de automatização.

De maneira geral, as soluções disponíveis não possuem um acesso à aplicação utilizada ou não possuem documentação do software, outra característica dessas soluções é que muitas vezes elas foram desenvolvidas para um contexto de pesquisa específico. A falta de documentação e a especificidade das ferramentas praticamente impossibilita mensurar se poderiam ser adaptadas para pesquisas de maneira geral.

Apesar da impossibilidade da adaptação das ferramentas existentes para a pesquisa, elas geram acesso a experiências e reflexão válidas para o desenvolvimento de uma solução genérica. Com o objetivo de superar problemas definidos nas experiências

anteriores que poderiam inviabilizar o presente trabalho, realizou-se um estudo bibliográfico em outras áreas do Conhecimento, como é o caso do desenvolvimento de personagens inteligentes.

### 2.7.2 Similaridade entre estudos cognitivos e personagens inteligentes

Os requisitos necessários para o monitoramento em tempo real de uma sessão de jogo, a geração de um *logfile* e a resposta a eventos dentro do jogo, são problemas abordados em várias pesquisas que utilizam jogos digitais. A área de Inteligência Artificial (IA), e mais especificamente de personagens inteligentes, vem desenvolvendo soluções neste sentido há várias décadas.

Para essa área da computação, quando aplicada aos jogos digitais, diversos modelos de simulações abordando ambientes ricos e realistas, contendo muitas propriedades existentes no mundo real, têm sido tratados com o intuito de desenvolver personagens virtuais inteligentes e automatizados.

Os pesquisadores da área de inteligência artificial muitas vezes se interessam em jogos digitais como ambientes de experimentação para avaliação do comportamento gerado por técnicas de inteligência artificial, utilizando uma abordagem baseada em um modelo de interface de comunicação entre os jogos digitais e as técnicas experimentadas no contexto da pesquisa.

Problemas similares ao dos personagens virtuais inteligentes e automatizados se mostram presentes no desenvolvimento de uma interface de comunicação entre jogos digitais e softwares de análises psicofisiológicas. Ambas as áreas apresentam requisitos de software muito parecidos em determinados aspectos. Portanto, tendo em mente que as soluções existentes atualmente para os personagens virtuais inteligentes e automatizados podem servir como base para o presente trabalho, pela forma que eles acessam as informações do jogos serem muito similares. O próximo capítulo introduz as soluções existentes para que possamos fazer um paralelo com a área de estudos cognitivos.

## Capítulo 3

### Soluções de IA aplicadas a estudos cognitivos

Na última década, particularmente, os jogos digitais também passaram a ser considerados como uma forte fonte de *benchmarks* para o estudo da inteligência humana (YANNAKAKIS, 2014). Jogos são desenvolvidos para serem desafiadores e interessantes para os seres humanos, portanto, muitos problemas presentes neles são relevantes para a compreensão da cognição humana e estímulo à criatividade.

Quando esses problemas são simulados, permitindo serem resolvidos com o auxílio de técnicas de Inteligência Artificial (IA) (RUSSELL e NORVIG, 2009), passam a se mostrar como laboratórios de grande contribuição com o processo de resolução de problemas do mundo real.

Para a área de Inteligência Artificial (IA), especificamente, os jogos digitais fornecem uma infinidade de desafios interessantes, incluindo o desenvolvimento de personagens inteligentes automatizados (do inglês *Non-Player Characters* - NPC), sistemas computacionais criativos para geração de conteúdo, agentes automatizados que analisam o ambiente onde vivem e se adaptam ao comportamento de jogadores humanos, entre outros. A área de IA já tem algumas décadas de experiência com a aplicação de jogos em suas pesquisas, em um papel similar ao utilizado em pesquisas em Neurociência Cognitiva.

Este capítulo visa investigar diversos jogos e suas ferramentas voltadas ao desenvolvimento de personagens inteligentes automatizados para direcionar a proposta que será apresentada no presente trabalho. Esse direcionamento se deve ao fato que as ferramentas utilizadas pela comunidade de IA e jogos digitais possuem características similares ao problema aqui tratado e por essa razão merecem ser investigadas.

#### 3.1 Personagens Inteligentes em Jogos de FPS

Os desenvolvedores de jogos digitais têm gasto seus esforços no design de ambientes 3D realistas a fim de atrair cada vez mais jogadores. Além disso, visando

enriquecer a experiência do jogador, eles também têm evoluído nos seus estudos das técnicas para construção dos personagens automatizados presentes nos jogos, o que tem fomentando o interesse pela área de IA.

No contexto acadêmico, o interesse dos pesquisadores de IA também se dá cada vez mais pela área de jogos digitais, como pode ser demonstrado pelo número de eventos específicos e periódicos importantes que incluem a *IEEE Conference on Computational Intelligence and Games (CIG)*, a *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, bem como a *IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG)*, entre outros.

As primeiras experiências do desenvolvimento de personagens automatizados (NPCs) eram de simples interação, por exemplo, correr em direção do avatar do jogador e atacá-lo. As limitações dos personagens podiam ser explicadas tanto por questões de processamento como pela pequena quantidade de tempo e esforço na criação de NPCs.

Entretanto, após o lançamento do influente DOOM, o mercado foi inundado por outros títulos de FPS que tentavam replicar o sucesso do seu predecessor. Alguns desenvolvedores perceberam que a IA desempenhava um fator importante a fim de criar experiências de jogo mais longas e criar desafios mais escaláveis. Havia a preocupação em oferecer ao jogador um sentimento de desafio a fim de mantê-lo engajado, apesar do desafio ter efeito paradoxal, ou seja, se o jogador tivesse uma sensação que o desafio seria algo impossível, ele se sentiria injustiçado. Esse conceito é conhecido como teoria do fluxo cognitivo (do inglês *Cognitive Flow Theory*) (SNYDER, 2005).

Com essa perspectiva se tornava importante que os NPCs simulassem o comportamento de um humano, oferecendo um desafio justo ao jogador. O aprimoramento de personagens mais inteligentes, capazes de desafiar os jogadores de maneira coerente, tornou-se um diferencial e as soluções para desenvolver esse tipo de estratégia seguem uma abordagem arquitetural em camadas (LUCAS, 2012) o que permite definir componentes de software e atribuir funcionalidades a eles, facilitando a manutenção.

Cada solução adota um modelo específico de camadas nas quais aquelas dos níveis inferiores (com menor nível de abstração) lidam com as tarefas elementares, tais como determinar o melhor trajeto para alcançar o alvo ou executar uma sequência de animação de personagens.

As camadas de níveis superiores (com maior nível de abstração) são responsáveis pela realização do processo de tomada de decisão do personagem, tal como a definição das ações que devem ser executadas a cada momento. Essas ações devem manter uma conformidade com a estratégia escolhida, por exemplo, o agente deve patrulhar uma área determinada, atacando um oponente que surgir no seu campo de visão ou explorar o terreno em busca de adversários (GRZYB, 2005).

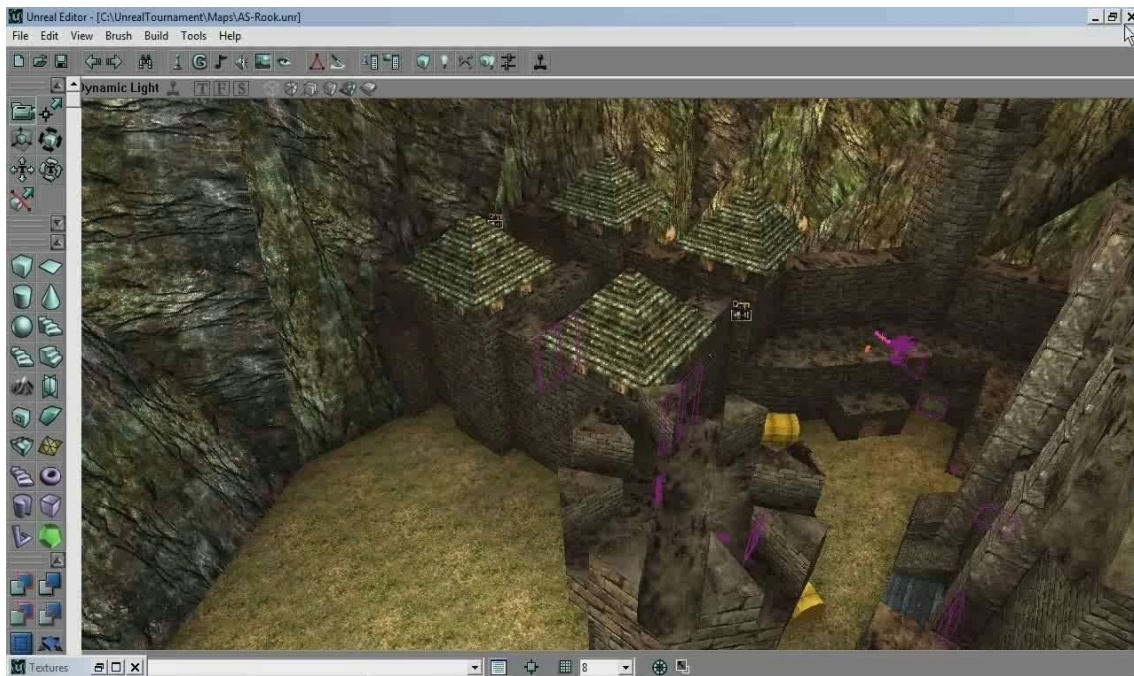
A comunidade acadêmica interessada no desenvolvimento de personagens inteligentes automatizados vem adotando alguns jogos como *benchmarks* a fim de avaliar o desempenho desses agentes. Os jogos de FPS possuem destaque nesse contexto, alguns bons exemplos sendo Half-Life 1, Half-Life 2, Counter-Strike, Doom III e Quake IV.

Half-Life 1, lançado em 1998, apresentava forte inovação quanto ao comportamento dos seus personagens, diferentemente dos FPS anteriores (MORA, 2010). Por exemplo, os personagens automatizados, inimigos ou aliados, tinham reações únicas ao levar um tiro, ao escolher o momento de lançar uma granada e até mesmo um modelo de consciência parecido às ações de um jogador humano.

Unreal Tournament 2004 (UT2004) é outro jogo que adiciona algumas vantagens como *benchmark* para a comunidade acadêmica. Além de ser uma opção mais recente que o Half-Life 1, outra vantagem é sua grande capacidade de modificação e expansão facilitada (WRAY, 2004).

Este jogo foi desenvolvido utilizando a Unreal Engine 3, uma *engine* moderna que suporta a utilização de uma linguagem simples e poderosa chamada de UnrealScript. Os desenvolvedores do Unreal Tournament desenvolveram UnrealScript (SWEENEY, 1998), uma linguagem de scripts orientada a objetos, que fornece uma interface de modificação de alto nível para o servidor de UT2004. Esta se assemelha a linguagens de programação como Java e C, sendo usada para programar as mecânicas de jogo.

Isso permite a edição facilitada de diversos componentes do jogo. Além disso, é possível modificar ou gerar um mundo completamente novo usando seu editor de cenários (ver Figura 10), ao ponto de criar jogo derivado do jogo original que substitui virtualmente todos os recursos artísticos do jogo original.



**Figura 10 – Unreal Level Editor: o editor de níveis do Unreal Tournament.**

Sua capacidade de modificação instigou o *Institute at the Naval Postgraduate School* a utilizar o UT2004 para desenvolver o jogo *America's Army*. John Laird e seu grupo de pesquisa da Universidade de Michigan também migraram para UT2004 a fim de dar continuidade aos seus estudos sobre personagens inteligentes e experiência de jogo (MAGERKO, 2004) (SENTRY STUDIOS, 2003).

Existe uma forte comunidade que utiliza o Unreal Tournament 2004, em conjunto com o *middleware Pogamut*, para o desenvolvimento de personagens inteligentes. A combinação das diversas ferramentas de edição e modificações do UT2004, além da capacidade de análise e desenvolvimento de personagens do Pogamut, serviu como base para pesquisas com jogos e IA. O uso do *middleware* tem referência em mais de 100 artigos publicados por 23 grupos (GEMROT, 2009).

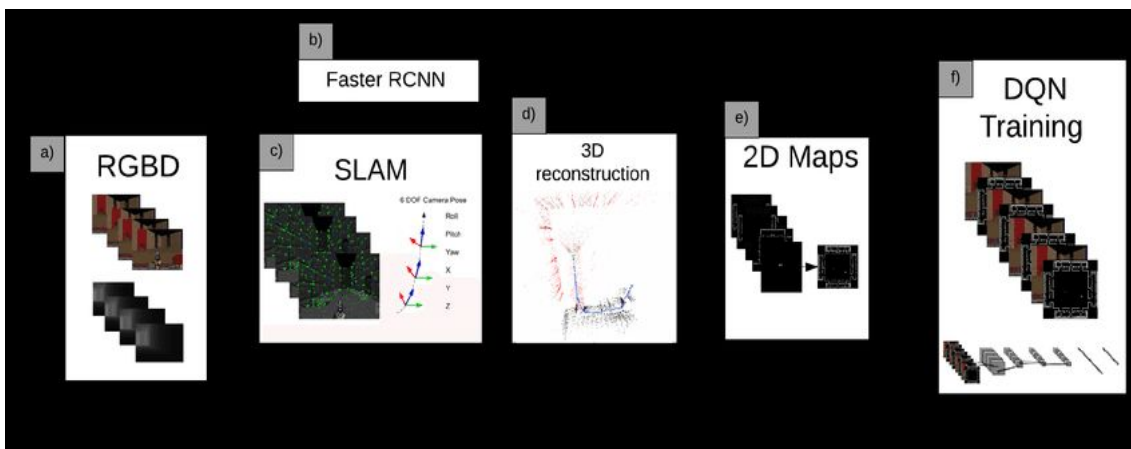
O objetivo do Pogamut é ser uma plataforma para desenvolvimento de personagens inteligentes em ambientes virtuais, gerar abstração de detalhes de implementação e permitir que os desenvolvedores se concentrem no desenvolvimento dos algoritmos referentes ao processo de tomada de decisão.

Existem publicações recentes utilizando essa plataforma como a *Hierarchical First Person Shooter Bot Using Multiple Sarsa( $\lambda$ ) Reinforcement Learners* (GLAVIN, 2018). Entretanto, recentemente a comunidade de desenvolvimento de agentes inteligentes tem voltado a sua atenção para outra ferramenta bastante interessante

conhecido com VizDOOM. Ferramenta essa, presente em mais de 150 publicações (VizDOOM, 2019).

Atualmente, a comunidade é muito ativa em pesquisas sobre "DOOM" por envolver técnicas como *deep reinforcement learning* e *visual reinforcement learning*. A utilização dessas técnicas permite o desenvolvimento de *bots* que jogam DOOM usando apenas as imagens da tela do jogador como informação de entrada para o processo de tomada de decisão.

O VizDOOM trabalha apenas com informações visuais “cruas”, o que pode ajudar os pesquisadores no fornecimento de informações nativas da plataforma para a IA (KEMPKA, 2016).



**Figura 11 – Processamento visual VizDOOM**

No artigo *Training an Agent for FPS DOOM Game using Visual Reinforcement Learning and VizDOOM* é exemplificado como esse processamento visual (ver Figura 11) é realizado (ADIL, 2017). Jogos de FPS vistos até o momento, como o Unreal Tournament 2004, Half-Life 1, Half-Life 2, Counter-Strike, DOOM III e Quake IV, estudam agentes com um maior número de informações tais como posições das paredes, inimigos, locais de itens, etc., que são inacessíveis para jogadores humanos que se guiam apenas pelas imagens na tela.

### 3.2 Aplicação de uma plataforma de desenvolvimento de personagens inteligentes para uma ferramenta de análises psicofisiológicas

A partir das plataformas analisadas, podemos constatar duas possíveis

alternativas pela solidez das suas comunidades e publicações, que são o UT2004 com o Pogamut (GEMROT, 2009) com 128 citações, e o DOOM com VizDOOM (KEMPKA, 2016) com 188 citações.



**Figura 12 – Cena do jogo DOOM (acima) e do jogo UT2004 (abaixo).**

Na Figura 12 é possível comparar os estilos gráficos dos dois jogos. O UT2004 possui um estilo gráfico mais realista e contém mais detalhes que o DOOM. Considerando que experimentos em Neurociência necessitam se aproximar de experiências mais realistas, o UT2004 pode ser visto como a melhor opção nesse

aspecto.

A API do VizDOOM foi implementada em linguagem C++ com suporte a *bindings* para Python e Java, já o Pogamut é implementado em Java. Ambas as soluções possuem editores de mapas e permitem customizações nos projetos com códigos “parcialmente abertos” ou totalmente abertos como no caso do DOOM.

**Tabela 2 DOOM+ VizDOOM vs UT2004 + Pogamut**

	<b>DOOM + VizDOOM</b>	<b>UT2004 + Pogamut</b>
Quantidade de publicação	188 citações	128 citações
Qualidade Gráfica	Gráficos simples	gráficos mais elaborados, efeitos de luz/sombra
Capacidade de Edição	Edição completa inclusive motor gráfico, licença open-source	Edição limitada a mods e código, com licença proprietária
API	API com suporte a múltiplas plataformas	API com suporte a JAVA

Nos aspectos abordados até então o DOOM permite uma capacidade de edição maior por ser totalmente aberto, no entanto, a implementação de *bots* é mais complexa que o UT2004 pois este último pode ser modificado utilizando a UnrealScript. Parte substancial do UT2004 é programada em linguagem de *script*, exceto o motor de renderização de gráficos, o motor de modelagem física e alguns componentes de comunicação com o sistema operacional como o acesso a rede ou o sistema de arquivos.

Por outro lado, a capacidade de edição do motor gráfico do DOOM permite (com alguma complexidade técnica) ter um controle fino da exibição de estímulo, algo impossível de fazer com o UT2004, já que os componentes internos da *engine* possuem código fechado.

Existe uma literatura razoável que disponibiliza documentação quanto à capacidade de edição do UT2004, indicando como construir *mods* e cenários com qualidade. Tutoriais e fóruns na internet são facilmente acessíveis, além de livros como o "*Focus creating mods Unreal Tournament*", de Smith (2002).

A principal diferença entre as soluções consiste na abordagem de como a ferramenta funciona, com o VizDOOM criando bots que tentam simular a interação humana e tendo somente a tela como informação disponível sobre o jogo, por outro lado o Pogamut consegue obter informações internas do jogo a todo momento, bem como detectar mudanças nos seus valores.

No caso específico dos experimentos em Neurociências, a abordagem do Pogamut não se mostra somente como mais vantajosa, mas essencial por isso.

### 3.2.1 Detalhes do Pogamut

Compreender o processo de construção de personagens inteligentes utilizando o Pogamut é necessário para entender como este pode contribuir com a criação de uma ferramenta para análises psicofisiológicas. A construção de personagens inteligentes com o Pogamut segue normalmente cinco etapas de desenvolvimento:

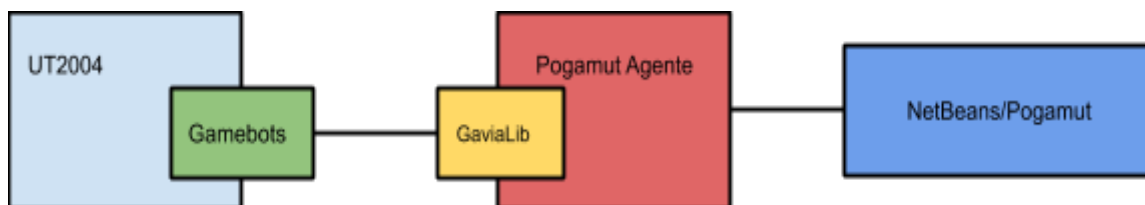
- Criação do personagem;
- Execução do personagem;
- Depuração do personagem;
- Ajuste dos parâmetros do personagem;
- Validação do personagem por meio de testes sucessivos.

Cada uma dessas etapas gera requisitos de software complexos que devem ser tratados por uma plataforma que auxilia o desenvolvimento, desse modo no Pogamut foi adotada uma arquitetura para cobrir os requisitos de software requeridos por cada uma dessas etapas, contando com os componentes necessários:

- API de *bindings* para o mundo virtual do jogo UT2004;
- Ambiente de desenvolvimento integrado (IDE) com suporte para depuração;
- Biblioteca nativa de código que garante ao agente (personagem do jogo) suporte sensorial e motor, algoritmos para planejamento de trajetórias (*pathfinding*), e suporte para comportamentos de tiro e manuseio de armas,

- Ligação ao POSH, que é um planejador reativo para controlar comportamento dos agentes, com suporte a um editor gráfico. Monitoramento da execução e simulação, incluindo suporte a execução distribuída em um terreno discretizado em formato de grade (*grid*).

Tecnicamente, a *suite* de ferramentas Pogamut integra cinco componentes principais: UT2004, GameBots2004, a biblioteca GaviaLib, o agente Pogamut, e o IDE. Estes componentes mencionados permitem ao usuário criar, depurar e avaliar de forma adequada os seus agentes.



**Figura 13 – Componentes da arquitetura do Pogamut**

Os componentes do Pogamut (ver Figura 13) podem ser utilizados de maneira independente e podem ser intercambiáveis, por exemplo, pode-se usar GameBots2004 independentemente, bem como uma parte da biblioteca GaviaLib ou o IDE.

Teoricamente, é possível utilizar o Pogamut com outros jogos, além do UT2004 contudo para que o Pogamut funcione com outro jogo ele depende da implementação de uma interface entre o jogo em questão e o GaviaLib, papel desempenhado no UT2004 pelo GameBots2004 isso se deve à necessidade de utilização da API do Pogamut (Bida, 2012).

### 3.2.2 Gamebots

O projeto Gamebots tem, desde o seu início, a ambição de ser um ambiente rico para o desenvolvimento de personagens inteligentes. O projeto foi iniciado por Andrew N. Marshal e Gal Kaminka no Instituto de Ciências da Informação da Universidade do Sul da Califórnia (BIDA, 2012).

Explorando o fato do Unreal Tournament 2004 ser um jogo multijogador com arquitetura cliente-servidor, o Gamebots atua como uma interface entre o servidor e os clientes e trata-se basicamente de uma modificação (*mod*) escrita em UnrealScript. O

projeto começou utilizando a versão do Unreal Tournament 2000, mas foi atualizado para a versão 2004 posteriormente.

Essa modificação permite obter informações sobre o ambiente do jogo e controlar avatares presentes no mesmo por meio de protocolo TCP/IP. As informações sensoriais tais como a localização e a direção dos personagens, tanto para aqueles controlados pelo jogador quanto para os que não, são enviadas para o servidor que centraliza todas essas informações do mundo do jogo.

O servidor envia as informações para as instâncias clientes, sendo a comunicação realizada através de mensagens síncronas e mensagens assíncronas. Os clientes, por sua vez, enviam as mensagens contendo os comandos de ação que devem ser executados pelos personagens (MARSHALL, 2002). Todos os comandos e mensagens compartilham um formato de texto similar a um JSON, sendo a primeira palavra o tipo da mensagem e, em seguida, os dados da mensagem são determinados como identificadores e valores.

As mensagens síncronas tem um escopo determinado pelas instruções BEG (início) e END (fim), entre essas instruções, o lote síncrono contém informações sobre o perímetro do *bot* e sobre o estado do personagem (saúde, armadura, arma atual, etc.).

As mensagens assíncronas podem ser enviadas a qualquer momento, geralmente referindo-se a algum evento especial que aconteceu no ambiente do jogo, como correr em direção da parede ou ouvir algum ruído, por outro lado, também pode ser uma resposta a alguma solicitação de comando.

O Gamebots aceita dois tipos de comando: controle do servidor e controle de um adversário automatizado. O primeiro tipo é utilizado para controlar a mecânica do jogo, obtendo informação sobre o estado atual do mundo de jogo e sobre a existência de personagens. O segundo tipo é usado para controlar e configurar os NPCs e quando possível, todos os comandos são executados em paralelo.

O projeto Gamebots foi continuado pelos pesquisadores Joe Manojlovich, Tim Garwood e Jessica Bayliss (BIDA, 2012). Eles coordenaram o projeto para levá-lo a uma nova versão do jogo Unreal Tournament 2004. Posteriormente, foi adicionada uma API de alto nível, chamada JavaBot, com base no protocolo Gamebots (MARSHALL, 2002).

A fim de facilitar o desenvolvimento, foi criado um *fork* do Gamebots, chamado

*Pogamut*, englobando todo o projeto do Gamebots e o JavaBot, e adicionando suporte à depuração. *Pogamut* refatora grande parte do código do Gamebots, adicionando novas funcionalidades e corrigindo erros de prorrogação no protocolo de texto. Essas melhorias foram tão importantes que o *Pogamut* passou a ser o ramo principal desta solução, sendo adotado pela comunidade de UT.

### 3.2.3 API para o Mundo Virtual

É possível desenvolver personagens inteligentes apenas com a UnrealScript. No entanto, por questões educacionais, experimentais e de generalização, é mais vantajoso utilizar outra abordagem. A solução adotada pelo *Pogamut* para desenvolver personagens inteligentes consiste em usar uma biblioteca em linguagem Java, que é mais generalizável, promovendo uma ligação (*binding*) para UT2004.

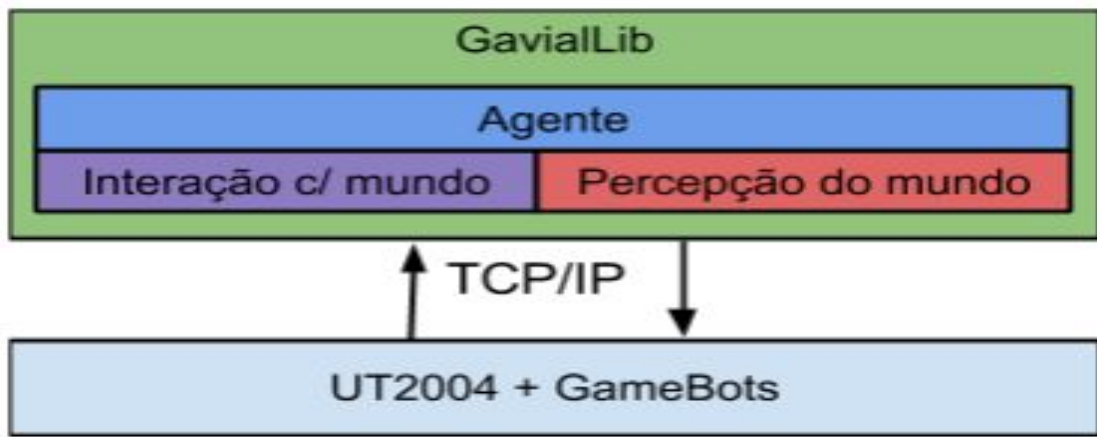
A API *GaviaLib* funciona como uma interface para acesso a uma vasta gama de informações dos elementos do mundo de jogo, facilitando tanto a interação com os objetos no ambiente virtual quanto o acesso às informações de eventos que ocorrem durante a partida. A biblioteca fornece as seguintes funcionalidades:

- Modelos de abstração de personagens, que contém um ciclo de vida e permite seu controle através das chamadas a interface JMX, tecnologia Java que fornece ferramentas para o monitoramento de aplicações;
- Métodos para o personagem poder acessar o mundo de jogo, manipular objetos, identificar eventos (por exemplo, a criação, destruição ou alteração de um objeto no cenário);
- Estrutura XML / XSLT para a definição de objetos e eventos do mundo.

Um detalhe importante é a resposta a eventos proporcionada pela API que utiliza *callback* para registrar eventos e retornar instâncias de objetos.

Arquiteturalmente existe a utilização de um padrão de projeto *observer* para a manipulação do agente dentro do sistema. No entanto, o *GaviaLib* propaga eventos de acordo com esta ontologia. Por exemplo, quando um evento é disparado no objeto da classe "maçã", é propagado também a "fruta" e o "item". Isso faz com que o modelo de evento de *GaviaLib* seja flexível visto que o usuário pode escutar em todos os eventos que são disparados em qualquer instância.

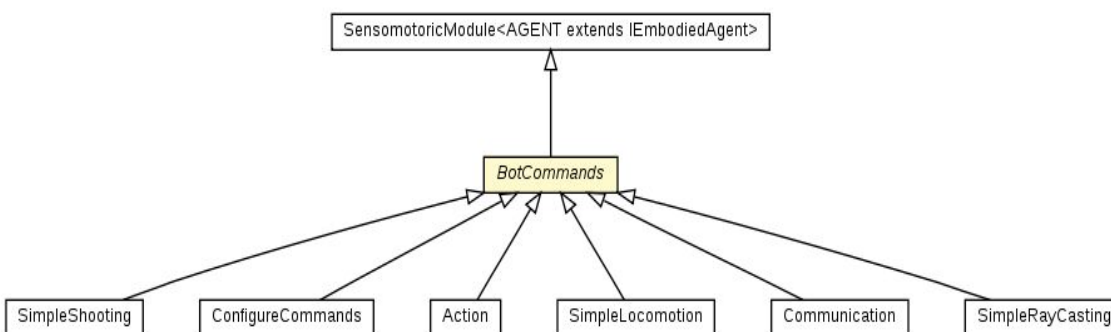
Este recurso não está disponível ao usar o GameBots2004 sozinho, esta aplicação também permite extensões personalizadas para o código existente e quando um designer cria uma nova classe "banana" que se estende de "fruta", todos os ouvintes (*observer*) existentes sobre a "fruta" serão automaticamente capazes de relatar eventos que acontecem em todas as bananas.



**Figura 14 - Camadas da biblioteca GaviaLib**

Como a comunicação entre a camada do GaviaLib e qualquer outra pode ser realizada por protocolo TCP/IP (ver Figura 14), é possível intercambiar as camadas vizinhas por outras que usem o mesmo protocolo.

Para representar o modelo de personagem, a API possui um conjunto de classes e interfaces Java com características específicas tais como a capacidade sensório motor no ambiente de realidade virtual (presentes no diagrama da Figura 15), sistema de navegação e classes auxiliares que fornecem informações sobre as regras do jogo.



**Figura 15 - Herança sensório motor do GaviaLib**

É possível desenvolver com o Pogamut utilizando qualquer IDE, apesar disso, a ferramenta possui uma integração bastante madura com o NetBeans pois a comunidade

desenvolveu um *plug-in* que permite a IDE controlar a execução de personagens por meio de mensagens (sobre protocolo JMX), possibilitando testes unitários.

### 3.3 Conclusões sobre a aplicação de soluções em estudos cognitivos

Os obstáculos encontrados nos trabalhos anteriores referentes aos estudos cognitivos e os jogos digitais apresentam princípios similares aos enfrentados pela área de estudo de personagens inteligentes, portanto, uma análise de como a comunidade acadêmica de IA lida com esses problemas, bem como as ferramentas que foram desenvolvidas para isso, nos possibilita gerar alguns entendimentos.

O desenvolvimento de personagens inteligentes e sua integração com jogos de tiro em primeira pessoa possui um corpo técnico bastante maduro, ademais a solução desenvolvida com Unreal Tournament 2004 e Pogamut permite resolver vários problemas identificados nas publicações apresentadas neste trabalho. Além disso, a arquitetura da solução é modular e genérica, e possui uma documentação de ampla cobertura.

Apesar do VizDOOM utilizar o DOOM quem tem a vantagem de ter código totalmente aberto não oferece acesso às variáveis internas do jogo, sendo possível apenas analisar o fluxo de vídeo, por isso a sua utilização foi descartada como infraestrutura para a solução a ser adotada no presente trabalho.

Logo, chegou-se à conclusão que é possível usar o Pogamut como alternativa para as pesquisas em psicofisiologia. A vantagem mais óbvia dessa alternativa é evitar o retrabalho de criar funcionalidades já desenvolvidas pela comunidade científica. De maneira mais sutil, essa abordagem garante certa segurança de uso de ferramentas já testadas por uma comunidade relevante, oferecendo solidez ao *benchmark* e o respaldo científico de uma suíte contando com uma série de publicações científicas.

Como última consideração é importante indicar que, apesar da aproximação dos jogos digitais com a neurociência cognitiva já possuir um relativo corpo, a associação dessas áreas de estudo no contexto do desenvolvimento de personagens inteligentes é um caminho de pesquisa inédito que se mostra bastante promissor.

## Capítulo 4

### *Neuro Game Logger*: uma proposta de ferramenta para análise cognitiva com jogos

Nas seções anteriores apresentou-se o potencial da combinação da neurociência cognitiva com os jogos digitais a fim de fornecer uma visão coerente dos processos psicofisiológicos. Percebeu-se que é possível facilitar a integração das áreas distintas, utilizando a experiência dos experimentos desenvolvidos com as ferramentas desenvolvidas pela comunidade científica de IA.

Pode-se assim propor uma solução de software usando esse conhecimento, responsável por gerar um *log* dos eventos de jogo para ser sincronizado com o registro de gráficos do EEG, assim, conectando eventos do jogo com respostas comportamentais humanas de maneira mais fácil e com alta resolução temporal.

Inicialmente, neste capítulo do trabalho será apresentada a metodologia de construção do *Neuro Game Logger*, uma proposta de ferramenta para análise cognitiva com jogos. Em seguida será apresentado o resultado do levantamento de requisitos em forma de uma análise e uma reflexão quanto à viabilidade de utilização do UT2004 e Pogamut como infraestrutura para a solução de software proposta.

A ideia é que além da capacidade de explorar novas abordagens experimentais, essa solução de software permitirá eliminar a tarefa de marcação manual de métricas de jogo, já que isso passará a ser feito automaticamente pelo sistema. Em um segundo momento será apresentado o projeto de software de forma sintetizada, bem como, detalhes de implementação.

Além disso, será validada a solução proposta por meio de um experimento que servirá como prova de conceito, dessa forma será possível detalhar os pontos positivos na utilização da ferramenta na atividade de pesquisa, mas também suas limitações. Além do conhecimento que a experimentação da proposta gerará, será possível fazer uma comparação com métodos utilizados em outras pesquisas utilizando jogos.

## 4.1 Metodologia

O desenvolvimento da ferramenta se deu inicialmente com a investigação e pesquisa na área e experiências relatadas anteriormente nesse documento que permitem identificar os requisitos de software necessários para o desenvolvimento da solução proposta.

Outras informações foram adquiridas por meio de entrevista para se avaliar requisitos mais granulares de como a ferramenta deveria se comportar em experimentos. Essas informações foram cedidas pelo corpo técnico do Laboratório de Sono, Sonhos e Memórias do Instituto do Cérebro da UFRN.

Munido o projeto de toda informação necessária, foi possível criar um documento de levantamento e análise de requisitos, sendo possível a definição das funcionalidades do projeto seguindo os padrões disponíveis na literatura.

Foi avaliada durante a análise, a possibilidade de utilizar como base da solução um arcabouço de software que evitasse o esforço de implementação de funcionalidades já existentes em outras plataformas, mantendo assim o foco na contribuição principal deste trabalho, obviamente em caso de não existir prejuízo para a análise.

A validação da ferramenta ocorreu por meio de experimentos que tinham como objetivo provar o conceito de uso da ferramenta e demonstrar suas capacidades e limitações. Essa prova de conceito segue os testes de benchmark da área e foi acompanhada pelo corpo técnico do Instituto do Cérebro.

Após a etapa de validação, foram apontadas características que, em um novo ciclo de implementação, agregaram novas capacidades e sanaram as pendências identificadas. Os registros gerados nessa prova de conceito foram analisados pelo corpo técnico do laboratório, gerando diversos gráficos e imagens que serão apresentadas e discutidas neste trabalho.

## 4.2 Análise de Requisitos

Requisito funcional define uma função de um sistema de software ou seu componente (SOMMERVILLE, 2007). Os requisitos funcionais podem ser cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades

específicas que definem o que um sistema será capaz de idealmente realizar e para o presente projeto, foram definidos alguns requisitos.

A primeira necessidade evidenciada é a capacidade de detectar a interação dentro de um ambiente virtual, influenciando diretamente os eventos. Para identificar os tipos de eventos que ocorrem no ambiente do jogo e sua ação no cérebro do jogador, seguem alguns exemplos de eventos que podem ser considerados como significativos em um ambiente de jogo de tiro em primeira pessoa:

- Dano no personagem do jogador;
- Pontuação do jogador;
- Coleta de item;
- Dano em outro personagem;
- Morte do personagem do jogador;

A ocorrência de eventos deve ser notificada para o equipamento de EEG através do envio de um marcador na porta paralela do computador e, simultaneamente, o registro desse evento em um arquivo de log, concomitantemente as informações dos arquivos de *log* precisam conter uma descrição do evento e a marca temporal de data e hora (*timestamp*) da sua ocorrência.

A ferramenta necessita marcadores psicofisiológicos precisos durante a fase de aquisição de dados, portanto, um sistema de registros que opere na velocidade de tempo de execução de um jogo e envie os eventos relacionados com a mesma resolução temporal do EEG é o ideal para os pesquisadores que estudam detalhadamente a experiência dos jogadores.

O EEG possui uma resolução temporal na ordem de milissegundo (MULERT & LEMIEUX, 2010) e com uma precisão fidedigna, sendo assim de alto desempenho e alta precisão, o que se traduz como uma característica importante para pesquisas psicofisiológicas dado que as mudanças do estado neural ocorrem rapidamente, isso é especialmente importante na detecção de eventos.

É importante deixar claro a relação entre sistema de alta precisão temporal (ou tempo real) e de alto desempenho. Para um sistema de tempo real a previsibilidade é uma característica mais importante que o desempenho em si, ou seja, ele pode aceitar uma certa latência desde que ela seja mensurável e razoavelmente constante.

O cenário ideal seria a ferramenta conseguir detectar eventos e marcá-los com o mesmo desempenho e precisão temporal que o equipamento de EEG suporta. Entretanto, manter o alto desempenho associado a uma precisão em milissegundos é uma tarefa que se apresenta como de grande complexidade computacional, não sendo o foco deste trabalho.

Dessa forma a proposta é que o sistema possa apresentar uma latência de registro e notificação (ao equipamento de EEG) de eventos em relação à ocorrência do evento, dispondo assim de uma característica mais próxima a um sistema de tempo real.

Outro ponto importante é a avaliação de quanto o sistema possui característica de tempo real, ou seja, que exige o controle de hardware de baixo nível similar ao que encontramos em sistemas embarcados (BURNS, 2009).

No mundo dos jogos, os primeiros videogames como o Atari 2600 tinham o tempo real como característica forte pela natureza de como o hardware temporizado gerava seus gráficos. Entretanto, em jogos comerciais modernos, computadores e consoles atuais, essa característica foi perdida (MENYCHTAS, 2009).

Essas particularidades conferem a natureza de tempo real à ferramenta, na qual eventuais variações de atraso são toleráveis, transformando o registro e a notificação do evento em um evento probabilístico.

O ponto chave é atingir uma baixa variância da latência para ter uma maior probabilidade de registrar o evento em suas sucessivas repetições, permitindo assim, por meio de análise estatística, encontrar o início dos eventos psicofisiológicos através do registro de EEG.

Tendo em vista todas essas questões, se faz necessário não só que a implementação considere esses aspectos, mas também que seja realizada uma bateria de testes para avaliar estatisticamente a precisão da ferramenta, tal avaliação necessária para a ferramenta é de conhecimento prático de laboratórios de pesquisa em neurociências, como o laboratório de Sono, Sonhos e Memórias do Instituto do cérebro.

Baseando-se em trabalhos anteriores, acreditamos que atrasos de um tempo máximo de 110 milissegundos sejam aceitáveis no envio de marcadores na porta paralela (SCOTT, 2015). Dessa forma é determinante para o bom funcionamento do sistema, que o tempo de transmissão esteja dentro do esperado.

Um segundo requisito importante é o registro síncrono de cada passo do personagem do jogador no mapa e o estado de seus atributos, sua posição (nos eixos x, y, z), para onde a sua câmera está direcionada, sua velocidade, entre outras informações. Dessa forma seria possível remontar a experiência de jogo por completo com dados atômicos e analisáveis.

Um terceiro requisito é a capacidade que o sistema funcione em um ambiente com múltiplos jogadores, isso é importante porque existem muitos experimentos que utilizam interações entre vários participantes, normalmente para medir os impactos sociais ou ecológicos nos processos psicofisiológicos. Entretanto nessa configuração os eventos entre jogadores devem ser possíveis de ser sincronizados em um mesmo quadro temporal para serem avaliados.

As regras de negócio refletem políticas para satisfazer clientes, fazer bom uso dos recursos, e obedecer às leis ou convenções gerais do negócio (SOMMERVILLE, 2007). Uma regra de negócio percebida é a necessidade de fácil utilização da solução, uma vez que, na maioria das vezes, o neurocientista não é um técnico da área de tecnologia.

#### 4.2.1 Reflexões sobre a Análise para uso do Unreal Tournament 2004

Os requisitos analisados apontam a possibilidade de utilizar o Unreal Tournament 2004 como plataforma para experimentação em conjunto com o Pogamut 3, que oferecem uma base interessante para a construção de um ferramenta para análises psicofisiológicas.

Tanto a capacidade de gerar registros quanto a de detectar eventos de maneira eficiente são problemas normalmente enfrentados no desenvolvimento de personagens inteligentes dos jogos digitais. Neste sentido, os pesquisadores do Pogamut despenderam muito trabalho para desenvolver uma infraestrutura de gravação de registros e contabilização automática de eventos.

A Robusta API em Java do Pogamut 3 permite acesso a informações muito úteis, lista de mensagens e comandos suportados pelos pacotes utilizados e o pacote padrão para Unreal Tournament 2004 é GameBots2004 que garante um bom número de sensores para os eventos do Pogamut é:

- **Item**: um item modificou de estado;
- **IncomingProjectile**: projétil atirado em direção a um alvo;
- **Jump**: algum jogador pulou;
- **Landed**: algum jogador escalou;
- **ShootingStarted**: algum jogador começou a atirar;
- **ShootingStopped**: algum jogador parou de atirar (arma automática);
- **Spawn**: algum jogador renasceu para voltar ao jogo;
- **WallCollision**: algum jogador colidiu com a parede;
- **HearNoise**: algum jogador escutou um barulho;
- **ItemPickedUp**: um item foi pego;
- **PlayerJoinsGame**: um jogador entrou na partida;
- **PlayerLeft**: um jogador saiu da partida;
- **PlayerScore**: um jogador pontuou;
- **PlayerKilled**: um jogador foi morto;
- **PlayerDamaged**: um jogador sofreu danos.

Além da detecção desses eventos é possível identificar também a cada momento, de maneira síncrona, as seguintes informações dos jogadores:

- **ID**: Identificador exclusivo do jogador;
- **Name** (String): Nome do personagem do jogador;
- **Ação** (String): Ação que o personagem do jogador está realizando;
- **Visível** (Boolean): Indica se o jogador está no campo de visão de outro jogador;
- **Rotation** (Rotação): Direção que o jogador está apontando em termos absolutos;
- **Localização** (Location): Localização absoluta (eixos x, y e z) do avatar dentro do mapa;
- **Velocity** (Velocidade): Velocidade absoluta do avatar como um vetor de movimento;
- **Team** (Int): Equipe que o jogador participa. O código 255 significa nenhuma equipe, 0 para vermelho, 1 para azul, 2 para verde, 3 para ouro;
- **Arma** (String): Classe da arma que o jogador está usando;

- **Agachado** (Boolean): Verdadeiro se o *bot* está agachado;
- **Firing** (Int): 0 significa que não está disparando; 1 significa disparar no modo primário; 2 significa disparar no modo secundário (alto disparo);
- **Anim** (String): Para UE2 significa a atual animação do jogador.

Ademais, a tarefa de sincronização pode se tornar muito mais precisa no contexto de um sistema de registro automatizado, enquanto que o registro manual pode introduzir erros no registro.

Esse acesso é fornecido a partir do servidor, logo é possível ter acesso às informações de todos na partida e refere-se a uma característica muito importante em experimento com múltiplos voluntários, permitindo sincronização de eventos em múltiplos setups de experimento.

Outra questão é a possibilidade de editar qualquer mapa do UT2004 usando o editor *UnrealEd*, adicionando e removendo itens no mapa, modificando a arquitetura, entre outras possibilidades, para ajustar o jogo para atingir diferentes objetivos do experimento, inclusive existe uma série de mapas concebidos para competições.

Uma funcionalidade adicional oriunda da utilização do Pogamut é a capacidade de criar personagens controlados por inteligência artificial (bots), além de poder utilizar implementações de bots existentes desenvolvidos pela comunidade, é possível ainda implementar bots que repliquem comportamentos específicos de forma convincente (KARPOV, 2013).

Por exemplo, um experimento desenhado para se analisar o comportamental de como funciona a mente humana em um papel de predador, pode se beneficiar de um *bot* que tem um comportamento de presa, sendo alvo para voluntários do experimento.

Usando tal sistema é possível, por exemplo, modelar um experimento gerador de certos estímulos que possam ser utilizados para mensurar quantitativamente a emoção e excitação provocada por um determinado evento de jogo.

Apesar do Pogamut possuir suporte a captura de uma diversidade de eventos do UT2004, é necessário implementar um *KeyLogger* e *MouseLogger* auxiliares a fim de gerar um log auxiliar, tendo registro de ações que não necessariamente se traduzem em eventos no jogo ou que não sejam capturados pela API do Pogamut.

## 4.3 Descrição do projeto arquitetural e desenvolvimento

A descrição da arquitetura do software envolve os princípios e as práticas de modelagem e representação de arquiteturas, usando mecanismos como: linguagens de descrição de arquitetura, pontos de vista de arquitetura e estruturas de arquitetura (CLEMENTS, 2011).

O entendimento alcançado permitiu a esse trabalho definir uma proposta de arquitetura para essa solução. O momento inicial desta seção descreve uma síntese da arquitetura com foco na visão estrutural do software e visão de ação do usuário da ferramenta.

Existe uma representação de alguns fragmentos do software descritos utilizando a linguagem UML a Unified Modeling Language que é uma linguagem de modelagem de desenvolvimento e propósito geral no campo da engenharia de software, que se destina a fornecer um padrão de design de um sistema (BOOCH, 2005). Essa mesma linguagem foi utilizada na descrição de casos de uso e as suas funcionalidades foram realizadas e se encontram nos anexos deste documento.

Nesta seção também serão relatados alguns detalhes do desenvolvimento, como as ferramentas utilizadas, tecnologias e a experiência de implementação.

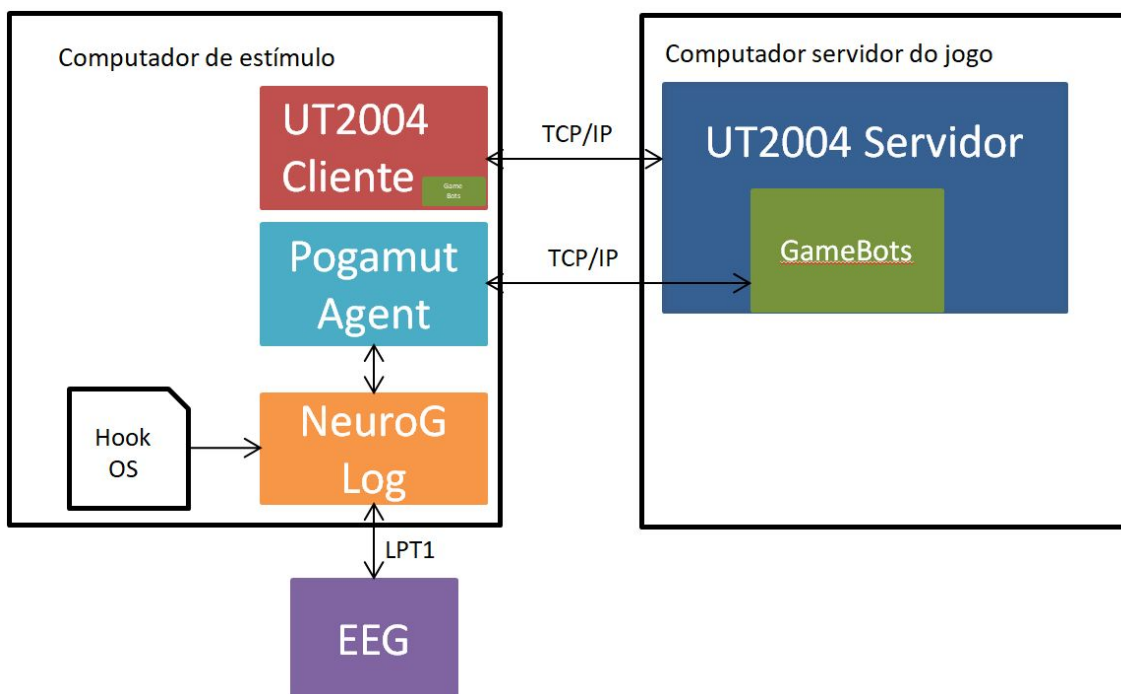
Pelo fato da solução utilizar componentes do Pogamut e Gamebots, sendo o primeiro uma API em Java utilizada para desenvolvimento e o outro um mod do UT2004 que envia diversas informações sobre a partida por sockets TCP/IP, suas arquiteturas serão tratadas em detalhes.

### 4.3.1 Visão da Estrutura

Essa seção trata do aspecto estrutural tanto do ponto de vista do sistema quanto das classes. Os aspectos estáticos desse sistema de software abrangem a existência e a colocação de itens como classes, interfaces e componentes, bem como as escolhas de arquitetura de software.

A utilização do Pogamut, como infraestrutura para aquisição de dados, obriga a utilização do UT2004 em seu modo multijogador. Nesse modo o jogo adota uma arquitetura cliente servidor, permitindo que vários computadores com instalação do cliente de jogo interajam entre si.

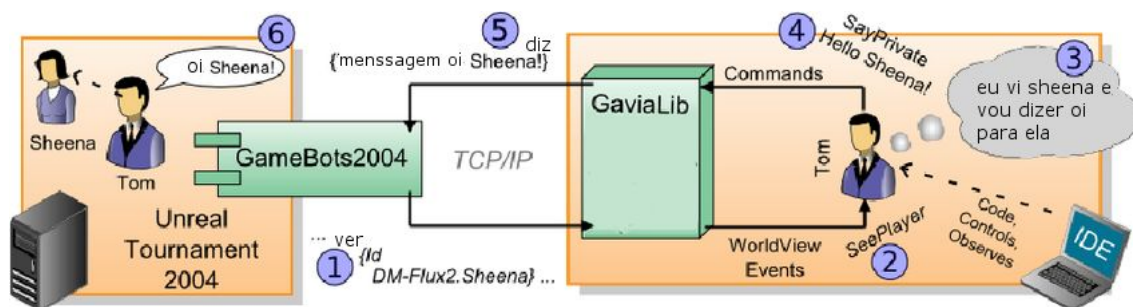
Mesmo em situações nas quais só exista um único jogador, se faz necessário executar o software do servidor (mesmo que seja na mesma máquina) para executar o jogo. O servidor do UT2004 deve ser carregado com o *mod* do Gamebots, o mesmo *mod* que deve estar instalado nos clientes, o cenário que descreve a interação dos componentes do projeto pode ser encontrado na Figura 16.



**Figura 16 - Arquitetura proposta para o *Neuro Game Logger*.**

O UT2004 Cliente é a instância do cliente que irá se conectar com o servidor, que está sendo executado no computador de estímulo. A imagem cliente do servidor está presente em computadores diferentes, mas nada impede que os softwares do cliente e do servidor sejam executados na mesma máquina pois a comunicação entre esses softwares se dá por protocolo TCP/IP.

De uma forma muito parecida o Gamebots executados no servidor também se comunicam com o Pogamut Agent, entretanto o Gamebots troca mensagens com o Pogamut Agent que é responsável por processar essas mensagens (uma espécie de analisador léxico e sintático), fazendo uma espécie de tradução dos dados do mundo de jogo para a API do Pogamut.



**Figura 17 - Pogamut 3 – Humanos Virtuais Simplificados (GEMROT, 2009)**

O software que tem acesso a API do Pogamut pode, dessa forma, alterar o modelo de dados fazendo com que o Agent envie mensagens para o Gamebots atualizando o estado do jogo para uma nova versão, permitindo interação com o jogo, essa relação é descrita na Figura 17, que acompanha o artigo *Pogamut 3 – Humanos virtuais simplificados* (do original, *Virtual Humans Made Simple*) (GEMROT, 2009).

#### 4.4 Neuro Game Logger

O Neuro Game Logger é a contribuição direta deste trabalho, essencialmente sendo a implementação de um software em Java que utiliza a API do Pogamut de maneira similar ao que os pesquisadores da área de agentes inteligentes fazem para desenvolver seus bots, a fim de testar algoritmos de inteligência artificial. Contudo, não sendo aqui o interesse criar outro personagem jogador, a solução é carregada no servidor por meio de um observador (*observer*).

Esse observer é um tipo de personagem que é intangível e inerte no mundo de jogo, seu objetivo sendo o de simplesmente capturar os inputs e estado das variáveis de outro personagem, assim toda a interação e eventos que ocorrem com o jogador no cliente do UT2004 podem ser capturados e salvos em um log.

No diagrama de classes da Figura 18 estão descritas as classes principais do projeto, mostrando como se dá sua interação.

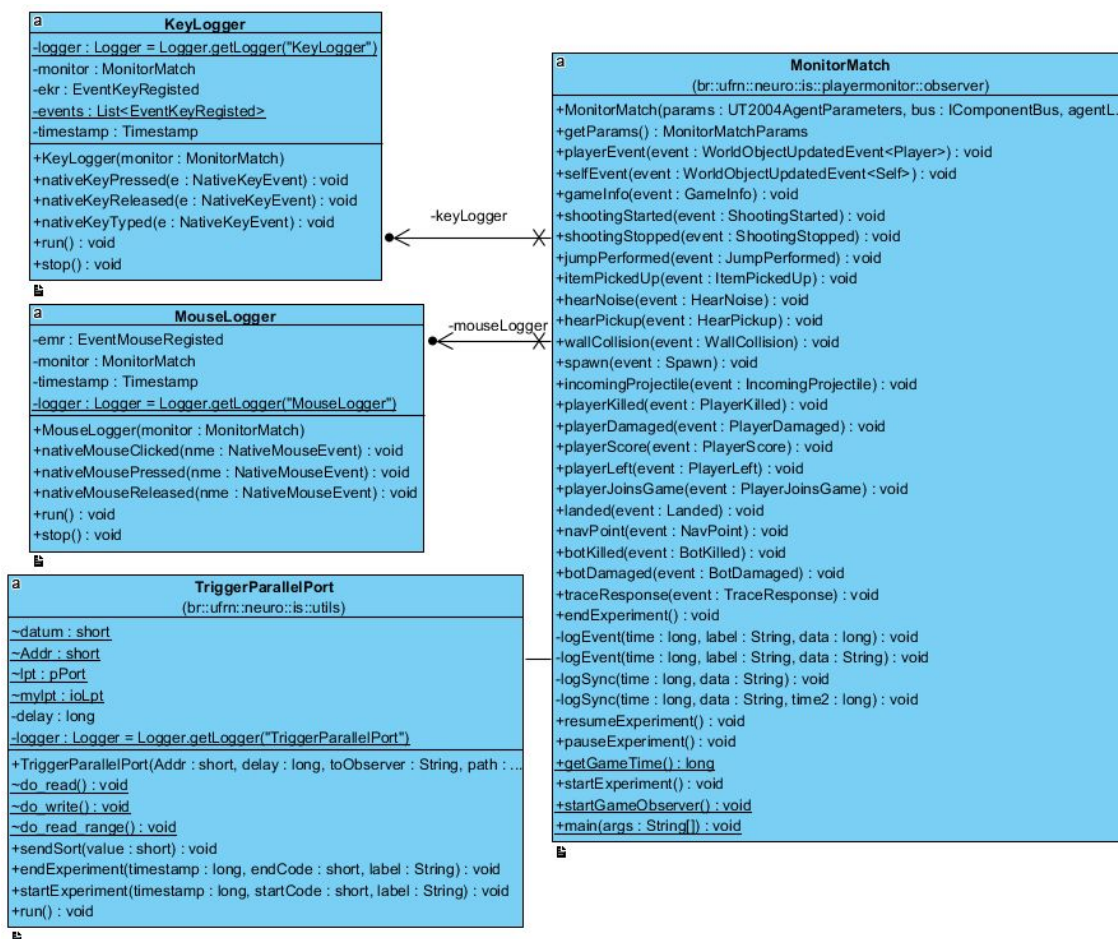


Figura 18 - Fragmento principal diagrama de Classe

A classe *MonitorMatch* é o coração da aplicação, inspirada na classe *Observer* do Pogamut. Quando a aplicação inicia, essa classe carrega no método principal todas as configurações tais como o IP do servidor, quais eventos devem ser registrados, quais eventos devem enviar marcadores, qual o marcador enviado para cada evento, além de outras configurações efetuadas por arquivos de configuração.

Ao ser instanciado, o objeto da classe instância três outros objetos e uma *thread* de execução para cada um desses objetos. O *MouseLogger* captura as informações vindas do mouse, o *KeyLogger* captura as entradas do teclado e o *TriggerParallelPort* é responsável pelo envio de marcadores pela porta paralela.

O *MonitorMatch* possui métodos que são *listeners* para os diversos eventos cobertos pela API do Pogamut. Quando eles ocorrem, o escopo de código desses métodos é chamado e os dados do evento que o disparou são passados como parâmetro.

Assim é possível gerar o *log* com essas informações bem como enviar marcadores para a porta paralela.

Toda a execução do *MonitorMatch*, *MouseLogger*, *KeyLogger* e *TriggerParallelPort* é monitorada, gerando *logs* específicos para cada um dos fluxos. Dessa forma é possível detectar problemas de execução ou ainda gerar algum dado útil para experimentos.

#### 4.4.1 MouseLogger e KeyLogger

As classes *MouseLogger* e *KeyLogger* são implementações utilizando a biblioteca *JNativeHook*, distribuída sobre as licenças GNU General Public and GNU Lesser General Public, essa biblioteca desenvolvida por Alex Barker fornece *listener* para teclado e mouse, permitindo escutar toda a interação do hardware (BARKER, 2018).

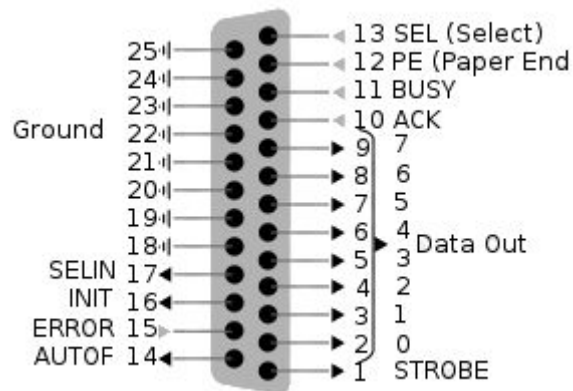
*JNativeHook* é um *wrapper*, ou seja, uma sub-rotina em uma biblioteca de software cuja finalidade principal é chamar um código nativo em C através da interface Java, para criar hook. Segundo a Microsoft, um hook é um ponto no mecanismo de tratamento de mensagens do sistema operacional, em que um aplicativo pode instalar uma sub-rotina para monitorar o tráfego de mensagens no sistema e processar certos tipos de mensagens antes que elas atinjam o procedimento de janela de destino (MICROSOFT, 2018).

Os eventos capturados com essa biblioteca são os seguintes:

- Key Press Events (pressionar uma tecla do teclado);
- Key Release Events (soltar uma tecla do teclado);
- Key Typed Events (digitar uma tecla do teclado);
- Mouse Down Events (pressionar o botão do mouse);
- Mouse Up Events (soltar o botão do mouse);
- Mouse Click Events (clique do mouse);
- Mouse Move Events (movimento do mouse);
- Mouse Drag Events (arrastar do mouse);
- Mouse Wheel Events (scroll do mouse).

#### 4.4.2 TriggerParallelPort

Em estudos utilizando EEG é comum enviar gatilhos para marcar o registro de data e hora para marcar eventos significativos (por exemplo, o início de um estudo, apresentação de um estímulo específico, etc.). Triggers são tipicamente 1 byte enviado pela porta paralela ao aparelho EEG, sendo utilizado 8 pinos para envios de dados, os outros pinos são usados para controle (como pode ser visto a figura 19).



**Figura 19 - Pinagem da porta paralela**

A maneira óbvia de utilizar essa porta seria desenvolver um software de baixo nível para controlar esse hardware. Para se abstrair toda essa complexidade o projeto faz o envio de dados pela porta paralela a classe TriggerParallelPort utiliza um driver nativo de código aberto que dá acesso direto às portas de hardware (porta paralela ou serial de programas) e foi originalmente desenvolvida pelo grupo Logix4U e mantido por Phillip Gibbons (GIBBONS, 2015).

Pelo fato de usar uma biblioteca dinâmica de dados é necessário fazer sua instalação manual no diretório específicos do Windows, para que a aplicação em Java consiga carregá-los:

- No Windows 32 bit é necessário colocar o arquivo `inpout32.dll` no diretório `C:\Windows\System32\`.
- No Windows 64 bit é necessário colocar o arquivo `inpout32.dll` no diretório `C:\Windows\SysWOW64\`, e o arquivo `inpoutx64.dll` no diretório `C:\Windows\System32\`.

## 4.5 Visão de ação do usuário



**Figura 20 - UT2004 e Neuro Game Logger em execução**

A interface de usuário é uma questão singular dessa ferramenta. Não foi desenvolvida uma interface gráfica para ela, por receio que uma janela saltitante (*pop-up* em inglês) pudesse estragar a experiência de jogo ou até mesmo problemas para o experimento.

Então foi idealizado um modelo sem interface gráfica que é executado em segundo plano ao se lançar a aplicação, esperando entradas do teclado. A tecla F2 executa o servidor do UT2004 (caso a execução seja local), a tela F3 executa o cliente do jogo conectando automaticamente com o servidor e carregando o modo de experimento. O F4 aciona o monitoramento gerando um sinal sonoro quando ele está funcionando, dando início ao experimento.

Antes da execução é importante que o aparelho de EEG já esteja fazendo o registro. Quando o monitoramento do jogador começa, ele envia um marcador ao EEG

indicando que o experimento foi iniciado. Nesse mesmo momento os *logs* são iniciados sincronizando o registro do EEG e os *logs* temporalmente.

Caso seja apertado o botão *Pause* do teclado, o jogo é pausado, bem como o registro do *log* e o envio de marcadores, voltando a ser executado quando o botão é acionado uma segunda vez. Pressionar o botão *Insert* realiza o envio do marcador de final do experimento para o EEG, fecha os arquivos de *log* e encerra o monitoramento do cliente do jogo, do servidor e do Neuro Game Logger.

## 4.6 Logs gerados

Cada execução gera um novo diretório no local onde está o executável do Neuro Game Logger seguindo o seguinte formato: LOG\_EXP\_Player\_ANO.MES.DIA.Hora.Minuto.Segundo (considerando o início do experimento). Nesse diretório encontra-se *logs* no formato CSV:

- **LOG\_Player\_EVENT:** Registro dos eventos realizados pelo personagem observado e capturados pelo Pogamut;
- **LOG\_Player\_KEY\_EVENT:** Registro de todas as teclas pressionadas no teclado durante o experimento;
- **LOG\_Player\_MOUSE\_EVENT:** Registro de todas as ações feitas pelo mouse realizadas durante o experimento;
- **LOG\_Player\_SYNC:** Registro da posição do personagem observado, a cada passo, bem como velocidade de deslocamento da posição da câmera, arma utilizada, entre outras informações;
- **LOG\_Player\_TRIGGER\_EVENT:** Registro de todos os marcadores enviados pela porta paralela.

Os logs registrados têm um formato similar a cada linha. O primeiro campo é um *timestamp* da entrada do registro, o segundo campo é um rótulo identificando o tipo do registro, e terceiro são as informações do evento que originou a entrada no registro. Segue um exemplo de registro no log de quando o jogador coleta um item no mapa:

```
23.32.56.0167,HealthPickedUp,itempickedup("DM-TrainingDay.MiniHealthPack0id1",
[2996.0, -2048.0, -104.0], false,
```

### Figura 21 - Registro no log

Nesse registro é possível observar o timestamp em azul, o rótulo em laranja e as outras informações em amarelo. As informações têm o tipo de item que o jogador coletou, sua posição no mapa, se ele coletou um item que brotou naturalmente no mapa ou se foi pela morte de outro jogador.

## 4.7 Validação

A validação é uma etapa prevista pela Engenharia de Software e se dá para assegurar que o software foi entregue atendendo às expectativas do cliente. Ou seja, se os requisitos, independentemente do que foi planejado, estão sendo implementados para atender a regra de negócio do cliente.

Normalmente a validação de um sistema é realizada pelo próprio cliente de acordo com o que foi especificado, todavia neste trabalho a validação é uma tarefa complexa em razão da sua natureza científica e de requisitos dificilmente testáveis fora de um laboratório de neurociências.

A validação desta ferramenta ocorreu *in loco* do laboratório de EEG do Instituto do Cérebro com o suporte e supervisão do responsável técnico pelo laboratório e aluno de doutorado em Neurociências Daniel Soares Brandão. Este propôs os experimentos que são capazes de possibilitar a validação da aplicação, deu o suporte técnico na utilização dos equipamentos para os mesmos e realizou as análises dos resultados.

Foram desenvolvidos dois experimentos que permitiram algumas análises para cada um. O primeiro experimento permite mensurar a precisão temporal máxima que a ferramenta possui. Como descrito na análise dos requisitos, a precisão temporal é essencial para o sucesso de experimentos de análise psicofisiológica, o resultado devendo apresentar gráficos de sinais gerados pelos eventos e a acurácia da solução por meio de um histograma.

Após a confirmação da precisão temporal que a solução oferece, foi realizado um segundo experimento para se provar o conceito de sua utilização, se dando com um participante jogando contra outro personagem controlado por inteligência artificial (ou *bot*). As análises foram feitas em relação ao percurso do avatar no ambiente virtual e a

potência de teta que um eletrodo específico mediu no jogador e o ERP do evento de disparo.

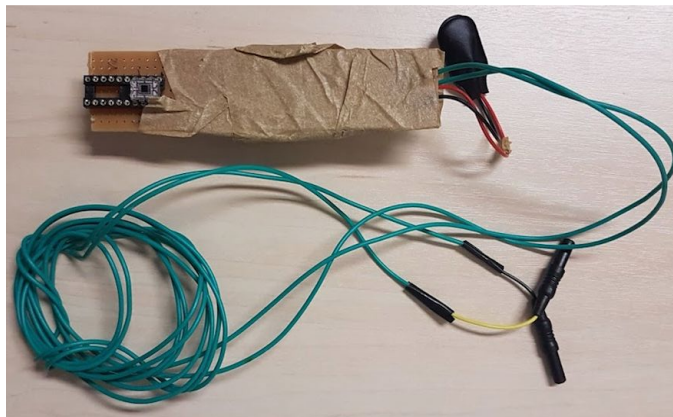
Essas análises foram realizadas utilizando o Matlab, software capaz de ler os arquivos de *log* gerados pela ferramenta proposta neste trabalho e os arquivos do EEG, e extrair os dados dessas duas fontes de maneira a torná-los comparáveis. Com scripts gerados especificamente para comparar esses dados é possível analisá-los e plotar gráficos com certa facilidade utilizando a toolbox disponível no Matlab. Foram utilizadas as seguintes metodologias experimentais em cada um dos experimentos propostos.

#### 4.7.1 Metodologia experimental para determinar a precisão temporal

A fim de definir uma metodologia experimental para se determinar a precisão temporal máxima da ferramenta, foi empregado o conhecimento técnico em experimentos psicofisiológicos que já define estratégias e orientações de como estabelecer essa precisão.

O capítulo 16 do Handbook of psychophysiology (CACIOPPO, 2017) descreve que a melhor maneira de testar o tempo de um sistema de apresentação de estímulo (no caso o UT2004) é registrar o estímulo com o sistema de aquisição de dados. No caso de estímulos auditivos, por exemplo, pode-se colocar um microfone na frente dos alto-falantes e gravar a saída do microfone como se fosse o EEG e utilizar um software de análise de dados para ver se o início do estímulo ocorre exatamente em um determinado momento.

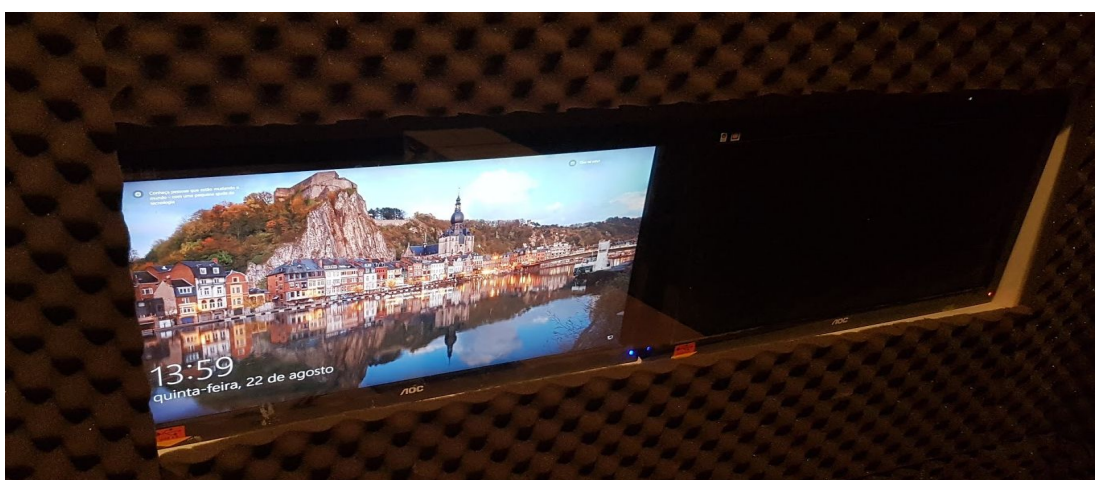
Isso se torna um pouco mais difícil com estímulos visuais. Nesse caso, para a validação, foi necessário o uso de um fotosensor (Figura 21) que gera uma voltagem que varia de acordo com a intensidade da luz recebida.



**Figura 22 - Fotosensor**

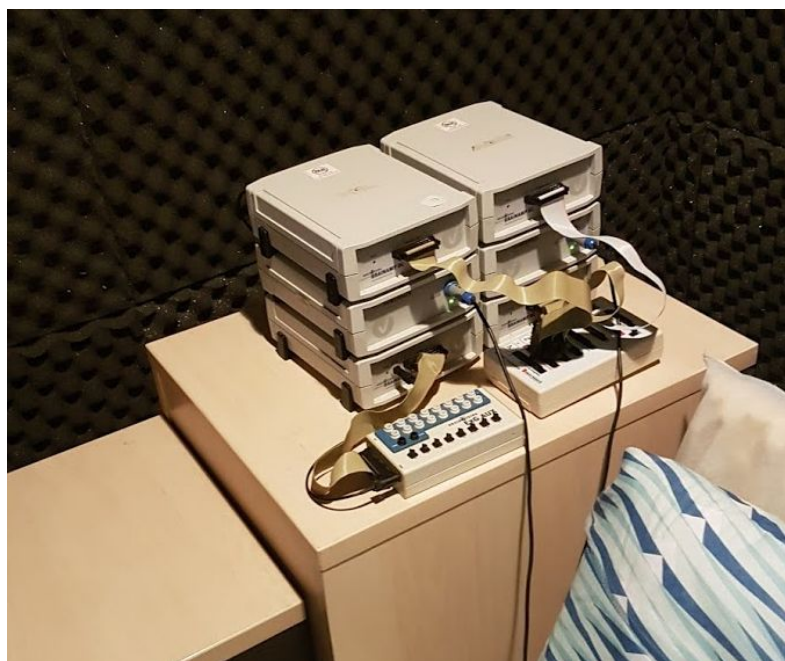
Este sensor pode ser apontado para a parte da tela na qual os estímulos aparecem, e a saída pode ser gravada pelo sistema de aquisição de dados (CACIOPPO, 2017), dessa forma, é possível verificar se a emissão da luz ocorre em sincronia ao que o evento ocorre.

Utilizando essa técnica, seguindo as orientações do corpo técnico do laboratório, foi montado um desenho experimental: em um computador de geração de estímulos dispoñdo de uma porta paralela ligada ao equipamento de EEG, foi instalado o Unreal Tournament, o Pogamut e a ferramenta de análise. Esse computador contou com 2 monitores (Figura 27) espelhados, sendo o primeiro dentro de uma sala vazia, com luz controlada e um fotosensor. Esse controle da luminosidade é importante para que apenas a luz do evento influencie a leitura do sensor.



**Figura 23 - Monitor dentro da sala escura**

O segundo monitor ficou fora da sala, local no qual os experimentadores puderam controlar o computador. Além desse monitor, há outro computador fora da sala em que os experimentadores também puderam acompanhar o registro do EEG em tempo real. O fotosensor ficou ligado ao equipamento de EEG (Figura 28) e seu sinal foi gravado pelo equipamento, permitindo o EEG registrar as mudanças de iluminação dos pixels na tela do monitor, mensurada com uma precisão temporal.



**Figura 24 - EEG**

A ideia de usar um fotosensor é de simular, de forma facilmente identificável, a mudança de padrão de onda que o cérebro teria ao realizar o evento. Assim, utilizando a informação da mudança de iluminação da tela como o momento zero de um evento no jogo, por exemplo, com o personagem atirando em uma parede escura e contabilizando após esse momento o registro do evento na ferramenta, se torna possível saber o atraso para, assim, mensurá-lo.

Apesar da ferramenta de análise oferecer um grande número de diferentes eventos de jogos, foram escolhidos alguns eventos específicos, conforme citados a seguir.

O primeiro evento consiste em pressionar o botão ESC do teclado. Essa escolha foi feita por 3 motivos:

- Ao se pressionar o botão ESC aparece um menu escuro que diminui a luz na tela, o que pode ser facilmente detectável pelo fotosensor;

- Não há uma animação para o aparecimento do menu, ou seja, não há atrasos gráficos que podem comprometer o teste;
- Uma comparação com KeyLogger pode ser feita.

O segundo evento avaliado consiste no disparo de um tiro em uma parede escura, podendo da mesma forma ser identificado pelo fotossensor para permitir uma comparação com o MouseLogger. Os loggers internos da ferramenta (chamados de KeyLogger Java), foram comparados com os da ferramenta de Logger escrita em Python utilizada no laboratório de EEG do Instituto do Cérebro.

Como a precisão e o desempenho são características críticas neste contexto, foi adicionado ainda uma outra ferramenta de Logger escrita em linguagem C++. O objetivo foi comparar o desempenho conforme o esperado no benchmark da área e, assim, sanar qualquer dúvida sobre o ganho em desempenho ao se adotar uma tecnologia de implementação de mais baixo nível e compilada nativamente.

Esperando determinar a latência média e sua variância foi realizado um número consecutivo de eventos contendo aproximadamente 50 repetições ao se pressionar ESC e, no caso dos disparos de tiros, tentando se aproximar de 100 eventos. Os dados do equipamento de EEG foram extraídos e analisados usando um script específico para um tipo de teste em Matlab produzido por membros do laboratório de EEG do Instituto do Cérebro.

Foram realizados 4 casos de teste, um para cada evento (ESC e disparo), utilizando cada forma de capturar esses eventos pelo Pogamut, ou pela captura das teclas do teclado e do mouse feita na própria ferramenta, pelo logger em Python e em C++.

O equipamento gera três arquivos para cada um dos respectivos casos de teste, sendo um com extensão ".eeg" referente aos dados brutos, um com extensão ".vhdr" referente ao cabeçalho e um outro com extensão ".vmrk" referente aos marcadores registrados. Todos os arquivos foram tratados com Matlab com o objetivo de formatar os dados de uma forma legível e visual.

Os resultados devem apresentar gráficos de sinais gerados pelos eventos e a acurácia da solução por meio de histogramas. Após a confirmação da precisão temporal que a solução oferece, a solução está apta a ser utilizada no experimento de provar o conceito de sua utilização.

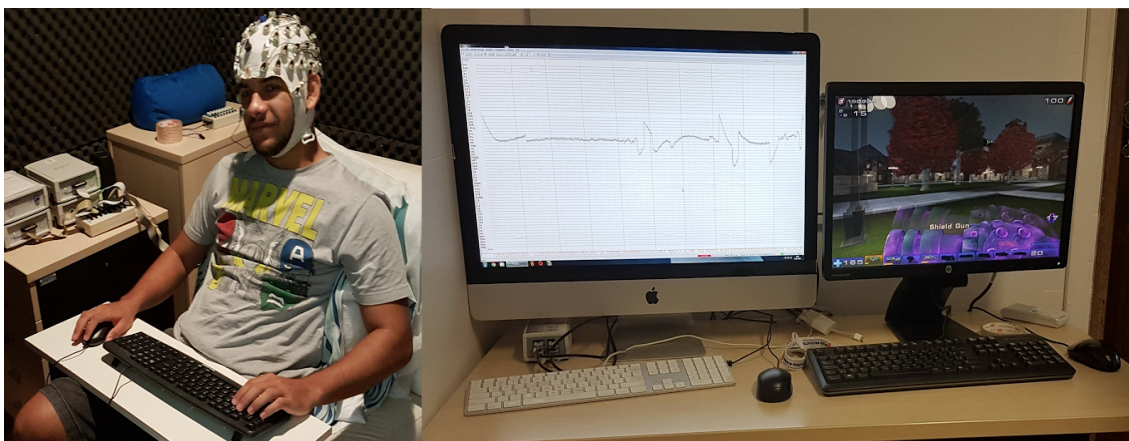
#### 4.7.2 Metodologia experimental usada na prova de conceito

A prova de conceito tem por objetivo estabelecer que um método ou conceito teórico possui viabilidade, seus potenciais e se podem ser exploradas de maneira útil. Nesta outra fase de validação o objetivo consistia em realizar uma prova de conceito a fim de estabelecer se a solução tem à capacidade de ser utilizada para produzir uma análise usando um voluntário e o equipamento de EEG.

Apesar das análises realizadas sejam similares às utilizadas em diversos artigos da área, essa prova de conceito não espera necessariamente que os resultados das análises realizadas corroboram a literatura. Isso se dá porque esse estudo de caso não tenta replicar qualquer experimento em sua totalidade e sim demonstrar a possibilidade de se fazer essas análises.

A primeira análise escolhida para esse estudo de caso foi um mapa de calor que consiste em gerar uma mapa bidimensional da partida (visto de cima) sendo marcado a movimentação do avatar do jogador, assim é possível fazer o rastreo do mesmo no ambiente virtual. Adicionalmente a esse *plot*, além da posição x e y do avatar, a potência do teta do jogador durante aquele deslocamento. Isso permite fazer uma correção espacial com a potência de teta já que em alguns artigos expõem uma correlação entre essas duas grandezas. A segunda análise realizada foi ERP do evento do disparo.

Um dos bolsistas do laboratório se voluntariou a ser participante, estando bem de saúde e descansado. Uma touca foi montada e ligada no EGG.



**Figura 25 - Voluntário no *setup* de experimento e sistema de monitoramento do experimentador**



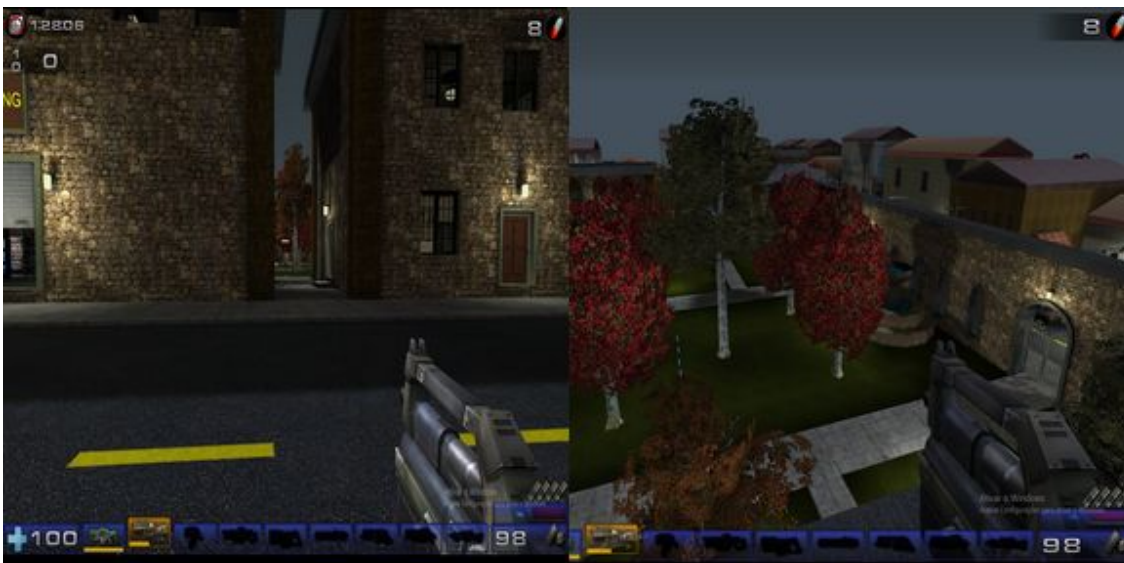
baseada na região cerebral coberta por este eletrodo e o número indicando a sua lateralização. Desta forma é utilizado Fp para fronto-polar, F para frontal, C para central, P para parietal, T para temporal e O para occipital.

Para diferenciar áreas homólogas em cada um dos hemisférios se utiliza números pares no hemisfério direito, ímpares no esquerdo, os eletrodos da linha média receberam a denominação Z e os eletrodos auriculares são chamados A30 (a esquerda) e A30 (a direita).

O Voluntário ficou isolado durante o experimento, sendo possível acompanhar as suas decisões através de um monitor espelhado fora da sala, bem como a leitura do EEG (monitorando o funcionamento de cada canal) e o envio automático de marcadores pela ferramenta.

Utilizamos um protocolo no qual o participante tinha a oportunidade de percorrer o cenário e coletar itens (forrageio) ao mesmo tempo em que disputava uma duelo contra um *bot*. O nível de dificuldade do *bot* oponente era acima da média, sendo considerado assim um aprendizado em situação de estresse, em um experimento de 1 hora de duração. O mapa escolhido para o experimento foi DM-UnrealVille que são 4 ruas que se cruzam em formato retangular com um grande prédio no meio (ver Figura 26).





**Figura 27 - Pontos do mapa DM-UnrealVille**

Essa construção tem várias entradas e corredores formando uma espécie de labirinto, com vários itens espalhados nos cômodos e várias oportunidades de exploração. O Neuro Game Logger foi configurado para fazer log de todos os eventos, mas em relação ao envio de marcadores foi decidido que fosse enviado sempre que o personagem do jogador se movesse.

Um outro marcador enviado foi quando o personagem jogador realizasse o disparo de tiros. O envio de marcadores do deslocamento foi enviado a uma frequência aproximada de 50 ms, já o do disparo foi enviado pelo MouseLogger presente no Neuro Game Logger.

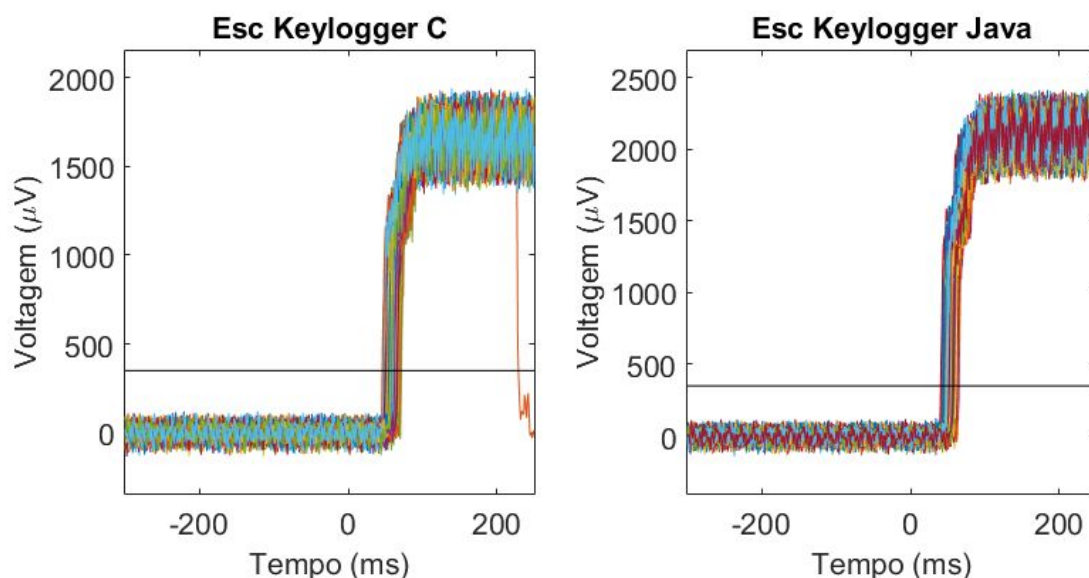
### 4.7.3 Resultados sobre a precisão temporal

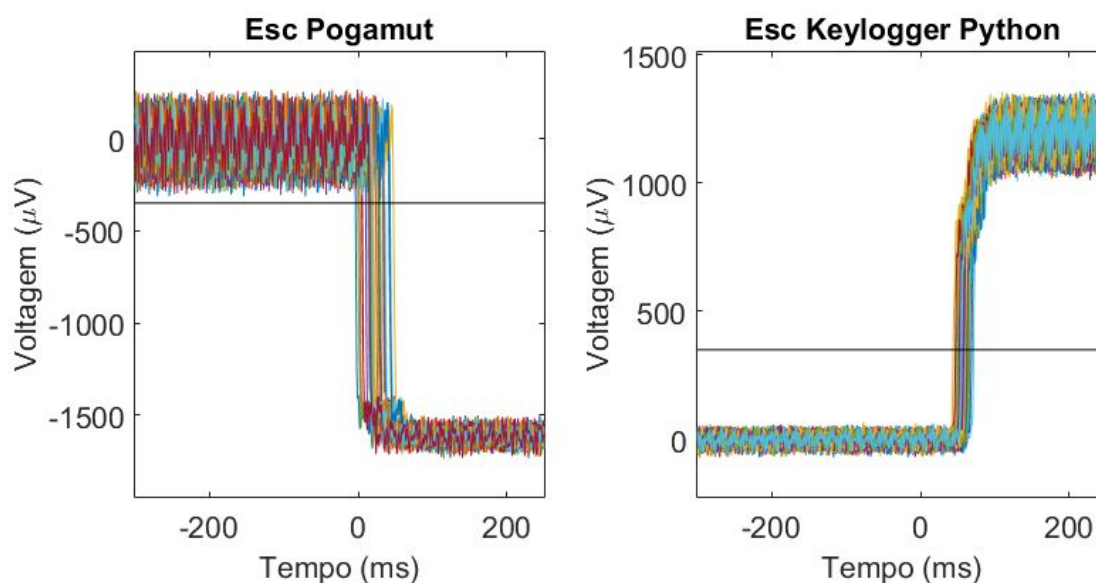
Depois da formatação dos dados dos arquivos é possível designar o momento do *input* determinado pelo *log* da ferramenta e o momento que ocorreu a mudança de tensão no fotosensor registradas pelo EEG. Assim é possível determinar o atraso entre o *input* registrado no *log* e a mudança de iluminação da tela em cada tentativa.

Identificar o quanto é esse atraso e se ele se mantém constante ou é variável, é de suma importância para determinar a chance estatística do envio do marcador representar temporalmente o evento. É natural que esse atraso não seja constante já que estamos lidando com dezenas de tempo muito pequenas e computadores comuns não garantem tempo real nessa escala.

Entretanto podemos determinar a variação desse atraso com uma certa precisão se comparamos todos os atrasos. Ao sobrepor os sinais (cada um com uma cor diferente) de mudança de tensão de todas as tentativas em um eixo X em relação a um eixo Y sendo o tempo, sendo o momento 0 a detecção do evento por meio da ferramenta analisada, podemos determinar as faixas de tempo e os limites máximos nos quais todos os sinais ocorreram.

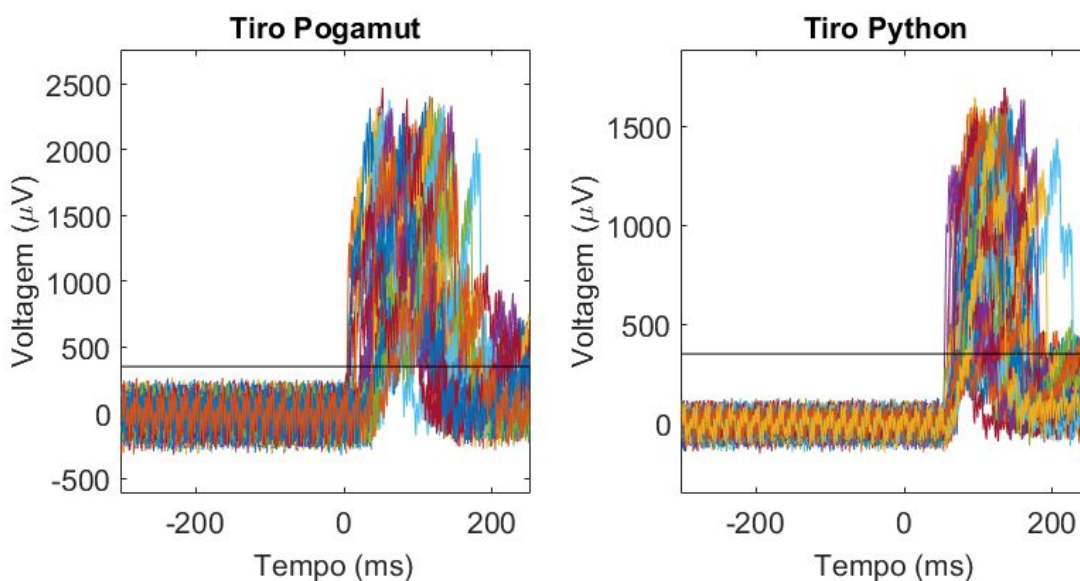
Os limites foram estabelecidos 200 ms antes e depois do momento do evento, assim como também é possível estabelecer os limites de microvoltagem, referentes à luminescência do fotosensor, definindo a mudança no sinal que indica que o evento ocorreu estabelecido em  $3500\mu\text{V}$ .

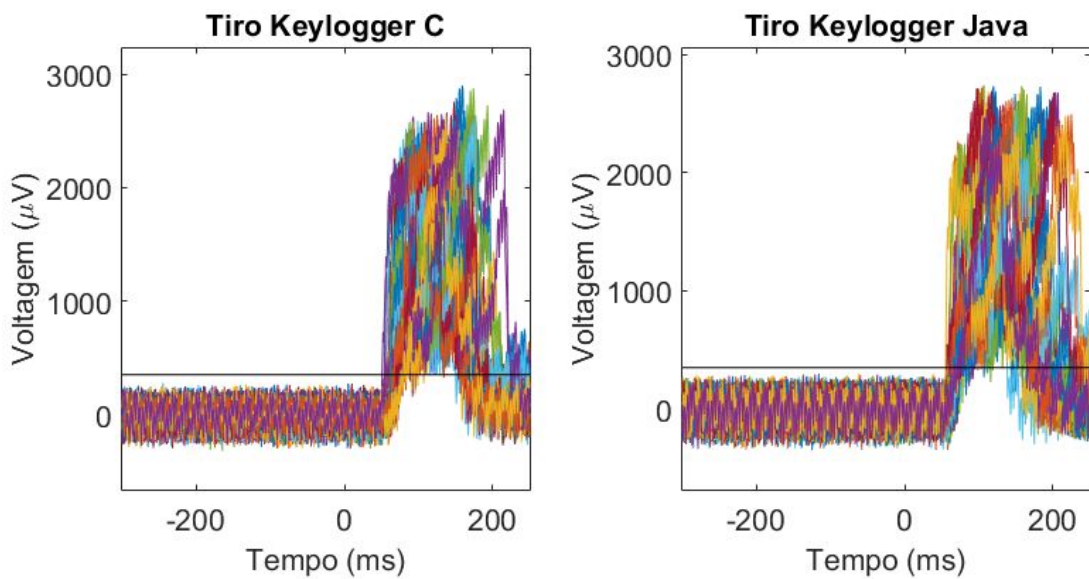




**Figura 28 - Sinais do trails sobrepostos do evento ESC**

Em alguns casos de teste do experimento houve uma inversão da voltagem do espectro positivo para o negativo, que foi causada apenas por um detalhe da implementação do jogo. A tecla ESC abre a janela de menu e a sua ativação causa uma pausa na execução da partida, fazendo com que o Pogamut só reconheça o evento quando ocorre a desativação do menu. Apesar disso, as ordens de grandeza se mantiveram em todos os casos de teste, sendo necessário apenas inverter os limites para não causar prejuízo ao resultado.

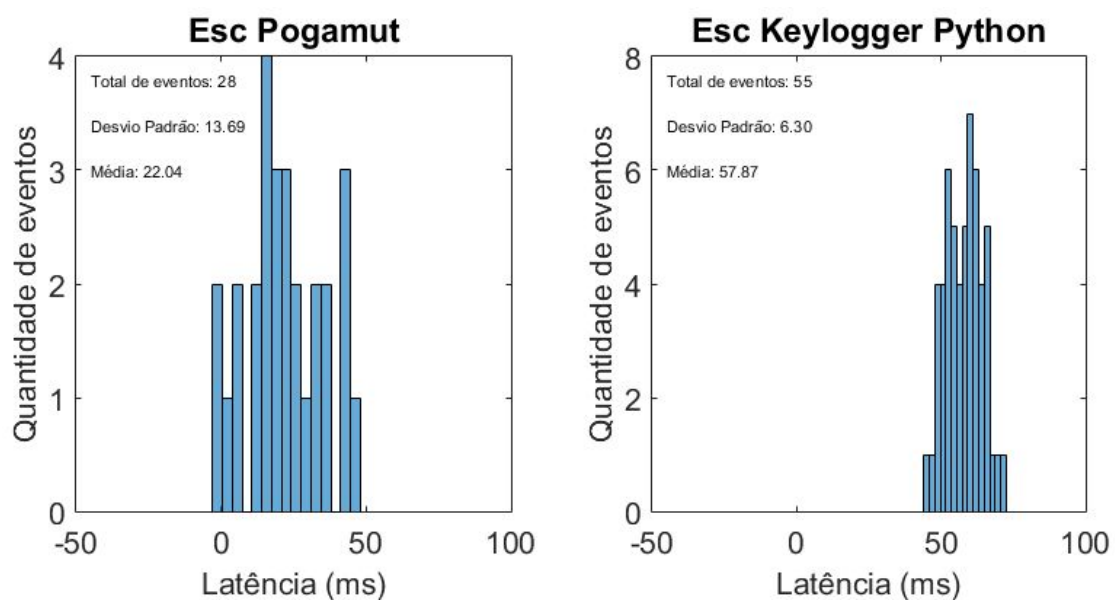


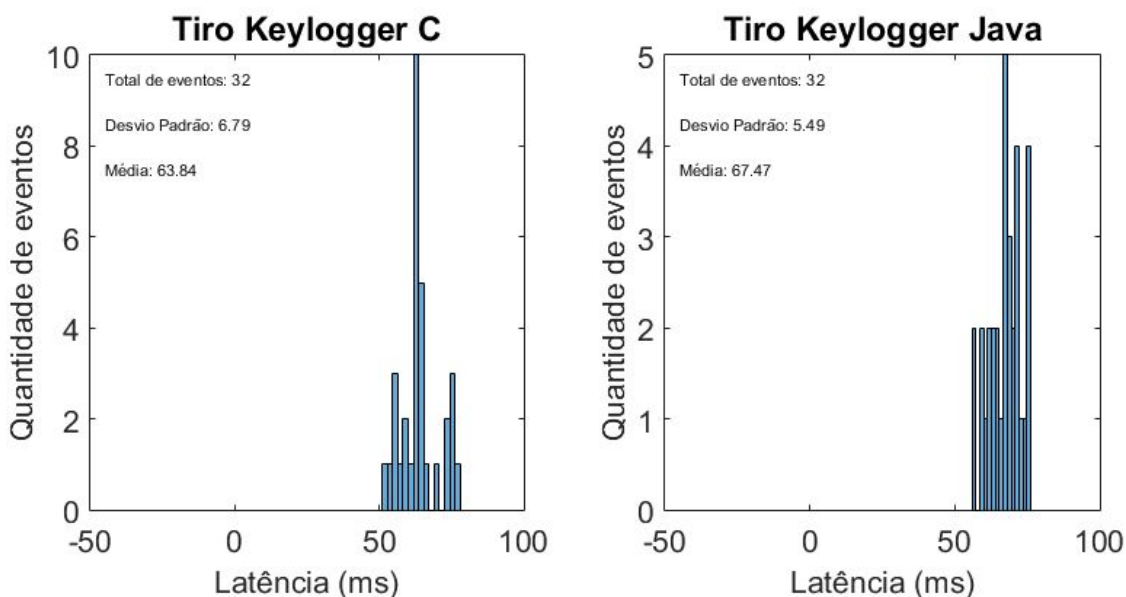


**Figura 29 - Sinais do trails sobrepostos do evento tiro**

Quanto ao evento de disparo de tiro não existiu nenhuma diferença entre os casos de teste, conforme pode ser visto na Figura 28, sendo um que o sinal se apresenta em um formato de pulso. Com os parâmetros mapeados dos tempo de atraso dos eventos e da distribuição das suas frequências em classes de forma gráfica gerou-se um histograma.

Distribuindo a frequência dos eventos no eixo y, sendo uma quantidade de eventos agrupados em classes determinadas pela a latência em milissegundos no eixo x na figura 34.





**Figura 30 - Histograma do evento ESC**

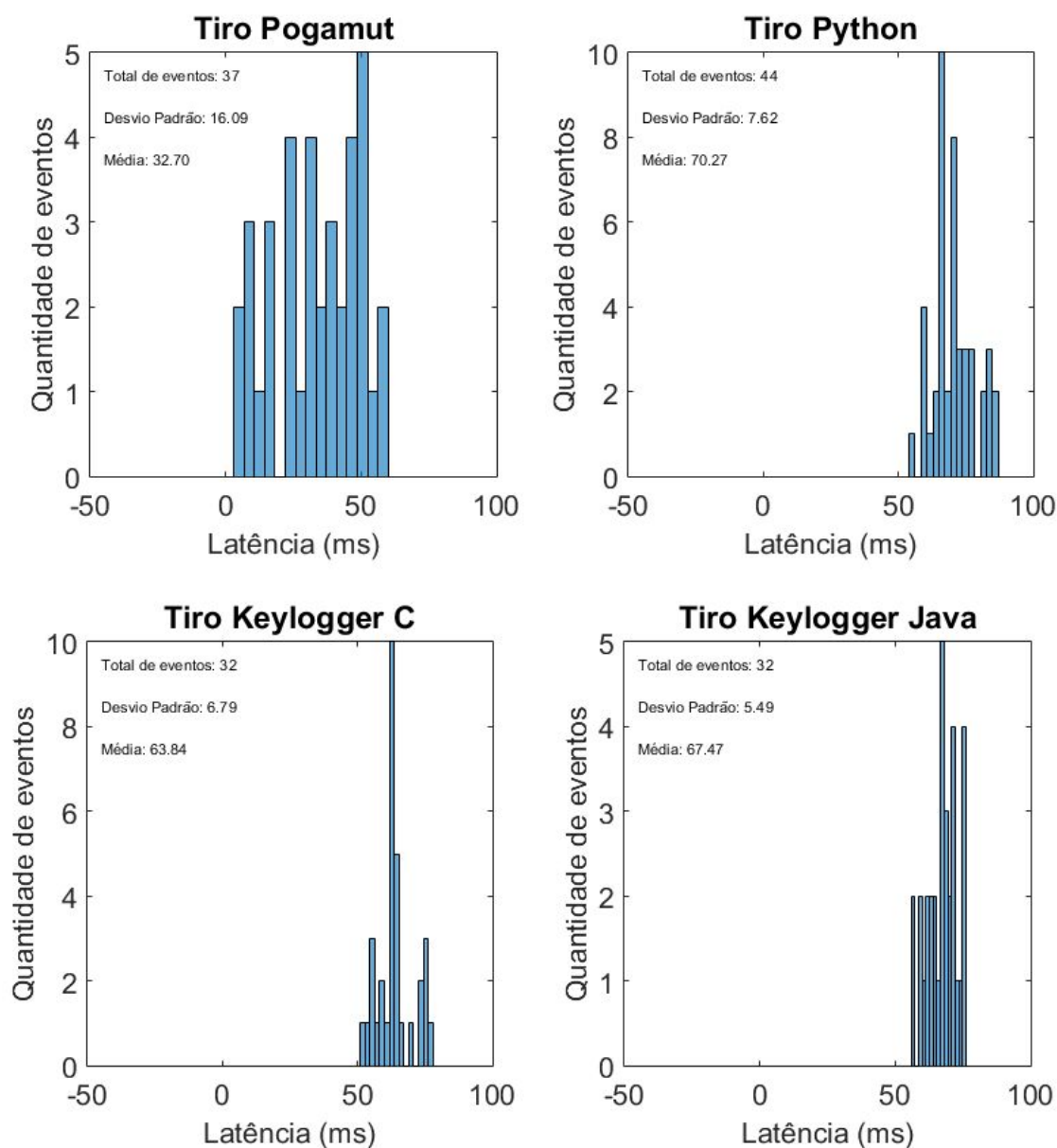
O histograma revela uma certa variância no atraso tanto dos keyloggers quanto da ferramenta quando envia marcadores pelo seu keylogger interno ou com eventos capturados pelo Pogamut, evento esperado dado a resolução temporal analisada e a natureza dos sistemas computacionais modernos.

O eixo da latência indica quanto tempo antes do evento (mudança de iluminação da tela) o marcador foi enviado, dessa forma, quando mais longe do 0 mais próximo o registro se encontra do evento que regrou a mudança dessa iluminação. Foi constatado nos gráficos dos casos de teste que um evento de pressionar o botão ESC tem performance inferior no Pogamut quando comparado aos keyloggers.

Além disso, a captura de eventos pelo Pogamut se mostrou com maior variância em relação aos keyloggers, dando assim aos keyloggers uma maior previsibilidade que garante uma maior assertividade em determinar o momento que o evento ocorre, relevando o atraso. Essa conclusão fica mais clara quando comparamos o desvio padrão.

O desvio padrão de uma amostra indica a dispersão dos dados em torno de uma média amostral, quanto mais baixo o desvio padrão, mais os pontos tendem a estar próximos à média, logo os keyloggers com valores de desvio padrão de 5,49; 6,30; 6,79 são opções melhores quando sua previsibilidade que a captura de eventos pelo Pogamut com o desvio padrão de 13,69.

É possível dizer que as diferenças entre os keyloggers são praticamente desprezíveis entre si, mas de toda forma existe uma leve vantagem demonstrada pelo valor de desvio padrão para o keylogger da solução, o que é um ponto menos problemático para a ferramenta. O ideal seria se a detecção do eventos pelo Pogamut tivesse uma vantagem temporal em relação ao feito pelos keyloggers, dado que é um registro mais completo e contextualizado na experiência de jogo.



**Figura 31 - Histograma do evento de tiro**

Da mesma forma, os casos de teste de captura de evento nas mesmas plataformas seguem um padrão similar ao visto nos casos anteriores. O desvio padrão é

maior para a detecção de eventos pelo Pogamut de título “Tiro Pogamut” (16,09) que o desvio padrão do registro feito com o MouseLogger em Python de título “Tiro Python” (7,62) e o MouseLogger em C++ de título “Tiro KeyLogger C” (6,79). Mais uma vez o MouseLogger da ferramenta possui o menor desvio padrão em relação às outras opções.

#### 4.7.4 Investigação da maior variância detectada em eventos utilizando o Pogamut

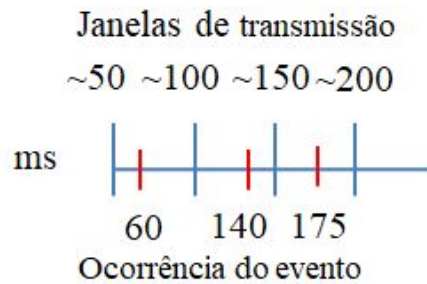
O fato da captura de eventos pelo Pogamut ter um desempenho muito inferior ao visto pelo KeyLogger poderia até ser esperado dado que o evento tem um registro com uma quantidade maior de informações e uma complexidade de obtenção oriunda de ser obtida por protocolo de rede ser esperada, no entanto uma variância tão elevada gerou certa desconfiança.

Com o objetivo de tentar identificar a causa da limitação, a fim de encontrar uma solução, foram realizados testes e análises em seguida do código do Gamebots e do Pogamut.

A partir do momento que o jogo inicia, o *mod* do game *bots* começa a trocar mensagens com o Pogamut via *sockets* TCP/IP. Logo, foi feito um teste forçando o Gamebots a enviar mensagens o mais rápido possível, porém mesmo diminuindo para o menor valor possível (um nanosegundo), os “*delays*” no Gamebots só enviavam mensagens em uma janela de aproximada 50 milissegundos.

Mesmo enviando uma quantidade maior de marcadores o envio só acontecia aproximadamente a cada 50 milissegundos, nesta configuração quando executado um evento de tiro, por exemplo, ao invés de mandar o evento predeterminado, era enviado o marcado do evento de tiro, o que explica a latência percebida no teste de precisão temporal realizado.

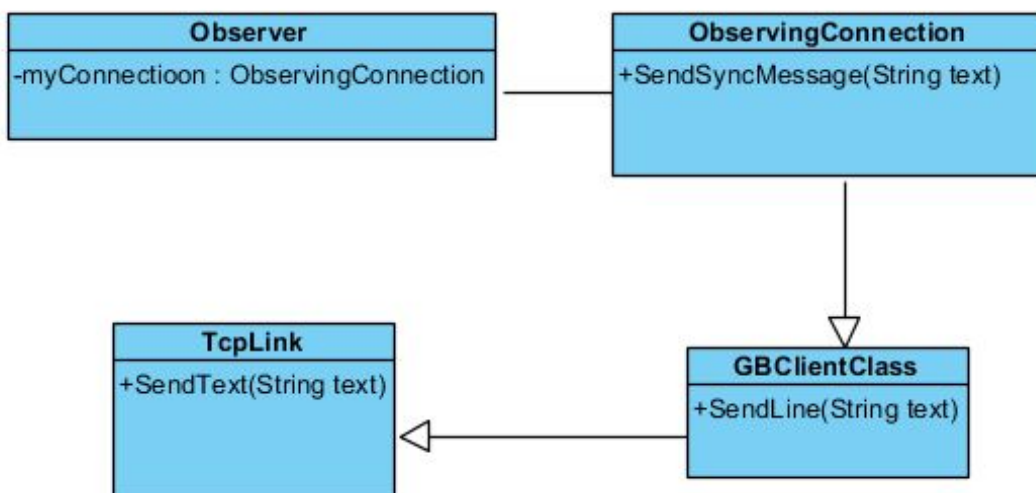
O evento ocorrido no mundo de jogo só foi notificado pelo Gamebots para o Pogamut na próxima janela de transmissão de aproximadamente 50 ms, dessa forma a latência depende de quando o evento ocorre dentro do intervalo da janela de transmissão do Gamebots, como está representando na Figura 31.



**Figura 32 - linha do tempo de eventos e transmissão.**

Nessa linha do tempo a ocorrência de evento está indicada em marcadores vermelhos, um primeiro evento ocorre aos 60 ms, ele só seria notificado para o Pogamut no momento 100 ms que é o intervalo da próxima janela de transmissão. Nesse exemplo os atrasos seriam de 40 ms, 10 ms e 25 ms respectivamente, sendo o mesmo comportamento evidenciado pelo teste de precisão.

Durante a análise de código fizemos o “*tracking*” do evento de disparo nas classes do Gamebots, a fim de entender qual parte do software é responsável por esse comportamento. Quando um personagem monitorado realizar um disparo, um objeto (do tipo Observer) específico para esse personagem tem a responsabilidade notificar.



**Figura 31 - Diagrama de Classes do Gamebots**

Esse objeto compõe outro objeto do tipo *ObservingConnection* que é o responsável por enviar a mensagem por rede notificando o disparo do tiro. O método utilizado é denominado *SendSyncMessage*, que na verdade corresponde a um *facade* para o método *SendLine* da sua classe pai *GBClientClass*.

O método *SendText* é outro *facade* (um objeto que serve como uma interface que mascara um código mais complexo) para o método *SendText* presente na sua classe pai *TcpLink*, entretanto *TcpLink* é uma classe de código nativo do Unreal Tournament 2004. Infelizmente o código dessa classe não é aberto e inviabilizou a investigação como tentativa de encontrar um meio para contornar essa limitação.

#### 4.7.5 Reflexão quanto ao resultado da validação de precisão temporal

O primeiro ponto de reflexão consiste em assumir que a captura de eventos do Pogamut possui uma variância de atraso maior pela sua natureza de implementação. Essa alta variância temporal pode atrapalhar a análise que necessita de precisão temporal, contudo, os *loggers* embutidos na ferramenta podem garantir a capacidade de enviar marcadores com a precisão necessária e a captura de eventos pelo Pogamut pode gerar as informações adicionais.

Infelizmente, essa alternativa inviabiliza a análise de eventos que não dependem de entrada do mouse e do teclado em alta precisão temporal, por exemplo, se em um experimento for desejado fazer o ERP do pós-morte do personagem do jogador, a variância de detecção dos eventos com o uso da ferramenta seria um problema para a pesquisa.

Apesar de tudo, o Pogamut oferece uma série de informações precisas sobre o ambiente de jogo para todos os eventos, portanto, o Neuro Game Logger poder oferecer precisão temporal com os eventos, surgindo a partir de inputs por meio dos seus *loggers*, ainda garantindo uma correção temporal dos eventos capturados pelo Pogamut. Essa conclusão habilita a ferramenta para uma segunda etapa de validação, em que será tratada essa forma de correção.

## 4.8 Resultados da Prova de conceito

As ondas teta hipocampais, com uma faixa de frequência de 6 a 10 Hz, aparecem quando um rato está envolvido em comportamento motor ativo, como caminhar ou farejar exploratório, e também durante o sono REM (LEGA, 2012). Em humanos há pesquisas que indicam efeitos similares, se espelhando em achados de efeitos similares em humanos realizamos a seguinte análise.

A primeira análise que foi realizada nos dados gerados neste experimento se refere ao mapa de calor quanto à potência de teta no eletrodo T7. Existem estudos que observam um aumento da potência de teta (ou seja um ritmo cerebral entre 4-7 Hz) durante a tarefa motora exigida para navegação e fazem uma associação entre o teta e o desempenho de tarefas de navegação em humanos (BISCHOF, 2003), (ARAÚJO, 2002) e (KAHANA, 1999).

A segunda análise foi o ERP após o evento de disparo de tiro, que é um procedimento pelo qual é possível fazer inferência entre eventos e processos psicofisiológicos. Existem ainda artigos que confirmam que jogar jogos de FPS altera os processos neurais que suportam atenção seletiva espacial, estabelecendo uma relação causal entre esse gênero de jogo digital e a mudança na plasticidade cerebral, percebidas por meio de ERPs antes e depois da jogatina (WU, 2012).

As análises foram realizadas pelo corpo técnico do laboratório do Instituto do Cérebro utilizando o mesmo ferramental da validação anterior, no entanto, foi pensada uma estratégia de análise para evitar o problema do atraso sofrido.

O primeiro passo para a realização das análises foi alinhar temporalmente os marcadores enviados com os eventos capturados pelo Pogamut e salvos em *log*. Desta forma, o algoritmo percorre todos os marcadores enviados e tenta conjecturar o atraso dos marcadores enviados pelo Pogamut em relação os eventos capturados pelos keylogger.

Depois que esses marcadores estão alinhados na escala de tempo, eles são pareados com as entradas no *log* referente, associando o marcador e o evento com um tempo mais próximo limitado ao atraso máximo observado na validação anterior. No contexto dos testes realizados, essa abordagem teve sucesso em encontrar todos os marcadores, obtendo o deslocamento com precisão temporal.

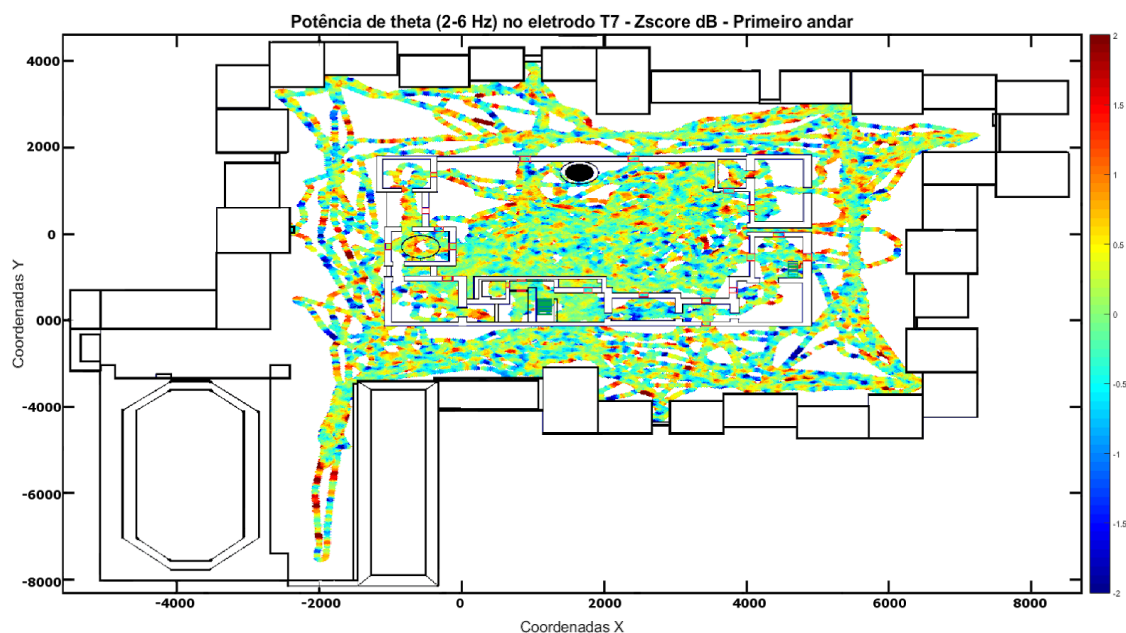
#### 4.8.1 Análise da potência de teta

A análise da potência de *teta* investigou a intensidade do sinal no eletrodo T7 durante toda a navegação do personagem do jogador realizada no mapa, os passos dessa análise se deram da seguinte forma:

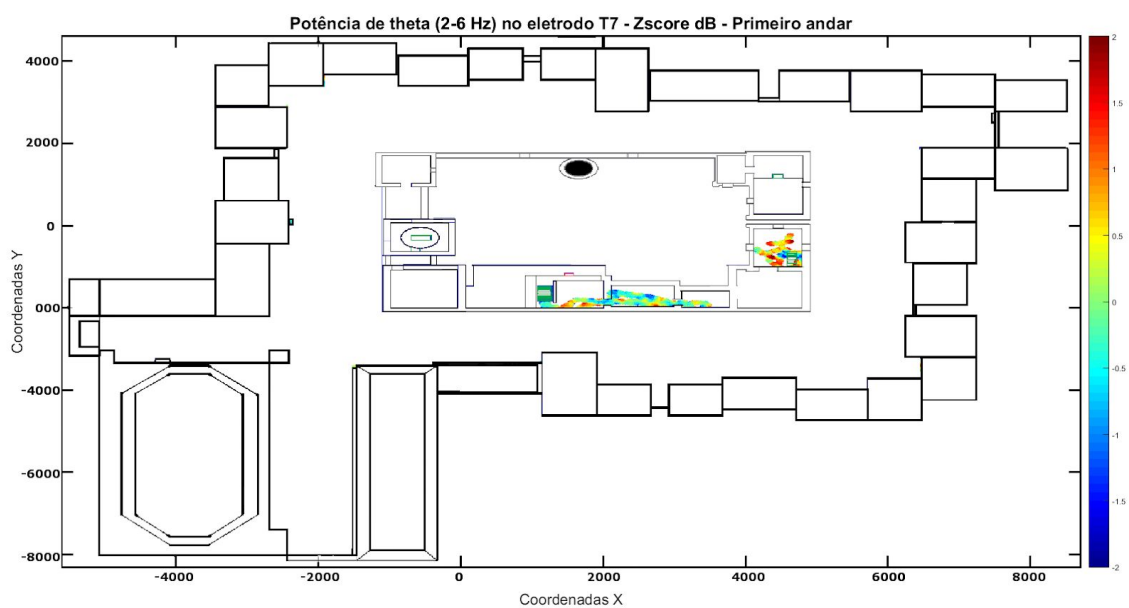
1. A criação de um novo canal de referência dado pela média de todos os outros 64 canais, habilitando a possibilidade de permitir a análise de todos os canais;
2. A filtragem dos sinais apenas do canal T7 na faixa de frequência de 2 Hz até 6 Hz;
3. A realização de um envelopamento da modulação por meio da transformação de Hilbert;
4. Alguns tratamentos dos sinais foram convertidos à magnitude em decibéis ( $20 \log_{10}(y)$ );
5. As ondas foram normalizadas usando o método Zscore que é utilizado para o desvio padrão da média.

A pontuação bruta é uma medida de quantos desvios padrão ocorreram abaixo ou acima da população, o Zscore variou de um desvio padrão de -2 (na extrema esquerda da curva de distribuição normal) até um desvio padrão de +2 (na extrema direita da curva de distribuição normal). Para apresentar em um formato de representação visual, foi escolhida uma escala de cores sendo a maior potência na cor vermelha, a menor potência na cor azul e neutro na cor verde.

Usando os marcadores foi possível fazer um cruzamento dos dados de posição e potência do teta a cada instante, que foram usados para plotar dois mapas correspondentes ao primeiro (mais baixo) e segundo (mais alto) pisos (ver Figuras 29 e 30). Para fazer essa distinção entre os pisos foi analisado o eixo Z para definir em qual piso o personagem do jogador estava a cada instante. Há uma versão ampliada disponível nos anexos deste documento.



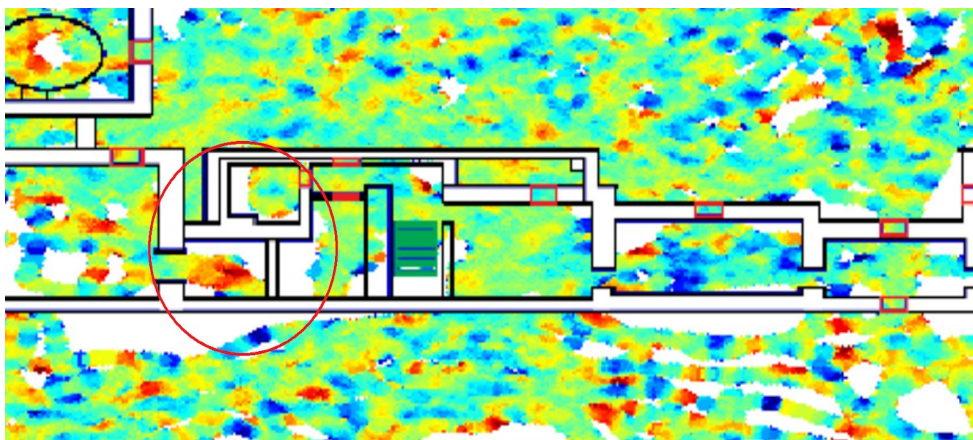
**Figura 34 - Mapa de calor da potência de teta (2-6 Hz) no eletrodo T7 -  
Primeiro piso**



**Figura 35 - Mapa de calor potência de teta (2-6 Hz) no eletrodo T7 -  
Segundo piso**

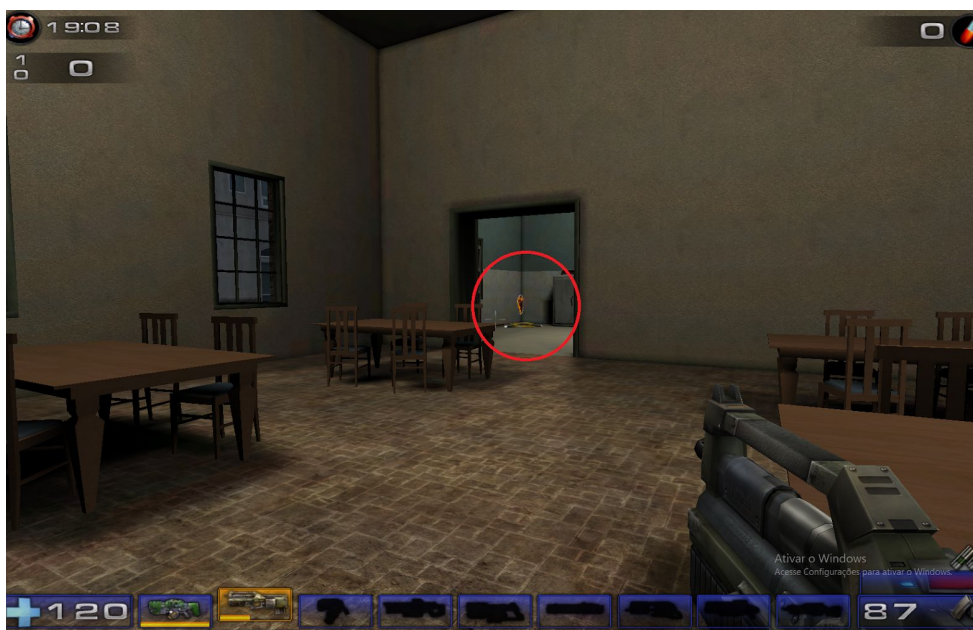
Na pesquisa realizada por Walter F Bischof (BISCHOF, 2003) foi indicado que a análise realizada no espaço de tempo das bandas teta é capaz de mostrar que os episódios de teta ocorrem mais provavelmente em pontos em um labirinto onde novos corredores são vistos, ou após erros de navegação terem sido realizados.

A partir das informações do mapa de calor gerado nos experimentos, algumas áreas demonstram um aumento da potência de teta. Por meio de uma análise visual é possível encontrar uma sala sem saída, com manchas vermelhas, ou seja, indicando um aumento na potência de teta, conforme descrito no artigo de Bischof (ver Figura 35).



**Figura 36 - Mapa de calor gerado a partir dos experimentos no qual um círculo indica a localização de uma sala com alto valor teta**

A visão que o jogador teve fora da sala pode ser vista na Figura 36, na qual é possível observar que não permitia tirar nenhuma conclusão sobre a inexistência de saída da sala, após entrar, ele percebe que a sala não tem saída indicando um aumento da potência do teta.



**Figura 37 - Visão de fora da sala**

A hipótese seria que o aumento da potência de teta, seria o jogador parado tentando se orientar no mapa. Nesse cenário, ele estaria parado no momento após perceber que a sala não teria saída, diferente de outras manchas vermelhas em áreas abertas que poderiam ser justificadas pela aceleração do deslocamento (LEDBERG, 2011).

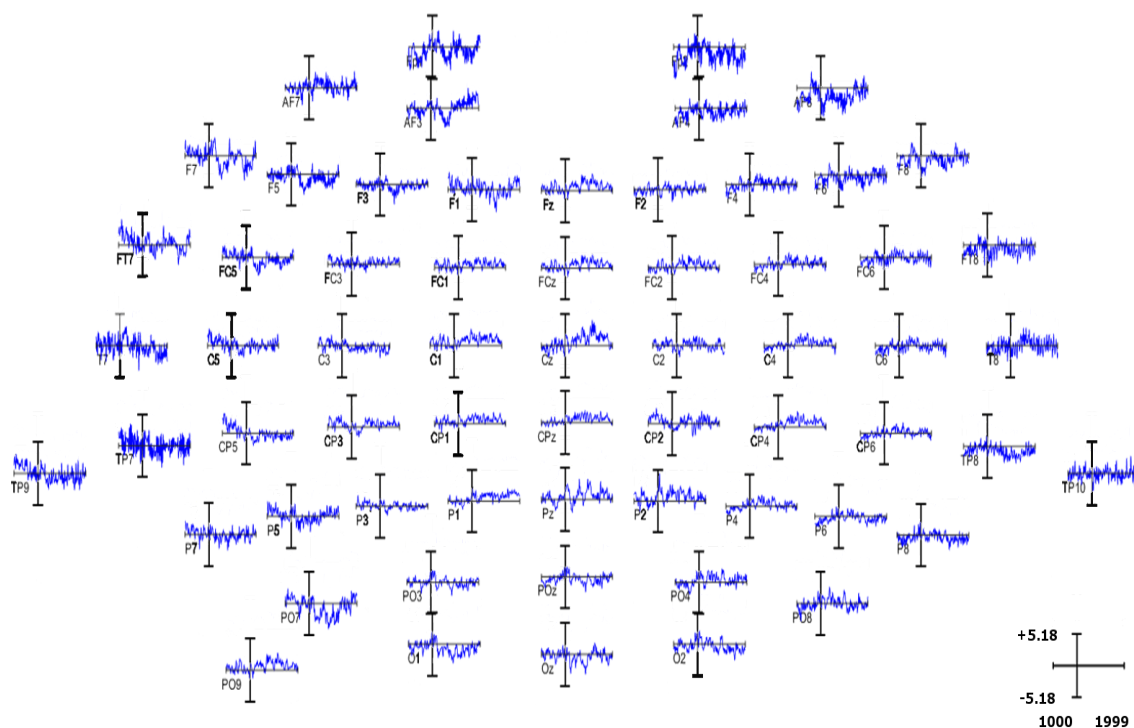
#### 4.8.2 ERP após disparo de tiro

Potenciais relacionados a eventos (ERPs) são voltagens muito pequenas geradas nas estruturas cerebrais em resposta a eventos ou estímulos específicos (Blackwood e Muir, 1990), e são amplamente utilizados em pesquisas em neurociência, psicologia cognitiva, ciência cognitiva e psicofisiológica. Psicólogos e neurocientistas experimentais descobriram muitos estímulos diferentes que provocam ERPs confiáveis dos participantes.

Para a análise do ERP foram utilizados aos potenciais evocados (que é um sinal elétrico registrado do sistema nervoso seguido um estímulo) e foi aplicada uma transformada off-line com a média do sinal de todos os eletrodos.

Em seguida, os sinais foram filtrados (com um filtro passa-banda, um dispositivo que permite a passagem das frequências de uma certa faixa e rejeita frequências fora dessa faixa) de 0.5 a 30 Hz. Depois disso foram selecionados os intervalos de interesse, correspondentes de 1 segundo antes até 2 segundos após cada disparo de tiro.

Nos casos em que dentro do intervalo de interesse ocorreu mais de um disparo de tiro, esse intervalo foi ignorado e o sinal de cada disparo foi subtraído da média dos 400 milissegundos precedentes ao tiro. Todos os eventos de tiro que tivessem voltagem maior que 100 microvolts ou menor que -100 microvolts, para qualquer canal e para qualquer instante dentro do intervalo de interesse, foram descartados.

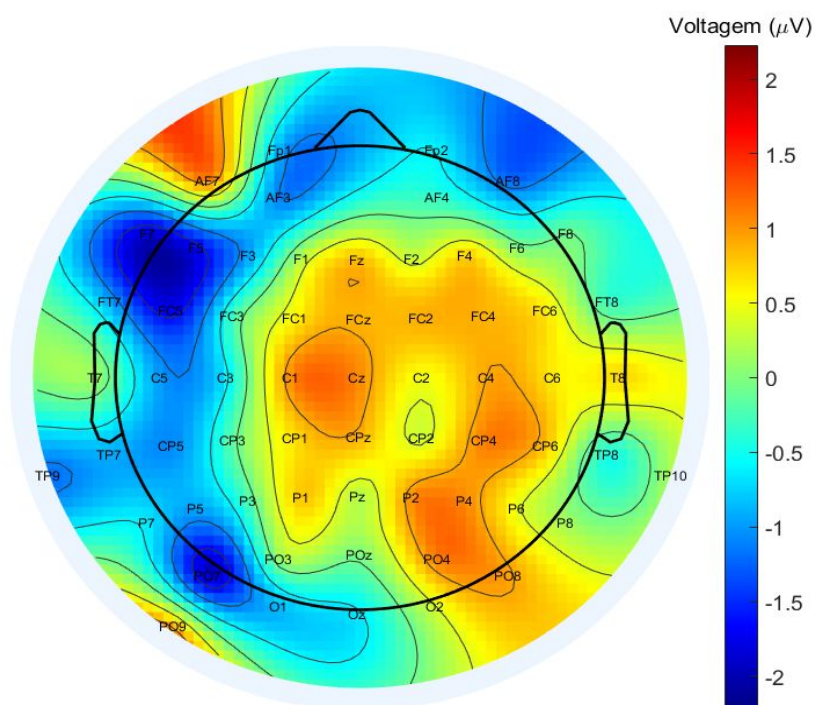


**Figura 38 - ERP após o tiro para todos os eletrodos**

O ERP foi realizado em todos os canais como pode ser observado na Figura 37 (uma versão ampliada está disponível nos anexos). Além da plotagem do ERP de todos os canais, foi montada uma topografia, que se trata de uma técnica de neuroimagem na qual um grande número de eletrodos do EEG é visualizado de uma perspectiva de cima, seguindo uma matriz geométrica de pontos espaçados.

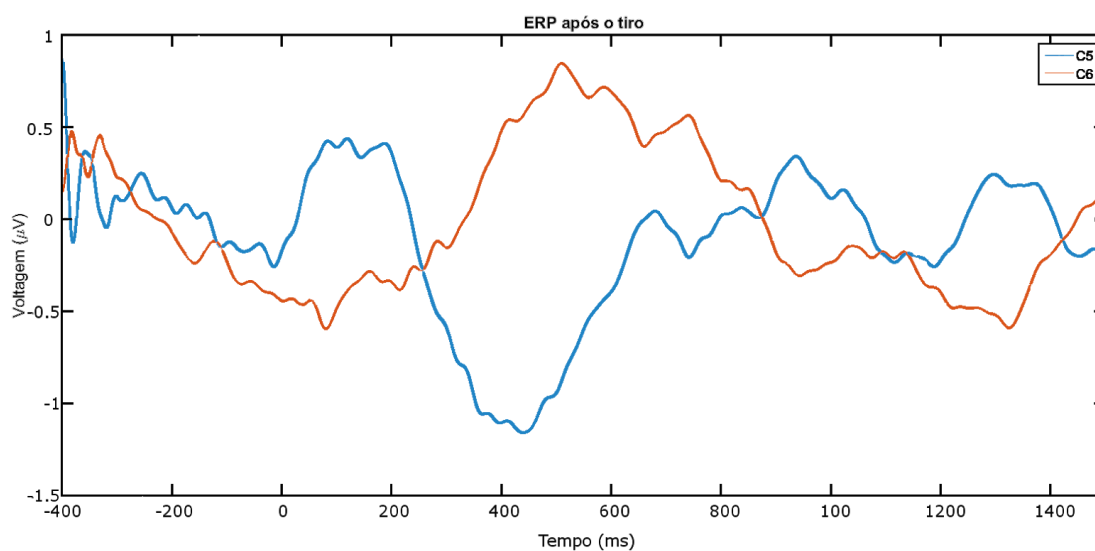
Mapas topográficos de tensão de cabeça inteira são uma forma visual de observar a ativação em áreas específicas, o mapa (na figura 38) de tensão do EEG foi plotado com interpolação e mostra a distribuição das tensões instantâneas no pico positivo máximo do período médio de resposta médio grande, essa representação foi montada com uma escala de cor referentes a com escores da tensão, do vermelho para a maior ativação e azul para o menor ativação.

Topografia de 300-600 ms após o tiro



**Figura 39 - Topografia do ERP após o disparo de tiro**

Dessa forma que é possível verificar os canais simétricos com maiores diferenças de tensão. Foram escolhidos os canais C5 e C6 para a sobreposição dos sinais, selecionando o sinal desses dois eletrodos em cada tiro e sobrepondo os sinais obteve-se uma evidente mudança após o momento zero da escala, referente ao evento do disparo. É preciso destacar que foi feita uma suavização através de uma média móvel com janela de 200 ms.



**Figura 40 - ERP após o tiro C5 e C6**

## 4.9 Conclusão da validação

Variância é uma medida importante para determinar o sucesso da solução apresentada neste trabalho. Na teoria das probabilidades e estatística, a variância é a expectativa do desvio de uma variável aleatória sua média ou seja "o quão longe" em geral os seus valores se encontram do valor esperado (HINES, 2008). A variância tem um papel central na estatística, onde algumas idéias que a utilizam incluem estatística descritiva, inferência estatística e teste de hipóteses.

A validação da precisão temporal demonstrou uma maior variância para determinar o momento que os eventos ocorreram no jogo quando sua detecção foi feita pela API do Pogamut, entretanto, foi possível corrigir essa variação durante as análises. Além disso, a precisão temporal garantida pelo monitoramento do mouse e do teclado, realizada pela ferramenta proposta neste trabalho, garante uma baixa variância, mesmo quando comparado a códigos compilados.

A validação quanto à capacidade de análise realizada exibiu êxito em gerar material para validação, em conformidade com os métodos utilizados em pesquisas psicofisiológicas por meio de EEG. Levando em consideração os padrões médicos adotados pelos equipamentos psicofisiológicos, é possível que a mesma ferramenta possa ser utilizada sem modificação por meio de outros equipamentos com capacidades de análise diferentes.

O primeiro caso de análise explorou a capacidade de mapeamento espacial da ferramenta e a sincronização espacial com o envio de marcadores do EEG e se mostrou além de possível, precisa, ao ponto que os valores de posição marcados representaram fidedignamente locais do mapa.

É possível verificar eventos curtos que podem corresponder à literatura, no entanto, por meio de um único experimento não seria possível chegar a um resultado tão significativo, apesar disso a capacidade de análise por meio de experimentos desse tipo se confirma. É claro que para replicar uma pesquisa completa da área de neurociências será necessário mais estudos.

O segundo caso de análise foi o ERP do evento de disparo do tiro, que se revelou discreto, mais com uma perceptível mudança de estados pós-eventos, principalmente entre os canais C5 e C6. Da mesma forma que no caso da análise

anterior, pode-se afirmar que resultados mais acentuados poderiam ser obtidos através da condução de mais experimentos.

Os resultados indicam o uso promissor do Unreal Tournament 2004 como ferramenta de estímulo em experimentos psicofisiológicos e a utilização do Pogamut/Neuro Game Logger na aquisição automatizada das informações da experiência de jogo.

## Capítulo 5

### Conclusão e trabalhos futuros

Segundo a contextualização feita no capítulo 2, existem inúmeras utilizações de jogos de tiro de primeira pessoa como estímulos complexos em pesquisas científicas que utilizam análises psicofisiológicas em humanos, dando à ferramenta proposta neste trabalho diversos contextos para sua utilização.

Apesar do contexto amplo da proposta, a conjuntura de validação - o Laboratório de Sono, Sonho e Memória do Instituto do Cérebro - forneceu uma perspectiva concreta de utilização desta ferramenta em casos de uso específicos de estudos de memória sono e sonhos. Algumas hipóteses apresentam sonhos como espécies de máquinas biológicas que criam cenários futuros baseados unicamente na experiência do passado, orientando as ações de vigília de modo a maximizar a aptidão. Este aspecto de previsão do futuro, ou mais exatamente de adivinhação futura, é provavelmente a explicação para a crença generalizada em sonho premonição entre sociedades antigas.

As diferentes abordagens quanto à função onírica, possuem indícios e argumentos fortes que intensificam a discussão, o que demonstra que não existe um entendimento exato do sonho ou de sua função, sendo um linha científica que ainda necessita de intensa investigação.

Este capítulo tem como primeiro objetivo pontuar como o ferramental desenvolvido neste trabalho pode ser aplicado nesse cenário de pesquisas em neurociências, apontando sua relevância para análises psicofisiológicas e de neurociências. Posteriormente a conclusão do capítulo trará as perspectivas futuras para esse trabalho.

A motivação da utilização de jogos de tiro em primeira pessoa como estímulo em pesquisas psicofisiológicas já foi tratada neste trabalho, todavia envolver o elemento do sono no experimento ressalta em certas características. Por exemplo, é sabido que o sono e o sonho consolidam a aprendizagem de habilidades sensório-motoras em uma tarefa na qual não se pode depender apenas da aprendizagem mecânica.

A utilização de jogos de FPS para o treinamento sensório-multimodal já se encontra bem presente na literatura, a fim de avaliar a consolidação de novas habilidades e memórias durante o sono, e foram utilizados em experimentos realizados por Brawn (2008) e os jogos Unreal Tournament 2004 e Quake II.

A capacidade do Neuro Game Logger registrar automaticamente os eventos diversos que ocorrem durante a jogatina e a indicação de cada ocorrência destes para o EEG, permite determinar com mais facilidade o padrão de onda de cada evento. Dessa maneira, o registro eletroencefalográfico gerado durante o sono permite ao pesquisador buscar padrões de onda similares aos encontrados durante a experiência de jogo.

Esse tipo de análise descrita em Wamsley (2010), que utilizou um modelo de estimulação visual ativa tal qual os Fliperamas, simulou a experiência de descer uma montanha esquiando como rotina pré-sono. Essa pesquisa visou entender a consolidação das memórias e do aprendizado durante o sono REM, chegando à conclusão de que há provas cognitivas fortes que as redes de memória são ativadas durante o sono.

A automatização proporcionada pela solução proposta neste trabalho se mostra promissora para esse tipo de análise, possibilitando investigações sobre a relação da atividade onírica com a realizada na vigília, pois sabe-se que o sonho é muitas vezes modulado por experiências que ocorrem durante a vigília (GACKENBACH e KURUVILLA 2008).

É possível oferecer uma experiência de vigília com estímulos marcantes que terão mais chances de influenciar e de serem percebidos na análise durante o sonho.

Isso pode permitir uma melhor elucidação sobre um possível papel que o sono e o sonho poderiam ter em relação aos *insights*. Existem relatos anedóticos que ligam o conteúdo dos sonhos com o contexto da vigília do sonhador, o ajudando na obtenção de *insights*. Podemos citar diversos exemplos importantes de insights históricos que ocorreram após sonhos: as descobertas científicas da estrutura circular do benzeno por Kekulé, a tabela periódica por Dmitri Mendeleev ou ainda a ideia da teoria da neurotransmissão química por Otto Loewi (MAZZARELLO, 2000).

Ribeiro e Pantoja (2009) relataram a utilização do jogo DOOM como protocolo de estímulo pré-sono para investigar os efeitos de aprendizado durante o sono, depois de jogar durante uma hora, antes de dormir, alguns dos participantes da pesquisa passaram

a ter sonhos com o tema do jogo, enquanto jogadores experientes apresentaram áreas frontais ativadas, os jogadores inexperientes apresentaram áreas motoras ativadas.

O desenvolvimento de *game designs* específicos com a ferramenta de edição do UT2004 permitiria desenvolver mapas com vários *puzzles*. Estes desafios mentais seriam uma ótima fonte de *insights* nos quais o padrão de onda pode ser mapeado durante a jogatina e ser comparado com a atividade cerebral durante o sono.

Outra linha no estudo dos sonhos seria o estudo de sonhos lúcidos, uma vez que há um paralelo entre o que acontece nos sonhos lúcidos e nos jogos digitais, pois ambos se caracterizam por simulações que representam realidades alternativas. Os jogadores dos jogos digitais controlam o ambiente do jogo, de modo que isso pode se traduzir em sonhos (GACKENBACH e KURUVILLA 2008).

Trabalhos relacionando as duas formas de simulação têm demonstrado que as pessoas que jogam jogos digitais frequentemente são mais propensas a ter sonhos lúcidos em que se veem fora de seus corpos físicos, além disso, são mais capazes de influenciar seus sonhos, como se estivessem controlando um ambiente de jogo.

Apesar de não serem experiências completamente idênticas – enquanto os jogos digitais são controlados por computadores e consoles, os sonhos surgem da mente humana – existe um potencial para o estudo relacionando essas duas áreas (GACKENBACH, 2006).

Uma contribuição que o Neuro Gamer Logger pode proporcionar às investigações relacionadas ao contexto espacial do ambiente de jogo seria a capacidade demonstrada na validação do mapa de calor da potência do teta. Montar um mapa de calor com as informações do arquivo de *log* é uma tarefa muito simples de grande valia em pesquisas utilizando labirintos virtuais e sono.

No trabalho de Wamsley (2010) foi utilizado um labirinto virtual em 3D, onde o participante deveria executar a tarefa de percorrê-lo antes e depois de efetuar um cochilo, sendo observado que o sonho tem um papel na melhoria da aprendizagem espacial e apresenta um melhor desempenho da memória pós-cochilo.

Em seus objetivos, esse trabalho estabelece como objetivo geral desenvolver um software para *logging* de eventos de jogos de tiro em primeira pessoa para uso em pesquisas em neurociência. Ainda na condição de dar suporte a sincronização temporal

com equipamentos de registro psicofisiológicos, o objetivo foi alcançado, pois a ferramenta cumpre essa função com determinadas limitações.

A primeira limitação é quanto ao atraso aleatório do envio causado pela camada de rede do Unreal Tournament 2004. A segunda limitação é a impossibilidade da própria simulação do jogo de gerar dados em escala de tempo menor que dezenas de milissegundos.

Quanto aos objetivos específicos, como a capacidade da ferramenta de detectar eventos específicos e a capacidade de ser facilmente legível foram satisfeitos, entretanto não significa que melhorias não possam ser realizadas, tais como implementar novos eventos ou melhorar a legibilidade do *log*.

O trabalho apresentado possui potencial de utilização em pesquisas na área das ciências cognitivas como a psicofisiologia e a neurociência cognitiva, que necessitam utilizar jogos de tiro em primeira pessoa como ferramenta, porque realizam avaliações espaciais, psicomotoras ou comportamentais. A percepção sobre esse potencial foi demonstrado não só pela experiência de validação, mas também por todo o conhecimento acumulado e demonstrado na síntese teórica deste trabalho.

Algo desafiador foi reunir o conhecimento de forma harmônica englobando as experiências de disciplinas tão diferentes em busca de atingir o desenvolvimento da ferramenta proposta. A conclusão deste trabalho se dá não só com a construção da ferramenta, mas com o fortalecimento de conexões acadêmicas entre as áreas de jogos digitais e das ciências cognitivas.

Além disso, foi estimulada a criação de uma ponte inusitada entre a área de desenvolvimento de agente inteligentes e da psicofisiologia, esse trabalho permitiu, assim, a utilização de agentes inteligentes em um novo caso de uso, possibilitando desenvolver *bots* com o objetivo de replicar comportamentos para avaliar como humanos podem reagir a eles.

As vantagens de avaliar o comportamento de um ser humano que se relaciona com *bots* é a eliminação de variáveis indesejáveis ao se comparar dois humanos como, por exemplo, nos níveis de habilidades diferentes ou ainda flutuações comportamentais por variáveis, que o pesquisador não tem controle. Esse novo modelo de caso de uso se mostra como uma contribuição acadêmica valiosa.

Algo que facilitaria sua adoção como plataforma de experimentação seria a formalização dos arquivos de *log* a fim de permitir uma análise que o pesquisador reduza o tratamentos das caracteres do arquivo, para tanto seria necessário um levantamento mais aprofundado de como tratar esses dados da melhor forma possível.

Ainda quanto aos *logs*, uma possibilidade de trabalho futuro seria que o Neuro Game Logger gerasse outros documentos para análise automática como, por exemplo, um mapa de calor da partida. Outra possibilidade seria adicionar um módulo de gravação de vídeo da partida para ser utilizado para análises auxiliares, pois da mesma forma que a ferramenta envia marcadores para o equipamento de EEG, ele poderia marcar o registro em vídeo associando aquele evento com evento ocorrido.

Algo importante detectado durante a validação foi à variância indesejada causada por um componente de software do jogo. Há várias soluções possíveis para isso, sendo desde implementar outra solução de conexão a rede para o jogo até solicitar acesso ao código fonte do componente em questão para poder tratá-lo.

Ainda seria possível adaptar a solução para a versão mais recente do Unreal Tournament, desenvolvida utilizando o motor Unreal Engine 4, ambos disponíveis em <https://github.com/EpicGames/Signup>. Mas para isso seria necessário desenvolver um componente similar ao Gamebots para essa nova versão do Unreal Tournament.

## Referências

ADIL, Khan; JIANG, Feng; LIU, Shaohui; *et al.* Training an Agent for FPS Doom Game using Visual Reinforcement Learning and VizDoom. **International Journal of Advanced Computer Science and Applications**: England, The Science and Information (SAI) Organization Limited. v. 8, n. 12, 2017. Disponível em: <http://thesai.org/Publications/ViewPaper?Volume=8&Issue=12&Code=ijacsa&SerialNo=5>. Acesso em: 4 fev. 2019.

AKYIL, K.; TUĞLU, H. E. **Design and implementation of a single-player first-person shooter game using XNA game development studio**. 2010. 97 f. Dissertação (Master of Science Thesis in the Department of Computer Science and Engineering) Chalmers University of Technology Department of Computer Science and Engineering Göteborg, Sweden, 2010.

ANDREASSI, J. L. **Psychophysiology: human behavior and physiological response**. 5. ed. New York, NY: Psychology Press, 2007.

ARAÚJO, Dráulio B. de; BAFFA, Oswaldo; WAKAI, Ronald T. Theta Oscillations and Human Navigation: A Magnetoencephalography Study. **Journal of Cognitive Neuroscience**: Cambridge, MA, MIT. v. 14, n. 1, 2002. p. 70–78.

ASERINSKY, E. & KLEITMAN, N.. Regularly Occurring Periods of Eye Motility, and Concomitant Phenomena, During Sleep. **Science**: Washington, DC, AAAS. vol. 118, n. 3062, 04 set. 1953. p. 273-274. DOI: 10.1126/science.118.3062.273.

BAKEMAN, R.; QUERA, V. ActSds and OdfSds: Programs for Converting INTERACT and The Observer Data Files into SDIS Timed-Event Sequential Data Files. **Behavior research methods**: Switzerland, Springer Nature. v. 40, n. 3, ago. 2008. p. 869–872.

BARKER, Alex. **Jnativehook**. Natal, 2018. Disponível em: <https://github.com/kwhat/jnativehook>. Acesso em: 31 jan. 2019.

BARRETT, D. **The committee of sleep: how artists, scientists, and athletes use dreams for creative problem-solving and how you can, too**. New York: Crown Publishers, 2001.

BRAWN, T. P. Consolidation of sensorimotor learning during sleep. **Learning e Memory**: Cold Spring Harbor, NY, CHS Press. v. 15, n. 11, 30 out. 2008. p. 815–819.

BIDA, M.; CERNY, M.; GEMROT, J. BROM, C. Evolution of GameBots project. *In*: Herrlich, M., Malaka, R., Masuch, M. (eds.) ICEC 2012. **LNCS**: Switzerland, Springer, vol. 7522, 2012. p. 397-400.

BISCHOF, Walter F.; BOULANGER, Pierre. Spatial Navigation in Virtual Reality Environments: An EEG Analysis. **CyberPsychology & Behavior**: USA, M-LIEBERT INC. v. 6, n. 5, 2003. p. 487–495.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **The unified modeling language user guide**. 2. ed. Upper Saddle River, New Jersey: Addison-Wesley, 2005.

BULKELEY, Kelly (Org.). **Dreams: a reader on religious, cultural, and psychological dimensions of dreaming**. New York: Palgrave, 2004.

BURNS, Alan; WELLINGS, Andy. **Real-time systems and programming languages: Ada, Real-Time Java and C/Real-Time POSIX**. 4. ed. Toronto: Pearson Education Canada, 2009. (International computer science series).

CACIOPPO, J. T.; TASSINARY, L. G.; BERNTSON, G. Chapter 1 - Psychophysiological science. In: **Handbook of psychophysiology**, Cambridge, United Kingdom; New York, NY, USA: Cambridge University Press, 2000.

CACIOPPO, John T.; TASSINARY, Louis G.; BERNTSON, Gary G. (ed.). **Handbook of Psychophysiology**. 3. ed. New York: Cambridge University Press, 2007.

CACIOPPO, John T.; TASSINARY, Louis G.; BERNTSON, Gary G. (Org.). Chapter 16 - Setting Up and Running an ERP Lab. In: **Handbook of psychophysiology**. 4. ed. Cambridge, United Kingdom; New York, NY, USA: Cambridge University Press, 2017.

CAVANAGH, J. F.; CASTELLANOS, J. Identification of canonical neural events during continuous gameplay of an 8-bit style video game. **NeuroImage**: USA: Elsevier, v. 133, jun. 2016. p. 1–13.

CHENIAUX, E. Os sonhos: Integrando as visões psicanalítica e neurocientífica. **Revista de Psiquiatria do Rio Grande do Sul** [online]: Porto Alegre, SPRGS. v. 28, n. 2, 2006. p. 169-177. ISSN 0101-8108.

CLEMENTS, Paul (org.). **Documenting software architectures: views and beyond**. 2. ed. Upper Saddle River, New Jersey: Addison-Wesley, 2011. (SEI series in software engineering).

DA MOTA GOMES, M.; QUINHONES, M. S.; ENGELHARDT, E. Neurofisiologia do sono e aspectos farmacoterapêuticos dos seus transtornos. **Revista brasileira de Neurologia**: Rio de Janeiro, UFRJ. v. 46, n. 1, 2010. p. 5–15.

DEMENT, W. & KLEITMAN, N. The Relation of Eye Movements During Sleep to Dream Activity: an Objective Method for the Study of Dreaming. **Journal of Experimental Psychology General**: Washington, DC: APA, v. 53, n. 5. 1957. p. 339-346. DOI:10.1037/h0048189.

DOMHOFF, G. W. **The Scientific Study of Dreams: Neural Networks, Cognitive Development and Content Analysis**. Washington: American Psychological Association Press. 2003.

EDWARDS, C. L. et al. Dreaming and insight. *In: Contrasting Dreaming and Wakefulness. **Frontiers in Psychology***: Lausanne, Switzerland, Frontiers Media. v. 4, 24 dez. 2013. p. 979. ISSN=1664-1078 DOI=10.3389/fpsyg.2013.00979.

ERIK, Fagerholt; MAGNUS, Lorentzon. **Göteborg**: Chalmers Tekniska Högskola, 2009.

FIELD, A. **Discovering Statistics Using SPSS**. 3. ed. London: SAGE Publications. 2009.

GACKENBACH, J. et al. Dream incorporation of video-game play as a function of interactivity and fidelity. **Dreaming**: Washington, DC, APA. v. 21, n. 1, 2011. p. 32–50.

GACKENBACH, J. & KURUVILLA, B. The Relationship Between Video Game Play and Threat Simulation Dreams. **Dreaming**: Washington, DC, APA. v. 18, n. 4, 2008. p. 236-256.

GAZZANIGA, M. S.; HEATHERTON, T. F. **Psychological science**. 2. ed. New York: W.W. Norton, 2006.

GEMROT, J. *et al.* Pogamut 3 can assist developers in building AI (not only) for their videogame agents. *In: **Agents for Games and Simulations***. [s.l.] Springer, 2009. p. 1–15.

GIBBONS, Phillip. **InpOutx64**. Natal, 2015. Disponível em: <http://www.highrez.co.uk/downloads/inpout32/>. Acesso em: 1 fev. 2019.

GLAVIN, Frank G.; MADDEN, Michael G. DRE-Bot: A Hierarchical First Person Shooter Bot Using Multiple Sarsa ( $\lambda$ ) Reinforcement Learners. **arXiv**: Ithaca, NY, Cornell University. v. 1. 13 jun. 2018. Disponível em: <http://arxiv.org/abs/1806.05106>. Acesso em: 4 fev. 2019.

HAAS, L F. Hans Berger (1873-1941), Richard Caton (1842-1926), and electroencephalography. **Journal of Neurology, Neurosurgery & Psychiatry**: London, UK, BMJ. v. 74, n. 1, 2003.

HALL, C. S. e VAN DE CASTLE, R. L. **The content analysis of dreams**. New York: Appleton-Century-Crofts, 1966.

HAZLETT, R. L. Measuring emotional valence during interactive experiences: Boys at video game play', *In: R. Grinter et al. (ed.) **SIGCHI Conference on Human Factors in Computing Systems (Proceedings)***: Montreal, Canada, 22–27 April, New York: ACM Press, 2006. p. 1023–26.

HEBB, D. O. **The Organization of Behavior**: a Neuropsychological Theory. Hoboken, NY: John Wiley & Sons, Inc. 1949.

- HERREWIJN, Laura; POELS, Karolien. The Effectiveness of In-Game Advertising: Examining the Influence of Ad Format. *In*: CAUBERGHE, VEROLIEN; HUDDERS (Orgs.). **Advances in Advertising Research IX**. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, p. 87–100. *E-book*. Disponível em: [http://link.springer.com/10.1007/978-3-658-22681-7\\_7](http://link.springer.com/10.1007/978-3-658-22681-7_7). Acesso em: 9 out. 2019.
- HILBERT, David M.; REDMILES, David F. Extracting usability information from user interface events. **ACM Computing Surveys**: ACM Digital Library, CSUR. v. 32, n. 4, 2000. p. 384–421.
- HINES, W.; MONTGOMERY, D.; GOLDSMAN, D.; BORROR, C. **Probability and statistics in engineering**. 4. ed. [s.l.]. Hoboken, NY: John Wiley & Sons, 2008.
- HOBSON, J. A.; HOFFMAN, S. A.; HELFAND, R. & KOSTNER, D. Dream bizarreness and the activation-synthesis hypothesis. **Human neurobiology**: Heidelberg, NY, Springer International. v. 6, n. 3, 1987. p. 157-164.
- HOBSON, J. A. & McCARLEY, R. W. The brain as a dream state generator: an activation-synthesis hypothesis of the dream process. **American Journal of Psychiatry**: Washington, DC, APA. v. 134, n. 12, 01 dez. 1977. p. 1335-1348.
- HOBSON, J. A. & PACE-SCHOTT, E. F. The cognitive neuroscience of sleep: neuronal systems, consciousness and learning. **Nature Reviews Neuroscience**: UK, Springer Nature Ltda. 3 set. 2002. p. 679-693.
- JUNG, C. G. **Dream Interpretation Ancient and Modern: Notes from the Seminar Given in 1936-1941**. Princeton: Princeton University Press, 2014. (Philemon Foundation Series).
- KAHANA, Michael J.; SEKULER, Robert; CAPLAN, Jeremy B.; *et al.* Human Theta oscillations exhibit task dependence during virtual maze navigation. **Nature**: UK, Springer Nature Ltda. v. 399, n. 6738, p. 781–784, 1999.
- KALLIO, K.; MÄYRÄ, F. and KAIPAINEN, K. At least nine ways to play: Approaching gamer mentalities. **Games and Culture**: New York, SAGE Publications. v. 6. n. 4, 2011. p. 327–53.
- KAMINKA, G. A. *et al.* Gamebots: a flexible test bed for multiagent team research. **Communications of the ACM**: ACM Digital Library, CSUR. v. 45, n. 1, p. 43–45, 2002.
- KARPOV, Igor V.; SCHRUM, Jacob; MIIKKULAINEN, Risto. Believable Bot Navigation via Playback of Human Traces. *In*: HINGSTON, Philip (org.). **Believable Bots**: Berlin, Springer Berlin Heidelberg, 2013, p. 151–170. Disponível em: [http://link.springer.com/10.1007/978-3-642-32323-2\\_6](http://link.springer.com/10.1007/978-3-642-32323-2_6). Acesso em: 4 fev. 2019.
- KEMPKA, Michał; WYDMUCH, Marek; RUNC, Grzegorz; *et al.* ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning. **arXiv**: Ithaca,

NY, Cornell University. v. 1. 20 set. 2016. Disponível em: <http://arxiv.org/abs/1605.02097>. Acesso em: 4 fev. 2019.

KLIMMT, C. Dimensions and determinants of the enjoyment of playing digital games: A three-level model. *In: Level Up: Digital Games Research Conference (Proceedings)*, Utrecht, Netherlands, 4–6 nov. 2003.

KIM, I. C.. UTBot: a virtual agent platform for teaching agent system design. **Journal of Multimedia**: London, UK, Hindawi. v. 2, n. 1, 2007. p. 48–53.

KIVIKANGAS, J. M. *et al.* A review of the use of psychophysiological methods in game research. **Journal of Gaming & Virtual Worlds**: Bristol, UK, Intellect. v. 3, n. 3, 13 set. 2011. p. 181–199.

KIVIKANGAS, M.; J.; NACKE, L.; RAVAJA, N. Developing a triangulation system for digital game events, observational video, and psychophysiological data to study emotional responses to a virtual character. **Entertainment Computing**: Amsterdam, the Netherlands, Elsevier. v. 2, n. 1, jan. 2011. p. 11–16.

KÖHLER, Wolfgang. **Blog da Psicologia da Educação**. p. 4, 2003. Disponível em: <https://www.ufrgs.br/psicoeduc/>. Acesso em: 4 fev. 2019.

KRUTCHEN, P. **The rational unified process**: an introduction. 3. ed. Boston: Addison-Wesley, 2004.

LEGA, Bradley C.; JACOBS, Joshua; KAHANA, Michael. Human hippocampal Theta oscillations and the formation of episodic memories. **Hippocampus**: New Jersey, NJ, Wiley Periodicals, Inc. v. 22, n. 4, 2012. p. 748–761.

LEDBERG, Anders; ROBBE, David. Locomotion-Related Oscillatory Body Movements at 6–12 Hz Modulate the Hippocampal Theta Rhythm. **PLoS ONE**: San Francisco, CA, PLOS. v. 6, n. 11, p. e27575. 2011.  
DOI.<https://doi.org/10.1371/journal.pone.0027575>

LUCK, S. J. **An introduction to the event-related potential technique**. 2. ed. Cambridge, Massachusetts: The MIT Press, 2014.

LUCAS, S. M. *et al* (ed.). **Artificial and Computational Intelligence in Games**. Wadern, Germany: Schloss Dagstuhl - Leibniz-Zentrum für Informatik GmbH, 2013. *E-book*. (121 p.).

MAYLOR, E. A.; LAVIE, N. The influence of perceptual load on age differences in selective attention. **Psychology and Aging**: Washington, DC, APA. v. 13, n. 4, dez. 1998. p. 563–573.

MANDRYK, R. L. and INKPEN, K. M. Physiological indicators for the evaluation of co-located collaborative play. *In: HERBSLEB, J. and OLSON, G. (ed.). Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work. CSCW 2004*. Chicago, Illinois, USA, November 6-10, 2004. New York: ACM Press, 2004.

MARKS, I. F. M.; NESSE, R. M. Fear and fitness: An evolutionary analysis of anxiety disorders. **Ethology and Sociobiology**: Amsterdam, the Netherlands, Elsevier. v. 15, n. 5–6, set. 1994. p. 247–261.

MAXWELL, S. E.; DELANEY, H. D. **Designing experiments and analyzing data: a model comparison perspective**. 2. ed. Mahwah, N.J: Lawrence Erlbaum Associates, 2004.

MAZZARELLO, P. What dreams may come? **Nature**: UK, Springer Nature Ltda. v. 408, n. 6812, nov. 2000. p. 523. DOI: 10.1038/35046170.

McCARLEY, R. W.. Neurobiology of REM and NREM sleep. **Sleep Medicine**: USA, Elsevier. v. 8, n. 4, 08 jun. 2007. p. 289-454. DOI: 10.1016/j.sleep.2007.03.005.

MCNAMARA, P.; BARTON, R.; NUNN, C. L. **Evolution of Sleep Phylogenetic and Functional Perspectives**. Leiden: Cambridge University Press, 2009.

MEDNICK, S. C.; NAKAYAMA, K. & STICKGOLD, R.. Sleep-dependent learning: A nap is as good as a night. **Nature Neuroscience**: UK, Springer Nature Ltda. v. 6, jul. 2003. p. 697-698. DOI: 10.1038/nm1078.

MENYCHTAS, Andreas; KYRIAZIS, Dimosthenis; TSERPES, Konstantinos. Real-time reconfiguration for guaranteeing QoS provisioning levels in Grid environments. **Future Generation Computer Systems**: USA, Elsevier. v. 25, n. 7, 2009. p. 779–784.

MICROSOFT. **Microsoft Hook**. Natal, 2018. Disponível em: <https://docs.microsoft.com/en-us/windows/desktop/winmsg/hooks>. Acesso em: 31 jan. 2019.

MORA, A. M. Evolving bot AI in Unreal™. *In*: Di Chio C. *et al.* (ed.). Applications of Evolutionary Computation. EvoApplications 2010. **Lecture Notes in Computer Science**: Springer, Berlin, Heidelberg. vol 6024, 2010. p. 171–180.

MULERT, C; Lemieux, L. (ed.). **EEG-fMRI: Physiological Basis, Technique and Applications**. Berlin, Alemanha: Springer, 2010

NACKE, L. E.; LINDLEY, C. A. Affective ludology, flow and immersion in a first-person shooter: Measurement of player experience. **arXiv**: Ithaca, NY, Cornell University. v. 1, 2010. DOI:1004.0248.

NACKE, L.; LINDLEY, C. A. Flow and immersion in first-person shooters: measuring the player's gameplay experience Proceedings of the 2008 Conference on Future Play: Research, Play, Share. **Anais...ACM**, 2008. Disponível em: <http://dl.acm.org/citation.cfm?id=1496998>. Acesso em: 19 ago. 2015.

NACKE, L.; LINDLEY, C.; STELLMACH, S. Log who's playing: psychophysiological game analysis made easy through event logging. *In*: Proceedings

of the 2nd International Conference on **Fun and games**. [s.l.] Springer-Verlag, Berlin, Heidelberg. 2008. p. 150–157. DOI: [org/10.1007/978-3-540-88322-7\\_15](https://doi.org/10.1007/978-3-540-88322-7_15).

NIELSEN, T. A., & Stenstrom, P. What Are the Memory Sources of Dreaming? **Nature**: UK, Springer Nature Ltda. v. 437, n. 7063, 27 out. 2005 1286-1289. DOI: [org/10.1038/nature04288](https://doi.org/10.1038/nature04288).

OSTERHOUT, L.; McLAUGHLIN, J.; KIM, A.; GREENWALD, R. e INOUE, K. Sentences in the brain: Event-related potentials as real-time reflections of sentence comprehension and language learning. *In*: CARREIRAS, M. e CLIFTON Jr, C. (ed.). **The on-line study of sentence comprehension: Eyetracking, ERP, and beyond**. Psychology Press, 2004. *E-book* (424 p.).

PAVLIDES, C. & WINSON, J. Influences of hippocampal place cell firing in the awake state on the activity of these cells during subsequent sleep episodes. **Journal of Neuroscience**: Washington, D.C, Society for Neuroscience. v. 9, n. 8, 01 ago. 1989. 2907-2918. DOI: <https://doi.org/10.1523/JNEUROSCI.09-08-02907.1989>.

PEARLMAN, C. & BECKER, M. REM sleep deprivation impairs bar-press acquisition in rats. **Physiology & Behavior**: Switzerland, Elsevier. v. 13, n. 6, dez. 1974. p. 813-817. DOI: [https://doi.org/10.1016/0031-9384\(74\)90267-4](https://doi.org/10.1016/0031-9384(74)90267-4).

PERACCHIA, Sara; TRIBERTI, Stefano; CURCIO, Giuseppe. Longer the Game, Better the Sleep: Intense Video Game Playing is Associated to Better Sleep Quality and Better Daytime Functioning. **ARCTT**: San Diego, CA, IMI, vol. 15, 2017. p. 4.

PETERSEN, K.; WOHLIN, C. The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study. **Empirical Software Engineering**: Switzerland, Springer Nature. v. 15, n. 6, p. 654–693, dez. 2010.

PIVIK, R. T. Tonic states and phasic events in relation to sleep mentation. *In*: ELLMAN, S. J.; ANTROBUS, J. S. (ed.). **The mind in sleep**: Psychology and psychophysiology. 2. ed. Oxford, England: John Wiley & Sons, 1991. p. 214–247. (Wiley series on personality processes.)

POLDRACK, R. The role of fMRI in cognitive neuroscience: where do we stand? **Current Opinions in Neurobiology**: Switzerland, Elsevier. v. 18, n. 2, abr. 2008. p. 223-227.

RAVAJA, Niklas; SAARI, Timo; SALMINEN, Mikko; *et al.* Phasic Emotional Reactions to Video Game Events: A Psychophysiological Investigation. **Media Psychology**: Oxfordshire, Taylor & Francis Group. v. 8, n. 4, 2006a. p. 343–367.

RAVAJA, Niklas; SAARI, Timo; KALLINEN, Kari; *et al.* The Role of Mood in the Processing of Media Messages From a Small Screen: Effects on Subjective and Physiological Responses. **Media Psychology**: Oxfordshire, Taylor & Francis Group. v. 8, n. 3, 2006b. p. 239–265.

RAVAJA, Niklas; SAARI, Timo; TURPEINEN, Marko; *et al.* Spatial Presence and Emotions during Video Game Playing: Does It Matter with Whom You Play? **Presence: Teleoperators and Virtual Environments**: Cambridge, MA, the MIT Press. v. 15, n. 4, 2006c. p. 381–392.

RAVAJA, Niklas; TURPEINEN, Marko; SAARI, Timo; *et al.* The psychophysiology of James Bond: Phasic emotional responses to violent video game events. **Emotion**: Washington, DC, APA. v. 8, n. 1, 2008. p. 114–120.

REVONSUO, A. The reinterpretation of dreams: An evolutionary hypothesis of the function of dreaming. **Behavioral and Brain Sciences**: Cambridge, UK, Cambridge University Press. v. 23, n.6, 27 fev. 2000. p. 85-110.

REVONSUO, A & Valli, K. How to test the threat-simulation theory. **Consciousness and Cognition**: Switzerland, Elsevier. v. 17, n. 4, 2008. p. 1292-1296.

REVONSUO, A. The reinterpretation of dreams: An evolutionary hypothesis of the function of dreaming. **Behavioral and Brain Sciences**: Cambridge, UK, Cambridge University Press. v. 23, n. 06, 2000. p. 877–901.

RIBEIRO, S. *et al.* Long-lasting novelty-induced neuronal reverberation during slow-wave sleep in multiple forebrain areas. **PLoS biology**: San Francisco, CA, PLOS. v. 2, n. 1, 20 jan. 2004. p. E24.

RIBEIRO, S.; GOYAL, V.; MELLO, C. & PAVLIDES, C. Brain Gene Expression During REM Sleep Depends on Prior Waking Experience. **Learning e Memory**: Cold Spring Harbor, NY, CHS Press. v. 06, 20 ago. 1999. p. 500-508.

RIBEIRO, S.; MELLO, C. V.; Velho, T.; GARDNER, T. J.; JARVIS, E. D. & PAVLIDES, C. Induction of Hippocampal Long-Term Potentiation during Waking Leads to Increased Extrahippocampal zif-268 Expression during Ensuing Rapid-Eye-Movement Sleep. **Journal of Neuroscience**: Washington, D.C, Society for Neuroscience. v. 22, n. 24, 15 dez. 2002. p. 10914-10923.

RIBEIRO, S. Sonho, memória e o reencontro de Freud com o cérebro. **Revista de Psiquiatria do Rio Grande do Sul [online]**: Porto Alegre, SPRGS. v. 25, n. 6, 2003. 59-63.

RIBEIRO, S. & NICOLELIS, M. A. L. Reverberation, storage, and postsynaptic propagation of memories during sleep. **Learning e Memory**: Cold Spring Harbor, NY, CHS Press. v. 11, n. 6, 2004. p. 686-696.

RIBEIRO, S. & NICOLELIS, M. A. L. 2006. The Evolution of Neural Systems for Sleep and Dreaming. **Evolution of Nervous Systems**: New York, Elsevier. v. 3, 451-464.

RIBEIRO, S.; SHI, X.; ENGELHARD, M.; ZHOU, Y.; ZHANG, H.; GERVASONI, D.; LIN, S.-C.; WADA, K.; LEMOS, N. A. M. & NICOLELIS, M. A. L. Novel

Experience Induces Persistent Sleep-Dependent Plasticity in the Cortex but not in the Hippocampus. **Frontiers in Neuroscience**: 1, 2007. p. 43-55.

RUGG, Michael D.; COLES, Michael G. H. **Electrophysiology of Mind**. [s.l.]: Oxford University Press, 1996. Disponível em:  
<http://www.oxfordscholarship.com/view/10.1093/acprof:oso/9780198524168.001.0001/acprof-9780198524168>. Acesso em: 21 out. 2019.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: a Modern Approach**. 3. ed. Pearson, 2009. *E-book*.

SALMINEN, M.; RAVAJA, N. Increased oscillatory theta activation evoked by violent digital game events. **Neuroscience Letters**: Switzerland, Elsevier. v. 435, n. 1, abr. 2008. p. 69–72.

SCOTT, Rafael Neia Barbosa. **Relação dos sonhos antecipatórios com desempenho cognitivo, afeto e comportamento da vigília**. 2015. 204 f. Tese (doutorado) – Universidade Federal do Rio Grande do Norte, 2015. Centro de Biociência. Programa de Pós-Graduação em Psicobiologia. p. 205. Disponível em:  
[https://repositorio.ufrn.br/jspui/bitstream/123456789/20577/1/RafaelNeiaBarbosaScott\\_TESE.pdf](https://repositorio.ufrn.br/jspui/bitstream/123456789/20577/1/RafaelNeiaBarbosaScott_TESE.pdf). Acesso em: 04 set. 2019.

SMITH, A. **Focus creating mods Unreal Tournament**. London: Pearson Education, 2002.

SMITH, C. & LAPP, L. Prolonged increases in both PS and number of REMS following a shuttle avoidance task. **Physiology & Behavior**: Switzerland, Elsevier. v. 36, n. 6, 1986. p. 1053-1057.

SMITH, C. & LAPP, L. Increases in number of REMS and REM density in humans following an intensive learning period. **Sleep**: Oxford, UK, Oxford University Press. v. 14, n. 4, jul. 1991. p. 325-330.

SMITH, C. & WONG, P. T. P. Paradoxical sleep increases predict successful learning in a complex operant task. **Behavioral Neuroscience**: Washington, DC: APA. v. 105. n. 2, 1991. p. 282-288. DOI:<https://doi.org/10.1037/0735-7044.105.2.282>.

SNYDER, C. R.; LOPEZ, Shane J. **Handbook of positive psychology**. New York: Oxford University Press, 2005. Disponível em:  
<http://public.eblib.com/choice/publicfullrecord.aspx?p=271581>. Acesso em: 9 out. 2019. *E-book*.

SKINNER, B.F.: **The Behavior of Organisms: An Experimental Analysis**. New York: D. AppletonCentury Company, incorporated, 1938.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

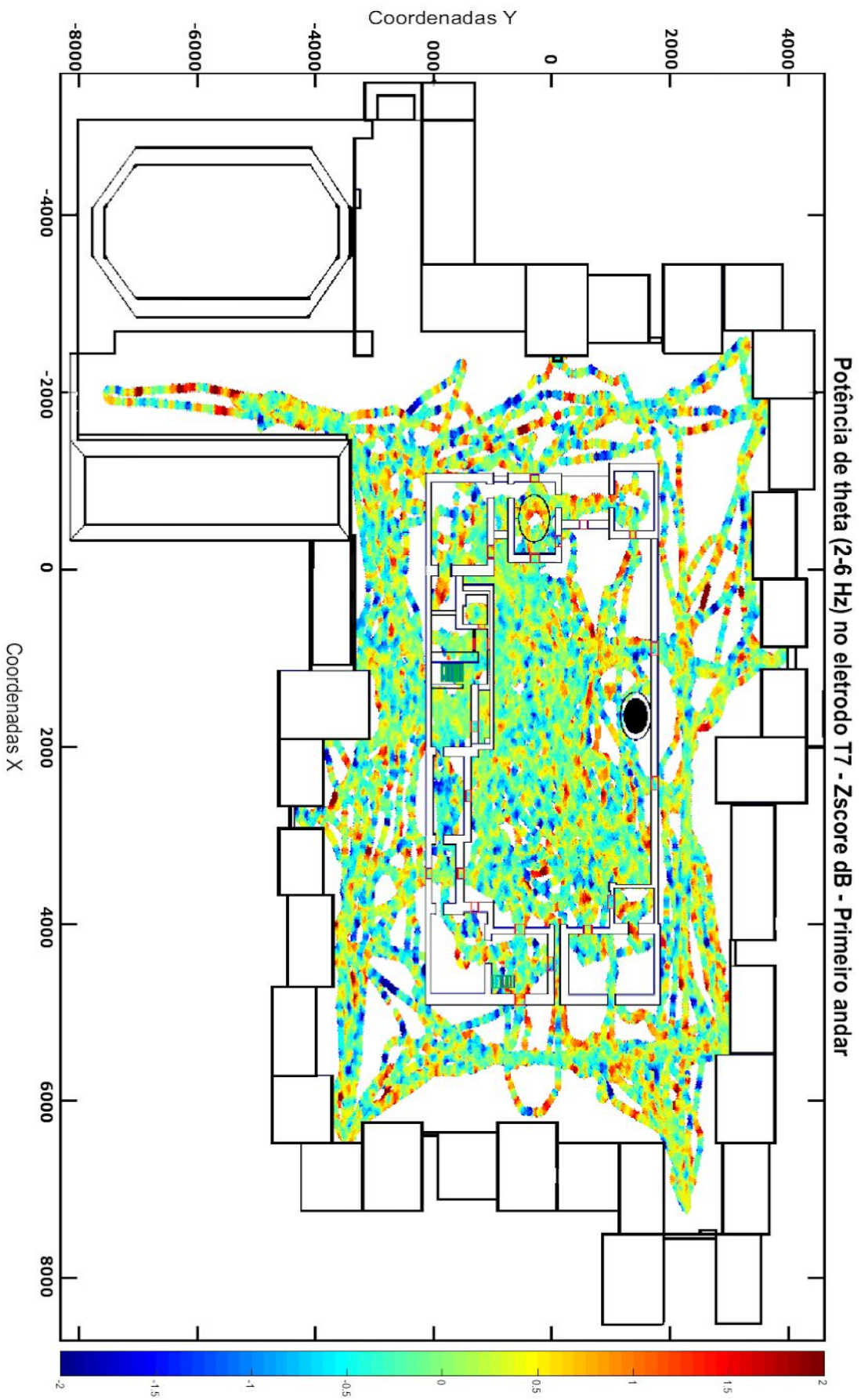
- SUR, Shravani; SINHA, V. Event-related potential: An overview. **Industrial Psychiatry Journal**: Alphen aan den Rijn, Netherlands, Wolters Kluwer v. 18, n. 1, 2009. p. 70.
- STICKGOLD, R.; JAMES, L. & HOBSON, J. A. Visual discrimination learning requires sleep after training. **Nature Neuroscience**: UK, Springer Nature Ltda. v. 3: n. 12, 2000a. p. 1237-1238.
- TERZIS, A. (2005). A explicação psicanalítica do mito e do sonho. **Revista da SPAGESP**: Ribeirão Preto, SPAGESP. v. 6, n. 2, 2005. p. 19-27.
- THIEL, G.; SCHOCH, S. & PETERSON, D. Regulation of synapsin I gene expression by the zinc finger transcription factor zif268/egr-1. **Journal of Biological Chemistry**: Rockville, MD, ASBMB. v. 269, n. 21, jun. 1994. p. 15294-15301.
- TUFIK, S. **Medicina e Biologia do Sono**. Barueri-SP: Manole, 2008. 483 p.
- VALLI, K.; REVONSUO, A.; PÄLKÄS, O.; ISMAIL, K. H.; Ali, K. J. & PUNAMÄKI, R. The threat simulation theory of the evolutionary function of dreaming: Evidence from dreams of traumatized children. **Consciousness and Cognition**: Switzerland, Elsevier. v. 14, n. 1, mar. 2005. p. 188-218. DOI: 10.1016/S1053-8100(03)00019-9.
- VizDOOM site. Disponível em: <<http://vizdoom.cs.put.edu.pl/research>>. Acesso em: 4 fev. 2019.
- WAGNER, U.; GAIS, S.; HAIDER, H.; VERGELER, R. & BORN, J. Sleep inspires insight. **Nature**: UK, Springer Nature Ltda. v. 427 jan. 2004. p. 352-355.
- WALKER, M. P. & Stickgold, R. Sleep, Memory, and Plasticity. **Annual Review of Psychology**: Palo Alto, CA. v. 57, 2006. p. 139-166.
- WAMSLEY, E. J. et al. Cognitive replay of visuomotor learning at sleep onset: temporal dynamics and relationship to task performance. **Sleep**: Oxford, UK, Oxford University Press. v. 33, n. 1, 2010a. p. 59.
- WAMSLEY, E. J. *et al.* Dreaming of a Learning Task Is Associated with Enhanced Sleep-Dependent Memory Consolidation. **Current Biology**: Massachusetts, USA, Cell Press, v. 20, n. 9, maio 2010c. p. 850–855.
- WILSON, M. A. & McNAUGHTON, B. L. Reactivation of hippocampal ensemble memories during sleep. **Science**: Washington, DC, AAAS. v. 265, n. 5172, jul. 1994. p. 676-679.
- WINSON, J. Patterns of hippocampal theta rhythm in the freely moving rat. **Electroencephalography and Clinical Neurophysiology**: Switzerland, Elsevier. v. 36, out. 1974. p. 291-301. DOI:[https://doi.org/10.1016/0013-4694\(74\)90171-0](https://doi.org/10.1016/0013-4694(74)90171-0).

WU, S. *et al.* Playing a first-person shooter video game induces neuroplastic change. **Journal of cognitive neuroscience**: Cambridge, MA, the MIT Press. v. 24, n. 6, jan. 2012 p. 1286–1293.

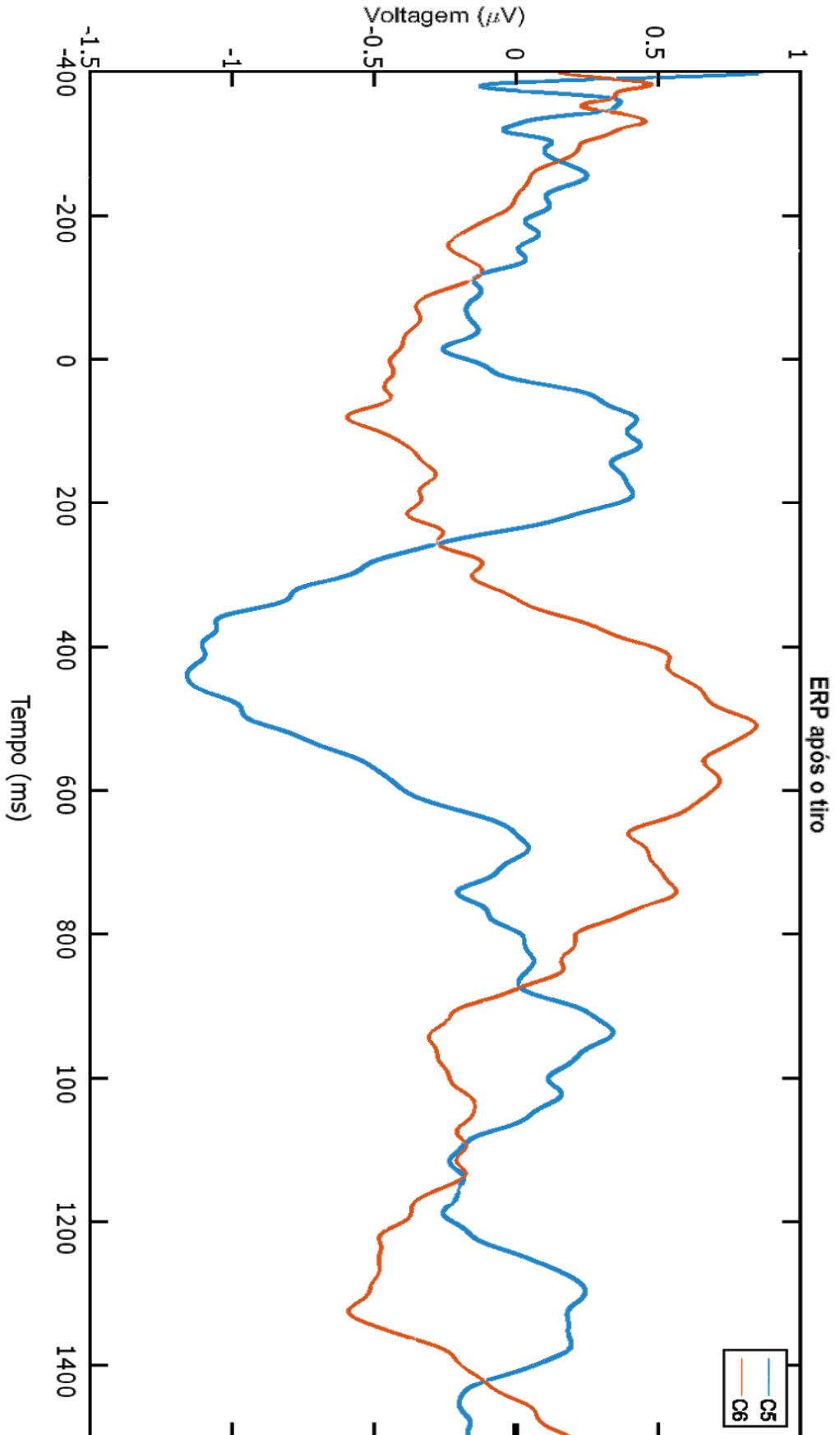
YANNAKAKIS, G. N.; TOGELIUS, J. A panorama of artificial and computational intelligence in games. **IEEE**: Digital Library, *IEEE Xplore*. v. 7, jan. 2014. p. 317-335. DOI: 10.1109/TCIAIG.2014.2339221.

ZIMBARDO, P. G., & RUCH, F. L. **Psychology and Life**. 9. ed. Glenview, IL: Scott, Foresman, 1975.

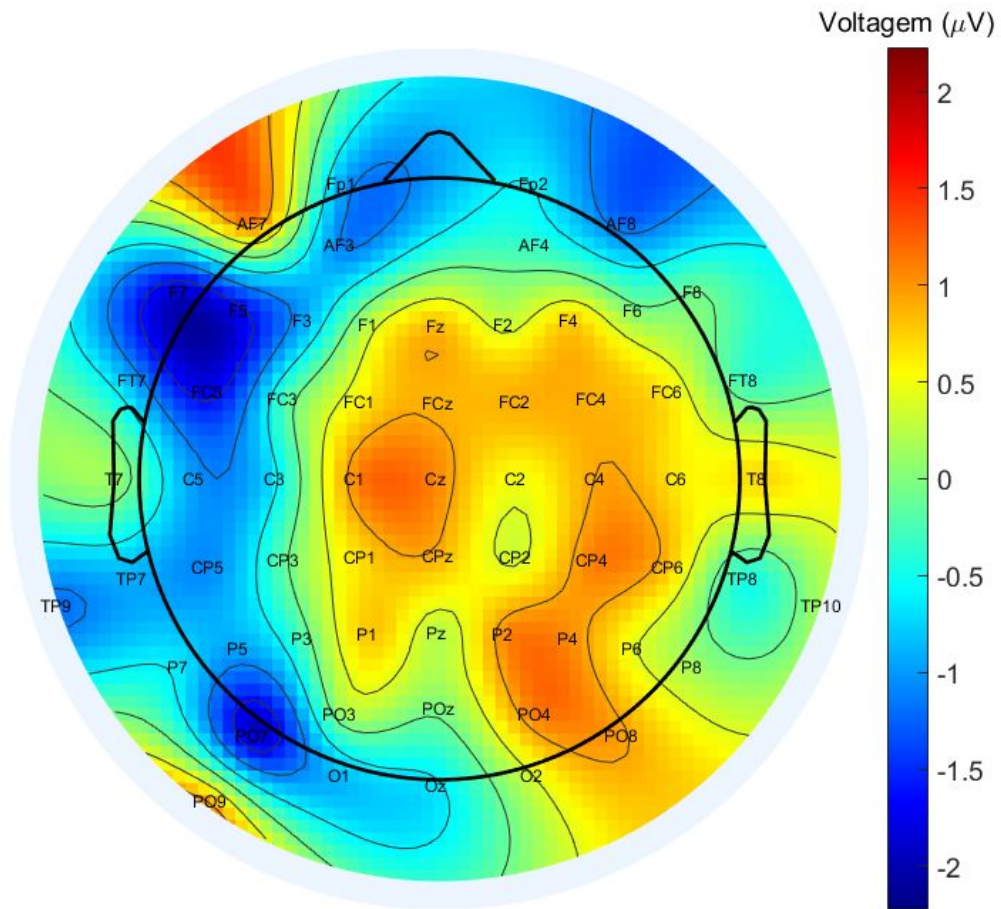
## Anexo 1 - Ampliação das Figuras dos resultados







## Topografia de 300-600 ms após o tiro





## Anexo 2

### Casos de Uso

Nesta seção são desenvolvidos os casos de uso, classificadores representando uma unidade funcional coerente provida pelo sistema, subsistema, ou classe manifestada por sequências de mensagens intercambiáveis entre os sistemas e um ou mais atores (SOMMERVILLE, 2007).

Casos de uso são narrativas em texto, descrevendo a unidade funcional, e são amplamente utilizados para descobrir e registrar requisitos funcionais nos sistemas. Os diagramas de casos de uso são representações dos casos de uso e podem ser representados por uma elipse contendo, internamente, o nome do caso de uso.

O apelo visual dessa ferramenta permite literalmente desenhar o processo de execução do negócio e visualizar a responsabilidade de cada participante, quando ele entrará em cena, qual será sua interação, a amplitude e a sequência em que o seu trabalho precisa ser realizado em relação às responsabilidades e tarefas dos demais integrantes do processo.

Casos de uso são tipicamente relacionados a “atores” (diagrama disponível nos anexos). Um ator é um humano ou entidade máquina que interage com o sistema para executar um trabalho significativo. É importante notar que não descreve como o software deverá ser construído, mas sim como ele deverá se comportar quando estiver pronto.

#### CSU01 – Iniciar experimento

##### **Descrição do caso de uso.**

O usuário deve ser capaz de iniciar um experimento por meio de comando na interface gráfica. Além disso, ele poderá interromper ou finalizá-lo ainda por essa interface.

##### **Atores**

Usuário Experimentador

##### **Prioridade**

Crítico

##### **Pré-condições**

O software deve estar aberto e executando.

#### **Fluxo de eventos**

- Ao acionar a interface para finalizar o sistema volta ao estado que estava antes de começar o experimento.

#### **Fluxo principal**

- O Usuário aciona o comando para começar o experimento;
- O sistema deve mudar seu estado para em experimento;

#### **Fluxos alternativos**

- O sistema pode ser interrompido por diversas vezes que seja antes de ser finalizado, entretanto não pode ser interrompido antes do iniciado.

#### **Pós-Condições**

O sistema muda seu estado para em experimento

#### **Pontos de extensão**

**CSU02 Interromper experimento**

#### **Casos de uso incluídos**

**CSU03 Finalizar experimento**

**CSU04 Monitorar Jogo**

### **CSU02 – Interromper experimento**

#### **Descrição do caso de uso.**

O usuário deve ser capaz de interromper um experimento por meio de comando na interface gráfica. Desse modo o sistema muda o estado para pausado, mas o experimento não está finalizado e pode ser retomado acionando a opção novamente.

#### **Atores**

Usuário Experimentador

#### **Prioridade**

Média

#### **Pré-condições**

O software deve estar aberto e executando e no estado de “em experimento”.

#### **Fluxo principal**

- O Usuário aciona o comando “pausar” o experimento;

- O sistema deve mudar seu estado para pausado;

#### **Fluxos alternativos**

- Se o sistema já está em estado pausado ele voltará para o estado em experimento.

#### **Pós-Condições**

O sistema muda seu estado para pausado.

### **CSU03 – Finalizar experimento**

#### **Descrição do caso de uso.**

O usuário deve ser capaz de iniciar um experimento por meio de comando na interface gráfica. Além disso, ele poderá interromper ou finalizá-lo ainda por essa interface.

#### **Atores**

Usuário Experimentador

#### **Prioridade**

Crítico

#### **Pré-condições**

O software deve estar aberto e executando, em estado em experimentação.

#### **Fluxo de eventos**

- Ao acionar a interface para finalizar o sistema volta ao estado que estava antes de começar o experimento.

#### **Fluxo principal**

- O Usuário aciona o comando para começar o experimento;
- O sistema deve mudar seus estados para em experimento;

#### **Fluxos alternativos**

- O sistema pode ser interrompido por diversas vezes que seja antes de ser finalizado, entretanto não pode ser interrompido antes do iniciado.

#### **Pós-Condições**

O sistema muda seu estado para em experimento

#### **Pontos de extensão**

**CSU02 Interromper experimento**

**Casos de uso incluídos****CSU03 Finalizar experimento****CSU04 Monitorar Jogo****CSU04 – Monitorar Jogo****Descrição do caso de uso.**

O sistema deve ser capaz de monitorar durante um loop de execução do jogo o estado do mundo de jogo da instância do game.

**Atores**

Game

**Prioridade**

Crítico

**Pré-condições**

O software deve estar aberto e executando, em estado em experimentação.

**Fluxo principal**

- Em um intervalo determinado (pelo arquivo de configuração), durante a execução da partida, o jogo deve enviar para o sistema o estado atual do mundo de jogo.
- O sistema deve direcionar os valores para monitorar e direcionar esse estado do mundo para a geração de log

**Fluxos alternativos**

- O sistema deve verificar se aconteceu alguns dos eventos cadastrados, caso tenha acontecido haverá o desvio para o fluxo alternativo de resposta de eventos.

**Pontos de extensão****CSU06 Responder a Evento****Casos de uso incluídos****CSU05 Gerar Log experimento****CSU05 – Gerar Log****Descrição do caso de uso.**

O sistema deve gravar os dados oriundos do mundo de jogo em um arquivo de log.

**Prioridade**

Crítico

**Pré-condições**

O sistema deve ter monitorado os dados do estado de jogo, em estado em experimentação.

**Fluxo principal**

- O sistema ao receber os dados do estado do mundo do jogo, criará um arquivo referente aquele experimento, caso o arquivo já exista ele o abrirá para edição;
- O sistema deve deverá inserir no final do arquivo os dados mudar seus estados para em experimento;

**CSU06 – Responder a evento****Descrição do caso de uso.**

O sistema deverá verificar se alguns dos eventos marcados ocorreu, caso isso tenha acontecido ele deverá determinar o marcador correto para ser enviado para o equipamento de leitura psicofisiologia.

**Prioridade**

Crítico

**Pré-condições**

O sistema deve ter monitorado os dados do estado de jogo, em estado em experimentação.

**Fluxo principal**

- O sistema deve identificar a ocorrência de um evento;
- O sistema deve determinar qual evento e o código referente aquele evento;
- O sistema enviará enviar o código à porta

**Casos de uso incluídos**

**CSU07 Enviar Sinal a Porta Paralela**

## CSU07 – Enviar Sinal a Porta Paralela

### Descrição do caso de uso.

O sistema ser capaz de comunicar-se e enviar os códigos em formato binário para equipamentos psicofisiológicos, bem como controlar o meio de comunicação.

### Atores

EEG

### Prioridade

Crítico

### Pré-condições

O software deve estar aberto e executando, em estado em experimentação.

### Fluxo principal

- O sistema converte o código para um binário de igual valor;
- O sistema verifica se o canal da porta paralela está disponível;
- O sistema envia o sinal na porta.

