DESENVOLVIMENTO DE UM SISTEMA PARA INSERÇÃO, ANÁLISE E GERENCIAMENTO DE DADOS DE ELETROENCEFALOGRAMA EM AMBIENTES VIRTUAIS

DEVELOPMENT OF A SYSTEM FOR THE INSERTION, ANALYSIS AND MANAGEMENT OF ELECTROENCEPHALOGRAM DATA IN VIRTUAL ENVIRONMENTS

Leandro Paiva Andrade[*]
Thaís Gaudencio do Rego[**]

RESUMO

A realidade virtual é uma tecnologia que possibilita a ação integrada entre computadores e cérebro humano. A viabilidade de reproduzir situações reais de forma controlada e a possibilidade de simular e visualizar ações impossíveis de serem percebidas no mundo real fazem com que este tipo de tecnologia encontre aplicações em diversas áreas, como na saúde, educação e entretenimento. No campo da neurociência, a realidade virtual, combinada à interfaces cérebro-computador, tem se tornado uma importante ferramenta para o estudo e entendimento da atividade cerebral, além de contribuir grandemente para a reabilitação de funções cerebrais debilitadas. Neste sentido, o objetivo deste trabalho foi a criação de um sistema denominado “NeuroReality”, o qual possui um componente para o ambiente Unity e uma interface simples de gerenciamento, onde pesquisadores da área de neurociência podem inserir, visualizar e manipular dados de eletroencefalograma em um ambiente virtual de teste. O produto obtido é um sistema que disponibiliza ao usuário todas as suas funcionalidades em uma só interface, simples e intuitiva, onde é possível acessar os dados de eletroencefalograma e, simultaneamente, realizar ações em objetos contidos no ambiente virtual. O sistema também possui a capacidade de capturar e salvar os dados de eletroencefalograma em função do tempo, em forma de tabelas, e registrar o ambiente virtual ao qual o componente está associado, em forma de vídeo, para possíveis futuras análises.

Palavras-chave: Realidade virtual. Neurociência. Interface cérebro-computador.

ABSTRACT

Virtual reality is a technology that enables the integrated action between computers and human brain. The feasibility of reproducing real situations in a controlled way and the possibility to simulate and visualize actions impossible to be perceived in the real world, make this type of technology have applications in several areas, such as health, education and entertainment. In the field of neuroscience, virtual reality, combined with brain-computer interfaces, has become an important tool for the study and understanding of brain activity, as well as contributing to the rehabilitation of weakened brain functions. In this sense, the objective of this work was the creation of a system called "NeuroReality", which is composed by a component for the Unity engine and a simple

[*] Mestrando em Informática. Faculdade Federal da Paraíba (UFPB leandropiox@gmail.com
[**] Docente da Faculdade Federal da Paraíba (UFPB) e Orientadora desta pesquisa. gaudenciothais@gmail.com

management interface, where neuroscience researchers can insert, view and manipulate electroencephalography data in a virtual test environment. The obtained product is a system that provides the user all its functionality in a single interface, simple and intuitive, where it is possible to access electroencephalography data and simultaneously perform actions on objects contained in the virtual environment. The system also has the ability to capture and save electroencephalography data over time, in the form of tables, and record the virtual environment to which the component is associated, in the form of video, for possible future analysis.

Keywords: Virtual reality. Neuroscience. Brain-computer interface.

Introdução

A realidade virtual (RV) é uma tecnologia que possibilita a ação integrada de computadores e cérebro humano (STEUER, 1992; LATTA; OBERG, 1994). Aplicações baseadas em RV permitem aos usuários navegar e interagir em tempo real, em ambientes virtuais gerados por computador, usando canais multissensoriais (BURDEA; COIFFET, 2003; TORI et al., 2006).

A aplicação de realidade virtual em diversas áreas, como na saúde, educação e entretenimento, tem contribuído para que esta tecnologia evolua em termos de *hardware* e *software*, de modo a suprir as demandas específicas de todas estas áreas. A viabilidade de reproduzir situações reais de forma controlada, trazendo todos os benefícios proporcionados por tais situações, sem gerar riscos aos usuários, e a possibilidade de simular e visualizar ações impossíveis de serem percebidas no mundo real, são motivações para que a RV constitua uma área de interesse crescente (NUNES et al., 2011).

Na neurociência, a tecnologia de realidade virtual vem se destacando por oferecer possibilidades inovadoras para a identificação de estruturas cerebrais durante cirurgias e para a reabilitação de funções cerebrais debilitadas por traumas e/ou doenças (DA COSTA, 2004). Estudos realizados nos últimos anos abordam o desenvolvimento de sistemas a partir da combinação de interfaces cérebro-computador (do inglês, Brain Computer Interface - BCI) e realidade virtual, visando a neuro-reabilitação motora e cognitiva (VOURVOPOULOS et al., 2013), reabilitação do transtorno de déficit de atenção e hiperatividade (ROHANI et al., 2014), reabilitação de membros superiores de pacientes pós-AVC (ACHANCCARAY et al., 2017), entre outros.

Vários estudos (CRESCENTINI et al., 2016; REN et al., 2016; BAKA et al., 2018; TROMP et al., 2018; ZHANG et al., 2018) foram realizados recentemente, utilizando a realidade virtual junto à interfaces cérebro-computador, com a finalidade de monitorar e decodificar a atividade cerebral, por meio de sinais de EEG. No entanto, pesquisas abordando o desenvolvimento de sistemas genéricos que permitam, de forma simultânea, a inserção e a manipulação de dados de EEG em ambientes virtuais de testes, precisam ser mais exploradas.

Atualmente, existem várias técnicas utilizadas para medir e registrar a atividade cerebral de forma não invasiva. No entanto, a técnica de eletroencefalografia (EEG) é a mais comumente utilizada para aquisição de sinais para BCI, pois apresenta uma excelente resolução temporal, capaz de medir a atividade cerebral a cada milésimo de segundo (VALLABHANENI et al., 2005).

Portanto, este trabalho teve como principal objetivo desenvolver um sistema denominado "NeuroReality", constituído por um componente para o ambiente Unity e uma interface simples de gerenciamento, onde pesquisadores da área de neurociência podem inserir, visualizar e manipular dados de eletroencefalografia em um ambiente virtual de teste.

1 Desenvolvimento

1.1 Arquitetura

Processos de engenharia de software, tais como elicitação, especificação e validação de requisitos (PRESSMAN, 2010; SOMMERVILLE, 2015), foram utilizados a fim de definir a arquitetura do sistema e a forma de comunicação entre os componentes e ferramentas. Os dois componentes do sistema se comunicam através de uma conexão UDP, utilizando o padrão de Cliente-Servidor, enquanto os dados de eletroencefalograma são adquiridos e enviados através do OpenVibe, através de uma conexão utilizando um servidor analógico fornecido pela Virtual Reality Peripheral Network (VRPN), como pode ser visto na Figura 1.

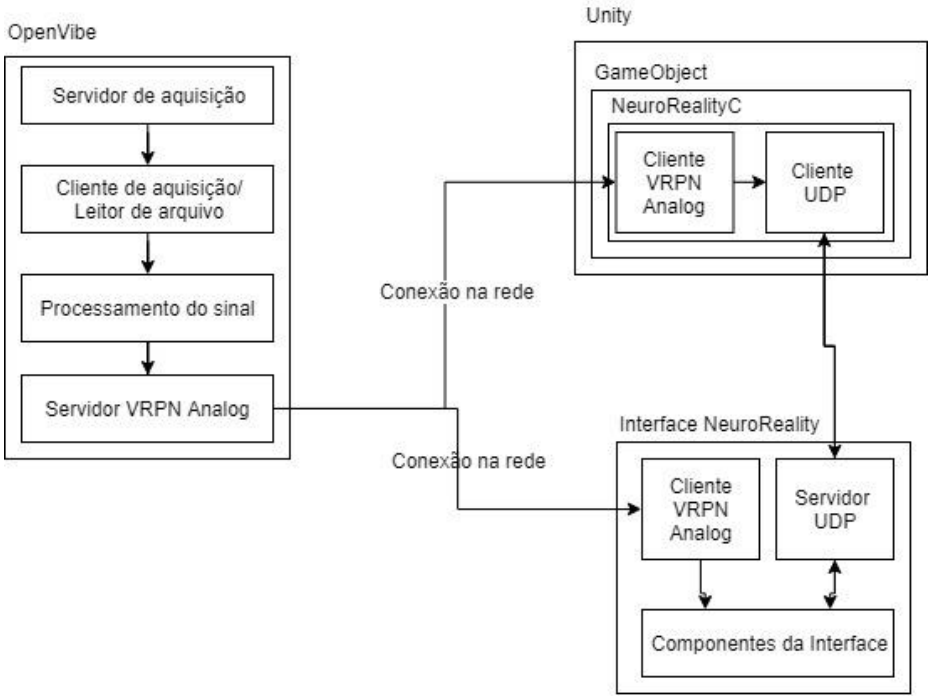

Figura 1 – Arquitetura do sistema
Fonte: o próprio autor

O OpenVibe é uma plataforma de *software* projetada para design, teste e uso de BCIs. Neste trabalho, ele foi utilizado especificamente para aquisição, processamento e envio dos dados de EEG para o componente do Unity. No entanto, é importante ressaltar que qualquer outro *software* BCI pode ser utilizado junto ao componente, visto que não existe dependência direta com o OpenVibe, sendo apenas necessária a criação de um servidor VRPN para o envio dos dados. O VRPN, por sua vez, foi escolhido por ser um sistema voltado para o contexto da realidade virtual, fornecendo uma interface unificada, independe de dispositivos, para acessar periféricos de realidade virtual (TAYLOR II et al., 2001).

1.2 Desenvolvimento do componente para Unity

Para o desenvolvimento deste componente, utilizou-se a linguagem de programação C#, o ambiente de desenvolvimento Visual Studio 2017, o motor de jogo Unity e uma versão empacotada do VRPN, denominada UVRPN. Algumas funcionalidades foram definidas para o componente, dentre elas: recebimento de dados via VRPN, recebimento e envio de dados via UDP, recebimento e envio de comandos para a interface, aplicação de transformações geométricas em objetos do ambiente,

captação e envio da lista de objetos existentes no ambiente, gravação do ambiente virtual em formato de vídeo.

O primeiro passo para o desenvolvimento foi a implementação da conexão através do VRPN. Para receber os dados enviados pelo OpenVibe, foi criado um cliente analógico VRPN, o qual possui três campos de configuração: *Host*, que é o gerenciador VRPN da classe VRPN_Manager; *Tracker*, que recebe o nome do servidor que está enviando os dados; e o *Channel*, utilizado para definir quais canais o cliente deve escutar. No caso de cliente analógico, sempre são escutados dois canais.

Para estabelecer a conexão entre o pacote Unity e a interface de controle foram utilizados o protocolo UDP e o modelo cliente-servidor, o qual foi implementado utilizando a linguagem C# e a classe *System.Net.Sockets*, nativa do Unity. Dentro da classe *System.Net.Sockets* existe uma subclasse, denominada *UdpClient*, que recebe como parâmetro um valor inteiro, referente a porta que vai ser utilizada na conexão. Esta subclasse foi utilizada para enviar e receber dados por meio da rede e através do protocolo UDP.

Para realizar as transformações geométricas nos objetos utilizando dados de EEG, fez-se a implementação de um método para ser utilizado em cada quadro da animação. Por meio deste método, quando um objeto é selecionado através da interface, o mesmo sofre uma transformação geométrica, como rotação, translação ou mudança de escala. Esta transformação é diretamente relacionada aos valores de EEG que são enviados ao componente, sendo apenas necessário realizar a normalização dos mesmos por meio de uma variável denominada "*Time.deltaTime*", a qual é relacionada ao tempo da animação. O dado recebido é multiplicado pelo valor desta variável, para que a transformação aconteça de forma contínua e suave, e posteriormente aplicado ao objeto.

Para enviar a lista dos nomes dos objetos contidos no ambiente virtual via UDP, inicialmente, foi necessário identificar todos os objetos existentes no cenário, extrair os seus identificadores e enviar os mesmos via UDP para a interface.

Para a gravação de vídeos apresentados no ambiente virtual, utilizou-se um pacote para Unity, denominado *VR Capture*. O *VR Capture* é capaz de capturar os quadros e o áudio que são apresentados no ambiente virtual, e os codifica utilizando o FFMPEG, que é uma coleção de bibliotecas e ferramentas para processar conteúdo multimídia, gerando assim um arquivo de vídeo.

Para utilizar o *VR Capture* dentro de um ambiente em Unity, fez-se o uso três *classes*: *VideoCapture*, *AudioCapture* e *VideoCaptureCtrl*. O *VideoCapture* tem como

função capturar os *frames* a serem utilizados para a criação do vídeo, o *AudioCapture* tem como finalidade captar o áudio da apresentação, e o *VideoCaptureCtrl* é responsável por gerenciar os dois anteriores e também o processo de codificação do vídeo. O vídeo resultante é salvo diretamente na pasta do projeto, no formato .mp4.

1.3 Desenvolvimento da Interface

Para o desenvolvimento da interface, utilizou-se a linguagem de programação Java, o ambiente de desenvolvimento NetBeans e a versão em Java da biblioteca VRPN. Algumas funcionalidades foram definidas para a interface, dentre elas: recebimento de dados via VRPN, apresentação de gráfico com dados de EEG em tempo real, recebimento e envio de dados via UDP, recebimento e envio de comandos para o componente, recebimento e apresentação da lista de objetos existentes no ambiente virtual, e capacidade de salvar dados de EEG em arquivo.

O VRPN utilizado no desenvolvimento da interface é uma versão empacotada, na linguagem Java. Os arquivos necessários para a utilização do VRPN foram criados através do código fonte, utilizando o Visual Studio, de forma que três arquivos foram gerados para serem utilizados na interface: *vrpn.jar*, *java_vrpn.dll* e *libjava_vrpn.so.* O primeiro arquivo foi adicionado diretamente nas configurações do projeto, enquanto os outros dois foram adicionados na pasta raiz.

O gráfico apresentado na interface foi desenvolvido com o auxílio da biblioteca *XChart*. Os dados recebidos via VRPN foram diretamente inseridos no gráfico, para a visualização em tempo real.

A conexão UDP da interface foi desenvolvida em Java. As principais classes utilizadas foram o *DatagramPacket*, que representa o pacote de dados a ser enviado ou recebido, e o *DatagramSocket*, que representa um soquete para enviar e receber pacotes de datagrama utilizando o protocolo UDP. Ambas são classes nativas do Java. Na parte da interface foi desenvolvido um cliente UDP que se comunica com o servidor presente no componente criado para Unity.

A interface, da mesma forma que o componente criado para Unity, é uma aplicação multitarefa. Desta forma, foi possível manter a interface e o cliente UDP ativos simultaneamente.

Ao final do processo de recebimento de dados via UDP, é preciso converter os dados recebidos em texto, a fim de descobrir qual comando está sendo enviado. A conversão de *bytes* foi feita utilizando a classe *String.*

Para a realização das tarefas, foram criados diversos comandos. Estes comandos foram enviados sozinhos ou como prefixos das mensagens utilizadas na conexão UDP, como pode ser visualizado na Tabela 1.

Tabela 1 - Tabela de comandos

Comando	**Descrição**
Download	Utilizado para requisitar os identificadores/nomes dos objetos contidos no ambiente virtual
stopDownload	Utilizado para informar que todos os identificadores/nomes dos objetos foram enviados
Vrpa	Prefixo para enviar o nome do servidor analógico VRPN, por exemplo: vrpnanalog0
Objn	Prefixo para enviar a identificação/nome do objeto selecionado
Sigv	Prefixo para enviar o valor mínimo que o sinal deve alcançar para aplicar uma transformação em um objeto
Path	Prefixo para enviar o local onde o vídeo deve ser salvo
Record	Comando para iniciar a gravação do ambiente de teste em formato de vídeo
stopRecord	Comando para a gravação de vídeo
rotate/rotateStop	Inicia/Cancela a rotação do objeto selecionado
translate/translateStop	Inicia/Cancela a translação do objeto selecionado
scale/scaleStop	Inicia/Cancela a mudança de escala do objeto selecionado
Connect	Comando para informar que a conexão UDP foi estabelecida
disconnect	Comando para informar que a conexão UDP foi finalizada

Fonte: o próprio autor

O processo de listagem dos objetos presentes no ambiente virtual está associado ao comando "*download*". Quando este comando é enviado da interface para o componente, o mesmo retorna a lista dos identificadores dos objetos em formato de texto, onde cada item é formado pelo comando *"objn"* e o seu nome ou identificador (por exemplo, *"objnsquare"*).

Para salvar os dados de EEG recebidos via VRPN utilizou-se a classe *FileWrite*, a qual é nativa do java e usada para escrever caracteres em arquivos. Ficou estabelecido

que devem ser salvos neste arquivo: os dados de EEG já processados (*"EEG Processed Signal"*) e o tempo a partir do início do envio do sinal (*"Time"*).

Para iniciar a gravação de vídeo no Unity, fez-se o envio do comando *StartCapture* ao componente, por meio da conexão UDP. Da mesma forma, para interromper a gravação, fez-se o envio do comando *StopCapture*. Os comandos *StartCapture* e *StopCapture* foram associados a botões que aparecem na interface.

2 Resultados

O NeuroReality é uma ferramenta que foi criada para o auxílio e controle de tarefas que utilizam EEG e Realidade Virtual. Esta ferramenta foi dividida em duas partes: um pacote para Unity e uma interface de controle. Cada parte foi construída em módulos que representam os requisitos do sistema.

Inicialmente, quando o usuário executa o arquivo da interface, uma imagem com o nome e logo da aplicação é exibida, como mostrado na Figura 2.

Figura 2 – Imagem de inicialização da interface
Fonte: o próprio autor

A Figura 3 apresenta a tela inicial da interface de controle, a qual aparece após a imagem de inicialização. Como pode ser observado, esta tela é dividida em quatro áreas: configurações, conexões, ambientes virtuais e sinal.

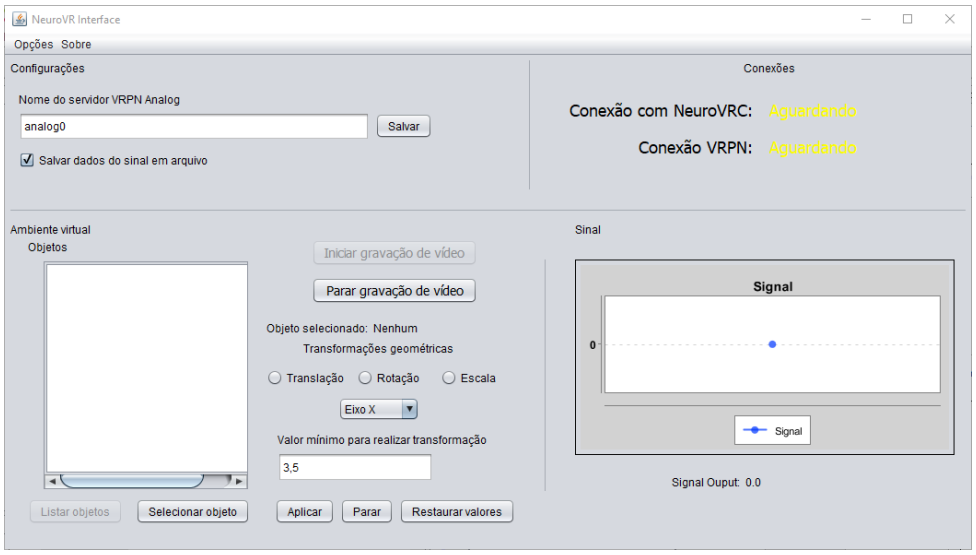

Figura 3 – Tela inicial da interface de controle
Fonte: NeuroVR Interface (2019)

A Figura 4 apresenta, mais especificamente, a área de configurações. Nesta área, localizada na parte superior esquerda da interface de controle, encontram-se os itens que podem ser configurados para o funcionamento da ferramenta e as opções de inicialização da mesma. O primeiro item é o nome do servidor *VRPN Analog* que está sendo utilizado para enviar os dados, sendo que o nome padrão adotado foi "*analog0"*, que é o mesmo do componente. Caso o usuário tenha preferência por utilizar outro nome de servidor, é necessário digitá-lo no campo de texto e pressionar o botão "Salvar". Vale ressaltar que esta configuração só pode ser efetuada se a conexão com o NeuroRealityC estiver correta.

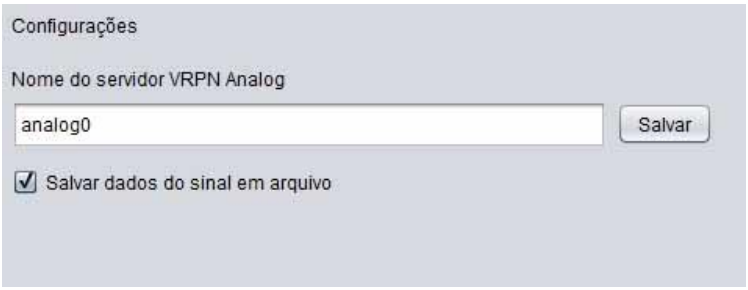

Figura 4 – Área da interface para configurações
Fonte: NeuroVR Interface (2019)

O segundo item que aparece na área de configurações é o *checkbox* para ativar ou desativar a opção de salvar os dados do sinal que estão sendo recebidos. Caso o usuário decida salvar os dados do sinal localmente, é necessário marcar esta opção antes de começar a receber o sinal. Neste caso, será criado um arquivo na mesma pasta em que se

encontra o executável da interface, e este terá o nome de "sinalDD-MM-YYYY HH-mm.csv", onde DD representa o dia em que o arquivo foi criado, MM o mês, YYYY o ano, HH a hora e mm os minutos. O arquivo consiste em duas colunas, sendo a primeira com os valores do sinal recebido pelo VRPN e a segunda com a data em que aquele dado foi recebido, no formato DD-MM-YYY HH:mm:ss.SSS, onde DD representa o dia, MM o mês, YYYY o ano, HH a hora, mm os minutos, ss os segundos e SSS os milisegundos. Na Figura 5, tem-se um exemplo de arquivo, com os valores do sinal recebido via VRPN e o momento do recebimento.

EEG Processed Signal	Time
-1.619347048	07-02-2019 01:05:28.146T
-1.399551708	07-02-2019 01:05:28.348T
-1.260336227	07-02-2019 01:05:28.349T
-1.085933809	07-02-2019 01:05:28.450T
-0.968945891	07-02-2019 01:05:28.652T

Figura 5 – Exemplo de arquivo, contendo os valores do sinal recebido via VRPN e o momento do recebimento
Fonte: o próprio autor

A Figura 6 apresenta a área das conexões (Conexão com NeuroRealityC e Conexão VRPN), a qual localiza-se na parte superior direita da interface. As conexões possuem quatro estados: "Erro", significa que houve algum erro ao iniciar as configurações de conexão da interface; "Aguardando", significa que os módulos de conexão com o NeuroRealityC e/ou VRPN foram inicializados com sucesso e estão aguardando uma conexão; "Conectado", significa que a conexão com o NeuroRealityC e/ou com o VRPN foi realizada com sucesso; e "Desconectado", significa que a conexão foi interrompida. Como pode ser visto na Figura 7, os estados de "Erro" e "Desconectado" são apresentados na cor vermelha, o estado "Aguardando" é apresentado em amarelo e o estado "Conectado", em verde.

Figura 6 – Área da interface para conexões
Fonte: NeuroVR Interface (2019)

Figura 7 – Os quatro estados das conexões
Fonte: NeuroVR Interface (2019)

A Figura 8 apresenta a área onde se encontram todas as funcionalidades relacionadas ao ambiente virtual. Esta área localiza-se na parte inferior esquerda da interface. Após a conexão com o NeuroRealityC, o botão "Listar Objetos" é habilitado e, ao clicar sobre ele, o usuário visualiza uma lista contendo todos os objetos que existem no ambiente virtual onde o NeuroRealityC foi adicionado. Esta lista é apresentada no componente logo acima do botão. Posteriormente, o usuário tem a opção de escolher um objeto para manipular utilizando os dados do sinal. Para isto, ele deve clicar no objeto de interesse e, em seguida, no botão "Selecionar Objeto".

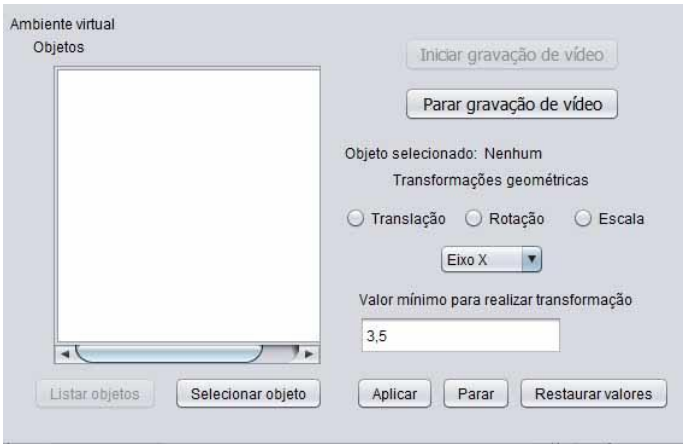

Figura 8 – Área onde se encontram todas as funcionalidades relacionadas ao ambiente virtual
Fonte: NeuroVR Interface (2019)

A Figura 9 apresenta: (a) a lista de objetos vazia e o botão "Listar Objetos" desabilitado, e (b) a lista de objetos recebidos ao clicar sobre o botão "Listar Objetos", após o mesmo ter sido habilitado.

(a) (b)

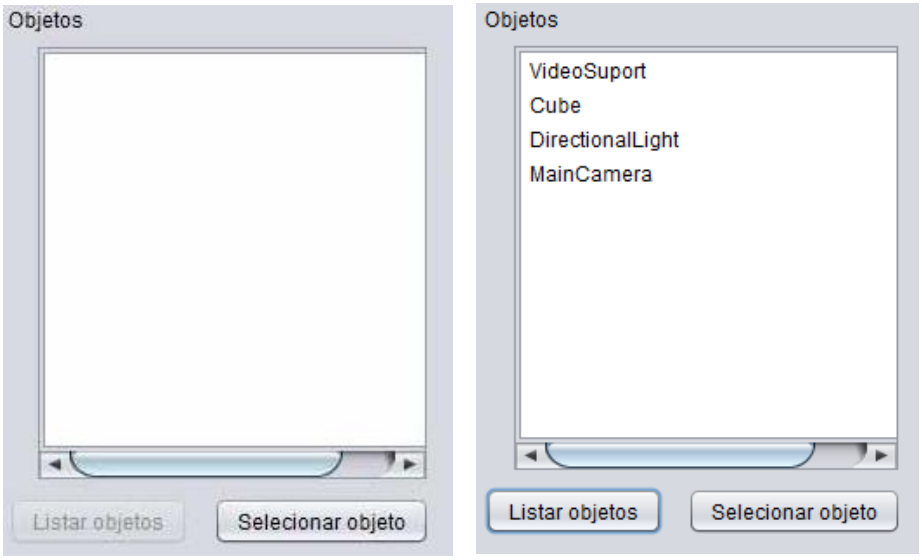

Figura 9 – (a) Lista de objetos vazia e o botão "Listar Objetos" desabilitado; (b) Lista de objetos recebidos ao clicar sobre o botão "Listar Objetos", após o mesmo ter sido habilitado
Fonte: NeuroVR Interface (2019)

Na Figura 10, tem-se as funcionalidades utilizadas para manipular um objeto no ambiente virtual, a partir dos dados provenientes do sinal de entrada. Após selecionar o objeto de interesse, como foi descrito anteriormente, o componente de texto "Objeto Selecionado: Nenhum" sofre uma atualização com o nome deste objeto e passa a apresentar-se como "Objeto Selecionado: Nome". Em seguida, a interface fornece três tipos de transformações geométricas: "Translação", que ocorre quando o objeto é deslocado sobre um dos eixos; "Rotação", que rotaciona o objeto em um dos eixos; e "Escala", que corresponde à mudança de tamanho do objeto. Nesta seção da interface, o usuário possui a opção de aplicar uma ou mais transformações em um objeto, caso o mesmo seja selecionado anteriormente. Junto à transformação, também é necessário especificar o eixo em que a mesma deve ser aplicada. Por padrão, o eixo selecionado é o X. Após estas etapas, é necessário informar o valor mínimo que o sinal de entrada deve alcançar para que a transformação aconteça no objeto, sendo assim, quando o sinal for igual ou maior ao valor, a transformação acontecerá de forma contínua. A Figura 11 apresenta os itens desta seção.

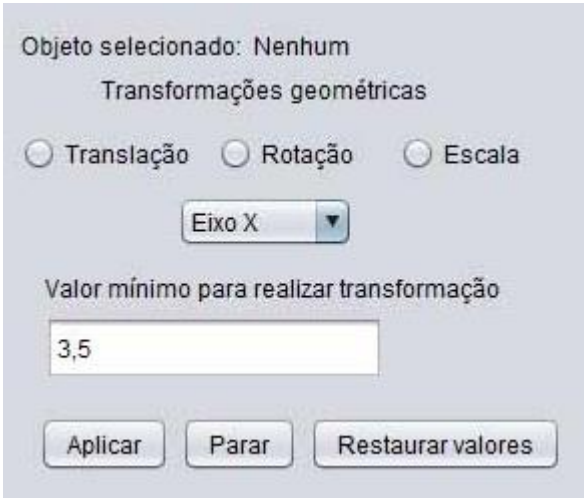

Figura 10 – Itens da interface relacionados à manipulação de objetos no ambiente virtual
Fonte: NeuroVR Interface (2019)

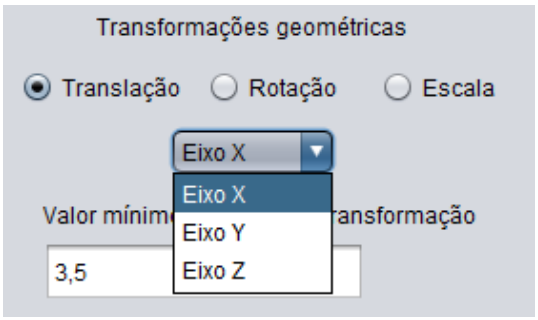

Figura 11 – Escolha do eixo X, para um movimento de translação
Fonte: NeuroVR Interface (2019)

Na Figura 12, tem-se as opções para iniciar e parar a gravação de vídeo do ambiente virtual. Da mesma forma que o botão "Listar Objetos", o botão "Iniciar gravação de vídeo" fica desabilitado enquanto não houver conexão com o NeuroRealityC. Quando a conexão é estabelecida, o botão é habilitado e, ao clicar sobre ele, inicia-se a gravação do ambiente virtual. Para que o vídeo seja devidamente gravado, deve-se clicar no botão "Parar gravação de vídeo" antes que a conexão com o NeuroRealityC seja interrompida.

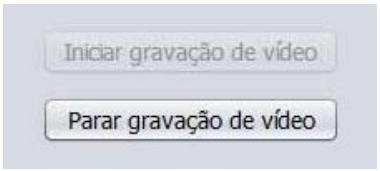

Figura 12 – Botões de iniciar e parar gravação de video do ambiente virtual
Fonte: NeuroVR Interface (2019)

O vídeo resultante é salvo no formato mp4, na mesma pasta que se encontra o arquivo executável da interface. A data com horas, minutos e segundos do momento em que deu-se início a gravação do video é utilizada para formar o nome do arquivo final. Além da data, também é adicionado no nome do arquivo, um ID único, criado de maneira aleatória.

A Figura 13 apresenta a seção associada ao recebimento do sinal via VRPN, a qual localiza-se no lado inferior direito da interface. Como pode ser observado, o sinal é plotado em forma de gráfico, onde o eixo X corresponde ao tempo e o eixo Y corresponde à intensidade do sinal. O gráfico é plotado em tempo real, de acordo com a chegada do sinal na interface. Adicionalmente, tem-se um componente de texto, localizado mais abaixo, responsável por mostrar o último valor de intensidade recebido.

(a) (b)

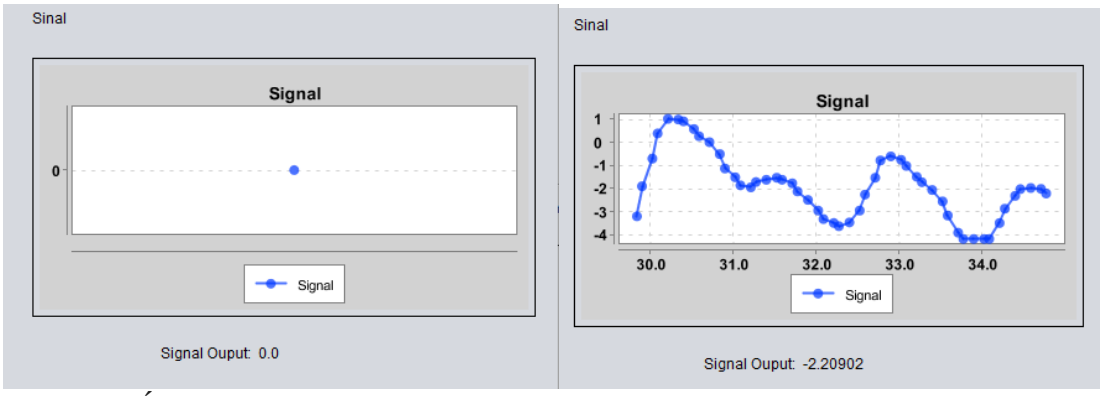

Figura 13– Área da interface associada ao recebimento do sinal via VRPN: (a) Antes do recebimento do sinal; (b) Plotagem do gráfico em tempo real, de acordo com o recebimento do sinal
Fonte: NeuroVR Interface (2019)

Em relação ao pacote desenvolvido para Unity, para sua utilização, é necessário adicioná-lo a ao projeto diretamente pela interface da aplicação Unity3D. Ao realizar a adição de um componente a um objeto da cena, uma nova janela é aberta (Figura 14), onde é possível ver todo o conteúdo que será importado ao projeto. A pasta "*Scripts*" possui os arquivos relacionados com a lógica do componente, enquanto as demais pastas são as dependências necessárias para o bom funcionamento do mesmo. Desta forma, com todas as pastas selecionadas, deve-se clicar no botão "*Import*".

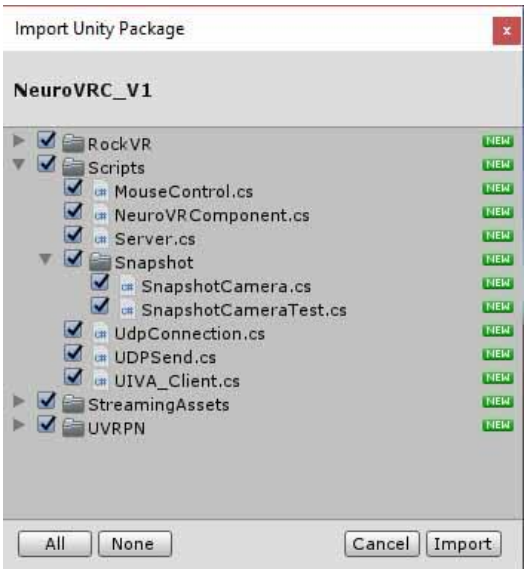

Figura 14 – Janela mostrando o conteúdo a ser importado
Fonte: Unity3D (2019)

Após a importação do componente, o menu de informações do objeto passa a possuir o item adicionado, como pode ser visto na Figura 15.

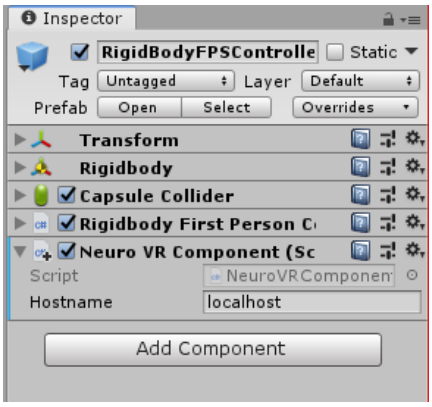

Figura 15 – Menu de informações do objeto após a adição do componente
Fonte: Unity3D (2019)

Considerações Finais

Este trabalho teve por objetivo desenvolver um sistema, onde pesquisadores da área de neurociência podem inserir, visualizar e manipular dados de eletroencefalografia em um ambiente virtual de teste. Esta é uma forma de contribuir com o bom andamento das pesquisas nesta área, tornando mais eficiente o processo de coleta e análise de dados.

Para trabalhos futuros, novos requisitos serão adicionados, como a junção dos dados de EEG e do vídeo de teste capturado, para facilitar o estudo e análise dos testes, e a captura de imagens dos objetos que existem no ambiente virtual, para facilitar a seleção.

Referências

ACHANCCARAY, D.; ACUÑA, K.; CARRANZA, E.; ANDREU-PEREZ, J. A virtual reality and brain computer interface system for upper limb rehabilitation of post stroke patients. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS. Naples, 2017, p. 1-5.

BAKA, E.; STAVROULIA, K. E.; THALMANN, N. M.; LANITIS, A. An EEG-based evaluation for comparing the sense of presence between virtual and physical environments. In: COMPUTER GRAPHICS INTERNATIONAL, **Proceedings…** Bintan Island, 2018, ACM, p. 107-116.

BURDEA, G.; COIFFET, P. **Virtual Reality Technology**. 2. ed. Nova York: John Wiley & Sons, 2003.

CRESCENTINI, C.; CHITTARO, L.; CAPURSO, V.; SIONI, R.; FABBRO, F. Psycological and physiological responses to stressful situations in immersive virtual reality: Differences between users who practice mindfulness meditation and controls. **Computers in Human Behavior**, v. 59, p. 304-316, 2016.

DA COSTA, R. M. E. M. A realidade virtual nas neurociências. In: KIRNER, C.; TORI, R. **Realidade Virtual: conceitos e tendências**. São Paulo: Romero Tori, 2004. p. 265.

LATTA, J. N.; OBERG, D. J. A conceptual virtual reality model. **IEEE Computer Graphics & Applications**, v. 14, n. 1, p. 23-29, 1994.

NUNES, F. L. S.; DA COSTA, R. M. E. M.; MACHADO, L. S.; DE MORAES, R. M. Realidade virtual para saúde no Brasil: conceitos, desafios e oportunidades. **Revista Brasileira de Engenharia Biomédica**, v. 27, n. 4, p. 243-258, 2011.

PRESSMAN, R. **Software engineering:** a practitioner's approach. 7. ed. Porto Alegre: McGraw-Hill, 2010.

REN, S.; BABILONI, F.; THAKOR, N. V.; BEZERIANOS, A. Real-Time Workload Assessment Using EEG Signals in Virtual Reality Environment. In: KYRIACOU, E.; CHRISTOFIDES, S.; PATTICHIS, C. In: MEDITERRANEAN CONFERENCE ON MEDICAL AND BIOLOGICAL ENGINEERING AND COMPUTING 2016, XIV., **IFMBE Proceedings…** Springer, 2016, v. 57.

ROHANI, D. A.; SORENSEN, H. B. D.; PUTHUSSERYPADY, S. Brain-computer interface using P300 and virtual reality: A gaming approach for treating ADHD. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY**,** 36th. Chicago, 2014, p. 3606-3609.

SOMMERVILLE, I. **Software Engineering**. 10. ed. Boston: Pearson, 2015.

STEUER, J. Defining virtual reality: Dimensions determining telepresence. **Journal of Communication**, v. 42, n. 4, p. 73-93, 1992.

TAYLOR II, R. M.; HUDSON, T. C.; SEEGER, A.; WEBER, H.; JULIANO, J.; HELSER, A. T. VRPN: A device-independent network-transparent VR peripheral system. In: ACM SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, **Proceedings…** Baniff, 2001, p. 55-61.

TORI, R.; KIRNER, C.; SISCOUTTO, R. **Fundamentos e tecnologia de realidade virtual aumentada**. Porto Alegre: SBC, 2006.

TROMP, J.; PEETERS, D.; MEYER, A. S.; HAGOORT, P. The combined use of virtual reality and EEG to study language processing in naturalistic environment**.** **Behavior Research Methods**, v. 50, p. 862-869, 2018.

VALLABHANENI, A.; WANG, T. Brain-computer interface. In: HE B. **Neural engineering**. Boston: Springer, 2005, p. 85-121.

VOURVOPOULOS, A.; FARIA, A. L.; CAMEIRÃO, M. S.; BADIA, S. B. RehabNet: A distributed architecture for motor and cognitive neuro-rehabilitation – Understanding the human brain through virtual environment interaction. In: INTERNATIONAL CONFERENCE ON E-HEALTH NETWORKING, APPLICATIONS AND SERVICES, 15th, Lisboa, 2013, p. 454-459.

ZHANG, F.; HU, M.; CHE, W.; LIN, H.; FANG, C. Framework for virtual cognitive experiment in virtual geographic environments. **ISPRS International Journal of Geo-Information**, v. 7, n. 1, p. 36, 2018.