

Framework para Controle e Debug de Simulation Sickness em Sistemas de Realidade Virtual

Vitor Balbio da Silva*

Alexandre da C. Sena

Celia Martins Cortez

Rosa Maria E. M. da Costa

Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística, Brasil

RESUMO

Apesar dos recentes avanços dos HMDs (*Head Mounted Displays*) de Realidade Virtual e o lançamento de diversos desses dispositivos no mercado internacional, muitos desafios ainda persistem para introduzir essa tecnologia ao público em geral, principalmente devido aos efeitos adversos que esses dispositivos podem causar como: desconforto, vertigem, fadiga visual e mais comumente náuseas, sintomas de um efeito conhecido como *Simulation Sickness*. Nesse contexto, o objetivo deste artigo é propor um *framework* que auxilie no desenvolvimento de aplicações em realidade virtual mais seguras e ergonômicas, características imprescindíveis para o uso em *games*. O *framework* disponibilizará para os desenvolvedores ferramentas de *debug* em tempo real do nível de Cinetose causada pela aplicação baseado em um modelo computacional da *Simulation Sickness*, assim como ferramentas de interação e locomoção de modo a mitigar os efeitos da Cinetose nessas aplicações, permitindo assim o uso dessa tecnologia com maior segurança e conforto em áreas como educação, entretenimento e jogos.

Palavras-chave: Realidade Virtual, Ergonomia, Segurança, Conforto Visual, *Simulation Sickness*.

1 INTRODUÇÃO

Os conhecimentos necessários para a construção de equipamentos de realidade virtual não são recentes, remontando à década de 40 do século passado. Mas, nos últimos anos, a redução dos custos de tecnologias, como Unidades Inerciais de Medição (IMU) e *displays* de alta densidade, além da possibilidade de utilização de *smartphones* como *displays* e unidades de processamento gráfico, tornaram possível o desenvolvimento e comercialização de HMDs (*Head Mounted Displays*) de realidade virtual para o consumidor doméstico. Esses fatos suscitaram em entusiastas a idealização do uso popular desses sistemas, que antes, só eram acessíveis para o meio acadêmico, aquecendo uma indústria que hoje prevê um faturamento de um bilhão de dólares, já em 2016 [8].

Apesar de todo apelo que essa tecnologia apresenta para o público, com a possibilidade de imersão em ambientes tridimensionais diferentes de qualquer outra tecnologia atual, ela ainda apresenta um grave problema em aberto. Esses dispositivos, em diversas situações, podem causar *Simulation Sickness*, que é um tipo de Cinetose causada por equipamentos de simulação que provoca náusea e, por vezes, pode inviabilizar totalmente a experiência de utilização desse tipo de dispositivo [2, 7].

Diversas são as propostas para reduzir os efeitos da *Simulation Sickness*, principalmente melhorias de *hardware*, como a implementação de *Positional Tracking*, *Displays* de baixa persistência, baixa latência e outros [9]. Porém, as perspectivas são de que somente melhorias de *hardware* não irão solucionar o problema, sendo necessário também propostas de implementações de

software que respeitem regras de boas práticas para conforto e ergonomia [11].

Este trabalho tem portanto como objetivo, apresentar um *Framework* que auxilia no desenvolvimento de sistemas de realidade virtual disponibilizando ferramentas de *Debug* e controle, de forma a mitigar os efeitos da *Simulation Sickness* nesses ambientes e aumentando portanto a segurança e o conforto na utilização dessas aplicações.

2 SIMULATION SICKNESS

Uma das teorias mais aceitas, representada na Figura 1, é a que descreve a *Simulation Sickness* como sendo a dissociação entre a informação sensorial que alcança o sistema nervoso central através das vias vestibular (A), visual (B) e proprioceptiva (C) [5, 9], ou seja, pela dissociação dos valores recebidos por diversos sensores. A via vestibular conduz informações sobre os movimentos da cabeça no espaço e pela proprioceptiva propagam-se sinais somestésicos gerados em músculos, tendões e articulações. O equilíbrio do corpo, seja parado ou em movimento, depende da coerência entre os dados oriundos das três vias sensitivas, que são processadas pelo sistema nervoso central [2, 3, 10].

É sugerido que a *Simulation Sickness* se apresente como uma forma específica de Cinetose (Motion Sickness) causada por equipamentos e dispositivos de simulação, sendo de uma variação conhecida como "Motion Sickness Visualmente Induzido" [5, 10]. Esse efeito foi e ainda é exhaustivamente estudado pela indústria aeroespacial [6], devido aos efeitos adversos que causa em simuladores de voo e, mais recentemente, como objeto de pesquisa também em dispositivos de realidade virtual, mais precisamente em HMDs [5].

Os sintomas da *Simulation Sickness* são diversos e incluem desconforto em geral, fadiga, dores de cabeça, dificuldade de concentração e, mais comumente, náuseas [6]. Encontrar meios de detectar e mitigar seus efeitos é fundamental para que desenvolvedores possam criar aplicativos de realidade virtual seguros e confortáveis para uso geral. De forma a viabilizar esse objetivo, faz-se necessário a formulação de uma modelagem computacional do efeito da Cinetose.

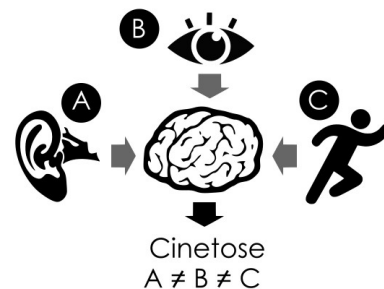


Figura 1: Cinetose causada pela dissociação sensorial das vias vestibular, visual e proprioceptiva no sistema nervoso central

*e-mail: vitorbalbio@gmail.com

3 UMA PROPOSTA DE MODELAGEM PARA *Simulation Sickness*

Nossa proposta de modelagem computacional da *Simulation Sickness* em HMDs de Realidade Virtual tem como objetivo principal disponibilizar ferramentas de *Debug* para que desenvolvedores de aplicações em realidade virtual possam estimar, em tempo de produção e em sessões de testes, com que intensidade, ao longo do tempo e também no instante atual, as suas aplicações estão produzindo as condições conhecidas que desencadeiam a Cinetose.

Atualmente a ferramenta mais popular para avaliação de *Simulation Sickness* é o questionário SSQ (*Simulation Sickness Questionnaire*) [6]. Apesar de sua importância reconhecida, a utilização de questionários para avaliação não permite a localização exata do momento do evento ou da ação que causou a Cinetose e nem a estimativa da variação de sua intensidade, sendo apenas avaliada a experiência geral de utilização. Desse modo, o uso do questionário torna inviável a tarefa de detectar os elementos causadores da Cinetose dentro das aplicações, sendo necessário uma outra abordagem. Este trabalho propõe a utilização de um modelo computacional do efeito da Cinetose que permita detectar na própria aplicação as condições que poderiam vir a causar Cinetose, bem como sua intensidade.

Para construção do modelo de Cinetose utilizou-se a definição mais aceita, apresentada na Seção 2, que afirma que a *Simulation Sickness* é causada pela dissociação dos sensores proprioceptivos, vestibular e visual [9]. Dessa forma, pode-se modelar a intensidade de Cinetose que um aplicativo provoca em um usuário como uma função da dissociação entre o que é exibido pelo *display* do HMD e o que é registrado pelo sistema vestibular do usuário como ilustrado na Figura 2.

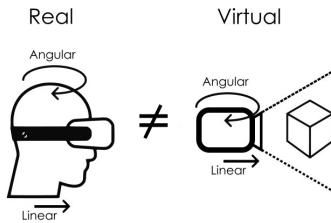


Figura 2: Cinetose e a dissociação entre os movimentos no Display do HMD e o registro vestibular

Tomamos como ponto central que a Cinetose é causada por disfunções relacionadas à movimento, principalmente aceleração, já que, nosso sistema vestibular não detecta a velocidade do corpo relativa a um sistema referencial qualquer e sim, a aceleração linear e angular no interior do osso temporal pela mudança da velocidade. A Cinetose associada à *Simulation Sickness* está portanto diretamente relacionada à mudança de aceleração vista pelo usuário no Display (Movimento este realizado pela câmera virtual) e a sua não correspondência com a estimulação do sistema vestibular.

Dessa forma, em nosso modelo, ignoramos as informações proprioceptivas e consideramos apenas duas variáveis relevantes: (1) o movimento da câmera virtual, que é um valor conhecido do sistema virtual e está diretamente relacionado ao que é visto com a aceleração nas imagens exibidas no HMD (aceleração visual); (2) a aceleração (linear e angular) do corpo do indivíduo (aceleração vestibular) que pode ser obtida através de instrumentos de medição inercial (IMU) como acelerômetros e giroscópios.

Considerando a intensidade da cinetose uma função de dissociação entre a aceleração visual e a aceleração vestibular adotamos a seguinte formalização:

$$D_{Lin} = \frac{|\vec{A}_{AcelLin} - \vec{U}_{AcelLin}|}{|\vec{M}_{MaxAcelLin}|}$$

$$D_{Ang} = \frac{|\vec{A}_{AcelAng} - \vec{U}_{AcelAng}|}{|\vec{M}_{MaxAcelAng}|}$$

$$Cinetose = D_{Lin} + D_{Ang} - 2VRLeg$$

Sendo D_{Lin} a dissociação de aceleração linear geral, $\vec{U}_{AcelLin}$ a do usuário e $\vec{A}_{AcelLin}$ a do avatar virtual. De forma análoga, D_{Ang} e seus termos respondem pela componente angular da dissociação. Para adequação matemática, os valores foram todos normalizados pelos termos $\vec{M}_{MaxAcelLin}$ e $\vec{M}_{MaxAcelAng}$ de forma que 0.0 seria a situação que se encontra o usuário ou avatar sem aceleração e 1.0 em aceleração máxima aceitável pelo sistema. A Cinetose portanto é calculada como o somatório das duas componentes subtraídas do termo $2VRLeg$.

O termo em inglês "VR Legs" é atribuído à Michael Abrash da Valve [4] e informalmente utilizado para designar a resistência biológica (que um usuário possui e potencialmente pode desenvolver com o tempo) aos efeitos da *Simulation Sickness*. Nosso termo $VRLeg$ necessita de uma adequação numérica para correção então multiplicamos pelo fator 2, sendo este termo também normalizado. Dessa forma é possível testar os efeitos da aplicação do modelo em diversos perfis de usuários, variando $VRLeg$ tal que: 0.5 seria um usuário comum; 0.0 um ultra sensível; e 1.0 um indivíduo com total resistência à *Simulation Sickness*.

Nesse ponto é importante notar que, com repetidas exposições à Cinetose nosso organismo cria adaptações neurológicas que, com o tempo, diminuem ou mesmo eliminam os efeitos da *Simulation Sickness* no organismo. Enquanto esse fato é um ponto positivo para a tecnologia, pois é possível "treinar" um indivíduo para utilizar as aplicações em realidade virtual, essa prática ainda encontra restrições éticas, por exemplo, é discutível se seria aceitável a exposição deliberada de crianças e idosos à esses ambientes para que propositalmente experienciem a *Simulation Sickness* visando a formação de suas "VR Legs". Essa prática que, sem uma análise aprofundada, nos parece nociva, tem se tornado comum atualmente para uma parcela de entusiastas de forma a permitir que fiquem engajados durante mais tempo em ambientes virtuais, principalmente em games [1].

Uma solução melhor parece ser a redução da Cinetose em si, com a disponibilização de ferramentas de *Debug* que permitam aos desenvolvedores detectar esses problemas em tempo de produção e com a disponibilização de ferramentas de interação, locomoção e outros que, seguindo boas práticas, reduzam os efeitos nocivos da *Simulation Sickness*. É com essa visão que propomos nosso *Framework*.

4 Framework PARA CONTROLE E *Debug* INTERATIVO DE *Simulation Sickness*

A partir do modelo de Cinetose apresentado na seção anterior, é proposto um *framework* para minimizar o problema através de duas abordagens:

- **VRF Tools:** Um conjunto de ferramentas de interação com objetos e locomoção que não provoque, ou provoque em menor grau, Cinetose, utilizando para isso regras de boas práticas definidas pelos últimos estudos de segurança e conforto da área.
- **VRF Debugger:** Uma ferramenta de *Debug* interativo que, através da modelagem proposta, disponibilize informações sobre o nível de Cinetose causada pela aplicação.

Conforme apresentado na Figura 3, o *framework* está sendo implementado com base em uma *Game Engine*. A Unity3D foi escolhida por ser atualmente a ferramenta de desenvolvimento de games

mais utilizada no mundo, correspondendo a 47% do mercado mundial [12]. Sua arquitetura foi estruturada de modo que possa ser utilizada de forma independente de sistema de renderização escolhido, assim, poderá vir a ser utilizado com os principais dispositivos no mercado como Oculus Rift, GearVR, SteamVR, Cardboard, Daydream e outros. Nossos testes estão sendo realizados na plataforma GoogleVR, utilizando a última versão disponível de seu SDK "GoogleVR 8.0" e o *headset* "BoboVR Z3", um similar plástico do Google Cardboard original. A Figura 3 apresenta a estrutura de sua arquitetura e a seguir serão apresentados os módulos principais VRF Tools e VRF Debugger.

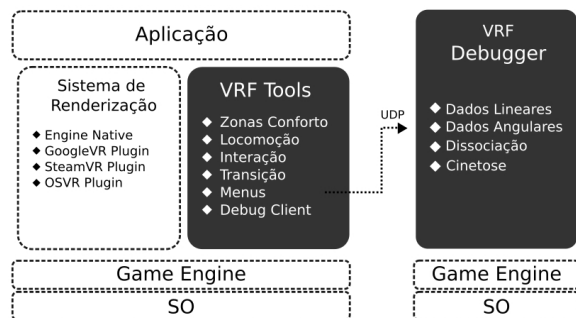


Figura 3: Arquitetura do sistema em relação a outros componentes de uma aplicação VR tradicional

4.1 VRF Tools - Ferramentas de Interatividade Utilizando Boas Práticas de Conforto e Ergonomia

Segundo a OculusVR apesar da *Simulation Sickness* estar associada muitas vezes à restrições de hardware como: baixa latência, falta de *positional tracking*, monitores com baixa persistência e outros, um importante componente está relacionado a implementações inapropriadas de software [11].

Grande parte das recomendações que são propostas, como o conjunto de boas práticas, advém de observações experimentais, realizadas pelos próprios desenvolvedores nos últimos anos, e que portanto ainda requerem um arcabouço teórico e técnico que comprovem suas presunções e observações. Apesar disso, essas recomendações tem obtido sucesso experimentalmente para uma grande parcela dos usuários, o que justifica a sua adoção em nosso trabalho, seguradas por essas devidas explicações. Desse modo, iremos nos ater à implementações baseadas nas afirmativas dispostas em guias de conforto e segurança das empresas que desenvolvem esses dispositivos, como a OculusVR, HTC e Google. A seguir apresentaremos cada módulo que compõe o sistema VRF Tools.

4.1.1 Visualização de Zonas de Conforto

Nossos olhos possuem zonas de conforto onde nos adaptamos melhor para tipos específicos de informação. Desrespeitar essas zonas de conforto em aplicações de realidade virtual podem levar rapidamente a um estresse visual o que impede uma experiência duradoura. Em nosso sistema desenvolvemos um visualizador em tempo real para zonas de conforto, dessa forma, desenvolvedores podem implementar elementos interativos a distâncias confortáveis durante a construção dos cenários. A Figura 4 apresenta como essa visualização se apresenta no editor da *game engine*.

4.1.2 Elementos Interativos

Atualmente existe uma grande variedade de dispositivos que podem ser utilizados como *input* em ambientes de realidade virtual: *gamepads*, controles de movimento, luvas, mouse, teclados entre outros. Ou seja, ainda está em aberto a criação de um padrão para interação nesses ambientes. Na busca de uma solução adequada, optamos

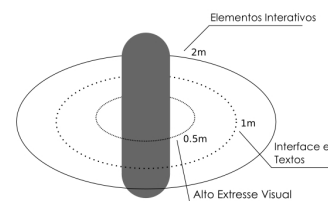


Figura 4: Visualização de Zonas de Conforto

por incluir em nosso sistema uma implementação de software que abarque o maior número de dispositivos atualmente disponíveis, o método escolhido foi o sistema de *Gaze Pointers*.

Gaze Pointers são sistemas que detectam a direção do olhar para o controle de um ponteiro virtual, nossa implementação utiliza o vetor que se projeta em Y, em relação à orientação da Câmera virtual e verifica se, em algum ponto, intercepta algum elemento interativo. Esse processo é conhecido como *Raycast*. O sistema de *Gaze Pointers* proposto é um aperfeiçoamento de propostas de como a GoogleVR SDK. Ele possui suporte para 2 modos de ativação: (1) por meio de um temporizador, também conhecidos como *Fuse Buttons* ou (2) pelo click de um dispositivo externo, ou seja, um botão do próprio HMD, um *gamepad*, *mouse* entre outras diversas opções. A Figura 5 apresenta o funcionamento do sistema de *Gaze Pointer*.

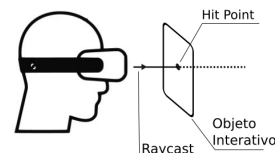


Figura 5: Método de Input por Gaze Pointer

A Tabela 1 foi elaborada para representar um comparativo entre os principais métodos de *Input* para VR e suas características conforme o suporte a múltiplas plataformas, custo, intuitividade de utilização e ergonomia. As estrelas representam os melhores classificados em cada categoria.

Método de Input	Plataformas	Custo	Intuitividade	Ergonomismo
Vive Controller	HTC VIVE	Alto	★ Alto	★ Alto
Daydream Controller	GoogleVR	N/D	★ Alto	Médio
Gamepads / Joysticks	★ Agnóstico	Baixo	Baixo	★ Alto
Mouse / Teclado	PC	Baixo	Baixo	Baixo
Gaze Pointer + Fuse	★ Agnóstico	★ Nenhum	★ Alto	Médio
Gaze Pointer + Click	★ Agnóstico	★ Nenhum	★ Alto	★ Alto

Tabela 1: Comparativo entre dispositivos e métodos de input para realidade virtual

4.1.3 Locomoção

Um dos grandes desafios da realidade virtual na área de *games* é desenvolver sistemas de locomoção confortáveis e ergonômicos. Em sessões onde o jogador está sentado e o personagem em movimento qualquer deslocamento do personagem claramente causará dissociação no sistema vestibular do jogador, levando então à Cinetose. Para buscar minimizar esses efeitos, nossa plataforma implementa um sistema de locomoção por "Teletransporte". Dessa forma o personagem virtual não se movimenta pelo ambiente, evitando portanto a dissociação sensorial causadora da Cinetose. Nossa implementação de Teletransporte é inspirada na demonstração "The Lab" da Valve [13] e suporta tanto o sistema de *Fuse* quanto de *Click*, como apresentado na Figura 6.

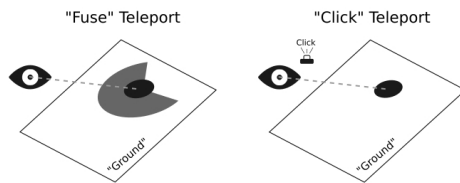


Figura 6: Modos de locomoção "Fuse Teleport" e "Click Teleport".

4.1.4 Menus Flutuantes

Nossa proposta para a implementação de menus é a criação de menus flutuantes posicionados na área de conforto do usuário e que somente são exibidos quando seu *Gaze Pointer* está apontado para um certo ângulo de ativação inferior ou superior. Botões e outros elementos virtuais são posicionados nesses menus e podem ser ativados por *Fuse Buttons* ou *Click*. A Figura 7 ilustra o funcionamento do sistema.

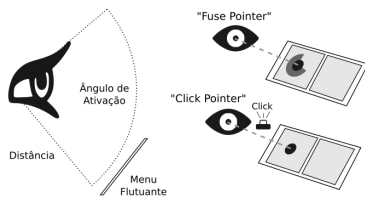


Figura 7: Menus Flutuantes ativados por "Click" ou "Fuse Pointers"

A seguir será apresentado o sistema de Debug Interativo que corresponde ao módulo VRF Debugger do sistema.

4.2 VRF Debugger - Ferramentas de Debug Interativo para Ergonomia e Conforto Visual

O processo de QA (*Quality Assurance*) e testes em games, tradicionalmente contam com uma bateria de testes feitos por profissionais (*Beta Testers*) e ferramentas que auxiliam na detecção de erros de código e/ou lógica (*Debuggers*). Para aplicações em realidade virtual o conforto do usuário necessita ser avaliado tanto quanto outros problemas de codificação, já que Cinetose e outros problemas fisiológicos podem inviabilizar a experiência do jogo.

O módulo de *debug* interativo proposto busca suprir essa necessidade fornecendo para os desenvolvedores uma estimativa do nível de Cinetose causado pelo sistema no usuário em tempo de desenvolvimento e testes. Essa ferramenta tem o objetivo de resolver as principais limitações dos testes de *Simulation Sickness* tradicionais como o SSQ, que não avaliam áreas específicas da aplicação e sim, a experiência como um todo, tornando difícil a identificação dos elementos causadores da Cinetose. O sistema proposto é baseado no modelo computacional da *Simulation Sickness* apresentado na Seção 3 e funciona como um aplicativo *Standalone* que se conecta por protocolo UDP à aplicação rodando no dispositivo local ou remoto recebendo, processando e apresentando as seguintes informações: Velocidade Angular, Velocidade Linear, Dissociação e Cinetose. Essas informações são apresentadas como gráficos na linha do tempo e através de vetores em um avatar reconstruído. Sua estrutura e conexões pode ser vistos na Figura 3.

A Figura 8 apresenta a interface proposta do sistema Debugger interativo. Ele é dividido em 3 regiões: (1) Uma seção de opções onde é possível selecionar opções de visualização para cada dado, (2) Uma área de exibição para gráficos na linha do tempo e (3) um ambiente 3D para visualização de vetores do personagem virtual.



Figura 8: Mockup do sistema de *debugger* interativo e suas regiões

5 CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou a proposta de um *framework* que tem como objetivo principal avaliar e mitigar os efeitos da *Simulation Sickness* em games e aplicações de realidade virtual. Para isso foram consideradas duas abordagens: (1) A detecção em tempo real do nível de Cinetose induzido pelo aplicativo e (2) a disponibilização de ferramentas ergonômicas de interação que diminuam os efeitos da *Simulation Sickness* nas aplicações. Resultados preliminares indicam que esta ferramenta pode aprimorar qualitativamente os processos de produção e *quality assurance*, permitindo desenvolver aplicações e games em realidade virtual mais confortáveis e seguros. Trabalhos futuros apresentarão a implementação do *framework* e uma avaliação mais detalhada de sua funcionalidade em um ambiente de produção.

REFERÊNCIAS

- [1] A. Aronsson. Simulator sickness and vr legs — a. aronsson's weblog. <http://andreasaronsson.com/2014/08/18/simulator-sickness-and-vr-legs/>, 08 2014. (Acessado em 05/07/2016).
- [2] Y. Fang, R. Nakashima, K. Matsumiya, I. Kuriki, and S. Shioiri. Eye-head coordination for visual cognitive processing. *PloS one*, 10(3):e0121035, 2015.
- [3] E. G. Freedman. Coordination of the eyes and head during visual orienting. *Experimental brain research*, 190(4):369–387, 2008.
- [4] T. Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Third Edition*. A K Peters/CRC Press, 2014.
- [5] P. Howarth and P. Costello. The occurrence of virtual simulation sickness symptoms when an HMD was used as a personal viewing system. *Displays*, 18(2):107–116, dec 1997.
- [6] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, jul 1993.
- [7] T. Kiryu and R. H. So. Sensation of presence and cybersickness in applications of virtual reality for advanced rehabilitation. *Journal of neuroengineering and rehabilitation*, 4(1):1, 2007.
- [8] P. Lee and D. Stewart. Virtual reality: a billion dolar niche. *TMT predictions* 2016, 2016.
- [9] G. Llorach, A. Evans, and J. Blat. Simulator sickness and presence using HMDs. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology - VRST*. Association for Computing Machinery (ACM), 2014.
- [10] S. Nishiike, S. Okazaki, H. Watanabe, H. Akizuki, T. Imai, A. Uno, T. Kitahara, A. Horii, N. Takeda, and H. Inohara. The effect of visual-vestibulosomatosensory conflict induced by virtual reality on postural stability in humans. *The Journal of Medical Investigation*, 60(3.4):236–239, 2013.
- [11] OculusVR. *Oculus Best Practices*. OculusVR, 310-30000-02 edition, 2016. (Acessado em 16/21/2016).
- [12] Unity. Unity - fast facts. <http://unity3d.com/pt/public-relations>, 2016. (Acessado em 05/21/2016).
- [13] Valve. The lab. <http://store.steampowered.com/app/450390/?l=portuguese>. (Acessado em 07/16/2016).