

UNIVERSIDADE FEDERAL DO PARANÁ

GUSTAVO HENRIQUE DA SILVA BARBOSA

JOGO EM REALIDADE VIRTUAL DE TREINAMENTO PARA COMPENSAÇÃO DE
PERDA DO CAMPO VISUAL PARA PACIENTES PÓS-AVC

CURITIBA PR

2024

GUSTAVO HENRIQUE DA SILVA BARBOSA

JOGO EM REALIDADE VIRTUAL DE TREINAMENTO PARA COMPENSAÇÃO DE
PERDA DO CAMPO VISUAL PARA PACIENTES PÓS-AVC

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2024

A alguém...

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, que me deu força e sabedoria para conseguir realizar esse projeto da melhor forma possível.

Aos meus familiares, sou profundamente grato pelo apoio incondicional que sempre me ofereceram, não apenas durante a realização deste trabalho, mas em todas as etapas da minha vida. Agradeço por serem meu alicerce e fonte constante de motivação.

Ao professor Todt, expresso minha sincera gratidão pela orientação e dedicação ao longo de todo o processo, cujos ensinamentos e conselhos foram fundamentais para a realização deste projeto. Estendo meus agradecimentos ao professor Renato e a toda a equipe da CETO, cuja participação ativa e contribuições foram essenciais para o desenvolvimento deste trabalho.

RESUMO

Anualmente, o Acidente Vascular Cerebral (AVC) afeta milhares de pessoas, sendo uma das principais causas de óbitos no Brasil. Entre os sobreviventes, muitos enfrentam sequelas permanentes, como a perda parcial do campo de visão, conhecida como hemianopsia. A estratégia de tratamento mais comum para essa condição, consistem em, por meio de diferentes exercícios, estimular o paciente a compensar de alguma forma essa perda do campo visual. Entretanto, assim como a maioria dos métodos de terapias tradicionais, esses exercícios podem ser pouco motivadores para os pacientes. Compreendendo os desafios da reabilitação voltada à compensação dessa perda visual, desenvolvemos o AeroVR, um jogo projetado para auxiliar na recuperação dos pacientes. O jogo integra dois conceitos fundamentais: a realidade virtual, que permite estimular uma ampla área do campo visual, e os jogos sérios, que promovem maior engajamento dos pacientes, incentivando o tratamento. Todo o projeto foi desenvolvido de forma iterativa em colaboração com a Clínica-Escola de Terapia Ocupacional da UFPR.

Palavras-chave: Realidade virtual. Jogos sérios. AVC. Hemianopsia. VR. App. Jogos. HMD

ABSTRACT

Annually, stroke (Cerebrovascular Accident or CVA) affect thousands of people, being one of the leading causes of death in Brazil. Among the survivors, many face permanent sequelae, such as partial loss of the visual field, known as hemianopia. The most common treatment strategy for this condition involves stimulating the patient, through various exercises, to compensate for this visual field loss. However, like most traditional therapy methods, these exercises can be demotivating for patients. Understanding the challenges of rehabilitation aimed at compensating for this visual loss, we developed AeroVR, a game designed to assist in patient recovery. The game integrates two fundamental concepts: virtual reality, which allows for the stimulation of a wide area of the visual field, and serious games, which promote greater patient engagement, encouraging treatment. The entire project was developed iteratively in collaboration with the Occupational Therapy School-Clinic of UFPR.

Keywords: Virtual Reality. Serious Games. Stroke. Hemianopsia. VR. App. Games. HMD

LISTA DE FIGURAS

1.1	Exemplo de óculos com prisma utilizado para o tratamento de um paciente com hemianopia no lado esquerdo. (Eye e Ear, 2013).	12
1.2	Exemplo de óculos VR com controle bluetooth. Reference: (Box, 2023)	14
2.1	Imagem do jogo <i>The Bradley Trainer</i> (AtariAge, 2024)	19
2.2	Google Cardboard (Google Developers, 2014).	22
2.3	Óculos de realidade virtual PlayStation VR2 (Sony, 2023)	22
2.4	Jogo de AR, Pokémon Go (Malik, 2016).	23
2.5	Imagem da campanha publicitária para o Meta Quest 3 (Meta, 2023)	23
2.6	Estudo realizado com monitores de 27". (Lidestam et al., 2019).	24
2.7	a) Estado do objeto (ativo ou desativado); b) Posição do objeto; c) Demais componentes	26
2.8	Tela principal do Blender, com modelo do avião	28
5.1	Cena do jogo <i>Blair Witch</i> (SA, 2019).	33
5.2	Cena do package <i>Shiratori's Low Poly Forest</i> (Shiratori, 2020)	34
5.3	Elementos de packages utilizados para compor o ambiente	34
5.4	Elementos modelados manualmente	35
5.5	Modelo do avião no software "Blender"	35
5.6	Ambiente apenas com as montanhas (em destaque) e o chão	36
5.7	Ambiente apenas com as vegetações	37
5.8	Ambiente final	37
5.9	Ambiente durante a partida	38
5.10	Óculos VR utilizado para o desenvolvimento do AeroVR	41
5.11	Avião com destaque para os movimentos possíveis	42
5.12	Controle bluetooth para óculos VR.	43
5.13	Diagrama de classes do AeroVR	44
5.14	Tela de menu principal	46
5.15	Tela para seleção do paciente	46
5.16	Tela de histórico de partidas paciente.	47
5.17	Tela para configurações gerais da partida	47
5.18	Tela para configurações das placas	48
5.19	Tela para configurações das posições das placas	48

6.1	Imagem capturada do jogo no modo VR	50
6.2	Grid configurado com os pesos tendendo a direita abaixo e os resultados correspondentes acima.	52
6.3	Grid configurado com os pesos apenas na parte central esquerda e direita e os resultados das medições acima	52

LISTA DE TABELAS

2.1	Comparação de motores de jogos	24
6.1	Especificações técnicas do smartphone	49
6.2	Medição do tempo de exibição das placas	51
6.3	Ciclos em que uma placa foi ou não exibida	51

LISTA DE ACRÔNIMOS

DINF	Departamento de Informática
PPGINF	Programa de Pós-Graduação em Informática
UFPR	Universidade Federal do Paraná
CETO	Clínica-Escola de Terapia Ocupacional da UFPR
VR	Virtual Reality
FoV	Field of vision
3D	Três dimensões
JSON	JavaScript Object Notation
min	Minutos
s	Segundos

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO.	13
1.2	OBJETIVOS	14
1.3	METODOLOGIA.	15
1.4	ESTRUTURA.	15
2	FUNDAMENTAÇÃO TEÓRICA.	17
2.1	ENGAJAMENTO DO JOGADOR	17
2.2	GAME DESIGN	18
2.3	JOGOS SÉRIOS	19
2.3.1	Gamificação e Jogos sérios	20
2.3.2	Game design para jogos sérios	20
2.4	REALIDADE VIRTUAL	20
2.4.1	Hardware para realidade virtual.	21
2.4.2	Realidade Virtual, realidade aumentada e realidade mista	21
2.5	MONITORES.	23
2.6	GAME ENGINES.	24
2.7	UNITY	25
2.7.1	GameObject.	25
2.7.2	Scripts.	26
2.7.3	Unity Asset Store	27
2.8	GOOGLE CARDBOARD XR PLUGIN	27
2.9	BLENDER	27
3	TRABALHOS RELACIONADOS	29
4	DESENVOLVIMENTO	30
4.1	CONTEXTO	30
4.2	ITERAÇÕES	30
4.2.1	Primeira versão	30
4.2.2	Segunda versão	31
4.2.3	Terceira versão	32
4.2.4	Quarta versão	32
5	PROJETO	33
5.1	AMBIENTE.	33
5.1.1	Composição	33

5.1.2	Música	35
5.1.3	Áreas do jogo	35
5.1.4	Geração do ambiente	36
5.2	CONFIGURAÇÕES	38
5.2.1	Configuração para múltiplos pacientes	38
5.2.2	Configurações do nível de dificuldade da partida.	39
5.2.3	Demais configurações.	40
5.3	MODOS DE JOGO	40
5.3.1	Controles	41
5.4	DADOS	42
5.4.1	Jogo	42
5.4.2	Dados coletados.	43
5.5	INTERFACE DO USUÁRIO	45
5.5.1	Menu principal	45
5.5.2	Seleção do paciente	46
5.5.3	Histórico	46
5.5.4	Configurações.	47
6	VALIDAÇÃO E TESTES	49
6.1	INFORMAÇÕES SOBRE OS TESTES	49
6.2	DISPLAY	49
6.3	CONFIGURAÇÕES	50
6.3.1	Tempo de exibição das placas.	50
6.3.2	Probabilidade de exibir uma placa	51
6.3.3	Distribuição das placas	51
7	CONCLUSÃO	53
	REFERÊNCIAS	54

1 INTRODUÇÃO

O Acidente Vascular Cerebral (AVC) é uma das doenças que mais geram óbitos no Brasil, sendo que apenas no ano de 2022 foram registrados 115.090 falecimentos gerados pelo AVC (M et al., 2023). Globalmente, cerca de 15 milhões de pessoas sofrem AVC anualmente, dessas 5 milhões vão a óbito e outros 5 milhões sobrevivem com sequelas (World Health Organization, 2024).

Apesar da alta taxa de mortalidade, é importante observarmos que muitas pessoas sobrevivem ao AVC, mesmo que para muitos a vida nunca volte a ser como antes. Um relatório da World Stroke Organization revela que, em 2022, havia mais de 100 milhões de sobreviventes de AVC em todo o mundo (World Stroke Organization (WSO), 2022). Muitos desses, após superarem o AVC, passaram a enfrentar o desafio de lidar com as sequelas dele, que em muitos casos podem ser permanentes.

O impacto dessa doença pode se dar de diferentes formas, e uma das áreas mais comumente afetada é a visão. Um estudo realizado na Universidade de Liverpool com pacientes de pós-AVC, mostrou que cerca de 92% deles tiveram a visão afetada, sendo que destes, 52% apresentaram perdas no campo visual (Rowe, 2017). Essa condição, conhecida como hemianopsia, pode se manifestar de diferentes maneiras e em diferentes graus, podendo chegar a comprometer até metade do campo visual do indivíduo.

As pessoas que se encontram nessa condição enfrentam desafios constantes e diários. Atividades simples, como caminhar, podem se tornar difíceis, pois a capacidade de evitar obstáculos diminui significativamente devido a problemas de visão. E atividades mais complexas, como dirigir, também são afetadas, aumentando o risco de acidentes e colocando em perigo o próprio indivíduo e os demais ao seu redor.

Atualmente, o tratamento para a perda do campo visual pode adotar diferentes estratégias (Pambakian et al., 2005). As três principais abordagens são: restauração, substituição e compensação (Lee, 2009). A terapia de restauração visual (*Visual Restoration Therapy*, VRT) visa restaurar o campo visual afetado por meio de estímulos visuais na área danificada (Kasten e Sabel, 1995). Embora disponível comercialmente nos EUA e aprovada pela FDA (NovaVision, 2024), sua eficácia é controversa, com estudos recentes questionando seus resultados (McFadzean, 2006). Já a estratégia de substituição, envolve o uso de equipamentos externos, como óculos com lentes especiais (ver Figura 1.1), para ampliar o campo visual do paciente (Rossi et al., 1990). Por fim, no método de compensação, por meio de diversos exercícios, é estimulado que o paciente consiga compensar a perda do campo visual de forma motora, seja adaptando o movimento ocular ou alterando a posição da cabeça (Nelles et al., 2001).



Figura 1.1: Exemplo de óculos com prisma utilizado para o tratamento de um paciente com hemianopsia no lado esquerdo. (Eye e Ear, 2013)

Ao escolher qual a metodologia utilizada, o terapeuta responsável irá levar em consideração diversos fatores como, por exemplo, a causa da hemianopsia (uma vez que esse distúrbio que pode ter diferentes causas) e o próprio estado do paciente. Entretanto, quando se trata de AVC, seguindo a linha de trabalho já aplicada na Clínica-Escola de Terapia Ocupacional da UFPR, a estratégia normalmente adotada é a compensação.

Uma vez que a estratégia foi decidida, assim como ocorre com muitos métodos de reabilitação tradicionais, engajar o paciente no tratamento passa a ser um dos grandes desafios desse processo. A motivação do paciente já é tido como um importante aspecto, tendo relação direta com os resultados obtidos (Maclean e Pound, 2000). Entretanto, terapia de reabilitação geralmente envolve exercícios repetitivos e pouco motivadores, tornando o tratamento ainda mais difícil para o paciente, que muitas vezes não se sente incentivado a continuar. Consequentemente, muitos pacientes acabam desistindo devido ao desinteresse ou à falta de percepção de melhorias em sua condição.

Analisando esse contexto atual, em que temos um problema real e que afeta milhares de pessoas no Brasil, aliada à grande necessidade de inovação nessa área, vemos uma oportunidade de desenvolver uma solução que busca complementar os métodos tradicionais, gerando uma maior eficiência à reabilitação, bem como um maior engajamento dos pacientes durante o tratamento. E isso se deve a duas tecnologias que estão sendo cada vez mais exploradas no âmbito da saúde e terapia ocupacional: realidade virtual e jogos sérios.

Com relação à realidade virtual, essa tecnologia está a cada dia se tornando uma opção mais acessível e comum no dia a dia. Atualmente existe uma grande gama de opções de equipamentos disponíveis para permitir o uso de VR, desde modelos de óculos VR mais simples, como o da figura 1.2 que deve ser utilizado com um smartphone, à modelos mais complexos que podem ser operados de forma independente.

Além disso, para o desenvolvimento proposto, uma grande vantagem que temos através desta tecnologia, é possibilidade de criar ambientes simulados como uma alta imersividade, uma vez que podemos estimular uma maior área do campo visual. Portanto, com um óculos VR simples, com o auxílio de um smartphone, podemos estimular uma grande amplitude do campo visual de um indivíduo de forma mais controlada, eficiente e barata.

Aliado à tecnologia de realidade virtual, a solução desenvolvida visa gerar maior engajamento dos pacientes. Pois, para uma reabilitação efetiva, não basta que a terapia seja boa, é necessário que a pessoa participante esteja motivada e ativa durante o processo (Maclean e Pound, 2000). Sendo assim, as técnicas de jogos sérios (*serious games*) são de fundamental importância.

O objetivo deste projeto é utilizar a tecnologia de realidade virtual, juntamente com técnicas de jogos sérios, para criar uma ferramenta que auxilie no tratamento de pessoas com alguma limitação no campo de visão ocasionado por um acidente vascular cerebral. Através do uso do óculos VR queremos exercitar uma maior área do campo visual do paciente. Já com o uso de técnicas de jogos sérios, temos como objetivo aumentar o interesse e engajamento do paciente ao tratamento, possibilitando, assim, uma maior taxa de sucesso do mesmo.

1.1 MOTIVAÇÃO

A principal motivação para este projeto vem justamente do grande potencial para o uso de novas ferramentas e tecnologias para a área da saúde, e mais especificamente, na terapia ocupacional. Para essa área em específico, os esforços para a inovação e melhores terapias com o uso de



Figura 1.2: Exemplo de óculos VR com controle bluetooth. Reference: (Box, 2023)

tecnologias têm começado apenas mais recentemente, portanto há uma grande demanda e necessidade de pesquisa no assunto.

Além disso, é importante frisar que, das ferramentas existentes, boa parte delas são proprietárias, pagas e pouco acessíveis à maioria das pessoas e instituições públicas brasileiras. Com esse projeto queremos não apenas criar uma ferramenta eficiente, mas também de código aberto e de baixo custo, facilitando assim o acesso à ferramenta, podendo trazer um grande benefício a diversas pessoas e profissionais da área da saúde.

1.2 OBJETIVOS

A proposta base é criar um jogo, chamado AeroVR. Os objetivos desse jogo são exercitar dois pontos principais:

- Percepção do usuário;
- Memória seletiva e compartilhada.

A mecânica do jogo pode ser simples: o usuário tem uma visão em terceira pessoa de um avião de papel, esse se desloca para frente e inicialmente pode se mover em duas direções (esquerda e direita) e conforme o ele se desloca são colocados obstáculos diante do jogador, dos quais ele terá que desviar. Juntamente com o avião, no campo de visão do jogador deverão ser exibidas algumas placas de trânsito, que ele terá que identificar durante o jogo.

Outros requisitos que o jogo deve cumprir são:

- O jogo deve estimular a percepção do jogador, de uma forma similar ao requerido para a condução de um veículo;

- Deve ser possível utilizar em um smartphone;
- Todos os dados devem ser salvos no celular, possuindo uma fácil visualização para o terapeuta;
- Deve possuir um modo que permita ser utilizado como um jogo em realidade virtual;
- No modo VR deve ser possível utilizar um controle bluetooth adequado;
- Além do modo VR, deve possuir um modo 3D tradicional;
- No modo 3D tradicional, deve ser possível espelhar a tela em um monitor maior e toda a interface de interação na partida deve se adaptar a isso.

1.3 METODOLOGIA

Todo o jogo deve ser desenvolvido de modo incremental, com os ajustes sendo realizados a partir da avaliação dos profissionais da Clínica-Escola de Terapia Ocupacional da UFPR (CETO).

Com base no levantamento de requisitos inicial, será realizada a análise dos mesmos e o desenvolvimento ocorrerá da seguinte forma:

Ambiente: Desenvolvimento do ambiente e dos objetos do jogo, importando seus modelos e definindo como eles serão gerados e como serão posicionados na cena;

Funcionalidade: Adição do controle do usuário, ou seja, a movimentação do avião, bem como as iterações com o ambiente através das colisões e placas que são exibidas durante a partida;

Parâmetros e dados: Adiciona a possibilidade do usuário configurar quais os parâmetros utilizados para configurar a partida. Além disso, realiza a coleta e exibição dados necessários para a avaliação.

Após o desenvolvimento serão realizados testes técnicos, validando se o comportamento está conforme os requisitos, e testes com o usuário (no nosso caso, pesquisadores da CETO) e a partir dessas avaliações, serão realizadas novas iterações para o desenvolvimento do projeto.

1.4 ESTRUTURA

Este trabalho está estruturado da seguinte forma:

1. Introdução: Capítulo atual

2. Fundamentação teórica: Conceitos que fundamentam esse trabalho: o que é gamificação e jogos sérios, e qual a sua importância para a área de terapia ocupacional, o que é realidade virtual, suas aplicações e benefícios. Do ponto de vista prático: qual a engine utilizada, ferramentas de software utilizadas e o Game Carboard XR Plugin.

3. Trabalhos relacionados: Trabalhos já realizados utilizando VR para o tratamento de hemiparesia.

- 4. Desenvolvimento:** Descrevemos qual foi a metodologia utilizada para o desenvolvimento do AeroVR, bem como suas etapas.
- 5. Projeto:** Neste capítulo descrevemos quais foram os requisitos principais para o desenvolvimento, como foi realizado a implementação e quais os desafios enfrentados e os resultados.
- 6. Validação e testes:** Experimentos e testes realizados para a validação do projeto. Neste descreveremos com mais detalhes os testes técnicos, seus resultados e os testes com os usuários e quais as informações colhidas.
- 7. Conclusão:** Com base nos resultados obtidos, é discutido quais os benefícios proporcionados por esse software e o que ainda pode ser realizado como melhorias e campos a serem explorados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ENGAJAMENTO DO JOGADOR

Os jogos sempre tiveram como objetivo primário entreter o jogador durante um determinado período. Portanto, quando tratamos de desenvolvimento de jogos, um questionamento que é sempre muito pertinente é qual o nível de engajamento que o jogador terá ao jogar o jogo que esta sendo produzido. Com o avanço dos jogos digitais, surgiu uma área específica para analisar essas questões, o estudo do Engajamento do jogador (ou, em inglês, *Player engagement*) (Schoenau-Fog, 2011).

Existem diversos motivos que levam um jogo oferecer uma melhor ou pior experiência para o jogador, e isso depende diretamente da proposta do jogo e qual seu público alvo. Isso significa que existem diferentes aspectos que podem ser explorados no processo de desenvolvimento de um jogo que pode aumentar ou diminuir o nível de engajamento em um jogo. Considerando o contexto do AeroVR, os principais conceitos relacionados ao *player engagement* que serviram de base para o seu desenvolvimento são: fluxo, fluxo de jogo (*gameflow*), presença, imersão e motivação (Schoenau-Fog, 2011).

Fluxo O conceito de fluxo foi enunciado pelo psicólogo Mihaly Csikszentmihalyi e pode ser aplicado a uma ampla gama de atividades, incluindo os jogos. Esse conceito pode ser definido como "Fluxo representa a sensação de foco completo e energizado em uma atividade, com um alto nível de prazer e satisfação"(Csikszentmihalyi, 1990). Enquanto no fluxo a pessoa possui o foco máximo na atividade que está realizando, perdendo a noção do tempo e das demais preocupações (Chen, 2007).

Fluxo de jogo Baseado no conceito de fluxo, Penelope Sweetser e Peta Wyeth propuseram um modelo chamado *gameflow* para avaliar o nível de satisfação e prazer de um jogador. Esse modelo se baseia em oito critérios fundamentais: concentração, desafio, habilidades, controle, objetivos claros, feedback, imersão e interação social (Sweetser e Wyeth, 2005).

Presença De forma geral, o conceito de Presença pode ser definida como a sensação de estar efetivamente presente em um ambiente de jogo, ou seja, o ponto em que o hardware ou o meio se torna o mais transparente possível para o jogador. Esse conceito pode ser dividido em três níveis principais: presença espacial (quando o jogador se sente presente em um local específico), presença social (experiência em que as interações com personagens virtuais são percebidas como interações com personagens reais) e auto presença (quando o jogador real se sente como se fosse o personagem virtual) (Tamborini e Skalski, 2006).

Imersão O conceito de imersão é de suma importância para toda a indústria de áudio-visual (música, cinema e televisão) e nos jogos não é diferente. Considerando o contexto do projeto, esse conceito pode ser definido simplesmente como, a mudança de estado de "estar jogando"para o "estar no jogo", o que, apesar de parecer algo simples, é fundamental para a experiência do *player*. A imersão pode se dar em três níveis diferentes: engajamento (*engagement*), que se refere ao interesse inicial e à disposição

do jogador em investir tempo e esforço no jogo; absorção (*engrossment*), que ocorre quando o jogador está profundamente envolvido no jogo, com atenção focada e pouca distração; e imersão total (*total immersion*), que é o estado em que o jogador se sente completamente dentro do mundo do jogo, perdendo a noção do tempo e do ambiente real ao seu redor (Jennett et al., 2008). Esses níveis de imersão são cruciais para proporcionar uma experiência envolvente e satisfatória para o jogador.

Motivação A motivação é essencialmente o motivo que leva as pessoas a jogarem. Diversos estudos psicológicos analisam o que torna um jogo motivador. Abordaremos a motivação de um ponto de vista mais prático, destacando sua importância fundamental para os jogos e os diferentes tipos de motivação. A motivação pode ser dividida em duas formas: a motivação intrínseca, que deriva do prazer e satisfação obtidos diretamente da atividade do jogo, como desafios intelectuais, exploração e narrativa; e a motivação extrínseca, que se relaciona a recompensas externas, como pontuações, conquistas e reconhecimento social (Vriend, 2017).

Embora o projeto tenha como princípio ser um jogo sério (o que será definido com mais detalhes na seção 2.3), é crucial considerar o aspecto do engajamento do jogador. Nosso objetivo é aumentar a motivação do paciente durante a reabilitação. Portanto, os aspectos mencionados foram cuidadosamente considerados no desenvolvimento do jogo e serão discutidos em maior profundidade nos capítulos seguintes.

2.2 GAME DESIGN

Desde os primeiros consoles lançados nos anos 70 os jogos evoluíram drasticamente, acompanhando uma indústria de áudio-visual. Essa evolução, não se resume a gráficos melhores, mas sim em toda uma cadeia técnica e criativa que aumentou em complexidade e competências. Uma das áreas que foram formalizadas é o *game design* (ou projeto de jogos).

Jesse Schell, em seu livro "The Art of Game Design: A Book of Lenses", define esse fundamento como "[...] *the act of deciding what a game should be.*" (Schell, 2008), ou seja, o game design refere-se ao processo de criação e desenvolvimento dos elementos (conceituais, funcionais, artísticos e etc.) de um jogo (Adams, 2013). Sendo assim, alguns processos que fazem parte dessa área são: a concepção de mecânicas de jogo, elaboração de desafios e recompensas, e a definição da estética e narrativa que guiarão a experiência do jogador. O objetivo do game design é criar uma experiência envolvente e cativante, garantindo que o jogo seja não apenas funcional, mas também agradável e imersivo para os jogadores.

O processo de game design tem uma particularidade muito importante que é o fato que ser um processo que envolve muitos elementos artísticos, mas também da computação e engenharia. Sendo assim, não existem normas e padrões bem definidos sobre como deve ser seguido esse processo, uma vez que cada profissional pode realizar da forma mais adequada para o seu contexto. Ainda assim, temos na literatura alguns pontos fundamentais para o game design.

Robert Zubek, por exemplo, define três diferentes níveis de se analisar um jogo, o nível de mecânicas, gameplay e experiência do jogador (Zubek, 2020). A mecânica corresponde as ações dos objetos no jogo, ou seja, é o mapeamento entre as peças individuais do jogo (objetos) e suas ações. Já a gameplay é processo de interação entre o jogador e o jogo em si, ou seja, o *player* atuando com base na mecânica do jogo. Por fim, como resultado dessa interação, temos

a experiência do jogador, que é subjetiva a cada jogador e está intimamente relacionado aos conceitos desenvolvidos no engajamento do jogador.

2.3 JOGOS SÉRIOS

Como descrito anteriormente, quando projetados e desenvolvidos de modo adequado, os jogos possuem uma capacidade ímpar em reter, motivar e engajar uma pessoa. Assim, com a popularização dos jogos eletrônicos, a ideia de utilizar um jogo com um propósito definido que vai além do entretenimento começou a ganhar força.

Um dos primeiros *serious games*, é o jogo *The Bradley Trainer*. Esse jogo foi produzido pela Atari em 1981, a pedido do exército norte americano. Ele é baseado no jogo *Battlezone*, e tinha como objetivo ser um tipo de simulador para a artilharia para o Bradley Fighting Vehicle, um veículo blindado de combate dos Estados Unidos (AtariAge, 2024).

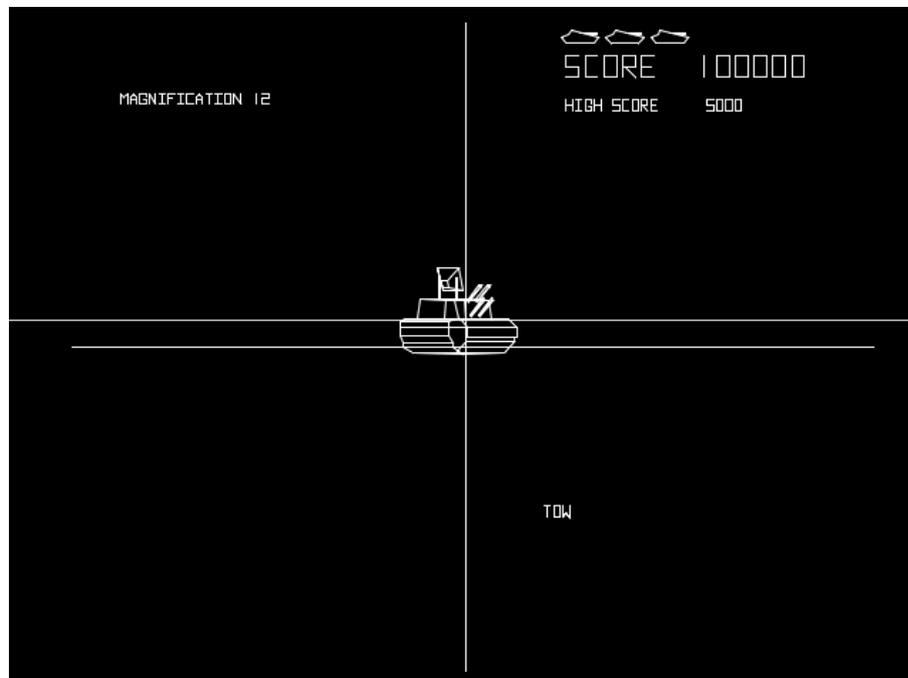


Figura 2.1: Imagem do jogo *The Bradley Trainer* (AtariAge, 2024)

Apesar de ser uma área amplamente explorada na indústria de jogos atualmente, a definição de jogo sério ainda não é bem estabelecida. No entanto, a maioria dos pesquisadores e profissionais da indústria concorda que a definição básica de um jogo sério é: um jogo cujo principal propósito é bem definido (como ensino ou treinamento) e não o entretenimento (Susi et al., 2007). Ainda assim é importante ressaltar que, mesmo que a diversão não seja o aspecto principal de um jogo sério, boa parte dos pesquisadores consideram que o entretenimento continua sendo um aspecto importante dentro do jogo (Laamarti et al., 2014).

O grande benefício dos jogos sérios, e a razão pela qual esse projeto explora essa estratégia de desenvolvimento, é justamente a alta eficiência quando aplicado a terapias e a área da saúde de modo geral. Já existem diversas aplicações que utilizam os jogos sérios na reabilitação locomotora de pacientes pós-AVC (Ma e Bechkoum, 2008) (Eichinger et al., 2020) ou na realização de movimentos gerais (González-González et al., 2019). Todos esses estudos

tiveram resultados positivos, tendo como resultado uma maior motivação e engajamento do paciente à reabilitação.

2.3.1 Gamificação e Jogos sérios

Quando falamos de jogos sérios, um conceito que está muito atrelado a ele, e que por muitas vezes é confundido com ele é a gamificação.

Gamificação é a aplicação de elementos de design de jogos em contextos não relacionados a jogos para aumentar o engajamento e a motivação dos indivíduos. Esses elementos podem incluir pontuações, níveis, desafios e recompensas, sendo amplamente utilizados em áreas como educação, marketing e saúde (Deterding et al., 2011).

Portanto, a gamificação em si não está ligada a um tipo de jogo em si, mas sim, em utilizar conceitos comuns a jogos em uma atividade já estabelecida. Diferentemente dos jogos sérios, que consistem em jogos digitais cujo propósito principal não é o entretenimento. Fazer essa diferenciação é importante para podermos entender quais os conceitos por trás do desenvolvimento desse projeto, bem como o que motivou certas decisões no seu desenvolvimento.

2.3.2 Game design para jogos sérios

O processo de design de jogos para jogos sérios segue as mesmas recomendações e regras de um jogo tradicional. No entanto, a literatura aponta algumas especificidades únicas aos jogos sérios que podem ser seguidas. Entre essas, destacaremos duas que são particularmente relevantes para o projeto em questão.

A primeira é com relação aos controles que devem ser adequados ao contexto em que o jogo será utilizado, ou seja, os existem casos em que controles específicos devem ser utilizados tendo em vista as limitações e necessidades dos jogadores (Machado et al., 2011).

Outra importante recomendação no desenvolvimento de jogos sérios é evitar consequências negativas ou punições severas. Estudos indicam que, em jogos sérios, penalizações por desempenhos abaixo do esperado não incentivam a melhoria; ao contrário, tendem a desmotivar os jogadores (Laamarti et al., 2014). Portanto, manter um ambiente de jogo positivo e encorajador é essencial para promover o engajamento e o aprendizado eficaz dos participantes.

2.4 REALIDADE VIRTUAL

A ideia de poder interagir com um ambiente diferente do mundo real através de um equipamento não é nova, e, como em diversos outros casos, podemos ver o “conceito” em histórias de ficção científica muito antes de se tornar algo real.

“[...] a movie that gives one sight and sound [...] now I add taste, smell, even touch [...]. Suppose I make it so that you are in the story, you speak to the shadows, and the shadows reply, and instead of being on a screen, the story is all about you, and you are in it. Would that be to make real a dream?”

Nesse trecho do conto de ficção “Pygmalion’s Spectacles” (Weinbaum, 1935), já temos um vislumbre do conceito básico da realidade virtual, ou pelo menos da direção em que queremos avançar. A meta da realidade virtual é criar um ambiente tridimensional crível e interativo, no qual uma pessoa possa explorar e interagir de maneira imersiva.

O que antes era apenas ficção já está presente em nosso cotidiano em certo nível. Atualmente existem diferentes dispositivos que permitem, em diferentes níveis de imersão, que seja experimentado a realidade virtual. E em um nível mais prático com o que temos hoje podemos definir a realidade virtual como "Realidade Virtual é o uso da tecnologia computacional para criar o efeito de um mundo tridimensional interativo, no qual os objetos possuem uma sensação de presença espacial"(Bryson, 2013).

O grande diferencial de um de realidade virtual é o seu alto nível de imersão. Isso é possibilitado por todos os sensores presentes na maioria dos dispositivos, mas principalmente pelo seu maior campo visual (*Field of Visio, FoV*), que em um modelo comercial como o Meta Quest 3 é de 110° horizontal e 96° vertical (Meta, 2023). Com isso, atualmente o uso do VR é muito presente em aplicações para treinamento, educação, simulação e jogos e entretenimento de modo geral (Strivr, 2023).

Com relação especificamente ao universo dos jogos, como descrito anteriormente, dois fatores cruciais para o engajamento do jogador são a imersão e a presença. A realidade virtual oferece uma vantagem significativa nesses aspectos, proporcionando uma experiência mais envolvente e realista. Nos últimos anos, houve um aumento notável no número de jogos desenvolvidos para VR, como os populares "Beat Saber" e "Assetto Corsa", que exemplificam essa tendência crescente e o potencial do VR em transformar a experiência de jogo.

Neste projeto em especial a realidade virtual permite que exercitemos uma maior área do campo visual do paciente. Portanto, temos como objetivo um ganho na flexibilidade de uso e eficácia do jogo quando estivermos no modo VR.

2.4.1 Hardware para realidade virtual

Para atingir o objetivo da realidade virtual, existem duas estratégias: através de Multi-project environments, que utiliza projetores ou monitores para criar a imersão, e os head-mounted displays (HMD), o mais comum, em que através de um óculos de realidade virtual as imagens são exibidas para a pessoa.

Os HMD's, mais conhecidos como óculos de realidade virtual, são o modo mais comum de pensarmos a realidade virtual atualmente. Começaram a ser desenvolvidos desde o final dos anos 70, e durante muito tempo era uma tecnologia muito limitada com relação ao acesso devido ao elevado custo e limitações. Entretanto, atualmente tem se tornado cada vez mais comuns e baratos.

O AeroVR foi projetado para ser utilizado com óculos de realidade virtual baseados no Google Cardboard (ver Figura 2.2). Esse foi um modelo de óculos VR muito simples, lançado pelo Google, e que tinha como proposta permitir uma experiência em realidade virtual de forma mais barata, utilizando o hardware dos smartphones. O Google Cardboard em si já não é mais fabricado, entretanto, ainda existem diversos modelos de óculos VR que se baseiam nesse mesmo princípio(Google Developers, 2014). Nesse modelo, o óculos em si possui apenas duas lentes funciona como um suporte para o celular. Assim, ao gerar uma imagem estereoscópica no aparelho, em conjunto com as lentes do óculos, o usuário terá uma visão imersiva, com um FoV podendo chegar a 100° dependendo do tamanho do celular.

2.4.2 Realidade Virtual, realidade aumentada e realidade mista

Com a popularização da realidade virtual novas formas de explorar esse novo universo foram sendo desenvolvidas. Entre essas podemos destacar a realidade aumentada e a realidade mista.



Figura 2.2: Google Cardboard (Google Developers, 2014)

Apesar de ser três paradigmas diferentes entre si, muitas vezes o limite entre o que seria a realidade virtual e aumentada, por exemplo, pode se tornar nebulosa. Assim, vamos diferenciar aqui essas modalidades e definir qual que é utilizada nesse projeto.

Para entender as diferenças entre esses paradigmas, primeiramente é importante definir. A realidade virtual é totalmente imersiva, portanto, a nesse caso o usuário interage com um mundo virtual que não têm relação com o ambiente em que o usuário está no momento. A figura 2.3 exemplifica um óculos VR.



Figura 2.3: Óculos de realidade virtual PlayStation VR2 (Sony, 2023)

Já na realidade aumentada (*Augmented Reality, AR*), as imagens são projetadas sobre os elementos do mundo real. Nesse modelo, o mundo real continua como um elemento central, mas é complementado com elementos virtuais (Intel, 2016). Com relação aos dispositivos de AR, existem óculos como o Microsoft HoloLens (Microsoft, 2019), mas também existem aplicações que utilizam apenas um smartphone para criar uma experiência de realidade aumentada, como por exemplo o jogo Pokémon Go lançado em 2016 (Malik, 2016) e que ainda é um dos maiores exemplos de jogo em realidade aumentada (ver figura 2.4).

Por fim temos a realidade mista, ou *Mixed reality*, em que temos o usuário interagindo com elementos reais e virtuais. Atualmente os dispositivos de realidade mista são apenas headsets, que possuem um conjunto de câmeras e sensores que permitem que o usuário esteja presente no



Figura 2.4: Jogo de AR, Pokémon Go (Malik, 2016)

mundo real, ao mesmo tempo que interage com elementos virtuais (Intel, 2016). Um exemplo de óculos de realidade mista é o Meta Quest 3 (Figura 2.5).



Figura 2.5: Imagem da campanha publicitária para o Meta Quest 3 (Meta, 2023)

Sendo assim, como o jogo idealizado para esse projeto não tem o objetivo de permitir que o paciente interaja com o mundo real, pelo contrário, queremos que ele esteja em total imersão em um ambiente controlado para permitir um melhor resultado, utilizamos apenas a realidade virtual.

2.5 MONITORES

Monitores tradicionais são ferramentas essenciais na reabilitação de pessoas com perda visual. Atualmente, muitos exames e terapias utilizam monitores como parte fundamental do processo de tratamento.

Apesar de não abrangerem a mesma área do campo de visão que os óculos de realidade virtual, se corretamente utilizados, um monitor apenas pode estimular uma grande área do campo de visão. Além disso, essa área pode ser ampliada com o uso de múltiplos monitores, chegando a resultados próximos ao FoV de um óculos VR.

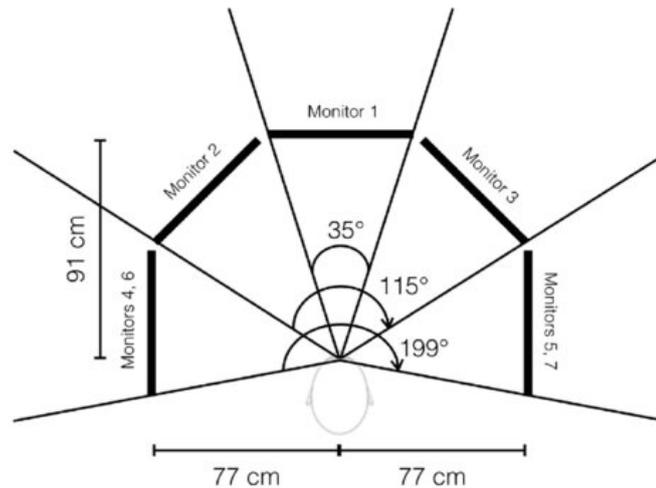


Figura 2.6: Estudo realizado com monitores de 27". (Lidestam et al., 2019)

2.6 GAME ENGINES

Podemos definir uma game engine como um ambiente de desenvolvimento de jogos, ou seja, um software que possui um conjunto de ferramentas que facilitam e simplificam o desenvolvimento de um jogo (Arm Limited, 2024). Essas aplicações permitem que jogos complexos sejam criados com muito mais facilidade, menos custo e de forma mais rápida.

Na tabela 2.1 apresenta um comparativo entre as principais game engines da atualidade (Douglas, 2024), destacando suas características relevantes para o projeto em questão. Observa-se que, entre as engines listadas, apenas o Unity e o Godot atendem, ao menos parcialmente, a todos os requisitos estabelecidos.

Nome	Jogos 3D	Multiplataforma (PC e Mobile)	VR para Mobile	Preço	Disponível para Linux
Unreal	Sim	Sim	Não possui suporte nas versões mais recentes	Grátis para jogos que possuem receita inferior à USD 1 milhão	Sim
Unity	Sim	Sim	Sim, através de plugins	Conta grátis com recursos limitados e para estudantes	Sim
Gamemaker	Sim, mas com restrições	Sim	Não possui suporte nativo	Possui uma versão gratuita	Sim, mas em versão Beta
Godot	Sim	Sim	Não possui suporte nativo	Grátis	Sim
Construct 3	Sim, mas com restrições	Sim	Não	Possui uma versão gratuita	Sim, é um editor online

Tabela 2.1: Comparação de motores de jogos

Na prática, vemos que a engine Godot possui um suporte muito limitado para o VR em smartphones, sendo que para o nosso projeto seria necessário utilizar plugins de terceiros

(que não recebem atualizações frequentes) ou criar um próprio. Sendo assim, o Unity, tendo um amplo suporte para realidade virtual em dispositivos móveis, com plugins oficiais, foi a opção escolhida para esse projeto.

2.7 UNITY

O Unity é atualmente o principal motor de jogo disponível no mercado. Desenvolvido e mantido pela empresa Unity Technologies, ele é amplamente utilizado por diversos estúdios de jogos e desenvolvedores independentes. De acordo com o relatório da Unity de 2021, 94 dos 100 maiores estúdios de jogos do mundo, classificados pela receita global, utilizam seus serviços (Unity Technologies, a).

A história da empresa se inicia em 2004, em Copenhague, quando três amigos, Nicholas Francis, Joachim Ante, e David Helgason, se uniram com o propósito inicial de criar uma game engine simples, mas ainda assim poderosa. O Unity 1.0 foi lançado em 2005, e mesmo ganhando notoriedade desde seu lançamento, os pontos-chave que tornaram essa a game engine mais popular globalmente ocorreram posteriormente com a ascensão do mobile e a decisão de adotar uma nova política de preços. Em 2007 ocorreram dois lançamentos importantes para o Unity: o primeiro iPhone e o Unity 2.0. Com o primeiro, o mercado de smartphones começou a se crescer anualmente, e desde o início o Unity era uma das poucas game engines com suporte a jogos mobile, o que possibilitou que se tornassem a principal escolha no universo de games mobile. Já em 2009, foi realizada a alteração na estratégia de precificação, permitindo que desenvolvedores amadores ou independentes utilizem o software de maneira gratuita, o que ampliou o acesso a plataforma (Cohen-Peckham, 2019).

Atualmente, a Unity é o *"the world's leading platform for creating and growing interactive, real-time 3D ("RT3D") content and experiences"* (Unity Technologies, b), ou seja, a empresa fornece diversas soluções para criação de conteúdo 3D não restrita à área de jogos, mas inclui a área de arquitetura, engenharia, educação e etc. (Unity Technologies, c).

Como mencionado anteriormente, o principal motivo pelo qual optamos por utilizar essa game engine no projeto, é pelo grande suporte oferecido para o desenvolvimento de jogos VR para smartphones. Entretanto, outros diferenciais do Unity quando comparado aos seus concorrentes, como, o grande suporte ao mobile, a comunidade em torno da plataforma, a vasta documentação disponível e facilidade de aprendizado, também foram fatores importantes para a decisão de utilizar o Unity nesse projeto.

Dentro do desenvolvimento de jogos, uma abstração básica que facilita significativamente o entendimento é o conceito de objetos e ações. Os objetos são todos os elementos que compõem um jogo (personagens, elementos do ambiente e etc.), sejam eles visíveis para o jogador ou não. Já as ações, são todas as transformações que podem ocorrer em um objeto em um estado do jogo. Esse conceito é importante para entendermos os dois conceitos principais do funcionamento do Unity na prática: os *GameObjects* e os *scripts*.

2.7.1 GameObject

No Unity, a classe *GameObject* representa tudo o que pode existir em uma determinada cena (Unity Technologies, e). Portanto, em um jogo desenvolvido com Unity, conceitualmente, tudo o que os jogadores visualizam são componentes que pertencem à classe *GameObject*.

Alguns métodos e propriedades importantes do GameObject são: posição em cena, tamanho, status (ativo ou não). Manipulando, as propriedades de um objeto, é possível determinar seu comportamento na cena.

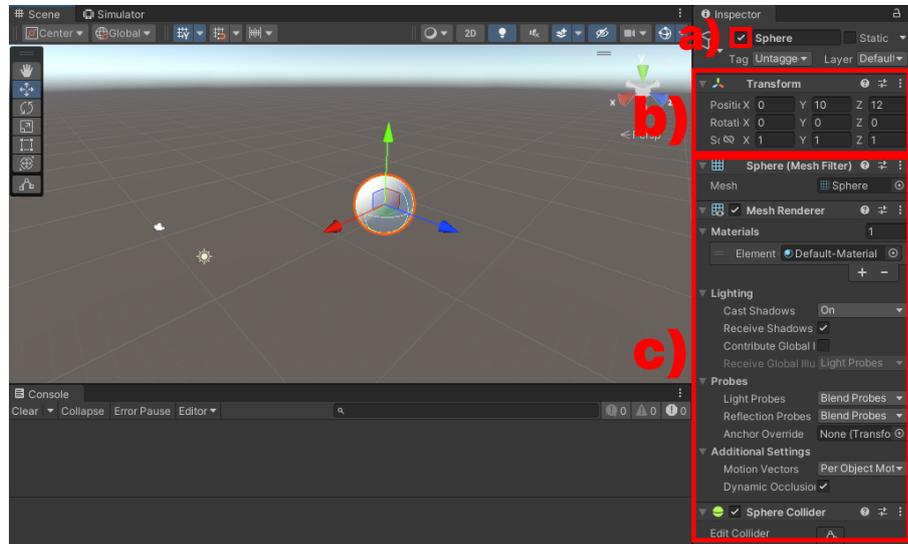


Figura 2.7: a) Estado do objeto (ativo ou desativado); b) Posição do objeto; c) Demais componentes

Em um jogo podemos ter um objeto esférico, como o exemplo da Figura 2.7. Esse GameObject está ativo e posicionado no eixo X no ponto 0, Y no ponto 10 e Z no ponto 12.

2.7.2 Scripts

O comportamento de um GameObject é determinado pelos componentes adicionados a ele. Esses componentes, por sua vez são criados via scripts (Unity Technologies, d).

A linguagem padrão utilizada para criar um script em Unity é o C#. Através dos scripts é possível alterar as propriedades do GameObject que ele pertence em resposta a uma entrada do usuário ou uma colisão, por exemplo.

A seguir temos um exemplo simples de script que apenas movimenta um objeto para frente:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript : MonoBehaviour {
5     public float speed = 10f;
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14        transform.Translate(Vector3.forward * speed * Time.deltaTime);
15    }
16 }

```

2.7.3 Unity Asset Store

A Unity Asset Store é uma plataforma que permite a comercialização de plugins, modelos 3D e 2D e etc. para o Unity.

Essa é uma plataforma oficial da empresa Unity, sendo totalmente integrado a game engine, ou seja, é possível importar um modelo adquirido dentro da loja diretamente no jogo. Mas é importante salientar, que essa plataforma facilita o processo de criação de um jogo, mas não é um componente obrigatório no desenvolvimento, ou seja, é possível criar e utilizar os próprios modelos 3D e plugins.

2.8 GOOGLE CARDBOARD XR PLUGIN

O Google Cardboard XR Plugin é uma ferramenta desenvolvida pelo Google para facilitar o desenvolvimento de experiências de Realidade Aumentada (RA) e Realidade Virtual (RV) utilizando dispositivos móveis e o visor de papelão do Google Cardboard. Esse plugin integra funcionalidades específicas para que desenvolvedores possam criar aplicações imersivas sem a necessidade de hardware sofisticado, aproveitando ao máximo as capacidades dos smartphones modernos.

O Google Cardboard XR Plugin permite aos desenvolvedores criar aplicações de RA e RV de forma eficiente e acessível. Esse plugin possui suporte as principais funcionalidade VR:

- Rastreamento de movimentos
- Interação via controles
- Renderização estereoscópica

O Cardboard SDK está disponível para: Android, iOS e Unity.

2.9 BLENDER

Blender é uma suíte de software livre e de código aberto para criação de gráficos 3D, animações, efeitos visuais, modelagem, escultura, rigging, simulações, renderização, composição e edição de vídeo. Desenvolvido pela Blender Foundation, o Blender é amplamente utilizado em diversas indústrias, incluindo animação, jogos, design gráfico, filmes e realidade virtual (VR).

Blender está disponível para os principais sistemas operacionais, incluindo Windows, macOS e Linux, tornando-o acessível a uma ampla base de usuários. Sua natureza de código aberto e a ativa comunidade de desenvolvedores e artistas contribuem para a constante evolução e melhoria do software. Além disso, a licença GNU General Public License (GPL) permite que usuários e empresas utilizem e modifiquem o Blender sem restrições de licenciamento

Para o projeto, foi utilizado o blender para criar os modelos 3D que não estavam disponíveis na Unity Asset Store, como por exemplo, o avião (ver Figura 2.8)

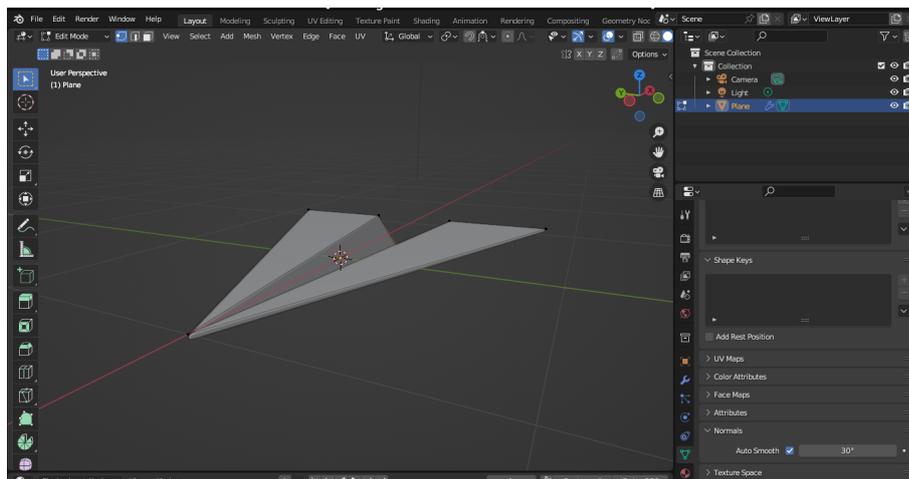


Figura 2.8: Tela principal do Blender, com modelo do avião

3 TRABALHOS RELACIONADOS

Apesar do crescente uso de tecnologia na área de reabilitação e terapia ocupacional, foi possível encontrar apenas dois trabalhos que tratam do uso da realidade virtual para o uso no tratamento da hemianopsia.

No artigo *Salzburg Visual Field Trainer (SVFT): A virtual reality device for (the evaluation of) neuropsychological rehabilitation* (Leitner et al., 2021) os autores propõem um sistema utilizando realidade virtual para auxiliar na avaliação da área do campo visual afetado, bem como para permitir que os pacientes realizem os teste de forma mais confortável e flexível. O trabalho mostrou que com um sistema construído utilizando um óculos VR no estilo do Google Cardboard (utilizando um celular como a tela principal), os resultados para avaliação e estímulo da área afetada foram muito elevados.

O artigo *An Audiovisual 3D-Immersive Stimulation Program in Hemianopia Using a Connected Device* (Daibert-Nido et al., 2021b) e o relatório *Case Report: Visual Rehabilitation in Hemianopia Patients. Home-Based Visual Rehabilitation in Patients With Hemianopia Consecutive to Brain Tumor Treatment: Feasibility and Potential Effectiveness* (Daibert-Nido et al., 2021a) tratam da mesma ferramenta para o tratamento de hemianopsia. A ferramenta desenvolvida se trata de um exercício realizado em um óculos VR, que consiste basicamente em acompanhar uma bola amarela se movendo no meio a outras. O objetivo desses trabalhos foi justamente mostrar a eficiência e a melhora da qualidade de vida do paciente em poder realizar o tratamento em casa com o equipamento.

Além desses existem outros trabalhos, relacionados ao uso de computadores para o teste e tratamento de problemas que afetam o campo visual. Entre esses podemos citar o artigo *A free and simple computerized screening test for visual field defects* (Hazelton et al., 2018) que propõe com sucesso um teste similar ao SFVT para identificar as áreas da visão afetadas, entretanto sem o uso de realidade virtual, apenas utilizando o monitor tradicional.

4 DESENVOLVIMENTO

Tendo em mente o projeto realizado, é importante pontuar como foi o processo para o seu desenvolvimento. Nesse capítulo descrevemos qual o modelo de processo de software utilizado e como ocorreu o desenvolvimento do AeroVR. No próximo capítulo, o projeto é apresentado em detalhes.

4.1 CONTEXTO

Um modelo de processo de software é uma representação do processo de software em si, ou seja, uma sequência bem definida de etapas e seus objetivos durante o processo de criação e desenvolvimento de um software. É importante frisar que um modelo não procura definir regras rígidas, mas sem estabelecer um framework que pode ser adaptado para cada contexto.

O processo de desenvolvimento do AeroVR possui como principal objetivo desenvolver um software que atenda as necessidades enfrentadas por profissionais da terapia ocupacional, e em específico, o grupo da Clínica-Escola de Terapia Ocupacional da UFPR. Sendo assim, foi de suma importância que, em todo o processo de desenvolvimento, os profissionais que irão utilizar o software pudessem ativamente participar do projeto, propondo melhorias identificando possíveis falhas que poderiam prejudicar o uso do aplicativo.

Tendo em vista esse contexto, utilizamos como base o modelo incremental para desenvolver o processo de software para esse projeto. Nesse modelo, a ideia é, com base na especificação do projeto, desenvolver uma versão inicial do software, e expor essa a avaliação do usuário. Assim, com o resultado dessa avaliação, realizar adequações e melhorias resultando em uma versão intermediária do software, após isso, é realizado novamente a validação com o usuário. Esse processo é repetido diversas vezes, até que se tenha uma versão final do software(Sommerville, 2010).

Esse modelo permite uma maior flexibilidade, permitindo que o usuário final seja uma parte importante do processo de desenvolvimento e diminui o custo de acomodar mudanças necessárias e não previstas no início do desenvolvimento.

4.2 ITERAÇÕES

O processo de desenvolvimento do AeroVR foi realizado em 4 iterações principais. A seguir descreveremos com mais detalhes o que foi gerado em cada uma dessas até a validação da versão final.

4.2.1 Primeira versão

A primeira versão corresponde a versão sobre a qual iremos realizar todas as alterações. Tendo em vista isso, a primeira versão do projeto foi implementada visando validar os principais conceitos do projeto. Portanto, o projeto consistia em uma versão de partida sem limites de tempo, apenas com a possibilidade do modo em realidade virtual, exibindo dois modelos de placa fixas.

Essa primeira versão foi disponibilizada para os profissionais da CETO validarem os principais conceitos do jogo. Nessa validação, o fundamento jogo foi aprovado, e foi apontado os

parâmetros do jogo que deveriam ser passíveis de configurações: tempo de exibição das placas, velocidade do jogo, tempo de partida e posição das placas. Além disso, foi definido que o modo de validação seria manualmente, com o auxílio do terapeuta. Também foi estabelecido que as placas utilizadas possuiriam um formato de uma placa de trânsito, mas exibiriam apenas números de 1 a 5.

4.2.2 Segunda versão

Com base nas melhorias apontadas como resultado da primeira versão, foram implementadas as seguintes melhorias no projeto:

- Adicionado suporte para múltiplos pacientes;
- Criada as telas de configuração, sendo que as principais configurações implementadas foram:
 - Velocidade do jogo em três níveis;
 - Tempo de duração do jogo;
 - Grid 3x3 para definir as posições das placas;
 - Tempo de exibição das placas em segundos.
- Implementado o modo normal do jogo, ou seja, passou o AeroVR passou a ter suporte para o modo VR e 3D;
- Durante a partida foi adicionado as placas numeradas de 1 a 5;
- Foi adicionado uma tela ao final da partida com as informações de colisão e quais placas foram exibidas;
- Adicionada telas de menu para navegar entre as partidas e configurações do jogo.

Na avaliação realizada pela CETO, foi verificado que uma possível melhoria seria a inclusão da lateralidade da colisão no resumo da partida, pois isso seria importante para avaliação do resultado do paciente. Já os problemas apontados nessa versão foram:

- Verificado que o grid para definir a posição das placas não possuía o número adequado de posições;
- O tempo de exibição de cada placa estava muito longo, não sendo adequado para os pacientes;
- O jogo poderia exibir placas com o mesmo número repetidas vezes, o que poderia induzir o paciente a cometer erros;
- A partida apresentava erro quando a velocidade era alterada;

4.2.3 Terceira versão

Para a terceira versão foram corrigidos os bugs de velocidade apontados, bem como foram realizadas melhorias na ambientação do jogo, adicionando cercas e pedras para delimitar o caminho que o jogador poderia passar. Além disso, outras melhorias foram:

- Aumentado o grid de posição das placas, passando de 3x3 para 3x5;
- o tempo de exibição das placas foi reduzido de segundos para milissegundos;
- Adicionado verificação para impedir repetição sequencial das placas;
- Adicionada telas com: autoria do jogo e histórico de partidas do paciente;
- Criada configuração com a quantidade de árvores e obstáculos na partida;
- Implementado o som ambiente do jogo.

Nessa avaliação não foram apontados erros, mas sim duas melhorias. A primeira seria a adição da informação da localização das placas ao final da partida e a segunda seria a possibilidade de reproduzir uma partida disputada, ambas as alterações buscam melhorar a avaliação do terapeuta.

4.2.4 Quarta versão

A quarta versão corresponde à versão final apresentada para validação. Foram realizadas as seguintes melhorias:

- Adição de tela anterior à partida, para permitir que o paciente se localize espacialmente antes de iniciar;
- Criação do playback com os momentos principais da partida, como colisões e placas exibidas;

5 PROJETO

Considerando todos os fundamentos de game design, engajamento do jogador e jogos sérios (descritos na seção 2), iremos analisar quais foram as decisões tomadas durante o projeto do AeroVR, levando em consideração seus objetivos e restrições. Além disso, iremos descrever quais os desafios que essas decisões levaram, no aspecto prático de desenvolvimento, e como esses foram superados.

5.1 AMBIENTE

A proposta do jogo é simples e divertida: controlar um avião de papel em um ambiente lúdico. Entretanto, mesmo assim, poderíamos ter diferentes ambientes ou combinações que poderiam afetar drasticamente a experiência do jogador. Sendo assim, o ambiente escolhido para o jogo é uma floresta, justamente por ser um ambiente mais leve e agradável para o jogo.

5.1.1 Composição

Mesmo com a proposta inicial definida, ainda assim, temos diferentes estilos de jogos, que poderiam gerar resultados muito diferentes. Para um exemplo simples, observe as diferenças entre as imagens 5.2 e 5.1



Figura 5.1: Cena do jogo *Blair Witch* (SA, 2019)

Ambas são imagens de florestas, entretanto, a primeira é de um jogo de terror diferente da segunda. Portanto, através deste simples exemplo, vemos que, independentemente da mecânica do jogo, o ambiente em que ele se ambienta é uma peça fundamental para a experiência do jogador.

Sendo assim, foi optado por utilizar o estilo conhecido como “low poly”, que traz um sentimento mais informal e divertido para o ambiente. Além disso, para simplificar o processo de desenvolvimento, utilizamos dois pacotes disponibilizados gratuitamente na Unity Asset Store: Low-Poly Nature e Low-Poly Simple Nature Pack. Na figura 5.3 é possível ver os itens utilizados no jogo provenientes de um package.



Figura 5.2: Cena do package *Shiratori's Low Poly Forest* (Shiratori, 2020)



Figura 5.3: Elementos de packages utilizados para compor o ambiente

Todo o ambiente do jogo é composto por:

- 3 modelos de árvores;
- 3 modelos para o chão e montanhas;
- 2 modelos de cerca;
- 3 modelos de arbusto;
- 1 modelo de flor.

Apesar desses pacotes facilitarem o processo de desenvolvimento do jogo, eles não possuíam todos os elementos necessários para a construção do jogo. Sendo assim, foi necessária a modelagem manual dos objetos utilizando o software Blender, para a construção do avião de papel em si e das placas que são exibidas.

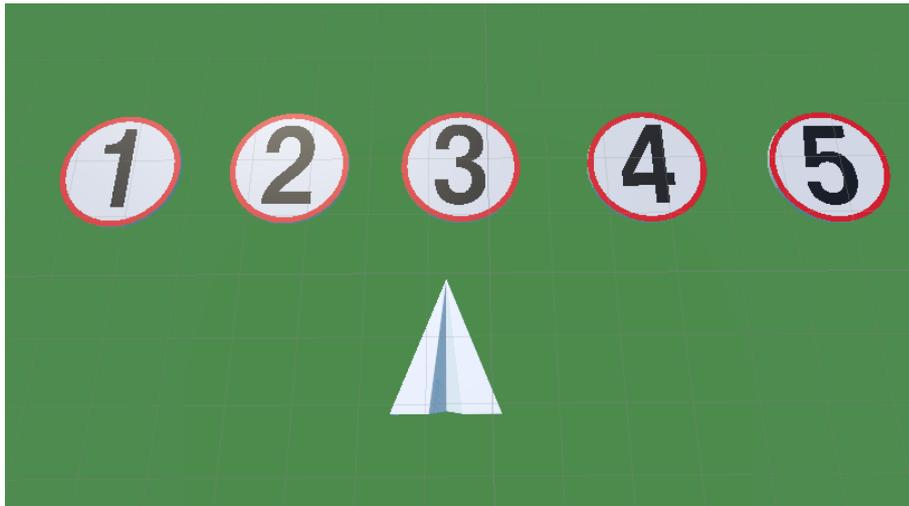


Figura 5.4: Elementos modelados manualmente

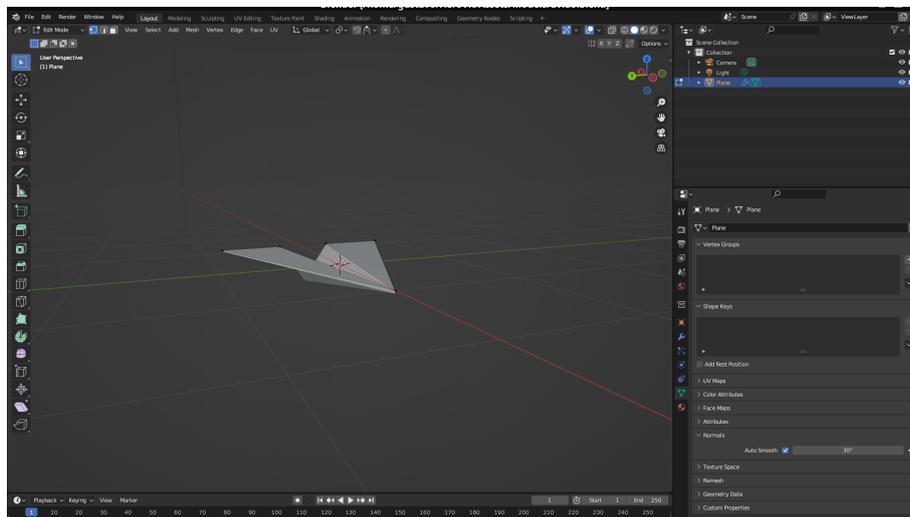


Figura 5.5: Modelo do avião no software "Blender"

5.1.2 Música

Outro aspecto de suma importância para o sentimento gerado durante a gameplay é a música e os efeitos sonoros. A trilha sonora de um jogo é um importante elemento para ajustar o sentimento da cena, já os efeitos sonoros fazem parte do feedback que o usuário recebe do jogo.

Para o AeroVR precisávamos de uma música simples, mas alegre, que não fosse uma distração, mas sim um complemento ao jogo. Portanto, optamos pela composição de Pyotr Ilitch Tchaikovsky, o Lago dos Cisnes, mais especificamente a peça de abertura.

Aliada a música, adicionamos uma camada de um som de vento, auxiliando a ideia de movimento do jogo, e também foram adicionados sons de batida a todo momento que o jogador colide com uma árvore.

5.1.3 Áreas do jogo

Sobre o ambiente, no jogo nós temos três elementos básicos: o caminho que o jogador pode percorrer, elementos de preenchimento da cena e elementos que limitam o mundo.

O Unity inicialmente cria um mundo infinito, ou seja, não existe uma limitação do tamanho do jogo. Portanto, a primeira coisa que precisamos fazer é adicionar elementos que limitem a região visível do jogo. Como o ambiente que queremos criar é de uma floresta, os elementos que limitam o mundo do jogo são: o chão verde para simular um gramado e montanhas para determinar a linha do horizonte.

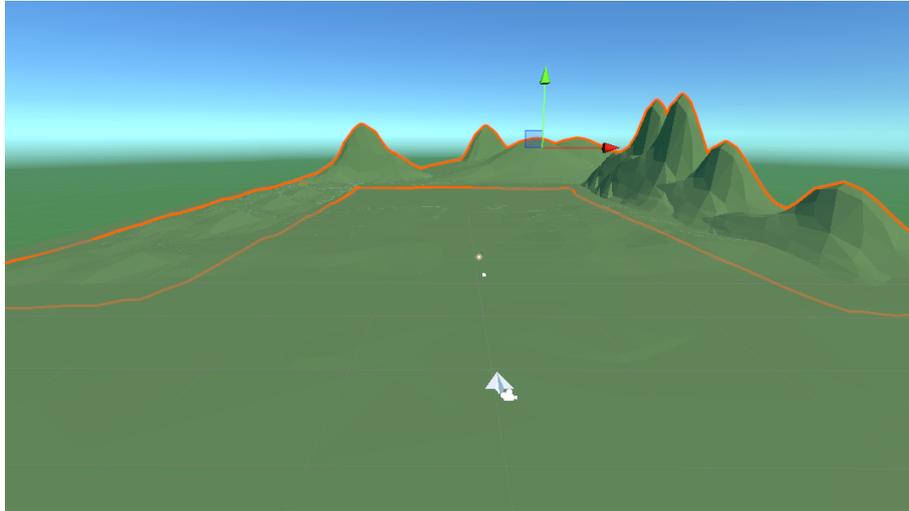


Figura 5.6: Ambiente apenas com as montanhas (em destaque) e o chão

Podemos comparar esses elementos a uma caixa que criamos para conter o jogo em si. Sendo assim, todo elemento que for adicionado ao jogo deve estar dentro dessa região, pois é a única área visível ao paciente.

É importante salientar que, para compor essa região foram utilizados apenas os elementos de chão. Assim, para criar as montanhas apenas é necessário deformar esses elementos.

Com o mundo delimitado, podemos adicionar os elementos que preenchem a cena. Observe que apenas com as montanhas a cena se tornaria muito desinteressante e pobre de elementos. E como queremos ambientar o jogo em uma floresta, é necessário adicionar um elemento muito importante: às árvores e outras plantas.

Boa parte desses elementos não exercem uma função no jogo, ou seja, não existe nenhuma interação com eles que não seja apenas visual.

Por fim, a última área do ambiente é justamente a área jogável. Como descreveremos em mais detalhes a seguir, os movimentos do jogador são limitados a desviar das árvores que aparecem em seu caminho, entretanto esse movimento não é por todo o mundo, apenas em uma região limitada, como pode ser observado na figura 5.8

Para delimitar visualmente essa região, utilizamos pedras para demarcar o chão e cercas ao lado. Assim, a experiência do paciente continua condizente com o ambiente em que ele está.

5.1.4 Geração do ambiente

Até este ponto foram citados apenas como os elementos que compõem o ambiente são organizados, não sendo detalhado como eles são posicionados e como se movimentam.

O movimento básico do jogador se dá no eixo Z, ou seja, o avião de papel está sempre indo para frente na visão do paciente. Entretanto, se movimentarmos o jogador, posicionar os demais elementos em tela seria problemático, uma vez que o estamos sempre em movimento.



Figura 5.7: Ambiente apenas com as vegetações



Figura 5.8: Ambiente final

Para superar esse desafio utilizamos um truque muito comum para esse tipo de jogo: não movimentamos o jogador em si, mas sim o mundo.

Assim, o jogador sempre está no mesmo ponto no eixo Z, mas os elementos da cena (caminho, árvores e montanhas), estão sempre se movimentando para atrás do jogador. Dessa forma, criamos a ilusão de movimento do jogador, ao mesmo tempo que mantemos as coordenadas fixas, facilitando posicionar novos objetos.

Observe, que com isso, temos ainda o seguinte desafio: como gerar diferentes ambientes para o jogador? Para isso, utilizamos uma técnica muito importante: geração procedural de conteúdo.

A geração de conteúdo processual (PCG) refere-se à geração algorítmica de conteúdo de jogo com limitada ou nenhuma contribuição humana (Togelius et al., 2013). Portanto, para o AeroVR, o objetivo era gerar ambientes que estivessem dentro dos padrões estabelecidos (ver seção sobre configurações) e que nunca se repetissem, ou seja, sempre organizar os elementos visuais de diferentes formas, mas que estes se mantivessem consistentes com o que foi configurado previamente.

Como não ainda não temos dados suficientes para tentar otimizar o ambiente para cada paciente utilizando técnicas mais sofisticadas de PCG com neuroevolução (Beukman et al., 2022) e inteligência artificial (Edwards et al., 2021), foi utilizada uma geração randômica com algumas restrições para compor uma seção do ambiente.

Ao combinar a geração procedural do ambiente com a movimentação da cena, obtemos o seguinte comportamento: criamos uma seção que se move em direção ao jogador. Quando essa seção atinge um ponto predeterminado, a próxima seção é gerada, e o processo se repete continuamente.

Na figura 5.9 é possível visualizar o ambiente do jogo durante uma partida a partir da visão do jogador.



Figura 5.9: Ambiente durante a partida

5.2 CONFIGURAÇÕES

Devido aos requisitos que o jogo precisa atender para ser utilizado por diversos pacientes diferentes, ele possui algumas características únicas quando nos referimos às suas possíveis configurações. Todos os parâmetros configuráveis da partida foram definidos e refinados, de forma iterativa em sucessivas reuniões, realizadas com equipe da CETO. Na seção "Desenvolvimento" é detalhado com mais detalhes como foi realizado o processo de desenvolvimento, na seção atual iremos detalhar o resultado das configurações do AeroVR.

5.2.1 Configuração para múltiplos pacientes

Uma das principais características de um jogo comercial, é que ele é voltado para possuir apenas um único jogador em cada conta ou instância do jogo. Portanto, mesmo quando pensamos nos exemplos mais clássicos de um jogo que suporta múltiplos jogadores ao mesmo tempo, ainda assim, as configurações e dados do jogo estarão associados ao dono do jogo em si. Entretanto, para o AeroVR, teremos múltiplos pacientes utilizando o mesmo dispositivo, e os dados devem ser salvos para cada paciente de forma consistente.

Portanto, o jogo foi desenvolvido com um menu especial para cadastrar ou selecionar pacientes já cadastrados anteriormente. Com isso, podemos cadastrar quantos usuários forem necessários e manter as configurações e dados destes isolados um do outro, assim os dados entre os pacientes não são misturados ou ligados, mantendo a consistência necessária.

5.2.2 Configurações do nível de dificuldade da partida

Outra característica que se diferencia de jogos comerciais tradicionais diz respeito aos níveis do jogo. Quando pensamos em jogos clássicos como, por exemplo, “Mario Bros.” e até em jogos mais recentes como “Subway Surfers”, temos mecanismos automáticos que aumentam a dificuldade conforme o jogador vai avançando. No caso do primeiro, esse mecanismo são os próprios níveis do jogo, ou seja, todo jogador se encontra em um nível e ao conseguir completar irá para o próximo que possui um maior nível de dificuldade. Já para o segundo exemplo, cada nível é “infinito”, ou seja, não existe um final para a partida que não seja o jogador perder. Entretanto, ainda assim, dentro desse nível infinito, a dificuldade aumenta gradativamente conforme o tempo de partida.

Existem jogos que utilizam outras estratégias, como definir campanhas ou histórias para o jogador, ou ferramentas, como o uso de inteligência artificial, para analisar o comportamento do jogador e definir a dificuldade do jogo (Corporation, 2024). Ainda assim, temos, na grande maioria dos casos, a estratégia de um aumento gradativo automático de dificuldade, em que pelo menos os parâmetros principais já foram definidos previamente e são aplicados para todos os jogadores de forma imparcial. E isso não seria adequado para o caso de uso do AeroVR justamente pelas particularidades de cada paciente.

Quando pensamos em casos de pacientes que sofreram AVC e possuem algum tipo de sequela, não podemos determinar um caso padrão justamente por que essa sequela pode ocorrer de diferentes formas. Existem pacientes que apresentam comprometimentos visuais (em diferentes níveis), outros podem apresentar comprometimentos motores e de fala. Portanto, não podemos determinar previamente qual o caso padrão para o paciente e qual é a sequência de níveis que pode melhor beneficiar ele e não possuímos ainda dados suficientes para que o uso de técnicas mais sofisticadas possam ser viáveis.

Sendo assim, o jogo foi projetado para que o profissional responsável e capacitado, possa configurar manualmente o nível de dificuldade do jogo para o paciente que ele está atendendo.

Quando falamos sobre a dificuldade de um jogo, existem diversas características que podem tornar um jogo mais fácil ou mais difícil. Como a premissa do AeroVR é ser um jogo mais simples, podemos controlar o desafio imposto pelo jogo em 3 pontos principais: ambiente, velocidade do jogo e configuração das placas.

Com relação ao ambiente do jogo, é possível controlar a densidade das árvores, ou seja, a quantidade de árvores geradas a cada segmento do jogo. No caso, a densidade possui níveis que vão de 0 a 10, e o número de árvores em cada segmento é de $20 \times$ nível, dessas $2 \times$ nível são as árvores que estarão dispostas no caminho como obstáculo. As árvores, são importantes elementos do jogo, pois além de serem utilizadas como obstáculos, elas também são um importante elemento que pode gerar oclusões parciais das placas, gerando um maior nível de dificuldade para o paciente.

Já a velocidade do jogo, na prática, diz respeito a velocidade que o ambiente se desloca, sendo assim, quanto mais rápido o ambiente se desloca, mais difícil é desviar dos obstáculos, uma vez que o paciente terá um menor tempo de reação. A velocidade é controlada em X níveis.

O elemento central do jogo é o paciente conseguir visualizar as placas exibidas no caminho. Sendo assim, é necessário que esse seja um dos aspectos do jogo com mais possibilidade de configuração. Nós permitimos que seja configurado:

- Tempo de exibição das placas durante uma partida, sendo um valor configurável em milissegundos;

- Tempo entre a geração das placas, ou seja, a cada quanto tempo será chamada a rotina que tenta adicionar uma nova placa no ambiente. Essa configuração se dá em segundos, e possui as seguintes possibilidades: 1s, 2s, 5s, 10s e 15s;
- Probabilidade de gerar uma placa, ou seja, quando é chamada a rotina que gera a placa periodicamente, ela deve decidir se adiciona ou não uma placa, e essa decisão será baseada nessa probabilidade configurada, que pode ser de: 30%, 50%, 70%, 90% e 100%;
- Tamanho da placa que será exibida, sendo definido em: normal, grande ou muito grande;
- Distribuição das placas na tela: por meio de um grid de 5 colunas e 3 linhas, podemos definir exatamente qual as áreas que a placa será gerada, bem como quais a distribuição, ou seja, qual área será mais exercitada e qual será menos exercitada.

5.2.3 Demais configurações

As outras configurações possíveis para o jogo dizem respeito ao modo do jogo e ao tempo de partida.

O objetivo principal do jogo é trazer maior conforto ao paciente, sendo assim, apesar de ser um jogo projetado para o ambiente VR, ele pode ser utilizado como um jogo 3D tradicional. Sendo assim, o modo VR é uma configuração que pode ser desabilitada dependendo das necessidades de cada paciente.

Além disso, é possível configurar o tempo de cada partida, sendo que cada uma pode durar de 1 minuto até 45 minutos.

5.3 MODOS DE JOGO

Como destacado anteriormente, mesmo tendo uma premissa simples, o AeroVR foi projetado para atender o maior número de pacientes possível, e isso implica também nas diferentes formas como o jogo pode ser utilizado.

O jogo tem em sua base o funcionamento mobile, portanto, independente do modo que se opte por utilizar, todo o processamento e funcionamento base deve ser realizado em um smartphone Android. Entretanto, a própria plataforma Android permite que seja explorada diferentes formas de se jogar, nesse caso nós destacamos outras duas além do uso no próprio smartphone: modo de realidade virtual e com monitor externo.

O primeiro e mais importante para o caso de uso é o modo de realidade virtual. Como descrito anteriormente, esse modo foi desenvolvido utilizando o “Google Cardboard XR Plugin for Unity”, que permite criar e configurar o smartphone como uma tela para realidade virtual. Sendo assim, com o auxílio de um óculos especial podemos utilizar o celular como um óculos de realidade virtual

Apenas a partida em si utiliza a realidade virtual, portanto todas as configurações da partida são feitas através do smartphone de modo convencional. Portanto, para auxiliar o paciente a se localizar espacialmente ao iniciar uma partida, foi criada uma tela, exibida anterior ao início da partida, com as indicações das direções (frente, atrás, esquerda e direita), assim ele pode facilmente se localizar antes de iniciar a partida efetivamente.

Um ponto muito importante a respeito do plugin utilizado é que ele permite que o possamos ativar ou desativar o modo VR. Sendo assim, foi possível utilizar a mesma cena tanto



Figura 5.10: Óculos VR utilizado para o desenvolvimento do AeroVR

para ambos os modos de jogo, e também permitiu que o modo em realidade virtual seja utilizado apenas na cena da partida em si, e não nas demais telas de configuração, por exemplo.

O segundo modo de uso é por meio de um monitor externo. Para esse modo não é realizada nenhuma configuração no jogo em si, apenas espelhar a tela do smartphone em uma tela externa com o auxílio de um computador. Apesar de simples, essa possibilidade é de suma importância, uma vez que mesmo os pacientes que não estão adaptados ao óculos VR podem utilizar o jogo como uma ferramenta para auxiliar em sua reabilitação. Assim como descrito na fundamentação teórica, os monitores tradicionais preenchem uma menor área do campo de visão, ou seja, possuem um menor grau de FoV. Entretanto, em conversas com os profissionais da CETO, foi percorrido que o uso do monitor se torna importante em dois momentos: o primeiro é para a própria adaptação do paciente, ou seja, antes de iniciar o uso do jogo em realidade virtual ele poderá aprender a jogar e a se ambientar utilizando o monitor; e o segundo é para os casos em que o paciente não consegue se adaptar ao uso do óculos VR de fato.

No aspecto prático, para realizar o espelhamento da tela do smartphone em um monitor externo, é necessário o intermédio de um computador. Nós utilizamos a ferramenta *scrcpy* (lê-se *Screen copy*), que está disponível para os principais sistemas operacionais atuais (Genymobile, 2024), conectando o aparelho ao computador é possível exibir facilmente as ações exibidas na tela do aparelho na tela do computador. Outra vantagem dessa ferramenta, é que ela permite que se utilize os periféricos do computador para interagir com o aparelho, permitindo que o paciente utilize o mouse ou o teclado para jogar.

5.3.1 Controles

Assim, vemos que o paciente pode utilizar o jogo de três formas: no próprio smartphone de forma independente, no mesmo aparelho juntamente com um óculos VR ou espelhando o smartphone em um monitor externo. Com essas opções de modo de jogo, também se fez necessário permitir diferentes interfaces de controle do jogo que se adéquem as necessidades do paciente.

Antes de especificar as interfaces utilizadas, é importante definir quais ações o usuário pode tomar durante uma partida. Conforme as mecânicas do jogo especificadas anteriormente, vemos que as ações disponíveis referem-se apenas à movimentação do avião, ou seja, para evitar colisões, o jogador pode deslocar-se apenas para a esquerda ou direita. Assim, em resumo, as únicas ações possíveis no jogo são: permanecer na posição atual, mover-se para a esquerda ou mover-se para a direita.

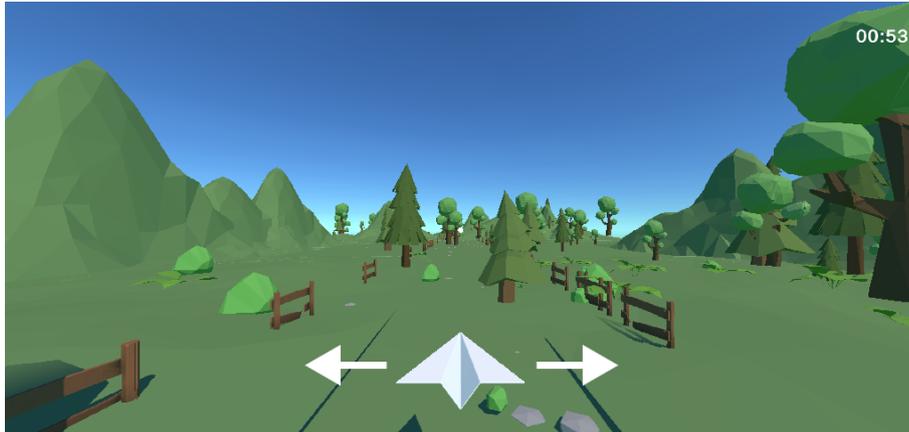


Figura 5.11: Avião com destaque para os movimentos possíveis

Tendo em vista as ações possíveis e as diferentes formas de se jogar o jogo em si, tivemos o desafio de projetar a melhor interface que se adapte a maioria dos pacientes. O público alvo do jogo são paciente pós AVC que possuem um comprometimento do campo visual, entretanto, o acidente vascular cerebral pode gerar múltiplas sequelas, ou seja, os pacientes podem potencialmente ter comprometimentos físicos ou de comunicação. Sendo assim, optamos por permitir diferentes interfaces para captar a escolha de ação do paciente.

A primeira e mais simples é através do próprio smartphone, mais especificamente através do touchscreen. Nesse caso, permitimos que o usuário controle a movimentação deslizando para esquerda ou direita para mover o avião. A segunda e principal interface é através do controle bluetooth para o óculos VR (ver Figura 5.12). Quando conectado à um computador, também é possível utilizar o teclado do computador utilizando as setas esquerda ou direita.

Apesar de estarmos utilizando um controle específico do óculos VR, do ponto de vista prático, esse controle funciona como um mouse. O maior benefício disso é que temos uma maior flexibilidade, podendo utilizar qualquer mouse bluetooth que seja mais adequado para o paciente.

5.4 DADOS

Dada a finalidade do jogo, um aspecto central é como os dados e informações são organizados e guardados.

5.4.1 Jogo

Na Figura 5.13 temos o diagrama com as classes que representam a parte central do jogo, guardando todos os dados de cada usuário, bem como as configurações. Claro que no jogo em si, temos diversos scripts responsáveis por controlar as diferentes ações dos GameObjects presentes na partida, entretanto, esses não são representativos do ponto de vista dos dados do jogo pertinentes para a avaliação do paciente.



Figura 5.12: Controle bluetooth para óculos VR

Com relação ao papel de cada classe no jogo, iniciamos destacando a classe responsável por guardar os dados a respeito dos usuários cadastrados no dispositivo, a classe *Session*. Os principais dados que essa classe é responsável por coordenar é: qual o usuário que está utilizando o jogo no momento e qual a lista de todos os usuários cadastrados no dispositivo.

Depois dessa, temos a classe que representa o paciente no jogo, a classe *User*. Essa é responsável por guardar os dados a respeito do usuário em si, como por exemplo, nome, id e qual a configuração atual do jogo para o paciente. Essa classe em específico, possui alguns métodos utilizados em outros scripts responsáveis pela partida em si.

Já a classe de configurações, a *Configuration*, possui todas as informações de configurações discutidas anteriormente. Essa classe se relaciona com o usuário e com a de status da partida, *GameStatus*. Essa última por sua vez, é responsável por guardar todos os dados de uma partida em específico, como: qual foi a configuração utilizada, quais as colisões que ocorreram, quais as placas que foram exibidas e etc. Todos os dados salvos são exibidos ao final da partida e são apresentados a fim de permitir que o profissional realize a correta avaliação do desempenho do paciente.

Um importante dado que é de responsabilidade da classe *GameStatus* é o score, ou seja, a pontuação do paciente durante uma partida. Essa informação, diferente das outras, não é recebida automaticamente, porque não existe uma forma automática, ou seja, através do jogo para contabilizar as placas que são exibidas e essa verificação é feita manualmente pelo terapeuta ocupacional que está acompanhando o paciente. Sendo assim, esse valor deve ser adicionado ao final da partida pelo profissional responsável.

5.4.2 Dados coletados

Todos os dados coletados pelo AeroVR a respeito de um paciente, são salvos na classe *GameStatus*, e sendo essa classe serializável, os dados dela são convertidos para um arquivo JSON. Portanto,

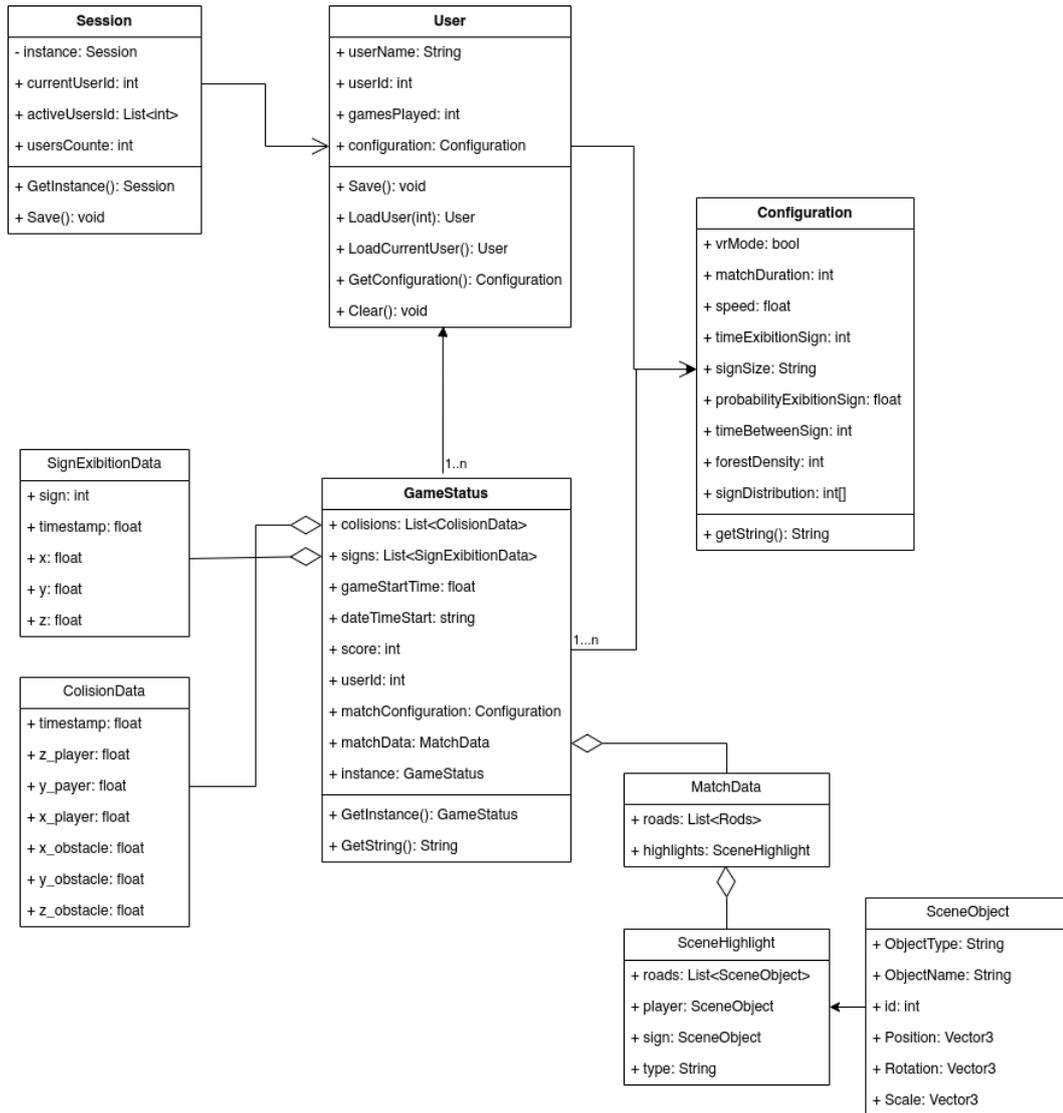


Figura 5.13: Diagrama de classes do AeroVR

para cada paciente a cada partida que ele finalizar será criado um arquivo com os dados dessa partida. Veja abaixo um exemplo simplificado do arquivo gerado:

```

1 {
2   "colisions": [
3     {
4       "timestamp": 8.533864974975586,
5       "x_player": 0.0,
6       "y_player": 2.0,
7       "z_player": -24.96925163269043,
8       "x_obstacle": -1.478133201599121,
9       "y_obstacle": -0.313737690448761,
10      "z_obstacle": -23.04137420654297
11    },
12    ...
13  ],

```

```

14 "signs":[
15   {
16     "sign":3,
17     "timestamp":7.835324764251709,
18     "x":4.441045761108398,
19     "y":17.06761360168457,
20     "z":-5.0
21   },
22   ...
23 ],
24 "gameStartTime":2.6061339378356935,
25 "dateTimeStart":"09/06/2024 20:24:59",
26 "score":7,
27 "game_settings":{
28   "vrMode":false,
29   "matchDuration":1,
30   "speed":1.25,
31   "timeExhibitionSign":200,
32   "signSize":"Grande",
33   "probabilityExhibitionSign":0.30000001192092898,
34   "timeBetweenSigns":1,
35   "forestDensity":2,
36   "signDistribution":[ 4,0,3,6,0,0,0,0,0,0,0,0,0,0,0]
37 },
38 "userId":1
39 }

```

É importante salientar que a biblioteca JsonUtility utilizada para criar todos os arquivos JSON é nativa do próprio Unity. Portanto, não foi necessário adicionar nenhum plugin ou biblioteca externa.

Assim, como temos os dados de cada partida, podemos consolidar os dados de histórico de cada paciente, como veremos a seguir.

5.5 INTERFACE DO USUÁRIO

Até o momento focamos em como foi realizada o desenvolvimento da partida e das parametrizações do jogo. Na sessão atual iremos descrever as demais telas do jogo, que visam definir a configuração da partida e exibir os dados dos pacientes.

Um ponto importante a ser observado é que essas telas foram desenvolvidas para serem utilizadas apenas no modo normal, ou seja, sem o uso da realidade virtual. Além disso, na idealização do fluxo de uso do jogo, foi definido que essas telas serão utilizadas terapeuta ocupacional responsável no momento da terapia, portanto, não são para o uso do paciente em si.

5.5.1 Menu principal

A primeira tela exibida ao entrar no jogo é a da tela de menu principal (ver Figura 5.14). A única informação exibida nessa tela é o do nome do paciente em si, as demais ações são para navegar

para outras telas, são elas: tela da partida (iniciando uma nova partida), configurações, histórico e seleção do paciente.



Figura 5.14: Tela de menu principal

5.5.2 Seleção do paciente

Na tela de seleção do paciente existem dois campos. O primeiro é uma caixa de texto juntamente com um botão indicado como "Adicionar", esse campo é utilizado para adicionar novos pacientes no jogo. Já no segundo campo, logo abaixo do primeiro, é onde selecionamos os pacientes. Para isso, basta selecionar o paciente desejado através da seleção dropdown e clicar no botão "Selecionar".



Figura 5.15: Tela para seleção do paciente

5.5.3 Histórico

A tela de histórico exibe os dados de todas as partidas do paciente em questão, como: quais foram as configurações da partida, quais foram as colisões, placas exibidas na partida e a data e hora de início. As únicas ações disponíveis são: visualizar o playback da última partida, iniciar uma nova partida ou navegar novamente para o menu principal.

Sobre a tela de histórico, é importante ressaltar que essa tela também é exibida ao final de cada partida, entretanto, exibindo apenas os dados referentes à partida. Ao final da partida também é permitido salvar um score, ou seja, a pontuação obtida, através dessa tela.

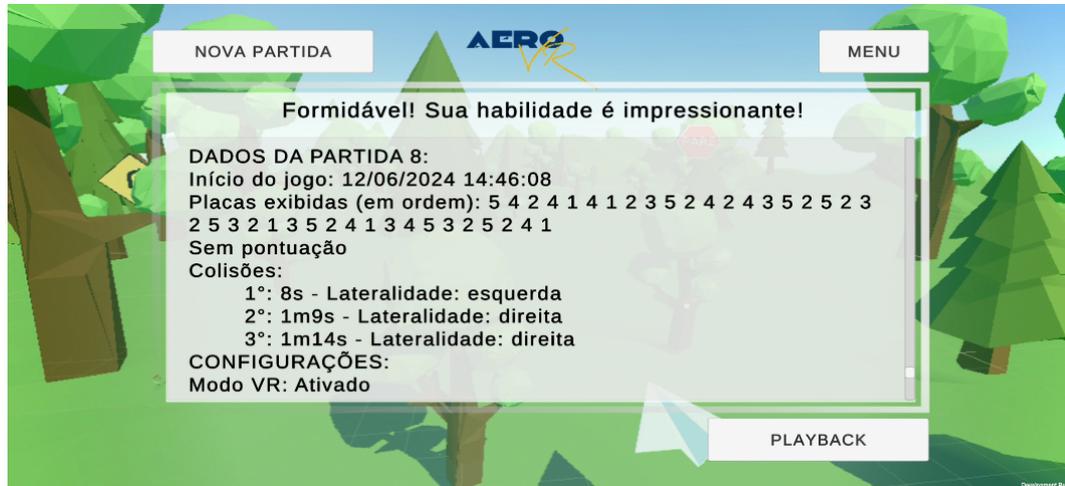


Figura 5.16: Tela de histórico de partidas paciente

5.5.4 Configurações

Sobre as telas de configurações, podemos separar elas em dois grupos: as configurações gerais e as das placas.

A primeira tela é a de configurações gerais (ver Figura 5.17). Através dessa é possível configurar: se o modo em realidade virtual está habilitado ou não, qual a duração da partida, velocidade de movimentação na partida e densidade da floresta. Portanto, são configurações do ambiente e dos aspectos globais da partida.

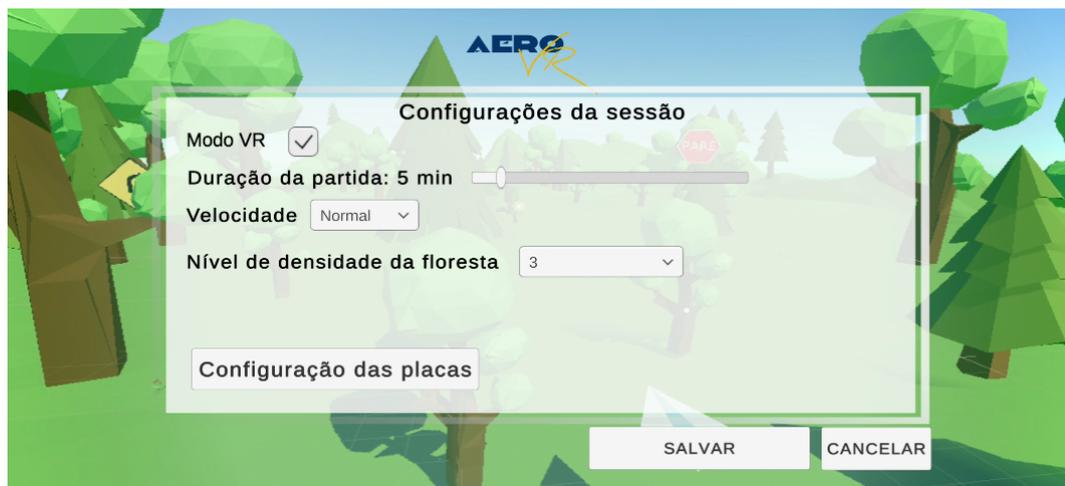


Figura 5.17: Tela para configurações gerais da partida

Por meio das telas de configurações das placas (ver Figura 5.18) é possível configurar: tempo em que uma placa será exibida, probabilidade de exibir uma placa, intervalo entre a geração das placas, tamanho das placas exibidas e navegar para o grid com as configurações referentes a posição das placas na tela (ver Figura 5.19).

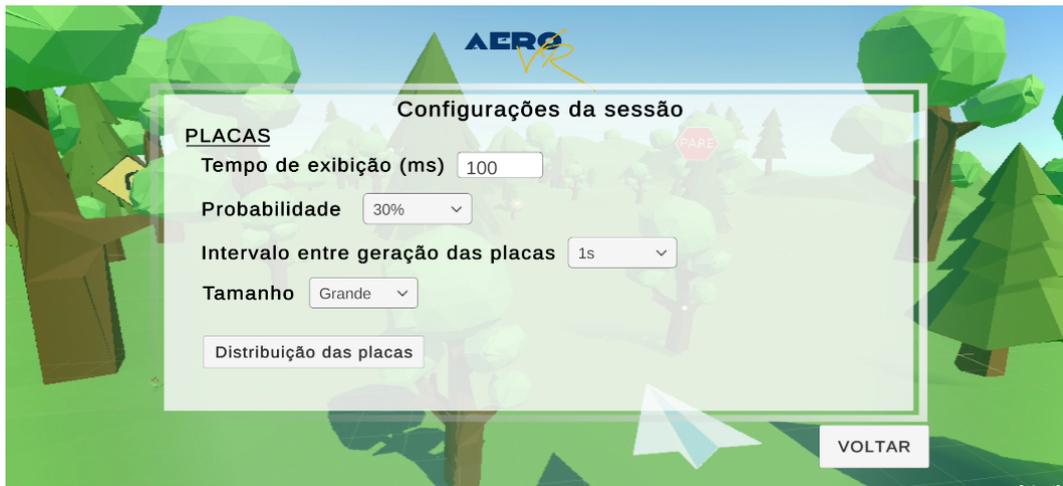


Figura 5.18: Tela para configurações das placas



Figura 5.19: Tela para configurações das posições das placas

6 VALIDAÇÃO E TESTES

Na área de desenvolvimento de software, existem diversos métodos e padrões para testar e validar o correto funcionamento de uma aplicação. Para o desenvolvimento do AeroVR, optamos por utilizar testes exploratórios. Esse método nos permitiu interagir de forma dinâmica com o software, identificando falhas e inconsistências de maneira mais intuitiva e adaptativa.

Os testes exploratórios, consistem em executar o software do ponto de vista do usuário, e com isso tentar encontrar falhas e inconsistências no sistema (Sommerville, 2010). Esse tipo de teste tem como principais vantagens a sua facilidade de aplicação, uma vez que possui uma menor quantidade de etapas formais se comparado a outras técnicas, e a rápida interação entre o teste e a correção dos erros encontrados.

A seguir descreveremos as principais validações do aplicativo que foram realizadas para responder três perguntas fundamentais: quando ativa, o display está aplicando a realidade virtual de forma adequada? Os parâmetros configurados estão sendo aplicados corretamente? Os dados coletados estão conforme o que foi apresentado durante a partida?

6.1 INFORMAÇÕES SOBRE OS TESTES

Todos os testes foram realizados utilizando um smartphone Samsung M21s, modelo SM-F415. As especificações desse dispositivo pode ser visualizada na Tabela 6.1

Característica	Especificação
Modelo	SM-F415
Fabricante	Samsung
Sistema Operacional	Android 11 Samsung One UI Core 3.0
Processador	4x 2.3 GHz Cortex-A73 + 4x 1.7 GHz Cortex-A53
Chipset	SAMSUNG Exynos 9611
GPU	Mali-G72 MP3
Memória RAM	4 GB
Armazenamento Interno	64 GB
Tela	6.4 polegadas
FPS	60Hz
Dimensões	159.2 x 75.1 x 8.9 mm
Peso	191 gramas

Tabela 6.1: Especificações técnicas do smartphone

Durante os testes não havia nenhum outro aplicativo aberto e o celular estava conectado ao Unity no modo debug para coleta de logs.

6.2 DISPLAY

Para validarmos se o jogo, ao estar no modo VR, apresenta uma imagem válida precisamos entender o princípio básico que queremos aplicar. Na estereoscopia, que é a base da realidade

virtual, queremos pegar duas imagens de pontos diferentes da cena. Assim, exibir uma imagem como se fosse o visto pelo olho esquerdo e outra pelo olho direito.

Esse efeito é mais facilmente observado nas bordas das imagens, como na Figura 6.1, em que temos uma imagem do aplicativo no modo VR. Observe que a imagem do lado direito da tela é ligeiramente diferente do lado esquerdo, ou seja, a partir do lado direito temos mais detalhes sobre o que está sendo exibido nesse lado, conforme o esperado.

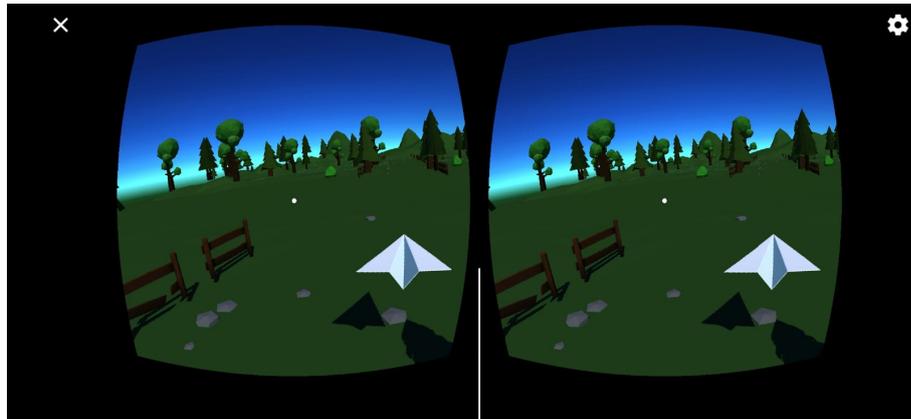


Figura 6.1: Imagem capturada do jogo no modo VR

6.3 CONFIGURAÇÕES

Um importante aspecto a ser validado no AeroVR é se as parametrizações estão funcionando corretamente. Aqui iremos falar sobre as principais parametrizações do sistema, ou seja, as que se relacionam as placas e impactam diretamente na avaliação do paciente.

6.3.1 Tempo de exibição das placas

Aqui queremos verificar dois pontos, se o intervalo de milissegundos configurado para a exibição da placa está sendo respeitado, ou seja, se uma placa não é exibida por mais ou menos tempo do que deveria.

Para realizar esse teste, utilizamos a ferramenta de logs (Technologies, a) e a função Time (Technologies, b), ambas disponibilizadas pelo Unity. Com os logs, podemos programar a escrita de mensagens que serão exibidas no logcat durante a execução do jogo. Já a função *Time.time* permite medir o tempo, em segundos, desde o início da aplicação até o início do frame atual (Technologies, b).

Estamos medindo dois momentos principais: o tempo em que uma placa é gerada e o tempo em que ela é apagada, ambos exibidos na tela de log. Portanto, tendo em vista a função *time* utilizada, é importante frisar que a taxa de atualização da tela é um fator que pode impactar a corretude desse parâmetro. No nosso caso, estamos testando em um dispositivo com uma taxa de atualização de 60 frames por segundo. Assim, podem ocorrer pequenas variações no tempo de exibição, que para valores muito pequenos podem se tornar significativos. Os resultados para 20ms, 100ms e 500ms podem ser visualizados na Tabela 6.2.

Configuração	Média	Desvio padrão	Moda
20ms	18ms	6,48	20ms
100ms	100,1ms	4,17	100,2ms
500ms	501,1ms	4,38	501ms

Tabela 6.2: Medição do tempo de exibição das placas

6.3.2 Probabilidade de exibir uma placa

Uma das configurações disponíveis para as placas é determinar a frequência com que essas são geradas. Essa funcionalidade é controlada por meio de dois parâmetros. O primeiro é o intervalo de tempo, em segundos, em que ocorre uma tentativa de gerar uma placa; o segundo é a probabilidade de que essa placa seja gerada. Portanto, ao configurar um intervalo de 1 segundo, significa que a cada ciclo de 1 segundo será realizada uma tentativa de gerar a placa, que ocorrerá com uma probabilidade definida.

Aqui queremos verificar se essa probabilidade está sendo seguida dentro da margem esperada. Para isso, foi utilizada uma estratégia similar ao caso anterior. Através da função *Debug.Log*, sempre que ocorria uma tentativa era impresso na tela de logs o resultado (placa será gerada ou não). Realizamos o teste para 600 tentativas em cada configuração possível, e o resultado obtido pode ser visualizado na Tabela 6.3.

Configuração	Exibiu	Não exibiu	Média
30%	167	433	27,83%
50%	316	284	52,67%
70%	434	166	72,33%
90%	544	56	90,67%
100%	600	0	100%

Tabela 6.3: Ciclos em que uma placa foi ou não exibida

Observe que apesar da variação, os resultados ficaram dentro do esperado.

6.3.3 Distribuição das placas

O último aspecto a ser validado é com relação à posição das placas na tela. Para essa validação, a partir de duas configurações de grid diferentes, sempre que uma placa era gerada, uma mensagem com o quadrante do grid que ela estava posicionada era exibida nos logs, com isso é possível verificar se está conforme o configurado.

Nas figuras 6.2 e 6.3 temos a configuração utilizada e o resultado obtido com as devidas porcentagens. Observamos que para todos os casos os resultados ficaram dentro do esperado, ou seja, com base na distribuição de pesos para cada quadrante, as placas foram exibidas conforme o configurado.



Figura 6.2: Grid configurado com os pesos tendendo a direita abaixo e os resultados correspondentes acima



Figura 6.3: Grid configurado com os pesos apenas na parte central esquerda e direita e os resultados das medições acima

7 CONCLUSÃO

Esse projeto nos deu a oportunidade de explorar tecnologias para o desenvolvimento de jogos em realidade virtual, com dois fatores muito importantes: os jogos sérios e a aplicação de jogos na terapia ocupacional.

Com relação ao projeto desenvolvido, foi possível completar todo o seu desenvolvimento. O resultado desse projeto foi um jogo, o AeroVR, que pode ser aplicado e utilizado na área de terapia ocupacional para auxiliar o treinamento de pacientes que tiveram perda do campo visual após um acidente vascular cerebral.

Como esse projeto foi totalmente desenvolvido em conjunto com profissionais da Clínica Escola de Terapia Ocupacional da UFPR, ele foi extensamente validado durante todo o seu desenvolvimento, e como resultado temos uma aplicação muito robusta e flexível, podendo ser utilizada em diferentes contextos. Do ponto de vista técnico, foram realizados diversos testes e validações, analisando o comportamento do jogo em diferentes configurações. Os resultados obtidos tanto a partir dos testes técnicos, quanto com a avaliação dos usuários, nos permitem afirmar o correto funcionamento do jogo.

Especificamente com relação ao uso da realidade virtual, os dois modos de jogos (VR e apenas 3D), nos permitiram experimentar o mesmo jogo em diferentes modalidades. Dessa forma, pudemos confirmar os ganhos que a realidade virtual traz com relação à imersão e ao campo de visão preenchido (FoV).

Ainda, o projeto tem o potencial para se constituir em valiosa contribuição para a clínica de Terapia Ocupacional da UFPR, podendo ser utilizado por outras clínicas futuramente.

Com relação aos desenvolvimentos futuros, no aspecto da terapia, passa a ser de suma importância a validação dos resultados obtidos com o AeroVR. Portanto, avaliar:

- Avaliar clinicamente, se o jogo de fato beneficiou o engajamento do paciente na reabilitação;
- Avaliar se o ganho de imersão e FoV permitido pela realidade virtual melhorou também o resultado do tratamento de reabilitação;
- Medir de forma sistemática a satisfação do paciente ao jogar o AeroVR em comparação com o uso aos métodos tradicionais de reabilitação.

Já com relação aos aspectos práticos e técnicos do jogo, as principais melhorias possíveis são:

- Criar uma versão do jogo para os modelos de óculos de realidade virtual mais novos que não dependam do uso do smartphone e possuem um maior FoV;
- A partir da coleta dos resultados com os pacientes reais, com o uso da inteligência artificial, melhorar como o nível de dificuldade da partida é definido, para se adaptar melhor a evolução do paciente;
- Criar novos ambientes para o jogo, melhorando assim a experiência dos jogadores;

REFERÊNCIAS

- Adams, E. (2013). *Fundamentals of Game Design*. Pearson Education.
- Arm Limited (2024). Gaming engines. Accessed: 2024-07-19.
- AtariAge (2024). Bradley trainer support in mame. <https://www.atariage.com/news/Bradley/>. Accessed: 2024-07-16.
- Beukman, M., Cleghorn, C. W. e James, S. (2022). Procedural content generation using neuroevolution and novelty search for diverse video game levels. Em *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22*, página 1028–1037, New York, NY, USA. Association for Computing Machinery.
- Box, V. (2023). Vr box. <http://www.vr-box.es/>. Acessado em 29/11/2023.
- Bryson, S. (2013). Virtual reality: A definition history - A personal essay. *CoRR*, abs/1312.4322.
- Chen, J. (2007). Flow in games (and everything else). *Commun. ACM*, 50(4):31–34.
- Cohen-Peckham, E. (2019). How unity built the world's most popular game engine. <https://techcrunch.com/2019/10/17/how-unity-built-the-worlds-most-popular-game-engine/>. Accessed: 2024-07-31.
- Corporation, V. (2024). The director. Accessed: 2024-06-18.
- Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper and Row, New York, NY.
- Daibert-Nido, M., Pyatova, Y., Cheung, K. G., Nayomi, C., Bouffet, E., Markowitz, S. N. e Reber, M. (2021a). Case report: Visual rehabilitation in hemianopia patients. home-based visual rehabilitation in patients with hemianopia consecutive to brain tumor treatment: Feasibility and potential effectiveness. *Frontier in Neurology*.
- Daibert-Nido, M., Pyatova, Y., Cheung, K. G., Reginald, A., Garcia-Giler, E., Bouffet, E., Markowitz, S. N. e Reber, M. (2021b). An audiovisual 3d-immersive stimulation program in hemianopia using a connected device. *American Journal of case reports*.
- Deterding, S., Dixon, D., Khaled, R. e Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*, 11:9–15.
- Douglas (2024). Top 5 game engines em 2024.
- Edwards, G., Subianto, N., Englund, D., Goh, J. W., Coughran, N., Milton, Z., Mirnateghi, N. e Ali Shah, S. A. (2021). The role of machine learning in game development domain - a review of current trends and future directions. Em *2021 Digital Image Computing: Techniques and Applications (DICTA)*, páginas 01–07.

- Eichinger, F. L. F., Soares, A. V., Noveletto, F., Sagawa Júnior, Y., Bertemes Filho, P. e Domenech, S. C. (2020). Serious game for locomotor rehabilitation of hemiparetic stroke patients. *Fisioterapia em Movimento*, 33:e003316.
- Eye, M. e Ear (2013). Peripheral prism glasses help hemianopia patients get around. Accessed: 2024-06-29.
- Genymobile (2024). scrcpy. <https://github.com/Genymobile/scrcpy>. Accessed: 2024-06-18.
- González-González, C. S., Toledo-Delgado, P. A., Muñoz-Cruz, V. e Torres-Carrion, P. V. (2019). Serious games for rehabilitation: Gestural interaction in personalized gamified exercises through a recommender system. *Journal of Biomedical Informatics*, 97:103266.
- Google Developers (2014). Google cardboard. <https://developers.google.com/vr/discover/cardboard>. Accessed: 2024-07-31.
- Hazelton, C., Pollock, A., Walsh, G. e Brady, M. C. (2018). Scanning training for rehabilitation of visual field loss due to stroke: Identifying and exploring training tools in use. *British journal of Occupational Therapy*, 0:1–10.
- Intel (2016). Virtual reality vs. augmented reality. <https://www.intel.com.br/content/www/br/pt/tech-tips-and-tricks/virtual-reality-vs-augmented-reality.html>. Accessed: 2024-07-16.
- Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T. e Walton, A. (2008). Measuring and defining the experience of immersion in games. *International Journal of Human-Computer Studies*, 66(9):641–661.
- Kasten, E. e Sabel, B. A. (1995). Visual field enlargement after computer training in brain-damaged patients with homonymous deficits: an open pilot trial. *Restorative Neurology and Neuroscience*, 8(3):113–127.
- Laamarti, F., Eid, M. e El Saddik, A. (2014). An overview of serious games. *International Journal of Computer Games Technology*, 2014(1):358152.
- Lee, M. S. (2009). Visual rehabilitation strategies for homonymous hemianopia. Accessed: 2024-06-29.
- Leitner, M. C., Guetlin, D. C. e Hawelka, S. (2021). Salzburg visual field trainer (svft): A virtual reality device for (the evaluation of) neuropsychological rehabilitation. *PLoS ONE*.
- Lidestam, B., Eriksson, L. e Eriksson, O. (2019). Speed perception affected by field of view: Energy-based versus rhythm-based processing. *Transportation Research*.
- M, M., LC, R., C, M., P, M., Pedatella, DC, B., R, P., OM, P.-N., J, O.-F., GR, F., GS, S., MC, L. e SMO, M. (2023). Números do avc. <https://avc.org.br/numeros-do-avc/>. Acessado em 30/11/2023.
- Ma, M. e Bechkoum, K. (2008). Serious games for movement therapy after stroke. Em 2008 *IEEE International Conference on Systems, Man and Cybernetics*, páginas 1872–1877.

- Machado, L. d. S., Moraes, R. M. d., Nunes, F. d. L. d. S. e Costa, R. M. E. M. d. (2011). Serious games baseados em realidade virtual para educação médica. *Revista Brasileira de Educação Médica*, 35:254 – 262.
- Maclean, N. e Pound, P. (2000). A critical review of the concept of patient motivation in the literature on physical rehabilitation. *Social science and medicine (1982)*, 50:495–506.
- Malik, O. (2016). Pokémon go will make you crave augmented reality. *The New Yorker*. Accessed: 2024-07-18.
- McFadzean, R. M. (2006). Novavision: vision restoration therapy. *Current Opinion in Ophthalmology*, 17(6):498–503.
- Meta (2023). Meta quest 3. <https://www.meta.com/quest/quest-3/>. Accessed: 2024-07-16.
- Microsoft (2019). Microsoft hololens 2. <https://www.playstation.com/pt-br/ps-vr2/>. Accessed: 2024-07-16.
- Nelles, G., Esser, J., Eckstein, A., Tiede, A., Gerhard, H. e Diener, H. (2001). Compensatory visual field training for patients with hemianopia after stroke. *Neuroscience Letters*, 306(3):189–192.
- NovaVision (2024). Vision restoration therapy (vrt). Accessed: 2024-06-29.
- Pambakian, A. M. R. C. P., Currie, J. M. B. B. S., e Kennard, C. P. (2005). Rehabilitation strategies for patients with homonymous visual field defects. *Journal of Neuro-Ophthalmology*, 25(2):136–142.
- Rossi, P. W., Kheyfets, S. e Reding, M. J. (1990). Fresnel prisms improve visual perception in stroke patients with homonymous hemianopia or unilateral visual neglect. *Neurology*, 40(10):1597–1599.
- Rowe, F. J. (2017). Vision in stroke cohort: Profile overview of visual impairment. *Brain and Behaviour*, 7(11).
- SA, B. T. (2019). Blair witch. https://store.steampowered.com/app/1092660/Blair_Witch/. Acessado em 17/06/2024.
- Schell, J. (2008). *The Art of Game Design: A Book of Lenses*. CRC Press, Boca Raton, FL.
- Schoenau-Fog, H. (2011). The player engagement process: An exploration of continuation desire in digital games. Em *Proceedings of the 2011 DiGRA International Conference: Think Design Play*, Ballerup, Denmark. Digital Games Research Association.
- Shiratori (2020). Shiratori's low poly forest. <https://docs.unrealengine.com/marketplace/en-US/product/shiratori-s-low-poly-forest>. Acessado em 17/06/2024.
- Sommerville, I. (2010). *Software Engineering*. Pearson Education, 9 edition.
- Sony (2023). Playstation vr2. <https://www.microsoft.com/pt-br/hololens>. Accessed: 2024-07-16.

- Strivr (2023). What is vr? <https://www.strivr.com/blog/what-is-vr>. Accessed: 2024-07-16.
- Susi, T., Johannesson, M. e Backlund, P. (2007). Serious games: An overview. *IKI Technical Reports*.
- Sweetser, P. e Wyeth, P. (2005). Gameflow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, 3:3.
- Tamborini, R. e Skalski, P. (2006). The role of presence in the experience of electronic games. *Playing video games: Motives, responses, and consequences*, páginas 225–240.
- Technologies, U. Android debugging on an android device. <https://docs.unity3d.com/Manual/android-debugging-on-an-android-device.html>. Accessed: 2024-07-29.
- Technologies, U. Time. <https://docs.unity3d.com/ScriptReference/Time.html>. Accessed: 2024-07-29.
- Togelius, J., Champanand, A., Lanzi, P. L., Mateas, M., Paiva, A., Preuss, M. e Stanley, K. (2013). Procedural content generation: goals, challenges and actionable steps. *Dagstuhl Follow-Ups*, 6:61–75.
- Unity Technologies. 2021 unity game report. <https://create.unity.com/2021-game-report>. Accessed: 2024-07-31.
- Unity Technologies. Unity gaming report. <https://unity.com/pt/resources/gaming-report>. Accessed: 2024-07-31.
- Unity Technologies. Unity industry. <https://unity.com/pt/industry>. Accessed: 2024-07-31.
- Unity Technologies. Unity manual: Creating and using scripts. <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>. Accessed: 2024-07-31.
- Unity Technologies. Unity manual: Gameobject. <https://docs.unity3d.com/Manual/class-GameObject.html>. Accessed: 2024-07-31.
- Vriend, S. (2017). Intrinsic and extrinsic motivation. <https://www.gamedeveloper.com/design/intrinsic-and-extrinsic-motivation>. Acessado em: 12 de julho de 2024.
- Weinbaum, S. G. (1935). *Pygmalion's Spectacles*.
- World Health Organization (2024). Stroke, cerebrovascular accident. Accessed: 2024-06-29.
- World Stroke Organization (WSO) (2022). Global stroke fact sheet 2022. Relatório técnico, World Stroke Organization. Accessed: 2024-06-29.
- Zubek, R. (2020). *Elements of Game Design*. MIT Press.