

Bruno Key Kawano

Treinamento para Radiologistas Utilizando Realidade Virtual

São Paulo, SP

2024

Bruno Key Kawano

Treinamento para Radiologistas Utilizando Realidade Virtual

Trabalho de conclusão de curso apresentado
ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Ricardo Nakamura

São Paulo, SP

2024

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Kawano, Bruno

Treinamento para Radiologistas Utilizando Realidade Virtual / B. Kawano
-- São Paulo, 2024.

53 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Realidade Virual 2.Radiologia I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Este trabalho é dedicado aos meus pais Carlos Alberto Kenji Kawano e Ana Claudia Shimofusa Kawano e a Leticia Miyuki Ito, sem eles, nada disso seria possível.

Agradecimentos

Os agradecimentos principais são direcionados aos meus pais, que sempre me apoiaram, a Leticia Miyuki Ito, que se manteve ao meu lado por tantos anos, ao professor Ricardo Nakamura, que me orientou e auxiliou na criação deste projeto e ao Sr. Fernando Razuck, que vem auxiliando o projeto em diversos aspectos.

Resumo

O objetivo do projeto será o de produzir um simulador, em realidade virtual, de exames radiológicos, inicialmente voltados para o raio X, mas que podem ser eventualmente expandidos a exames PET-CT. O software será posteriormente testado por profissionais de radiologia para validar a sua aplicação em treinamentos.

Palavras-chave: radiologia. realidade virtual. raio X. treinamento.

Abstract

The objective of the project is to develop a virtual reality simulator for radiological examinations, initially targeting X-rays but potentially extending to PET-CT scans. Subsequently, the software will undergo testing by radiology professionals to authenticate its suitability for training purposes.

Keywords: radiology. virtual reality. X-ray. training.

Lista de abreviaturas e siglas

RV	Realidade Virtual
RA	Realidade Aumentada
IRD	Instituto de Radioproteção e Dosimetria
HUD	Head-Up Display
TEP-TC	Tomografia por emissão de positrons - Tomografia computadorizada
MRD	Merck Sharp & Dohme
AMB	Associação Médica Brasileira
UI	User Interface

Sumário

1	INTRODUÇÃO	15
1.1	Motivação	16
1.2	Objetivos	16
1.3	Justificativa	16
1.4	Organização do Trabalho	16
2	ASPECTOS CONCEITUAIS	19
2.1	Seções do Trabalho	19
2.2	Conceitos e Tecnologias Relevantes para o Projeto	19
2.2.1	Realidade Virtual	20
2.2.1.1	Imersão	20
2.2.2	Motores de Jogos	21
2.2.3	Dispositivos de visualização imersiva em realidade virtual	21
2.2.3.1	Meta Quest	22
2.2.3.2	Head-Up Display (HUD)	22
2.2.4	Unity	22
2.2.4.1	Raycasting	23
2.2.4.2	Open XR	23
2.2.4.3	Pro Builder	23
2.2.5	Blender	24
3	MÉTODO DO TRABALHO	25
3.1	Processo de Implementação	25
3.1.1	Especificação	25
3.1.2	Desenvolvimento	25
3.1.3	Validação Inicial	25
3.1.4	Refinamento e Validação Final	25
3.2	Métodos de Implementação das Principais Features	25
4	ESPECIFICAÇÃO DE REQUISITOS	27
4.1	Especificação Geral	27
4.1.1	Principais Features	27
4.1.2	Cenários	27
4.1.3	O Exame	27
4.1.4	Itens Interagíveis	28
4.2	Requisitos Técnicos	28

5	DESENVOLVIMENTO DO TRABALHO	31
5.1	Tecnologias Utilizadas	31
5.1.1	Unity	31
5.1.2	Blender	31
5.2	Projeto e Implementação	32
5.2.1	Head's-Up Display (HUD)	32
5.2.2	Contador de Radiação Absorvida	34
5.2.2.1	Atualização Contínua e Dinâmica	34
5.2.2.2	Raycasting e Field of View	34
5.2.3	Visualizador de Radiação Ionizante	35
5.2.4	Realidade Virtual	38
5.2.5	Objetivos e Passo a Passo do Exame	38
5.3	Testes e Avaliação	41
5.3.1	Validação Inicial	41
5.3.2	Validação Final	41
6	CONSIDERAÇÕES FINAIS	43
6.1	Conclusões do Projeto de Formatura	43
6.2	Contribuições	43
6.3	Perspectivas de Continuidade	44
	REFERÊNCIAS	45

1 Introdução

Durante seu treinamento, médicos em formação precisam experienciar ao menos uma vez cada um dos campos da medicina. Por ser um trabalho delicado, que envolve diretamente o bem estar de outras pessoas, cada uma das frentes de pesquisa possui seus riscos associados. Para mitigar estes riscos, são estabelecidas diversas regras, como o acompanhamento dos alunos por profissionais graduados ou até mesmo instruções detalhadas da realização do exame.

No entanto, dentre todas as especializações da medicina, a radiologia se destaca como uma das de mais alto risco por envolver a manipulação de equipamentos capazes de emitir radiação ionizante. A complexidade dos equipamentos utilizados, unido à característica invisível da radiação ionizante, dificulta a compreensão dos verdadeiros riscos por alunos e médicos. Além de que pequenos deslizes na operação dos equipamentos podem ter graves consequências para os pacientes, mas principalmente para os operadores, que aplicam exames com frequência.

De acordo com o manual da MSD (Merck Sharp & Dohme), durante exames como radiografias, raio-X e tomografias, o paciente pode ser exposto a doses de radiação de 0,02 mSv (radiografia de tórax posterior) até 7,7 mSv (tomografia de abdome e pélvis). Considerando que a AMB (Associação Médica Brasileira) recomenda uma exposição anual máxima de profissionais da área radiológica de até 20 mSv, sem exceder uma média ponderada de 50 mSv em 5 anos, fica evidente que a realização inadequada desses exames, desde a configuração do equipamento, até o correto posicionamento do operador na sala, podem acarretar em riscos altos à saúde, especialmente em casos de múltiplas exposições. (A., 2024) (SCHWARCKE, 2024)

Além disso, profissionais do centro cirurgico do Hospital São Paulo da UNIFESP relatam que muitas vezes, principalmente quando exames raio X são realizados durante cirurgias, nenhuma das diretrizes de segurança são seguidas pelos médicos. Isso mostra o grau de despreocupação que esses profissionais tem com relação à radiação ionizante e seus perigos.

Neste documento, está descrito o projeto de formatura dos alunos autores, que visa sanar alguns dos problemas apresentados nesta introdução por meio de softwares que auxiliem os operadores em melhor compreender os riscos relacionados aos exames radiológicos.

1.1 Motivação

Considerando os riscos e desafios estabelecidos na introdução, o projeto busca aprimorar a compressão, tanto dos alunos em formação quanto dos operadores, sobre os riscos envolvidos nos exames e sobre o comportamento da radiação ionizante presente neles.

Além disso, mais uma das motivações do projeto foi a presença abundante de simuladores de operações médicas feita em realidade virtual, mas que apresentam uma falta de abordagem didática, que permita que os usuários compreendam mais profundamente o funcionamento dos exames.

1.2 Objetivos

O principal objetivo deste projeto e da presente monografia é, respectivamente, desenvolver um simulador de exames de raio-X que ofereça aos operadores em treinamento a possibilidade de vivenciar esses procedimentos em um ambiente virtual, eliminando os riscos associados à exposição real à radiação. O simulador busca não apenas recriar o ambiente de exames com precisão, mas também integrar funcionalidades inovadoras que permitam aos usuários visualizar de forma intuitiva e interativa o comportamento da radiação ionizante, promovendo uma compreensão mais profunda sobre sua dinâmica e seus efeitos.

Além do desenvolvimento do simulador em si, esta monografia tem como objetivo descrever detalhadamente a metodologia empregada, a especificação técnica utilizada e o processo de implementação do projeto. O documento destaca os aspectos técnicos e conceituais que fundamentaram o desenvolvimento, evidenciando as aplicações práticas do simulador e seu funcionamento, além de demonstrar como ele pode ser uma ferramenta efetiva no treinamento seguro e avançado de operadores e radiologistas em formação.

1.3 Justificativa

Principalmente durante seu treinamento, operadores de exames radiológicos ficam expostos a diversos riscos relacionados principalmente com a radiação ionizante. O projeto então, entra como uma forma de permitir que os operadores aprendam sobre o exame e sobre a radiação ionizante, sem que eles sejam inseridos em situações de risco.

1.4 Organização do Trabalho

A monografia observada está dividida nos seguintes capítulos:

-
- 1 - Introdução: contextualização e justificativa do projeto e monografia;
 - 2 - Aspectos Conceituais: apresentação das seções do trabalho e definição dos conceitos utilizados;
 - 3 - Método do trabalho: definir a metodologia de desenvolvimento e pesquisa utilizada na implementação do projeto;
 - 4 - Especificação de requisitos: introduzir todo o escopo e requisitos técnicos do projeto;
 - 5 - Desenvolvimento do trabalho: demonstrar como foi feita a implementação técnica do projeto, detalhando o desenvolvimento do software; e
 - 6 - Considerações finais: discutir o que foi alcançado ou não pelo projeto desenvolvido, além de demonstrar possibilidades para o futuro do projeto.

2 Aspectos Conceituais

2.1 Seções do Trabalho

As seções representativas do projeto têm como objetivo destacar os aspectos mais relevantes e significativos do trabalho desenvolvido, servindo como um panorama detalhado e coerente dos principais componentes do simulador e de sua aplicação prática. Essas seções foram estruturadas de forma a fornecer uma visão abrangente do projeto, desde sua concepção inicial até a implementação e validação, permitindo ao leitor compreender não apenas o funcionamento técnico, mas também a relevância e os resultados obtidos.

Entre as seções representativas, destaca-se a especificação técnica do simulador, que detalha os principais elementos implementados, como o HUD (Heads-Up Display), o contador de radiação em forma de barra de vida e o modo de visualização da radiação ionizante. Esses componentes são descritos em profundidade, com atenção especial às tecnologias empregadas, como o Unity e as APIs OpenXR, e às decisões de design que guiaram sua implementação.

Outra seção fundamental aborda o desenvolvimento do ambiente virtual, descrevendo como o software foi projetado para recriar cenários realistas de exames de raio-X, priorizando a interação intuitiva e a segurança do treinamento. Nessa parte, são discutidas as ferramentas utilizadas, os desafios enfrentados e as soluções adotadas para integrar elementos interativos de forma eficiente e funcional.

Por fim, a avaliação do simulador reúne os resultados dos testes realizados com alunos de medicina, evidenciando a usabilidade, o impacto no aprendizado e as possíveis melhorias para versões futuras. Essa seção reflete o alinhamento entre os objetivos propostos e os resultados obtidos, consolidando a utilidade do simulador no treinamento de radiologistas.

Essas seções representativas conectam os aspectos técnicos e conceituais do projeto, proporcionando uma visão clara de sua execução e de seu potencial para contribuir significativamente no campo do ensino e treinamento em radiologia.

2.2 Conceitos e Tecnologias Relevantes para o Projeto

Nesta seção estão definidos os conceitos mais utilizados ao longo do projeto, além de, de algumas features e tecnologias utilizadas no projeto.

2.2.1 Realidade Virtual

A Realidade Virtual é uma tecnologia que permite a criação de ambientes digitais imersivos, nos quais os usuários podem interagir de forma sensorial, visual e auditiva, como se estivessem presentes fisicamente no espaço virtual. Por meio de dispositivos como óculos de RV e controladores manuais, os usuários são transportados para cenários digitais que respondem aos seus movimentos e ações, proporcionando uma experiência de presença e imersão que não seria possível apenas por meio de uma tela tradicional.

Na RV, o ambiente é totalmente simulado por computador, diferentemente de tecnologias como a Realidade Aumentada (RA), que combina elementos virtuais com o mundo real. A RV cria uma sensação de “imersão”, onde o campo de visão, a percepção de profundidade e as interações em tempo real são projetadas para envolver o usuário completamente, muitas vezes enganando até os sentidos. Devido a esse nível de envolvimento, a RV é especialmente útil para treinamentos em áreas de alta precisão, como a saúde e a engenharia, onde é necessário praticar habilidades em ambientes controlados e realistas, porém sem riscos.

A realidade virtual já é amplamente utilizada na área da saúde como ferramenta de treinamento, principalmente em cirurgias de alta precisão, e também já é considerada uma possível ferramenta com aplicações na radiologia e também na medicina nuclear. (MACHADO, 2023)

2.2.1.1 Imersão

A imersão se refere ao grau de precisão do software em criar no usuário a ilusão de realidade. O grau de imersão é mensurado de acordo com as seguintes variáveis:

- Qualidade da imagem: realismo e fidelidade da renderização das imagens, incluindo resolução, taxa de atualização, qualidade das texturas e nível de detalhamento;
- Campo de visão: todo o campo de visão que o usuário tem;
- Estereoscopia: Possibilidade de fornecer imagens distintas para cada um dos olhos, criando a ilusão de profundidade;
- Rastreamento: graus de liberdade, precisão, tempo de resposta e outros atributos de qualidade do sistema de rastreamento;
- Abrangência: qualidade dos diferentes tipos de modalidade sensorial apresentadas ao usuário;
- Combinação: grau de congruência entre as diferentes modalidades sensoriais;

- **Envolvimento:** extensão em que os sentidos são envolvidos (ex.: rastreamento ou não da cabeça);
- **Interatividade:** capacidade do usuário interferir com o ambiente ao seu redor, além da resposta dos elementos do ambiente a essa interação; e
- **Enredo:** qualidade da narrativa apresentada.

(TORI, 2020)

2.2.2 Motores de Jogos

Motores de jogos fornecem ao desenvolvedor diversas ferramentas, funcionalidades e bibliotecas de código que simplificam o desenvolvimento de jogos e outras aplicações interativas. Eles são responsáveis por renderizar o mundo virtual, simular a física, reproduzir sons, interpretação de código entre outros.

Entre as características de motores de jogos, existem alguns principais que podem ser destacados:

- **Motor gráfico:** lida com a exibição de imagens, 2D ou 3D;
- **Motor de física:** simula a física do mundo virtual, desde a gravidade, até a reflexão de raios de luz;
- **Motor de áudio:** gerencia todos os sons do simulador, como música e efeitos sonoros;
- **Interpretador e compilador:** permite o desenvolvimento em linguagens de nível mais baixo de código, o que permite a criação de diversos tipos de interação; e
- **Motor de animação:** lida com toda a movimentação de objetos no mundo virtual;

(LOVERIDGE, 2024)

2.2.3 Dispositivos de visualização imersiva em realidade virtual

Como toda aplicação da área de processamento gráfico, a RV requer dispositivos de entrada, saída e de processamento.

Os dispositivos de entrada devem ser capazes de identificar diversos tipos de alterações no ambiente real, e, principalmente quando se trata de RV, é necessário que os dispositivos de entrada tenham um alto grau de liberdade, ou seja, precisam ser capazes de entender entradas complexas, como movimentos da cabeça e das mãos, distância do chão e das paredes, além dos próprios botões e gatilhos expalhados pelo hardware.

Por outro lado, quando tratamos dos dispositivos de saída, o foco se torna outro, a imersividade, cujos requisitos foram definidos anteriormente. Um detalhe importante da imersão gerada pelos equipamentos de visualização em RV é a estereoscopia, onde cada olho recebe uma imagem levemente diferente para dar a sensação de profundidade ao usuário. vale mencionar que esse é o mesmo mecanismo utilizado na natureza, por nós seres humanos e muitos outros animais, para gerar a perspectiva de profundidade e distância. (TORI, 2020)

2.2.3.1 Meta Quest

O Meta Quest é uma linha de headsets de RV desenvolvida pela empresa Meta Platforms, anteriormente conhecida como Facebook que se enquadra como um desses equipamentos de visualização imersiva. Os dispositivos Meta Quest, também chamados de "headsets VR standalone"(autônomos), são sistemas de VR all-in-one que não requerem conexão a um computador ou console de jogos para funcionar. A linha Meta Quest é composta por dispositivos de Rv de alta qualidade que têm como objetivo popularizar o uso da Realidade Virtual, tornando a tecnologia mais acessível e imersiva para consumidores e empresas.

O Meta Quest representa uma evolução significativa no mundo da Realidade Virtual, proporcionando uma experiência sem fio e autossuficiente, o que simplifica o acesso a conteúdo imersivo e reduz as barreiras tecnológicas que anteriormente existiam. Os dispositivos são equipados com sensores, câmeras e processadores embutidos, permitindo o rastreamento de movimento e controle intuitivo do ambiente virtual. Além disso, possuem uma loja digital própria, onde é possível adquirir jogos, aplicativos, experiências educacionais e outras utilidades.

2.2.3.2 Head-Up Display (HUD)

O Head-Up Display (HUD) é uma interface visual que permite a exibição de informações diretamente no campo de visão do usuário, de forma que ele possa acessar dados essenciais sem precisar desviar a atenção da tarefa em andamento. O conceito do HUD é proveniente da aviação e é amplamente utilizado no mundo dos jogos, que inspiraram a implementação desta feature. Uma das características mais importantes do HUD é que as imagens e dados são projetados em uma superfície transparente, o que permite que a apresentação da informação não interfira excessivamente com a visão do usuário.

2.2.4 Unity

O Unity é uma plataforma de desenvolvimento multiplataforma, amplamente conhecida e utilizada principalmente na criação de jogos, mas que também se estende a

aplicações em diversas outras áreas, como realidade aumentada (RA), realidade virtual (RV), simulações, treinamento e visualização de projetos arquitetônicos e de engenharia.

A estrutura do Unity se baseia em componentes modulares que facilitam a criação e manipulação de elementos 3D e 2D, além de permitir uma grande personalização e interatividade através de scripts de programação. O Unity suporta várias linguagens de programação, mas o C# é a linguagem principal para scripts, proporcionando flexibilidade e poder para desenvolver desde simulações físicas complexas até sistemas de inteligência artificial.

Além disso, o Unity conta com uma Asset Store, um repositório de conteúdos onde os desenvolvedores podem acessar modelos 3D, texturas, animações, scripts prontos e outros recursos que podem ser usados para agilizar o desenvolvimento. Essa vasta gama de recursos contribui para que o Unity seja uma das ferramentas mais versáteis do mercado, com aplicações em educação, medicina, simulação industrial, jogos, além de seu uso consolidado no Interlab, laboratório do qual o professor orientador deste projeto é membro. (STAFF, 2024)

2.2.4.1 Raycasting

O método `Physics.Raycast` é nativo do Unity e serve para emitir um raio no ambiente 3D ou 2D. Esse raio parte de um ponto de origem, segue em uma direção definida por um vetor e percorre até uma distância máxima. Durante seu trajeto, ele detecta todos os colisores atingidos e retorna diversas informações sobre eles. Esse método foi utilizado na implementação de algumas funcionalidades do Unity (API, 2024)

2.2.4.2 Open XR

O OpenXR é um padrão de API aberto isento de royalties da Khronos, fornecendo aos motores de jogos acesso nativo a uma variedade de dispositivos em todo o espectro de realidade mista. O Open XR é integrado diretamente com o Unity, e possibilita o desenvolvimento de jogos e simuladores em RV, com a integração aos dispositivos de visualização adequados. (TURNER, 2024)

2.2.4.3 Pro Builder

Pro Builder é um recurso do Unity que permite a modelagem de estruturas de forma simplificada. Ele também possui uma integração as APIs de Scripting do Unity, o que permite a criação de código em C# para customizar e adicionar novas funcionalidades ao pacote. (MANUALS, 2024)

2.2.5 Blender

O Blender é uma ferramenta de código aberto e gratuita para criação de conteúdo tridimensional (3D), amplamente utilizada para modelagem, animação, escultura digital, texturização, renderização e edição de vídeo e foi escolhido como principal ferramenta de modelagem do projeto. O Blender é um software altamente valorizado em diversas áreas, como cinema, design, jogos, engenharia, e arquitetura. Devido ao amplo uso do software por usuários e desenvolvedores, o Blender também conta com uma extensa biblioteca de modelos gratuitos, que podem ser encontrados online.

Entre as funcionalidades do Blender, destacam-se a criação de modelos complexos em 3D, o suporte a animações e rigging (configuração de esqueleto para animação de personagens), a escultura digital e a criação de texturas e materiais detalhados. O Blender também possui um motor de renderização próprio, chamado Cycles, que permite criar imagens e vídeos com alta fidelidade visual. Além disso, ele conta com suporte a scripts em Python, o que oferece uma enorme flexibilidade para automatizar tarefas e desenvolver extensões personalizadas.

3 Método do trabalho

3.1 Processo de Implementação

O desenvolvimento do projeto foi dividido em algumas etapas, as quais serão destacadas a seguir e detalhadas ao longo da monografia.

3.1.1 Especificação

Durante o primeiro semestre de 2024 foi realizado o detalhamento dos requisitos técnicos, além do mapeamento das principais features necessárias para a entrega do simulador.

3.1.2 Desenvolvimento

Neste passo, foi realizada a implementação do simulador no motor Unity, utilizando como base diversos conceitos provenientes do mundo dos jogos, como será detalhado mais tarde neste documento. Durante esta etapa, foi feito gradualmente o desenvolvimento com acompanhamento do Sr. Fernando Razuck.

3.1.3 Validação Inicial

Nesta etapa foram realizadas avaliações pelo professor Ricardo Nakamura e pelo Sr. Fernando Razuck. Ela ocorreu em paralelo com o desenvolvimento com o intuito de garantir que o simulador atenderia as necessidades dos radiologistas em treinamento.

3.1.4 Refinamento e Validação Final

Ao fim do desenvolvimento do projeto, ele será avaliado por profissionais da radiologia fornecidos pelo professor Fernando Razuck, essa avaliação será utilizada para refinar o software e para medir os resultados alcançados pelo projeto.

3.2 Métodos de Implementação das Principais Features

Foram utilizados diversos conceitos relacionados ao desenvolvimento de jogos. Nesta sessão serão evidenciadas as analogias utilizadas para criar cada uma das features do simulador.

- HUD: o conceito de HUDs é muito utilizado no desenvolvimento de jogos e foi utilizado para montar o indicador de radiação absorvida e adicionar instruções em tela para o usuário;
- Conceito de "Barra de vida": mais um conceito popular em jogos, que foi utilizado para gerar o contador de radiação absorvida; e
- Line of sight: é um tipo de comportamento muito aplicado no desenvolvimento de jogos que foi utilizado para definir as zonas seguras de radiação na sala de exame.

4 Especificação de Requisitos

4.1 Especificação Geral

Nesta subseção será explorada a especificação das funcionalidades e objetivos do projeto.

4.1.1 Principais Features

Foram observadas diversas possibilidades de features para o simulador, seguindo uma certa ordem de prioridade. A implementação dessas features será detalhada no capítulo 5. Desenvolvimento.

- Head's Up Display (HUD);
- Contador de radiação absorvida;
- Modo de visão para radiação ionizante; e
- Itens Interagíveis;

4.1.2 Cenários

O simulador terá alguns cenários específicos de exame:

- Ambiente "sandbox": o primeiro cenário será um onde o usuário poderá explorar as funcionalidades e comportamentos do simulador, sem que hajam pacientes ou quaisquer objetivos específicos.
- Exame comum: com o paciente já deitado e em posição, esse cenário envolve tirar o raio X torácico.

4.1.3 O Exame

Para termos em mão um simulador o mais fiel possível a realidade, serão tomados como referência alguns artigos sobre os exames, bem como sobre o comportamento da radiação. Para os exames, foram tomadas como base, referências e especificações fornecidas pelo Sr. Fernando Razuck, onde o passo a passo do exame será o seguinte:

- Passo 1: com o paciente já deitado na mesa de raio X, o operador deve levar a ele o colete de chumbo para tireoide;

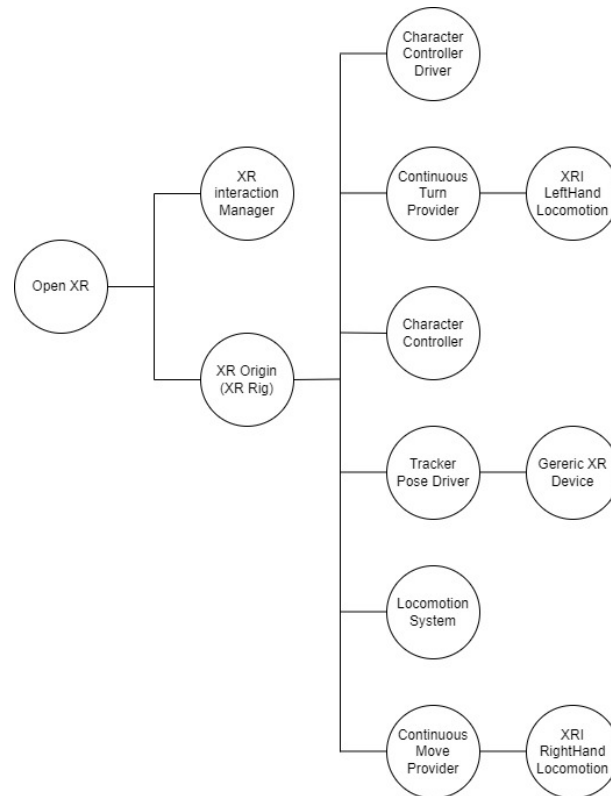


Figura 1 – Árvore de recursos para criação do ambiente em RV

- Passo 2: o operador deve sair da sala de raio X para a sala separada, de onde ele poderá ativar a câmera; e
- Passo 3: o operador deve ativar a câmera, se mantendo dentro da sala segura até o fim do exame.

4.1.4 Itens Interagíveis

Serão implementados itens no ambiente virtual com os quais o usuário poderá interagir, sendo o principal o colete de chumbo para tireoide.

4.2 Requisitos Técnicos

O projeto foi desenvolvido com o motor e jogos Unity, utilizando o conjuntos de APIs Open XR para realizar a implementação da realidade virtual. Além disso, foram utilizados diversos scripts de código em C# para implementar todas as lógicas e comportamentos essenciais ao simulador. Ademais, foi utilizado o Pro Builder para realizar a modelagem do ambiente de simulação, como a sala e alguns móveis.

Além do Unity e Pro Builder, o Blender foi utilizado como principal ferramenta de modelagem 3D para a criação de diversos objetos da cena.

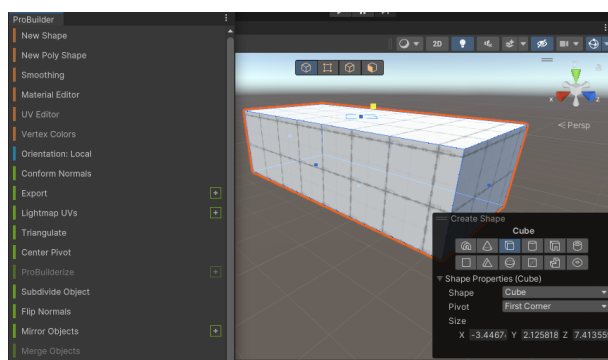


Figura 2 – Visualização do Pro Builder

5 Desenvolvimento do Trabalho

5.1 Tecnologias Utilizadas

Para o projeto, foram escolhidas algumas tecnologias principais, destacadas abaixo, junto da motivação para essas escolhas. A definição dessas tecnologias foi feita no capítulo 2. Aspectos Conceituais.

5.1.1 Unity

O Unity foi escolhido como motor de simulação do projeto, sendo o responsável por diversos aspectos fundamentais do simulador. Ele cuida da renderização dos ambientes e objetos, gerencia as características físicas do simulador - como gravidade, atrito e movimentação - e realiza a compilação dos scripts de código em C#, que definem a lógica de implementação das funcionalidades do simulador. Além disso, o Unity centraliza a gestão de recursos, como a importação de modelos 3D, texturas e materiais, facilitando a integração e o desenvolvimento do projeto.

5.1.2 Blender

O Blender foi utilizado no projeto para realizar a modelagem de objetos utilizados nos exames, além da edição de modelos obtidos da Asset Store do Unity. Além disso, as animações do projeto foram implementadas via um sistema de armaduras, nativo do Blender. As armaduras são uma feature do Blender que permite implementar animações de modelos 2D e 3D com o uso de "ossos"(bones) que funcionam como pontos de articulação dos modelos.

Esse foi o software escolhido para modelagem devido a presença ao amplo uso por outros usuários, que gera uma ampla rede de modelos pré prontos, tutoriais e conhecimento, permitindo um desenvolvimento mais fluido do projeto.

- Blender: foi escolhido como ferramenta de modelagem para realizar a edição de objetos e, caso necessário, a modelagem de objetos.
- Realidade Virtual: o treinamento será realizado em realidade virtual para gerar uma sensação maior de imersividade.
- Meta Quest 2 e 3: o projeto será testado e desenvolvido para o funcionamento para o óculos de realidade virtual da Meta. Estes foram os óculos escolhidos devido a sua disponibilidade no Interlab.

Vale ressaltar que a maioria dos objetos contidos no simulador serão obtidos online, a partir de repositórios de licença gratuita.

5.2 Projeto e Implementação

Nesta seção serão detalhados os metodos de implementação das principais features encontradas no simulador.

5.2.1 Head's-Up Display (HUD)

Foram implementados no total, 4 HUDs diferentes no simulador, que se apresentam ao usuário a medida que ele progride pelo exame. Entre eles estão os HUDs de:

- **Introdução:** exhibe ao usuário algumas informações sobre o simulador, incluindo o mapeamento dos botões, além das possíveis opções de cenários que podem ser selecionados;
- **Sandbox:** surge quando o usuário selecionar o modo sandbox, o informando sobre alguns detalhes desse modo;
- **Falha:** se no modo "Exame" o usuário cometer algum erro na ordem de ações, esse HUD se exhibirá, informando o usuário que o mesmo cometeu um erro e o convidando a reiniciar o simulador; e
- **Sucesso:** apresentado somente quando, no modo "Exame", o usuário conclui todo o passo a passo de forma correta, informando-o que realizou o exame corretamente e o convidando a reiniciar o simulador.

Esses HUDs foram implementados utilizando as ferramentas de UI integradas diretamente ao Unity, proporcionando uma base sólida e versátil para o desenvolvimento da interface. Essa integração permitiu que os elementos do HUD fossem adaptados de maneira eficiente às tecnologias de realidade virtual, garantindo uma experiência imersiva e funcional. As funcionalidades incluem tanto a visualização dos elementos em tempo real quanto a interação direta do usuário com os botões na tela, permitindo que ações e comandos fossem realizados de forma intuitiva e responsiva.

Na hierarquia do projeto, como demonstrado na Figura 4, os HUDs foram cuidadosamente posicionados abaixo do componente "XR Rig". Essa estrutura desempenha um papel fundamental no rastreamento preciso da posição do usuário no ambiente virtual, o que foi essencial para garantir que os HUDs permanecessem sempre dentro do campo de visão do operador. Essa configuração estratégica permitiu criar uma experiência fluida, eliminando a necessidade de ajustes manuais constantes e garantindo que as informações

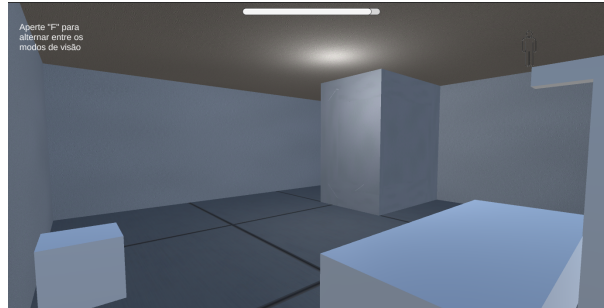


Figura 3 – Exemplo do HUD

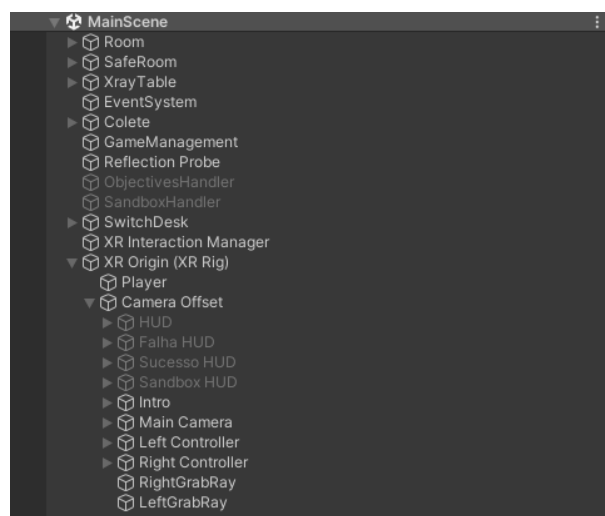


Figura 4 – Hierarquia do projeto

relevantes fossem exibidas de maneira consistente, independentemente do movimento ou orientação do usuário.

Além disso, o posicionamento hierárquico no projeto foi planejado para otimizar a integração entre os elementos da interface e o restante do ambiente virtual. Essa abordagem assegurou que o HUD mantivesse uma relação estável com o "XR Rig", adaptando-se automaticamente às mudanças no ponto de vista do operador, sem comprometer a clareza ou a funcionalidade. Dessa forma, o design contribuiu para uma navegação eficiente e para a redução da carga cognitiva durante a interação com o simulador, maximizando a usabilidade.

Com essa estrutura, foi possível criar uma experiência coesa em que os HUDs se tornaram uma extensão natural do ambiente virtual, proporcionando ao operador acesso constante às informações e ferramentas necessárias para o treinamento. A interação contínua e fluida entre o HUD, o XR Rig e as ferramentas de realidade virtual solidifica o projeto como uma solução prática e eficaz para simulações educacionais em radiologia.

5.2.2 Contador de Radiação Absorvida

O contador de radiação absorvida pelo operador, é uma das ferramentas visuais inseridas no Head-Up Display (HUD) do simulador. Esse contador simula a dose de radiação acumulada pelo operador ao longo dos procedimentos radiológicos, exibindo uma representação intuitiva e direta do impacto da exposição no operador em tempo real. Esse recurso é inspirado no conceito de “barra de vida” frequentemente utilizado em jogos digitais, que serve para monitorar a saúde do personagem, e aqui foi adaptado para simular a condição do operador em um ambiente radiológico.

O principal objetivo com a implementação dessa propriedade é o de educar o operador sobre as principais práticas de proteção radiológica, indicando uma maior absorção quando o operador estiver próximo a fonte de radiação, ou quando o profissional não se encontra protegido por barreira física. Além disso, o contador evidencia ao operador que apesar de da exposição em um único exame ser relativamente pequena, ao longo de diversos exames, essa radiação se acumula.

A implementação do contador foi feita utilizando-se de duas principais ferramentas do motor Unity.

5.2.2.1 Atualização Contínua e Dinâmica

A função "Update()" do Unity permite que o contador seja atualizado a cada frame da simulação. Dessa forma podemos fazer com que o contador esteja sempre atualizado no campo de visão do usuário.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DealDamage : MonoBehaviour
6 {
7     public void SendDamage (int dam)
8     {
9         PlayerHealth playerStats = GameObject.FindGameObjectWithTag("
            Player").GetComponent<PlayerHealth>();
10        playerStats.TakeDamage(dam);
11    }
12 }
```

5.2.2.2 Raycasting e Field of View

Para permitir que barreiras físicas reduzam a quantidade de radiação absorvida foi utilizada mais uma mecânica muito comum em jogos, o campo de visão. Caso o operador se encontre em uma posição fora da visão do emissor de raio X, o contador não indicará

absorção de radiação ionizante. Para implementar essa funcionalidade foram utilizadas as funções de Raycast() do Unity.

```
1     if (Physics.Raycast(XRayLight.transform.position, rayDirection,
2         out RaycastHit hit, 100.0f, layersToHit))
3     {
4         if (boolXRay & (hit.transform.position.x == player.transform.
5             position.x) & (hit.transform.position.z == player.transform.
6                 position.z))
7         {
8             damageManager.SendDamage(intDamage);
9         }
10    }
```

5.2.3 Visualizador de Radiação Ionizante

No simulador haverá um modo de visão para visualização da radiação ionizante, uma ferramenta implementada que permite que o operador visualize a presença da radiação ionizante associada aos exames de imagem. Esse recurso proporciona uma representação gráfica dos níveis de radiação no ambiente de simulação, tornando a radiação “visível” aos olhos do operador, o que normalmente seria impossível no mundo real. Esse modo de visão foi projetado para enriquecer a experiência de aprendizado dos radiologistas, auxiliando-os a compreender a distribuição da radiação e seu impacto no ambiente e nos indivíduos presentes, além de permitir a identificação de zonas e risco e zonas seguras com facilidade.

A implementação desse aspecto foi feita fazendo uso de um jogo de luzes no simulador, onde com um botão o operador pode alternar a renderização das fontes de luz, e com outro controlar o emissor. Isso permite que o usuário possa então "iluminar" a cena somente com a fonte que simula a radiação. Isso também permitiu que controlássemos quais objetos podem bloquear a radiação, simulando então barreiras de radiação e objetos transparentes à radiação.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System;
5 using UnityEngine.XR.Interaction.Toolkit;
6 using UnityEngine.InputSystem;
7
8 public class ToggleLight : MonoBehaviour
9 {
10
11     public GameObject XRayLight;
12     public GameObject SpotLight;
13     public GameObject RoomLight;
```

```
14     public int intDamage;
15     private bool boolXRay;
16     private bool boolRoomLight;
17     public DealDamage damageManager;
18     public GameObject player;
19     public LayerMask layersToHit;
20     public InputActionProperty XRayButton, LightButton;
21
22     // Start is called before the first frame update
23     void Start()
24     {
25         XRayLight.SetActive(false);
26         Spotlight.SetActive(false);
27         boolXRay = false;
28         RoomLight.SetActive(true);
29         boolRoomLight = true;
30     }
31
32     // Update is called once per frame
33     void Update()
34     {
35         var rayDirection = player.transform.position - XRayLight.
            transform.position;
36         //If que controla se o raio X est ligado ou n o
37         if(XRayButton.action.WasPressedThisFrame())
38         {
39             if(boolXRay == true)
40             {
41                 XRayLight.SetActive(false);
42                 Spotlight.SetActive(false);
43                 boolXRay = false;
44             }
45             else if (boolXRay == false)
46             {
47                 if(boolRoomLight == false)
48                 {
49                     XRayLight.SetActive(true);
50                     Spotlight.SetActive(true);
51                 }
52                 boolXRay = true;
53             }
54         }
55
56         //If que controla se a vis o normal ou do raio X
57         if(LightButton.action.WasPressedThisFrame())
58         {
59             if(boolRoomLight == true)
```

```
60     {
61         RoomLight.SetActive(false);
62         boolRoomLight = false;
63         if(boolXRay == true)
64         {
65             XRayLight.SetActive(true);
66             Spotlight.SetActive(true);
67         }
68     }
69     else
70     {
71         RoomLight.SetActive(true);
72         boolRoomLight = true;
73         if(boolXRay == true)
74         {
75             XRayLight.SetActive(false);
76             Spotlight.SetActive(false);
77         }
78     }
79 }
80
81 //Verifica se a luz est  ligada e causa dano ao player de
82 //acordo
83 if (Physics.Raycast(XRayLight.transform.position,
84     rayDirection, out RaycastHit hit, 100.0f, layersToHit))
85 {
86     if(boolXRay & (hit.transform.position.x == player.
87         transform.position.x) & (hit.transform.position.z ==
88         player.transform.position.z))
89     {
90         damageManager.SendDamage(intDamage);
91     }
92 }
93
94 public bool XRayState()
95 {
96     return boolXRay;
97 }
98 }
```

É possível observar no trecho de código acima, que o simulador observará qual das luzes está ligada ou desligada no momento. Tendo essa informação, o script então aguarda algum input do usuário para realizar ou a ativação do raio X, ou a troca do modo de visão. é possível observar também que a luz que representa o raio X só vai se apresentar como ativa, caso o modo de visão seja o correto.

5.2.4 Realidade Virtual

O ambiente em realidade virtual foi implementado utilizando-se do conjunto de APIs OpenXR, uma escolha que proporcionou grande flexibilidade e eficiência na criação de uma experiência imersiva e interativa. Por meio dessa tecnologia, foi possível mapear com precisão os movimentos do usuário, garantindo que cada ação realizada fosse replicada no ambiente virtual de forma fiel e responsiva. Esse mapeamento detalhado não apenas elevou o realismo do simulador, mas também permitiu a adaptação a diferentes contextos de treinamento, promovendo uma experiência rica e personalizada.

Além disso, o uso do OpenXR facilitou o desenvolvimento de interações naturais das mãos do usuário com os objetos presentes na sala virtual. Essa funcionalidade foi essencial para criar um ambiente onde os radiologistas em treinamento pudessem manipular ferramentas e realizar ações de maneira intuitiva, simulando as atividades que realizariam em situações reais. As mãos do usuário foram configuradas para interagir com diversos elementos, como botões, equipamentos e instrumentos médicos, permitindo uma prática realista e funcional.

Outra funcionalidade habilitada foi a integração com os controles do Meta Quest, que proporcionaram um meio adicional de interação no ambiente virtual. Através desses controles, foi possível observar e configurar o uso de diferentes botões para desempenhar funções específicas, como selecionar opções no HUD, ajustar parâmetros ou interagir com os objetos simulados. Essa integração garantiu que o simulador fosse não apenas tecnicamente avançado, mas também acessível e intuitivo para os usuários, contribuindo diretamente para o sucesso do treinamento.

Ao unir o mapeamento de movimentos, a interação precisa com objetos e a utilização otimizada dos controles, o ambiente virtual alcançou um alto nível de imersão, proporcionando aos radiologistas uma plataforma robusta e eficiente para aprimorar suas habilidades de forma segura e interativa.

5.2.5 Objetivos e Passo a Passo do Exame

Para o desenvolvimento dos objetivos foi utilizado um script que monitora constantemente todas as variáveis do exame, como a posição do operador, a presença de radiação ionizante e o posicionamento do colete de tireóide.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.SceneManagement;
6 using UnityEngine.XR.Interaction.Toolkit;
7 using UnityEngine.InputSystem;
```



```
8
9 public class ObjectivesHandler : MonoBehaviour
10 {
11     public GameObject objColete, Player, XRay, FailedCanva, HUDCanva,
        BemvindoCanva, SucessoCanva;
12     public TMPro.TextMeshProUGUI TextoColete, TextoXRay, TextoSala;
13     public bool boolColetePosicionado, boolPlayerPosicionado,
        boolXRayAtivado;
14     public LayerMask layersToHit;
15     private bool boolXRayLight, boolFailed, boolSucesso;
16     public InputActionProperty ResetButton;
17
18     // Start is called before the first frame update
19     void Start()
20     {
21         boolXRayAtivado = false;
22         boolFailed = false;
23         boolSucesso = false;
24     }
25     // Update is called once per frame
26     void Update()
27     {
28         var rayDirection = Player.transform.position - XRay.transform
            .position;
29         ToggleLight LightState = GameObject.FindGameObjectWithTag("
            XRay").GetComponent<ToggleLight>();
30         boolXRayLight = LightState.XRayState();
31
32         if (ResetButton.action.WasPressedThisFrame())
33         {
34             SceneManager.LoadScene("MainScene");
35         }
36
37         if (objColete.transform.position == new Vector3(1.6f, 1.3f,
            -4.75f))
38         {
39             boolColetePosicionado = true;
40             TextoColete.text = "- <s>Entregue o colete ao paciente</s
                >";
41         } else{
42             boolColetePosicionado = false;
43             TextoColete.text = "- Entregue o colete ao paciente";
44         }
45         if (Physics.Raycast(XRay.transform.position, rayDirection,
            out RaycastHit hit, 100.0f, layersToHit))
46         {
```

```
47         if((hit.transform.position.x == Player.transform.position
48             .x) && (hit.transform.position.z == Player.transform.
49                 position.z))
50             {
51                 boolPlayerPosicionado = false;
52                 TextoSala.text = "- Entre na sala segura";
53             }
54         else
55         {
56             boolPlayerPosicionado = true;
57             TextoSala.text = "- "+"<s>"+"Entre na sala segura"+"
58                 </s>";
59         }
60     }
61     if (boolXRayLight)
62     {
63         boolXRayAtivado = true;
64         TextoXRay.text = "- <s>Ligue o raio X</s>";
65     } else{
66         boolXRayAtivado = false;
67     }
68     if (((boolXRayAtivado && !(boolPlayerPosicionado)) || (
69         boolXRayAtivado && !boolColetePosicionado) || (boolFailed)
70         ) && !boolSucesso)
71     {
72         boolFailed = true;
73         FailedCanva.SetActive(true);
74         HUDCanva.SetActive(false);
75         BemvindoCanva.SetActive(false);
76     } else if ((boolXRayLight && boolColetePosicionado &&
77         boolPlayerPosicionado) || boolSucesso)
```

É possível observar acima que, além de monitorar a completude do exame, o script também é responsável por apresentar os diferentes HUDs ao usuário, dependendo da situação e controlar as dicas apresentadas ao operador.

O gerenciador de objetivos utiliza do método Raycast da mesma forma que o contador de radiação absorvida. Além disso, ele verifica o posicionamento do colete a partir da distância direta entre o colete e o paciente.

5.3 Testes e Avaliação

5.3.1 Validação Inicial

A princípio, com a ajuda do Sr. Fernando Razuck, validamos os principais aspectos técnicos do exame raio X. O Sr. Fernando Razuck também providenciou diversas referências para uso no projeto. Além de acompanhar seu desenvolvimento, para garantir a fidelidade da simulação.

5.3.2 Validação Final

Os testes foram conduzidos em múltiplos níveis para garantir a qualidade e a funcionalidade do simulador. Durante a fase de desenvolvimento, foi implementado um mecanismo de movimentação utilizando mouse e teclado, permitindo que os testes fossem realizados sem a necessidade do uso dos óculos de realidade virtual. Essa abordagem facilitou o processo iterativo, permitindo ajustes rápidos e eficientes no software antes da integração completa com os dispositivos de RV.

Após o desenvolvimento do software, alunos de medicina em formação foram convidados a participar de uma avaliação prática do simulador. Durante os testes, os estudantes puderam interagir com o sistema e experimentar as funcionalidades, fornecendo uma visão valiosa sobre sua usabilidade, precisão e realismo. Para coletar feedback estruturado, foi elaborado um formulário de avaliação, o qual foi aplicado logo após o uso do simulador.

Os resultados obtidos foram encorajadores. Todos os participantes (100%) consideraram o software simples de utilizar, destacando sua interface intuitiva. Além disso, 80% dos alunos acreditaram que o simulador facilitaria o aprendizado dos exames, enquanto cerca de 90% expressaram que o software tem utilidade prática e apontaram o potencial de soluções similares para outros tipos de exames radiológicos. Esses dados reforçam a relevância do projeto e seu impacto positivo no treinamento de futuros profissionais de saúde.

Por outro lado, os estudantes também identificaram oportunidades de melhoria. Entre os pontos mais mencionados estão a necessidade de aprimorar a modelagem do ambiente virtual e de ajustar alguns detalhes no fluxo dos passos do exame. Esse feedback fornece diretrizes valiosas para o refinamento do projeto, indicando caminhos para tornar o simulador ainda mais eficaz e alinhado às necessidades de seus usuários.

6 Considerações Finais

6.1 Conclusões do Projeto de Formatura

Com base no parecer dos alunos de medicina entrevistados, ficou evidente que o software desenvolvido possui grande utilidade prática nos treinamentos de radiologistas, especialmente no contexto de capacitação técnica para a operação de equipamentos radiológicos. A simplicidade e a intuitividade do sistema foram destacadas como pontos fortes, permitindo um aprendizado rápido e uma interação fluida por parte dos usuários.

Além da avaliação sobre a usabilidade do software, o processo de desenvolvimento do simulador também trouxe insights relevantes sobre as ferramentas utilizadas. Ficou particularmente claro o potencial do Unity como um motor de simulação versátil, capaz de atender a uma ampla gama de aplicações em diferentes áreas. Apesar de sua complexidade inicial, o Unity se mostrou uma plataforma robusta, com uma extensa biblioteca de recursos e funcionalidades que aceleraram o desenvolvimento.

Essas características permitiram que o projeto fosse conduzido com maior eficiência, reduzindo o tempo necessário para implementar componentes avançados e possibilitando a criação de um software sofisticado dentro do cronograma estipulado para este projeto de formatura. A experiência reforçou o papel do Unity como uma ferramenta indispensável não apenas para a criação de simuladores em realidade virtual, mas também como uma solução poderosa para outras áreas que demandam sistemas interativos e complexos.

6.2 Contribuições

Todo o desenvolvimento do código, assim como a modelagem dos elementos mais simples do ambiente virtual, foi realizado quase que inteiramente pelo grupo, com o auxílio de recursos externos disponíveis online. Durante o processo, fóruns da comunidade Unity voltados para realidade virtual foram uma importante fonte de suporte, especialmente para a resolução de dúvidas específicas e a exploração de funcionalidades avançadas. Além disso, playlists de vídeos no YouTube sobre a implementação do OpenXR no Unity também desempenharam um papel relevante, fornecendo exemplos práticos e guias que ajudaram a acelerar o progresso do projeto. ([TUTORIALS, 2024](#))

É igualmente importante destacar as valiosas contribuições externas recebidas ao longo do desenvolvimento. O professor Dr. Ricardo Nakamura e o Sr. Fernando Razuck foram figuras-chave nesse processo, fornecendo materiais técnicos e teóricos que foram fundamentais para a viabilização do projeto. Suas orientações não apenas ajudaram a

consolidar os fundamentos técnicos, mas também ofereceram insights que enriqueceram a implementação e a qualidade final do simulador. O suporte fornecido foi indispensável para superar os desafios encontrados e para garantir que o projeto alcançasse os padrões desejados.

6.3 Perspectivas de Continuidade

Tendo em mente as conclusões positivas obtidas a partir da implantação do projeto, há uma ampla gama de possibilidades para sua continuidade e expansão. Primeiramente, é possível identificar diversas oportunidades de melhorias no software desenvolvido especificamente para o exame de raio-X. Essas melhorias podem incluir a otimização da interface do usuário, a integração de novos parâmetros de simulação e o refinamento das métricas de desempenho para fornecer feedback ainda mais detalhado aos operadores durante o treinamento.

No entanto, as possibilidades de continuidade vão muito além do exame de raio-X. Há um vasto campo a ser explorado em outros procedimentos radiológicos que apresentam riscos similares ou até maiores. Um exemplo claro é o TEP-CT (Tomografia por Emissão de Pósitrons combinada com Tomografia Computadorizada), que envolve doses significativamente maiores de radiação e exige um treinamento ainda mais detalhado para garantir a segurança do operador e do paciente. Outra área promissora é a medicina nuclear, que engloba diversos procedimentos que utilizam materiais radioativos para diagnóstico e tratamento. A simulação desses exames em realidade virtual poderia oferecer uma plataforma segura para que os profissionais adquirissem experiência prática sem os riscos associados à exposição real.

Além disso, o projeto pode evoluir para incluir outros avanços tecnológicos, como a integração de inteligência artificial para análise automática de erros durante os procedimentos simulados ou a personalização das experiências de treinamento com base no desempenho individual do usuário. Também há a possibilidade de expandir o alcance do simulador para treinamentos multiprofissionais, permitindo a colaboração de equipes compostas por radiologistas, técnicos e outros profissionais de saúde em um ambiente simulado.

Por fim, a expansão do projeto também poderia contemplar o desenvolvimento de módulos educativos focados em aspectos específicos da proteção radiológica, como estratégias avançadas de minimização da dose ou o impacto acumulativo da radiação no longo prazo. Com essas abordagens, o simulador se tornaria uma ferramenta cada vez mais abrangente e indispensável no treinamento de profissionais da área médica.

Referências

A., M. F. *Doses típicas de radiação*. [S.l.], 2024. Disponível em: <<https://www.msmanuals.com/pt/profissional/pages-with-widgets/tabelas?mode=list>>. Citado na página 15.

API, U. S. *Physics.Raycast*. 2024. Documentação. Disponível em: <<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>>. Acesso em: 03 dez 2024. Citado na página 23.

LOVERIDGE, S. *The most crucial part of video-game development explained*. 2024. Blog post. Disponível em: <<https://www.gamesradar.com/what-is-a-game-engine-and-what-does-it-do/>>. Acesso em: 03 dez 2024. Citado na página 21.

MACHADO, D. M. The use of virtual reality as an aid tool in the implementation of a radiopharmacy center in a nuclear medicine service. *Brazilian Journal of Radiation Sciences*, v. 11, n. 4, p. 01–28, 2023. Citado na página 20.

MANUALS, U. *About ProBuilder*. [S.l.], 2024. Disponível em: <<https://docs.unity3d.com/Packages/com.unity.probuilder@6.0/manual/index.html>>. Acesso em: 03 dez. 2024. Citado na página 23.

SCHWARCKE, M. M. B. Níveis de taxa de dose para profissionais ocupacionalmente expostos durante exames com tecnécio-99m. *Colégio Brasileiro de Radiologia e Diagnóstico por Imagem*, v. 41, n. 0, p. 54–55, 2024. Citado na página 15.

STAFF, C. *Meta Quest*. 2024. Blog post. Disponível em: <<https://www.coursera.org/articles/what-is-unity>>. Acesso em: 03 dez 2024. Citado na página 23.

TORI, R. *Introdução a Realidade Virtual e Aumentada*. 3. ed. Porto Alegre: Editora SBC, 2020. Acesso em: 29 nov 2024. Citado 2 vezes nas páginas 21 e 22.

TURNER, A. *OpenXR*. 2024. Documentação. Disponível em: <<https://learn.microsoft.com/pt-br/windows/mixed-reality/develop/native/openxr>>. Acesso em: 03 dez 2024. Citado na página 23.

TUTORIALS, V. *OpenXR*. 2024. Vídeo. Disponível em: <<https://www.youtube.com/watch?v=fM0k2n7u8sc&list=PLpEoiloH-4eP-OKItF8XNJ8y8e1asOJud&index=1>>. Acesso em: 10 jan 2024. Citado na página 43.