

**UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

MAURÍCIO ALMEIDA DOS SANTOS

**UMA EXPERIÊNCIA IMERSIVA NO CENTRO DE CIÊNCIAS EXATAS E
TECNOLOGIA DA UNIVERSIDADE FEDERAL DO MARANHÃO**

**SÃO LUÍS-MA
2019**

MAURÍCIO ALMEIDA DOS SANTOS

**UMA EXPERIÊNCIA IMERSIVA NO CENTRO DE CIÊNCIAS EXATAS E
TECNOLOGIA DA UNIVERSIDADE FEDERAL DO MARANHÃO**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Tiago Bonini Borchardt

**SÃO LUÍS-MA
2019**

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Santos, Maurício Almeida dos.

Uma Experiência Imersiva no Centro de Ciências Exatas e
Tecnologia da Universidade Federal do Maranhão / Maurício
Almeida dos Santos. - 2019.

79 f.

Orientador(a): Tiago Bonini Borchartt.

Curso de Ciência da Computação, Universidade Federal do
Maranhão, Auditório do Departamento de Informática - UFMA,
2019.

1. Android. 2. Cardboard. 3. Realidade Virtual. I.
Borchartt, Tiago Bonini. II. Título.

Dedico este trabalho a minha família e meus amigos que sempre me incentivaram e apoiaram.

AGRADECIMENTOS

Agradeço primeiramente à minha família. Em especial, à minha esposa Emilly e minha filha Júlia, por serem minha motivação para estar sempre buscando aprimoramento e fortalecimento. À minha irmã Valéria, por estar sempre me apoiando e a meus pais pelas experiências que me ajudaram a chegar a ser quem sou.

Sou muito grato a meu Primo André e ao curso LUMUS por permitirem a aplicação da pesquisa em suas dependências.

Agradeço também ao orientador professor Tiago Bonini por toda a sua ajuda e paciência.

"Se um dia tiver que escolher entre o mundo e o amor lembre-se: se escolher o mundo ficará sem o amor, mas se escolher o amor com ele você conquistará o mundo."

Albert Einstein

RESUMO

As tecnologias para Realidade Virtual e Aumentada apresentam desenvolvimento acelerado nos anos recentes. O uso, implementação e possível impacto desta tecnologia no interesse de jovens pelas áreas que a possibilitaram guiam este estudo. O trabalho apresenta conceitos à partir dos quais se fez possível o surgimento dos dispositivos para a tecnologia de realidade virtual como conhecida atualmente, as exigências, métodos e softwares para o desenvolvimento de uma aplicação dessa natureza para a plataforma Android e Cardboard. Estuda ainda a impressão e o conhecimento de jovens acerca das ciências exatas e como se relacionam com a tecnologia de Realidade Virtual. O estudo é feito à partir de aplicação de questionário através do qual é possível verificar alterações do interesse em ciências exatas, avaliação da aplicação pelos usuários, assim como a capacidade do jovem de relacionar conceitos primordiais ao funcionamento da aplicação e suas respectivas disciplinas.

Palavras-chave: realidade virtual, android, cardboard.

ABSTRACT

Augmented Technologies for Virtual and Augmented Reality show accelerated development in recent years. The use, implementation and possible impact of this technology in the interest of young people in the areas that made it possible guides this study. The work presents concepts from which the emergence of devices for virtual reality technology as it is known today, the requirements, methods and software for the development of an application of this nature for the Android platform and Cardboard. It also studies the impression and knowledge of young people about the exact sciences and how they relate to the VR technology. The study is done by questionnaire application through which it is possible to verify changes in interest in exact sciences, assessment of the application by users, as well as the the young person's ability to relate essential concepts for the application operation and their respective disciplines.

Keywords: virtual reality, android, cardboard.

LISTA DE FIGURAS

Figura 1 – Gráfico para o interesse(número de buscas) no tópico Realidade Virtual no Brasil entre janeiro de 2015 e dezembro de 2017	14
Figura 2 – Gráfico para o interesse no tópico Realidade Virtual no Brasil entre dezembro de 2017 e fevereiro de 2018	15
Figura 3 – Ilustração da reta numérica e posição na reta	17
Figura 4 – Ilustração Ponto P no plano cartesiano	18
Figura 5 – Ilustração do cubo no plano tridimensional	19
Figura 6 – Imagem de exemplo básico de arquivo OBJ.	20
Figura 7 – Captura de Tela da importação resultante do Exemplo básico obj(cubo) no Blender. ¹	21
Figura 8 – Ilustração do pipeline de renderização 3D	22
Figura 9 – Foto de jarro do CCET	23
Figura 10 – Imagem de objeto com alta densidade de polígonos	23
Figura 11 – Imagem de objeto com otimizado, com baixo número de polígonos.	24
Figura 12 – Foto promocional do protótipo do Sensorama	26
Figura 13 – The Dawn of Interactive Computing (O Alvorecer da Computação Interativa)	29
Figura 14 – Imagem ilustrando a utilização de gesto para interação com celular	29
Figura 15 – Passos para escaneamento de objeto 3D por aplicativo	31
Figura 16 – Pipeline para criação de uma aplicação de RV interativa com modelos 3D	33
Figura 17 – Imagem Ilustrando o funcionamento de um Botão fusível(<i>Fuse Buttons</i>)	36
Figura 18 – Amostra do conjunto de plantas baixas da UFMA	38
Figura 19 – Imagem da UFMA, modelo tridimensional, visto de cima.	39
Figura 20 – Imagem da UFMA vista de cima, planta baixa.	40
Figura 21 – Foto à esquerda e imagem do modelo à direita.	41
Figura 22 – Fotos da UFMA	41
Figura 23 – Planta baixa do CCET ao fundo e arestas do modelo tridimensional	43
Figura 24 – Captura de Tela da interface de exportação do DWGSee	44
Figura 25 – Ajuste de Perspectiva no Gimp	45
Figura 26 – Amostra de pinceis do Krita	45
Figura 27 – Imagem mapeada em um cubo à esquerda e imagem aberta no Inkscape à direita	46
Figura 28 – Interface do 3DSMAX, localização da função de exportação	47
Figura 29 – Imagem das primitivas do Blender	49
Figura 30 – Exemplos de operações sobre a malhas no Blender	49
Figura 31 – Captura de tela do menu de operações para malhas no Blender.	50
Figura 32 – Modelo com modificadores desabilitados à esquerda e habilitados à direita	51
Figura 33 – Captura de tela do menu de modificadores para objetos no <i>Blender</i>	51

Figura 34 – Captura de tela do menu de pincéis para escultura no blender e dragão chinês esculpido	52
Figura 35 – Exemplos de otimização	53
Figura 36 – Modelo de caixa eletrônico otimizado à esquerda e original à direita	53
Figura 37 – imagem do Mapa UV à direita e do modelo tridimensional à esquerda	54
Figura 38 – Captura de tela do menu de opções de baking no blender com renderizador interno na versão 2.79b.	55
Figura 39 – Interface do <i>Awesome Bump</i> . Opções à esquerda, previsualização ao centro e mapa normal à direita	56
Figura 40 – Imagem encaixável criada no krita.	56
Figura 41 – Visualização do encaixe da imagem criada no krita.	57
Figura 42 – Exemplo de uso do <i>Inkscape</i> para aplicação de detalhes em texturas.	57
Figura 43 – Renderização do piso da entrada norte à esquerda e foto da entrada norte à direita	58
Figura 44 – Imagem de exemplo da propriedade cor e normal sendo determinadas por texturas mapeadas por mapas uv.	59
Figura 45 – Imagem do Estudio virtual para criação de objetos com iluminação básica.	60
Figura 46 – Foto de referência de iluminação	62
Figura 47 – Imagem renderizada da iluminação resultante	62
Figura 48 – Imagem demonstrando o esquema da cálculo da luz na <i>Unity</i>	63
Figura 49 – Captura de tela demonstrando a ativação da mira para o cubo lilás.	64
Figura 50 – Capturas de tela combinadas mostrando a sequencia de ativação da tela de informação.	65
Figura 51 – Captura de tela demonstrando a distribuição dos cubos verdes.	66
Figura 52 – Diagrama de estados que ilustra a jogabilidade para a aplicação.	67
Figura 53 – Captura de tela do <i>profiler</i> da <i>unity</i>	69
Figura 54 – H	71

LISTA DE TABELAS

LISTA DE QUADROS

Quadro 1 – Quadro com a descrição dos tipos de RV I	27
Quadro 2 – Quadro com a descrição dos tipos de RV II	27
Quadro 3 – Técnicas de Modelagem	48
Quadro 4 – Quadro com os Caminhos de Renderização e Descrições	61
Quadro 5 – Quadro com a descrição do hardware utilizado	68

LISTA DE ABREVIATURAS E SIGLAS

3D	Tridimensional
6DOF	Six Degrees of Freedom (Seis Graus de Liberdade)
API	Application Programming Interface (Interface de Programação de Aplicação)
CAD	Computer-aided design (Design Auxiliado por Computador)
CCET	Centro de Ciências e Tecnologia
COLLADA	Collaborative Design Activity (Atividade de Design Colaborativo)
CPU	Central Processing Unit (Unidade Central de Processamento)
DWG	Drawing (Desenho)
GLSL	OpenGL Shading Language (Linguagem de Sombreamento do OpenGL)
HDM	Head Mounted Display (Visor Mountado na Cabeça)
HTC	High Tech Computer Corporation (Corporação de Computadores de Alta Tecnologia)
IGES	Initial Graphics Exchange Specification (Especificação Inicial para Intercâmbio de Gráficos)
MAX	Maximum(Máximo)
MSAA	Multi-Sampling Anti-Aliasing (Anti-aliasing de amostras múltiplas)
NASA	National Aeronautics and Space Administration (Administração Espacial e Aeronáutica Nacional)
OBJ	Object (Objeto)
OPENGL	Open Graphics Library (Biblioteca de Gráficos Aberta)
PNG	Portable Network Graphics (Gráficos de Rede Portáteis)
RV	Realidade Virtual
SDK	Software Development Kit (Pacote de Desenvolvimento de Software)
STEP	Standard for the Exchange of Product Data (Padrão para o Intercâmbio de Dados de Produtos)
STL	Stereolithography (Estereolitografia)
UFMA	Universidade Federal do Maranhão
VR	Virtual Reality (Realidade Virtual)

LISTA DE SÍMBOLOS

Δ	Delta, quarta letra do alfabeto grego
\mathbb{R}	Conjunto dos números reais
\mathbb{R}^3	Espaço tridimensional com pontos denotados por triplas de números reais

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	ESPAÇO TRIDIMENSIONAL	17
2.2	OBJETOS TRIDIMENSIONAIS: DE ARQUIVOS PARA O MONITOR . .	19
2.3	REALIDADES VIRTUAIS	24
2.3.1	Conceitos e definições	25
2.4	APLICAÇÕES RV INTERATIVAS	28
2.4.1	Pipeline para uma RV Interativa	30
2.4.2	Aspectos Fisiológicos	34
2.4.3	Diretrizes para Movimentação e ambiente	34
2.4.4	Diretrizes para interatividade	35
2.4.5	Diretrizes para Áudio	36
3	DESENVOLVIMENTO DE UMA APLICAÇÃO PARA NAVEGAÇÃO VIRTUAL NO CCET	37
3.1	AQUISIÇÃO DE MATERIAL BASE	37
3.2	FERRAMENTAS UTILIZADAS	41
3.3	MODELAGEM	47
3.3.1	Otimização de Modelos	52
3.4	MATERIAL E TEXTURA	54
3.5	ILUMINAÇÃO E RENDERIZAÇÃO	59
3.5.1	Luz Real e Virtual	61
3.6	INTERAÇÃO	64
4	EXPERIMENTOS	68
4.1	Dispositivos Utilizados	68
4.2	AVALIAÇÃO DA APLICAÇÃO	69
5	CONCLUSÃO	72
	REFERÊNCIAS	73
	APÊNDICE A – QUESTIONÁRIO	76

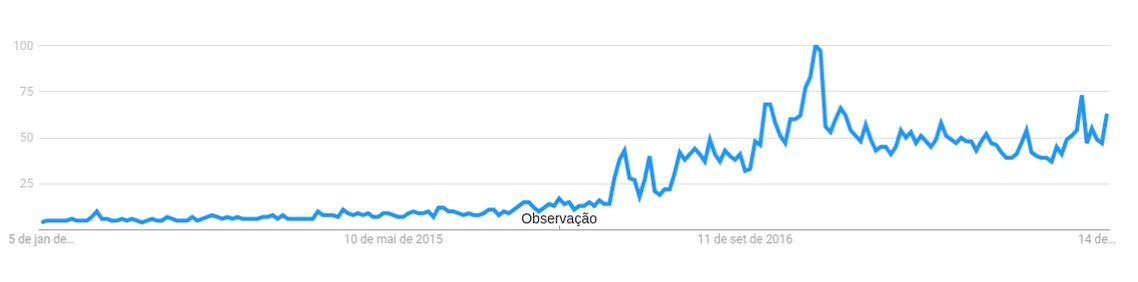
1 INTRODUÇÃO

Uma das apostas de tecnologia para essa década, a realidade virtual nos apresenta o resultado de esforços das mais diversas áreas acadêmicas. O computador, que a torna possível nos moldes modernos, tem suas bases na física, química, matemática, engenharia entre outras.

Muito antes da formalização de seu conceito, a realidade virtual, de certa maneira já se fazia presente em murais em 360° conhecidos como pinturas panorâmicas, fotos e visualizadores estereoscópicos que são a base para tecnologia disponível atualmente como o *cardboard*, visualizador de RV da Google¹. Entretanto, o acesso a essa tecnologia, assim como a compreensão dos princípios que a fazem funcionar ainda parece estar distante da maioria dos jovens brasileiros. Isso, no entanto, não significa que a curiosidade pela tecnologia e suas aplicações no Brasil seja baixa.

Como é possível observar nas Figura 1 e Figura 2, o interesse² pelo tema tem sido crescente nos últimos anos.

Figura 1 – Gráfico para o interesse(número de buscas) no tópico Realidade Virtual no Brasil entre janeiro de 2015 e dezembro de 2017

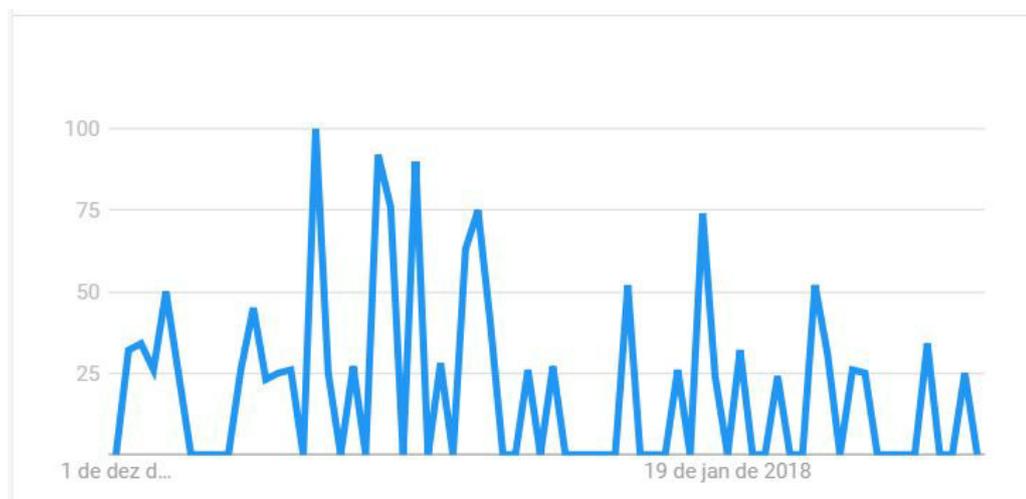


Fonte: *Google Trends*(2018)

¹ www.google.com.br/intl/pt-BR_br/about/

² Os números representam o interesse de pesquisa relativo ao ponto mais alto no gráfico de uma determinada região em um dado período. Um valor de 100 representa o pico de popularidade de um termo. Um valor de 50 significa que o termo teve metade da popularidade. Uma pontuação de 0 significa que não havia dados suficientes sobre o termo.

Figura 2 – Gráfico para o interesse no tópico Realidade Virtual no Brasil entre dezembro de 2017 e fevereiro de 2018



Fonte: *Google Trends*(2018)

Logo, percebe-se que há uma vontade de parte da população de se aproximar da tecnologia. Utilizar essa tecnologia para apresentar o Centro de Ciências e Tecnologia(CCET) aos possíveis futuros estudantes da Universidade Federal do Maranhão(UFMA), assim como informações sobre cursos nele ministrados pode despertar o interesse nas áreas que tanto contribuíram e contribuem para o avanço tecnológico do mundo. Pode, além de informar diretamente, trazer a experiência do potencial para compreender, criar, desenvolver e inovar de cada área do conhecimento diretamente envolvida na criação de uma Realidade Virtual(RV).

Este trabalho tem como objetivo explorar a criação de uma aplicação de RV, estudar a compreensão e o impacto, dessa tecnologia, em alunos do ensino médio em relação às áreas exatas do conhecimento. Busca ainda:

- Estudar como são armazenados, processados e representados em monitores, os modelos tridimensionais;
- Estudar técnicas de modelagem, mapeamento de texturas, iluminação, otimização e gerenciamento de recursos computacionais em uma aplicação RV;
- Estudar a implementação de tecnologias no motor de jogos *Unity*³.

A implementação da aplicação se dará por meio do uso de fotos de referência para criação de modelos tridimensionais e o uso de informações sobre os cursos retirados de domínios da

³ <https://unity3d.com/pt>

UFMA. Usando o motor de jogos *Unity*, da Unity Technologies⁴, em conjunto com o pacote de Desenvolvimento de Realidade Virtual da Google(Google VR SDK⁵), para as plataformas *Cardboard*⁶ e *Daydream*⁷ na Unity, modelos tridimensionais elaborados no *Blender*⁸, texturas à partir de fotos, criadas à partir de softwares como o Gimp⁹, Krita¹⁰ e Inkscape¹¹.

Por meio de questionário aplicado em duas partes, apresentado no Apêndice A, a aplicação desenvolvida será avaliada. Uma parte anterior à utilização da aplicação e uma posterior. Para que fiquem registrados os efeitos e mudanças da opinião sobre tecnologia.

O trabalho segue com o Capítulo 2 que apresenta os fundamentos para o desenvolvimento da tecnologia de realidade virtual e o Capítulo 3 que discorre sobre todos os princípios, estratégias e métodos utilizados para implementar a aplicação e a pesquisa; no Capítulo 4 são apresentados os resultados dos experimentos em torno da aplicação, assim como sua avaliação.

⁴ unity3d.com

⁵ disponível em github.com/googlevr/gvr-unity-sdk/releases

⁶ vr.google.com/cardboard/

⁷ vr.google.com/daydream/

⁸ blender.org

⁹ gimp.org

¹⁰ krita.org

¹¹ inkscape.org

2 FUNDAMENTAÇÃO TEÓRICA

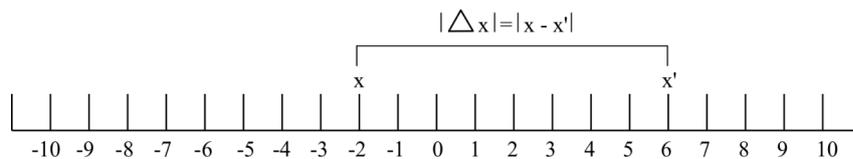
Neste capítulo são introduzidos os conceitos básicos e as tecnologias adotadas neste trabalho. Inicialmente, uma explicação sobre objetos tridimensionais, espaço tridimensional e sua projeção em monitores. Em seguida, são apresentados formatos de arquivos e conceitos de APIs, interfaces de programação para aplicações gráficas. Finalmente, é apresentado o fluxo básico para renderização de gráficos tridimensionais em um computador e conceitos de aplicação de realidade virtual.

2.1 ESPAÇO TRIDIMENSIONAL

“Matemática para gráficos tridimensionais é sobre medir posições, distâncias e ângulos”(DUNN; PARBERRY, 2002, p.5, traduzido pelo autor). Para isso, é necessário um sistema de coordenadas onde possam ser mapeados os pontos que compõem o objeto, sendo que o mais utilizado é o Cartesiano.

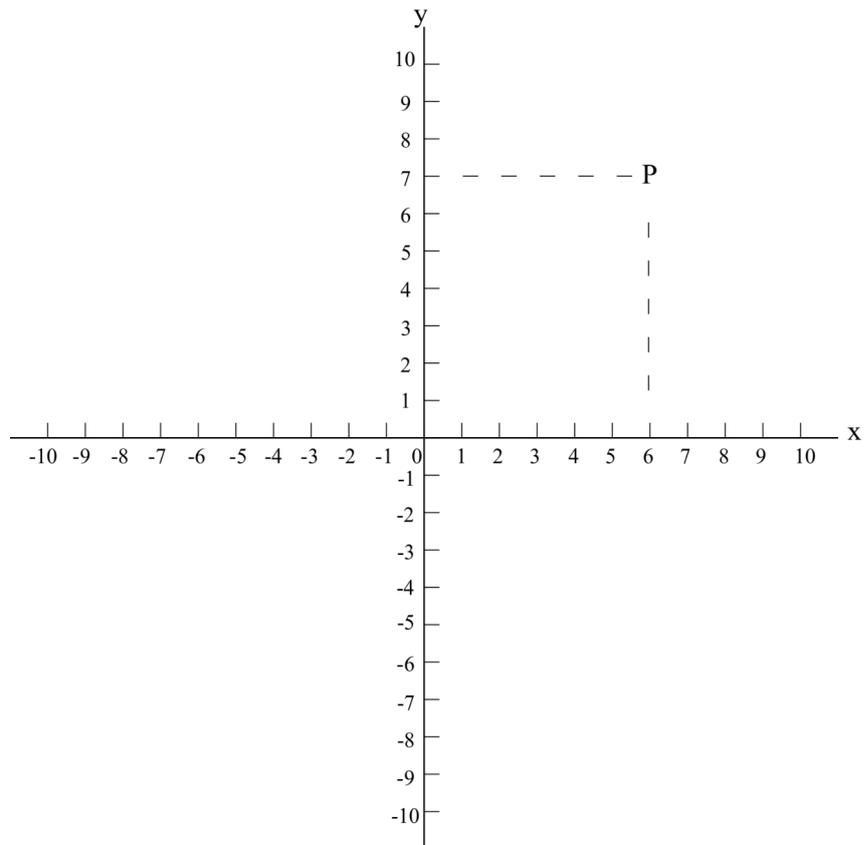
Segundo Nitecki (2012), a geometria para a reta numérica \mathbb{R} é muito clara e a posição de um ponto x nessa reta é definida em relação à posição 0 (zero) pela própria posição do ponto. Em relação a outros pontos, sua posição é obtida pelo módulo da diferença entre posição de x e de outro ponto x' . Essa diferença pode ser chamada Δx (delta x) e ser vista na Figura 3.

Figura 3 – Ilustração da reta numérica e posição na reta



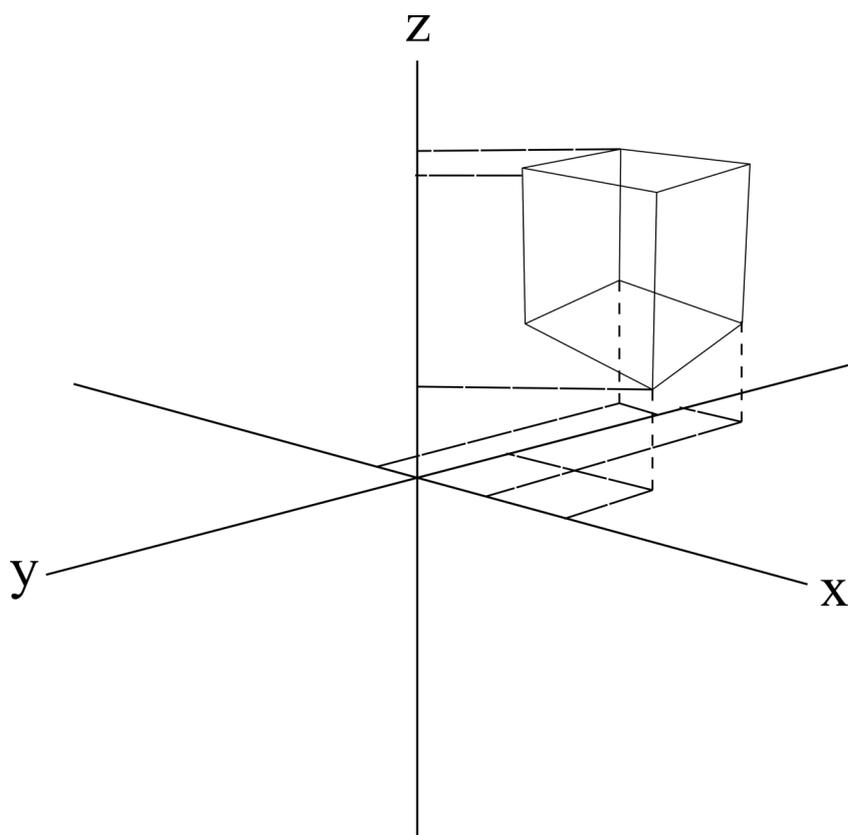
Ainda em Nitecki (2012), observa-se que há um ponto de origem à partir do qual pode-se desenhar dois eixos, geralmente chamados eixo x , horizontal, e eixo y , vertical. Cada eixo é uma reta dos reais. Dado um ponto P no plano, desenha-se um retângulo ao traçar um segmento de reta para cada eixo, perpendicular ao outro eixo, originado no eixo correspondente e finalizado no ponto P . Diz-se que as coordenadas (retangulares ou cartesianas) de P são os dois números (x,y) , com x anotado sobre o eixo x em $(x,0)$ e y anotado sobre o eixo y em $(0,y)$, como na Figura 4.

Figura 4 – Ilustração Ponto P no plano cartesiano



Ao inserir-se mais um eixo nesse modelo tem-se um espaço com coordenadas tridimensionais na Figura 5. Neste espaço é possível representar os objetos de interesse deste trabalho.

Figura 5 – Ilustração do cubo no plano tridimensional



“Isso leva à identificação do conjunto \mathbb{R}^3 de triplas (x,y,z) de números reais com pontos no espaço, o que às vezes chama-se de espaço tridimensional (ou 3-espaço).”(NITECKI, 2012, p.2, traduzido pelo autor)

Um objeto tridimensional pode ser mapeado em vários sistemas de coordenadas tridimensionais. Quando o modelo é carregado ele é mapeado para o sistema vigente dentro do software utilizado. A exemplo, quando no Blender, tem-se que o vetor Z aponta para cima nas coordenadas globais, já no Unity o vetor Y aponta para cima. Como resultado é necessário fazer adaptações para que um objeto seja representado da maneira desejada na tela. Além de sistemas de coordenadas diferentes, existem diferenças significativas entre monitores, como resolução, densidade de pixels, quantidade de cores representável, taxa de atualização e outros. A quase totalidade dessas diferenças é resolvida pelos softwares, atualmente.

2.2 OBJETOS TRIDIMENSIONAIS: DE ARQUIVOS PARA O MONITOR

Após a construção de um objeto e sua exibição, é necessário armazenar os dados. Para isso existem diversos modelos como o STL, COLLADA¹, OBJ, IGES e STEP². Esses modelos

¹ www.khronos.org/collada/

² www.steptools.com/stds/step/

são Open Source. Isto significa que seu código fonte é aberto para visualização, modificação e distribuição. Existem também os formatos MAX, 3DS e CAD que são proprietários, ou seja, de código fechado, não modificável ou distribuível.

A geometria (ou forma) de um modelo é geralmente guardada como um conjunto de pontos 3D(ou vértices). A superfície do modelo é então guardada como uma série de polígonos(ou faces) que são construídas ao indexar esses vértices.(MCHENRY; BAJCSY, 2008, p.1,traduzido pelo autor).

Arquivos, muitas vezes, guardam dados além daqueles para representação dos modelos tridimensionais como animações, esquemas de articulações, dados sobre outros objetos como lâmpadas e metadados. A exemplo pode-se ver na Figura 6 o conteúdo de um arquivo no formato OBJ e, na Figura 7 o objeto tridimensional resultante de sua importação no Blender. Deve-se notar que no Blender os índices iniciam em zero, enquanto no formato OBJ eles se iniciam no um.

Figura 6 – Imagem de exemplo básico de arquivo OBJ.

```
# Vertices: 8
# Points: 0
# Lines: 0
# Faces: 6
# Materials: 1

o 1

# Vertex list

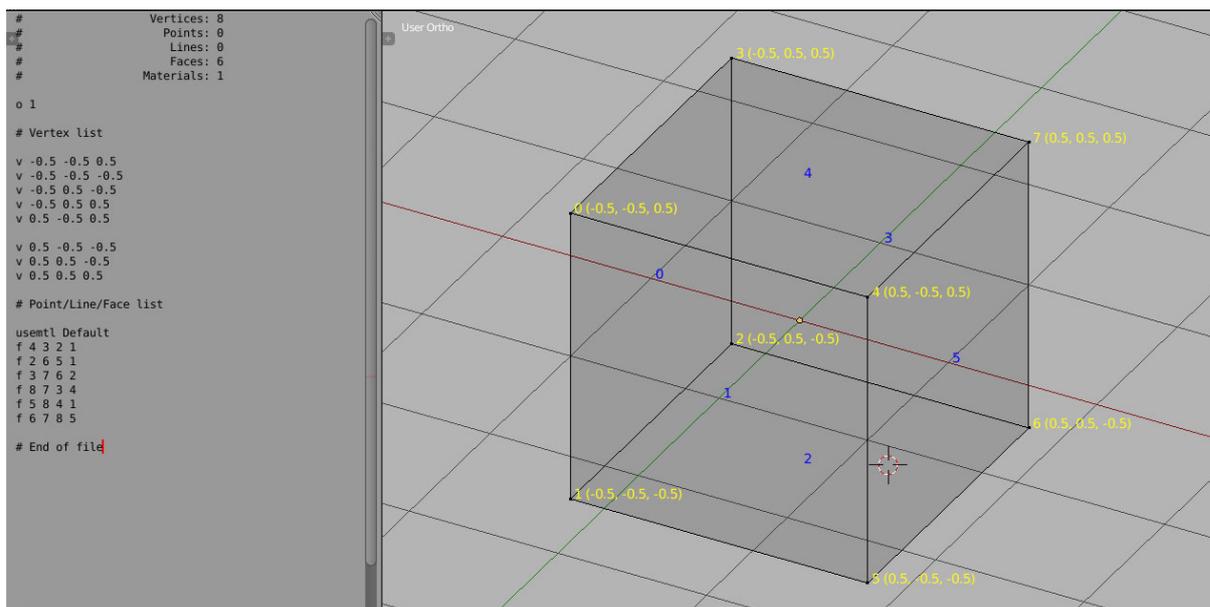
v -0.5 -0.5 0.5
v -0.5 -0.5 -0.5
v -0.5 0.5 -0.5
v -0.5 0.5 0.5
v 0.5 -0.5 0.5
v 0.5 -0.5 -0.5
v 0.5 0.5 -0.5
v 0.5 0.5 0.5

# Point/Line/Face list

usemtl Default
f 4 3 2 1
f 2 6 5 1
f 3 7 6 2
f 8 7 3 4
f 5 8 4 1
f 6 7 8 5

# End of file
```

Figura 7 – Captura de Tela da importação resultante do Exemplo básico obj(cubo) no Blender. ³



Para desenhar gráficos em um monitor atualmente são utilizadas placas gráficas conhecidas como Unidade de Processamento Gráfico(GPU). Antes, a Unidade de Processamento Central(CPU) era responsável pelo processamento gráfico e essa foi uma grande mudança como frisou Eck (2018). Essas placas recebem instruções vindas da CPU, escritas utilizando APIs como o OpenGL⁴, Direct3D⁵ e a mais nova Vulkan⁶. A API utilizada pelo Blender é o OpenGL e na Unity tem-se as citadas acima à disposição. Necessário, porém frisar que o hardware precisa dar suporte à API para que programas escritos com ela funcionem.

Os passos básicos da renderização do lado da CPU é dizer ao *pipeline* para utilizar um certo sombreador, dar a ele a informação necessária para renderizar e dar ao sombreador um comando de desenho.(JARNO, 2017, p.26, traduzido pelo autor).

Em seguida, como visto na Figura 8, algumas etapas devem ser executadas até a imagem chegar à tela.

Como explicado por Hock-Chuan (2012), no processamento de vértice são processados e transformados os vértices de cada objeto da cena. Já na rasterização cada primitiva é convertida num conjunto de fragmentos. O fragmento possui propriedades como posição, cor, normal e textura. No processamento de fragmento são processados os fragmentos para depois serem todos combinados em uma nova etapa onde são transformados de 3D para 2D para exibição na tela.

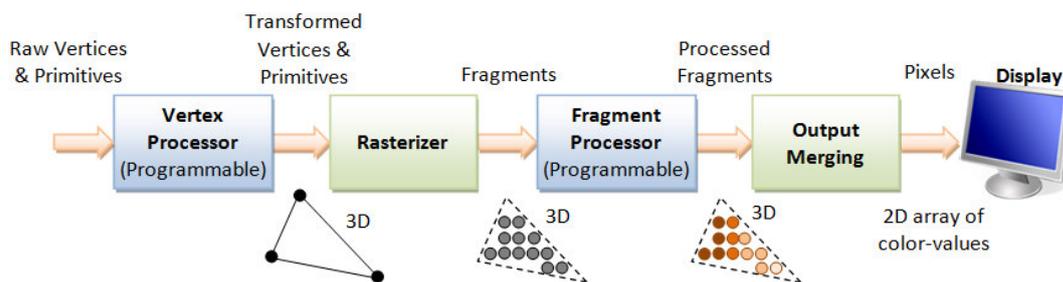
³ O primeiro vetor(v) do arquivo é representado no objeto pelo vetor de número 0 (zero). A primeira face (f) no arquivo é representada pela face de número 0 (zero) no objeto.

⁴ www.opengl.org

⁵ docs.microsoft.com/en-us/windows/desktop/directx

⁶ www.khronos.org/vulkan/

Figura 8 – Ilustração do pipeline de renderização 3D



3D Graphics Rendering Pipeline: Output of one stage is fed as input of the next stage. A vertex has attributes such as (x, y, z) position, color (RGB or RGBA), vertex-normal (n_x, n_y, n_z) , and texture. A primitive is made up of one or more vertices. The rasterizer raster-scans each primitive to produce a set of grid-aligned fragments, by interpolating the vertices.

Fonte: https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html

Seguindo os princípios da Unity Technologies (2018e), os modelos voltados para realidade virtual devem seguir um padrão que os torne o mais eficiente possível. Isso se traduz em objetos com baixo número de polígonos (*LowPoly*, de *Low Polygon Count*), com apenas o lado externo e as partes visíveis presentes, texturas de media e baixa resolução ou até mesmo a não utilização de texturas.

Modelos *LowPoly* são simplificações das representações de objetos de alta densidade de polígonos (*HighPoly* referecia a *High Polygon Count*). A implementação destes conceitos podem ser observada à partir de modelos criados para representar o objeto real visto na Figura 9. Um modelo de alta densidade pode ser visto na Figura 10. Modelos *HighPoly* são mais detalhados como os vistos em consoles de última geração e mais propriamente em computadores capazes de processar milhões de polígonos em frações de segundo. Uma versão *LowPoly* deste modelo é observada na Figura 11.

Figura 9 – Foto de jarro do CCET



Figura 10 – Imagem de objeto com alta densidade de polígonos

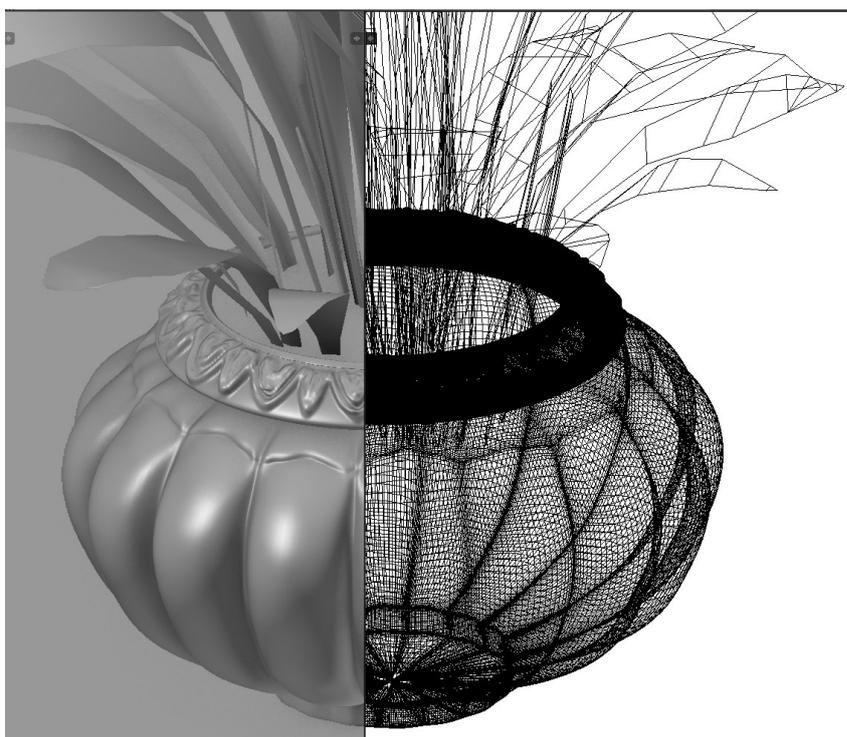
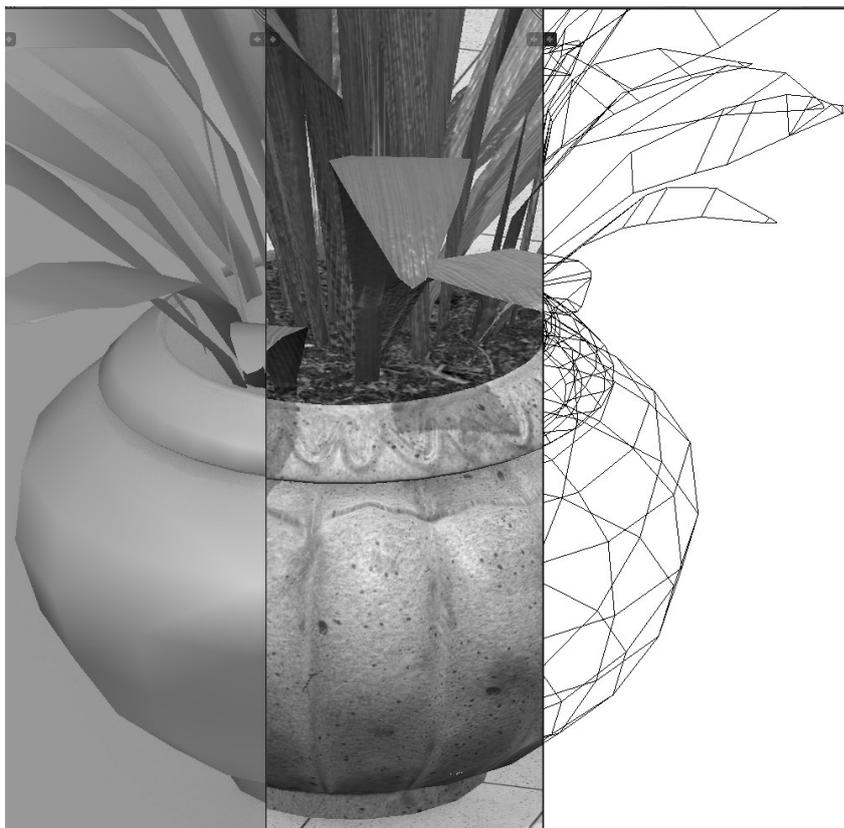


Figura 11 – Imagem de objeto com otimizado, com baixo número de polígonos.



Para facilitar a criação e manipulação de objetos em três dimensões atualmente são utilizados softwares como Blender, Autocad⁷, Maia⁸ e 3DS Max. Cada suíte tem foco em diferentes tipos de aplicação. O Autocad é voltado para engenharia e arquitetura, já o Maia e o 3DS Max para modelagem e animações. Dos citados, apenas o Blender é *open source* e sua ideia central é ser adaptável a várias tarefas.

Neste trabalho todos os objetos são processados pelo Blender em alguma instância e a grande parte foi totalmente trabalhada dentro desse software. Alguns objetos foram portados de outros softwares como 3DS Max e Maia.

2.3 REALIDADES VIRTUAIS

Inventado por Jaron Lanier, cientista da computação e artista, o termo Realidade Virtual surgiu no final dos anos 80. De acordo com Araujo e Kirner (1996) esse termo foi criado para diferenciar as simulações tradicionais, monousuário, de simulações com múltiplos usuários em um ambiente compartilhado.

⁷ www.autodesk.com.br/products/autocad/overview

⁸ www.autodesk.com.br/products/maya/overview

2.3.1 Conceitos e definições

Vários pesquisadores, desenvolvedores de software e acadêmicos apresentaram sua definição para o termo RV. Como dito por Latta e Oberg (1994 apud NETTO et al., 2002) e Aukstakalnis et al. (1992 apud NETTO et al., 2002), “trata-se de uma interface que simula um ambiente real e permite aos participantes interagirem com o mesmo, permitindo às pessoas visualizarem, manipularem e interagirem com representações extremamente complexas”.

“Na prática, A RV permite que o usuário navegue e observe um mundo tridimensional, em tempo real e com seis graus de liberdade(6DOF).”(NETTO et al., 2002, p.5) Isso significa dizer que deve haver conteúdo para ser visualizado em 360 graus em relação a qualquer posição que o usuário possa assumir dentro do mundo virtual. Além disso, deve haver a capacidade do *hardware* para reconhecer movimentos em todas as direções.

Ainda em Netto et al. (2002), são citadas as implementações de RV ao longo do tempo. Por exemplo, o sensorama, que combinava filmes 3D, som estéreo, vibrações, movimentos de ar e aromas para proporcionar uma experiência muito rica para os sentidos. O sensorama já utilizava um visor estereoscópico e um protótipo pode ser observado na Figura 12 . A seguir, foi criado o HMD (Visor montado na Cabeça). Esse era um par de câmeras que enviavam imagens para um capacete com monitores.

Figura 12 – Foto promocional do protótipo do Sensorama



Fonte: Pimentel e Teixeira (1994 apud NETTO et al., 2002)

Um dispositivo capaz de detectar o movimento e a inclinação dos dedos, conhecido como *VPL DataGlove*, foi criado por Thomas Zimmerman em 1985 (LOWOOD, 2015).

No final de 1986 a equipe da NASA já possuía um ambiente virtual que permitia aos usuários ordenar comandos pela voz, escutar fala sintetizada e som 3D, e manipular objetos virtuais diretamente por meio do movimento das mãos. O mais importante é que esse trabalho permitiu verificar a possibilidade de comercialização de um conjunto de novas tecnologias, tornando mais acessível o preço de aquisição e desenvolvimento. A conscientização de que os empreendimentos da NASA poderiam gerar equipamentos comercializáveis deu início a inúmeros programas de pesquisa em RV no mundo inteiro. (NETTO et al., 2002, p.8)

À partir das pesquisas de Araujo e Kirner (1996 apud NETTO et al., 2002), foi montado o Quadro 1. Neste quadro são descritos os tipos de sistema RV considerados por eles.

Quadro 1 – Quadro com a descrição dos tipos de RV I

Tipo de Realidade Virtual	Descrição
Tele-colaboração	Ambiente multiusuário com capacidade de manipulação de objetos e sensação de peso destes.
Tele-presença	Possibilita ao operador realizar atividades em um ambiente remoto. Para isso, utiliza máquinas nesse ambiente que imitam seus movimentos e respondem com impulsos que, para o operador, transmitem as sensações proporcionadas pelas reações, às ações, e o ambiente
Visualização Científica	Representar visualmente grande volume de dados para facilitar a interpretação e a extração de resultados. Como, por exemplo, um conjunto de informações, geradas por uma simulação de líquidos, representado por um modelo tridimensional animado à partir desses dados.

Fonte: Adaptado de Araujo e Kirner (1996 apud NETTO et al., 2002)

Pimentel e Teixeira (1994 apud NETTO et al., 2002) apresentam outras classificações para RV que podem ser verificadas no Quadro 2.

Quadro 2 – Quadro com a descrição dos tipos de RV II

Tipo de Realidade Virtual	Descrição
RV de Simulação	Tipo mais antigo representado por simuladores de voo, corrida etc. Coloca o usuário numa cabine com controles, telas e monitores que traduzem para o mundo virtual os controles do usuários.
Realidade Aumentada	Permite ao usuário ver, projetado na realidade, gráficos, dados, diagramas e animações. Isso é possível por conta da capacidade do dispositivo visual de parecer transparente, ou seja, mostrar o mundo real ao mesmo tempo que exhibe os aspectos virtuais. da projeção virtual.
RV de Projeção	Captura as imagens e ações do(s) usuário(s) e as transporta para o mundo virtual sem que o usuário seja inserido no nele.
Telepresença	Câmeras de vídeo, microfones e acessórios capazes de transmitir sensações, como peso, são utilizados para mergulhar o usuário no mundo virtual.
<i>Displays</i> Visualmente Acoplados	Imagens exibidas diretamente ao usuário por dispositivo acoplado à sua cabeça. Tal dispositivo é capaz de acompanhar o movimentos da cabeça do usuário e mudar a imagem exibida adequadamente para manter a sensação de imersão
RV de mesa	Monitores ou sistemas de projeção apresentam o mundo virtual.

Fonte: Adaptado de Pimentel e Teixeira (1994 apud NETTO et al., 2002)

Outra tecnologia disponível é o sistema *CAVE*. “Uma *CAVE* (ou uma Caverna, em português) consiste de uma sala em que paredes, teto e chão são telas semitransparentes onde as imagens são projetadas, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual.” (SANTOS, 2015)

Atualmente, várias empresas investem nestes tipos de RV. Como exemplo, óculos virtuais de várias empresas, como o Cardboard da Google e o Gear Vr da Samsung, que oferecem variados níveis de qualidade visual e imersão. Há também *headsets* como o *Oculus Rift*⁹ da Facebook¹⁰, o *Vive*¹¹ da HTC e o Sony Playstation VR¹². E para realidade aumentada é possível destacar o produto da Microsoft¹³, *Hololens*¹⁴.

2.4 APLICAÇÕES RV INTERATIVAS

A interatividade com dispositivos computacionais evoluiu e continua evoluindo rapidamente. Fato verificável pela história, com o primeiro mouse, que pode ser visto na Figura 13, criado por Doug Engelbart, assim como o desenvolvimento do *Sketchpad*, por Ivan Sutherland, ambos em 1963 como lembrado pela Doug Engelbart Institute (2018). Em alguns anos, sistemas computacionais que anteriormente possuíam apenas a programação como maneira de interagir, implementaram a capacidade de receber comandos por diversos dispositivos e interfaces, como apontadores, canetas, toque, pressão, imagem, movimento e outros. Também se tornaram capazes de responder de novas maneiras como por som, cor, vibrações, imagem e, em alguns casos, movimentos. A exemplo tem-se os aparelhos celulares que apresentam grande variedade de meios de interação. Como dito por Dam (1997) os ambientes computacionais com interface *post-WIMP*(pós "janela, ícones, menus e ponteiros") abordam mais o uso de alternativas à interação através da interface gráfica quando apostam em interações por fala, desenhos, 3D e gestos como na Figura 14.

⁹ <https://www.oculus.com/rift/>

¹⁰ <https://pt-br.facebook.com/>

¹¹ <https://www.vive.com/us/>

¹² <https://www.playstation.com/pt-pt/explore/playstation-vr/>

¹³ <https://www.microsoft.com/pt-br>

¹⁴ <https://www.microsoft.com/en-IE/hololens>

Figura 13 – The Dawn of Interactive Computing (O Alvorecer da Computação Interativa)



Fonte: (DOUG ENGELBART INSTITUTE, 2018)

Figura 14 – Imagem ilustrando a utilização de gesto para interação com celular



Fonte: (MOTOROLA MOBILITY LLC, b)

De acordo com Hornbæk e Oulasvirta (2017 apud SAFFER, 2009), uma interação, a grosso modo, é uma transação entre duas entidades, tipicamente uma troca de informação, mas também pode ser uma troca de bens ou serviços. [...]. Eles apresentam tipos de interação como:

- Por diálogo: um processo de atos de comunicação e suas interpretações;
- Por transmissão: um remetente enviando uma mensagem através de um canal com ruído;
- Uso de objetos: um humano que utiliza ferramentas para manipular e agir.

Em termos de software, segundo o (ENCYCLOPEDIA.COM, 2018), sistemas interativos são sistemas de computador caracterizados por um número significativo de interações entre humanos e o computador . “Os sistemas interativos mais antigos foram os sistemas de linha de comando”(ENCYCLOPEDIA.COM, 2018). Dentro desse conceito cita-se os sistemas operacionais UNIX e DOS.

Sistemas por linha de comando recebem instruções e parâmetros para realizar atividades. Esta interação foi aprimorada com o passar dos anos pelo surgimento de menus dinâmicos, interfaces gráficas e o mouse em 1965. Até que surgem as telas sensíveis ao toque, com alta resolução e sensores de movimentos que deram suporte a novas tecnologias como a RV.

“Essas máquinas são equipadas com placas gráficas melhores e mais rápidas e seus preços caem rapidamente. Torna-se possível até para o usuário médio entrar no mundo dos gráficos de computador.”(MYERS, 2001)

Com um aumento no alcance dos dispositivos para RV como celulares, óculos, soluções para consoles como PSVR¹⁵, além da capacidade computacional, empresas como Google e Unity disponibilizaram materiais de referência em termos de interface, parâmetros de performance e interação para diversos tipos de aplicação. Conceitos de interação que são estudados há muito sob o nome IHC(Interface Humano Computador).

“O termo interface humano computador emergiu na segunda metade dos anos 80, como forma de descrever novo campo de investigação preocupado não somente com o design da interface de sistemas computacionais, mas, também, com o foco de interesse e de demandas do público.” (GUEDES, 2010)

2.4.1 Pipeline para uma RV Interativa

Um dos produtos deste desenvolvimento foi a Realidade Virtual com modelos 3D. Esta faz uso de vários dispositivos e técnicas da era *post-WINP* e apresenta algumas das novas formas de interação, dentre estas estão o toque e o acompanhamento de movimento.

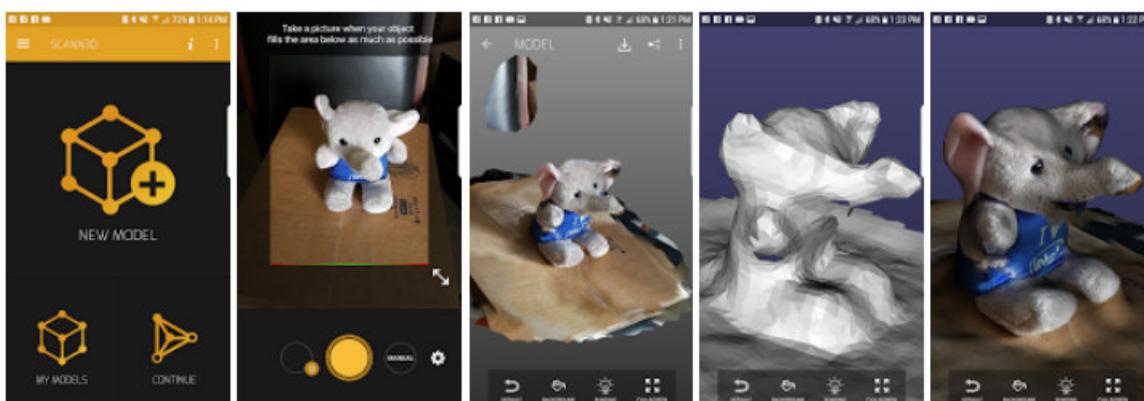
¹⁵ <https://www.playstation.com/pt-br/explore/playstation-vr/>

Duffy (2018), em seu artigo, apresenta um pipeline básico para o desenvolvimento de aplicações deste tipo, em termos de modelos tridimensionais. Segundo ele, os passos são:

- Criar ou adquirir material 3D
- Fazer a retopologia e mapeamento UV dos objetos
- Aplicar texturas e editar os materiais dos objetos
- Importação e exportação de objetos e outros fluxos necessários para motores de aplicações 3D

No primeiro passo é possível usar várias fontes como scaneamento 3D, visto na Figura 15, modelagem, compra de modelos e escultura digital.

Figura 15 – Passos para escaneamento de objeto 3D por aplicativo



Fonte: (DUFFY, 2018)

O segundo passo é crucial para aplicações que demandam alto número de quadros produzidos por segundo, pois "Retopologia é o processo de reconstruir uma malha 3D para que seja menor e melhor planejada para animação"(DUFFY, 2018, traduzido pelo autor). Já o mapeamento UV é aplicado posteriormente sobre as malhas otimizadas.

Utilizando o mapeamento UV é possível aplicar texturas, o próximo passo, com detalhes não representados nos modelos com menor número de polígonos. Seguindo-se da edição de materiais, na qual são definidas as propriedades de interação com a luz.

É possível editar algumas das texturas e materiais em ferramentas de edição 3D profissionais como Blender. Blender oferece tudo o que é necessário para isto, especialmente se você encontrar materiais para renderização baseados em física, online e queira usá-los em seus objetos.(DUFFY, 2018, traduzido pelo autor)

O artigo termina explicando que a tarefa final é levar seus objetos para dentro do motor de jogos e configurar as texturas e materiais dentro deste. Além disto é necessário controlar o que acontece com os objetos através de *scripts*.

Para seguir os passos não abordados no artigo, deve-se escolher um motor de jogos que apresente suporte para a plataforma desejada. Dentro dele, é possível organizar os objetos e configurar elementos de iluminação para se construir uma cena. A implementação da iluminação pode variar entre os motores de jogos. As mais comum demandam posicionamento de luzes virtuais, pré-computação e construção de mapas.

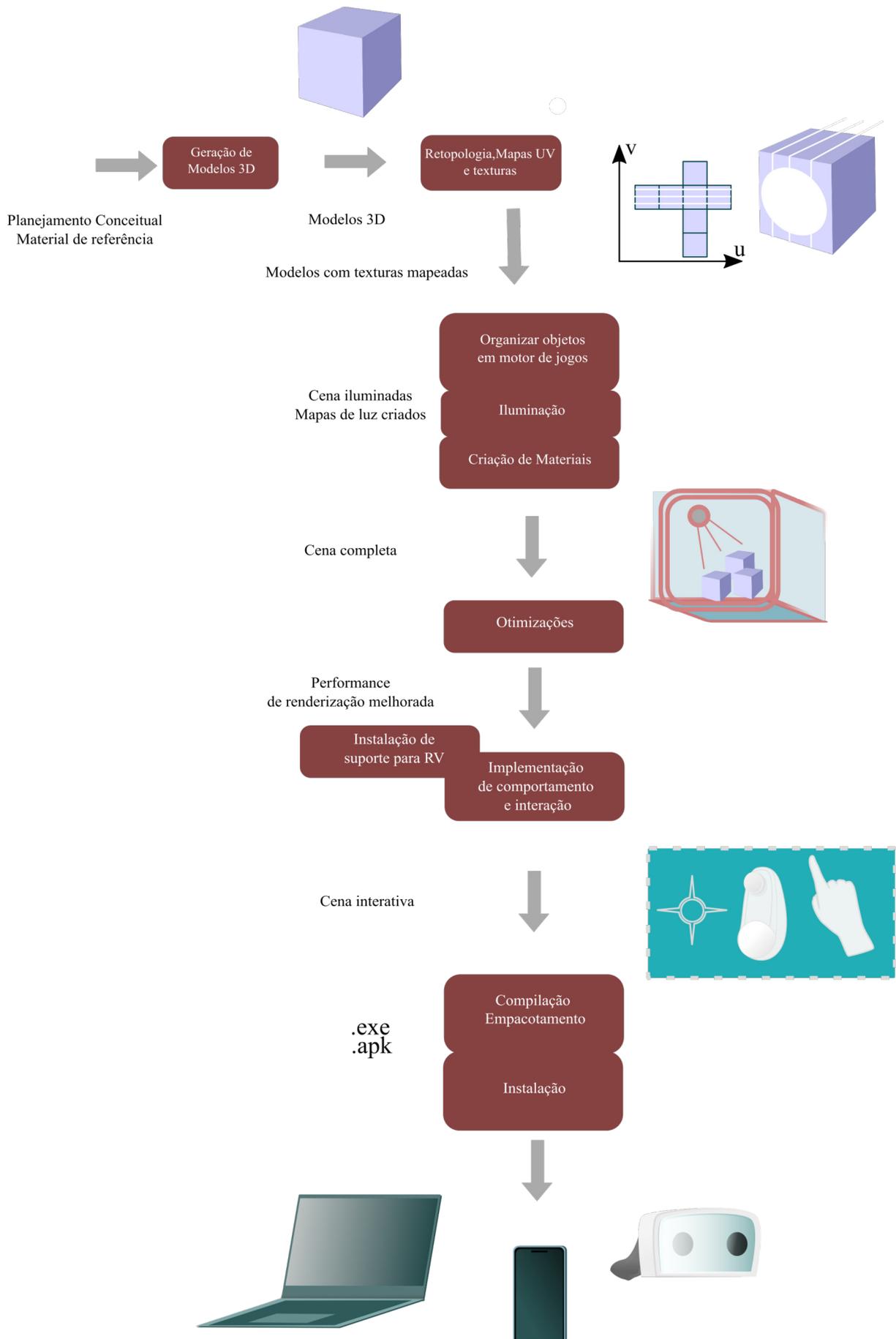
Com a cena construída é necessário configurar a maneira como o conteúdo será renderizado para que apresente as características para RV na plataforma desejada. Como exemplo, no Carboard a tecnologia utilizada para gerar a profundidade é a estereoscopia.

Uma imagem estereoscópica, como explica a StereoGraphics Corporation (1997), apresenta o ponto de vista do olho direito e esquerdo do observador, como acontece na realidade. Os pontos de vista apresentam diferenças por serem separados por uma distância de 64 milímetros. Tais disparidades são processadas pelo cérebro que monta a imagem final a ser apresentada.

Para gerar as imagens estereoscópicas, duas câmeras são posicionadas de maneira que a distância entre elas seja equivalente à distância entre os dois olhos de um observador. Dentro do motor de jogos estas câmeras são virtuais e podem ser controladas por script. Isso possibilita ao usuário locomoção dentro do espaço virtual. No visor, um conjunto óptico pode aplicar ajustes, como fazer a imagem parecer mais distante e diminuir possíveis distorções.

Na Figura 16 pode-se verificar um pipeline básico para o desenvolvimento de uma aplicação dessa natureza.

Figura 16 – Pipeline para criação de uma aplicação de RV interativa com modelos 3D



2.4.2 Aspectos Fisiológicos

De acordo com a Google (2018), compreender os efeitos fisiológicos do design de Realidade Virtual e seguir essas diretrizes é crítico para tornar seu aplicativo um sucesso e assegurar que usuários não sintam o mal estar da simulação.

Realidade virtual introduz um novo conjunto de considerações fisiológicas para design. Como simuladores de voo utilizados por pilotos em treinamento a realidade virtual tem o potencial para apresentar disparidade entre as sugestões, físicas e visuais, de movimento. Essa disparidade pode produzir náusea, conhecida como “enjoo de simulação”, quando seus olhos pensam que você está se movendo, mas seu corpo não.(GOOGLE, 2018)

A experiência de RV na plataforma Cardboard é construída à partir do acoplamento de um celular a um visor, feito de papelão e lentes de vidro. O celular apresenta o conteúdo e recebe dados de entrada, o visor possui lentes para ajustar o conteúdo, de maneira a fazer o cérebro ter a sensação de profundidade, e um botão para interação. Sobre essa plataforma são apresentados pontos que devem ser observados para uma melhor experiência.

"O rastreamento de cabeça possibilita aos objetos no espaço virtual a manutenção sua posição independente de como você move sua cabeça, criando a percepção de um mundo virtual ao seu redor.”(GOOGLE, 2018, traduzido pelo autor)

2.4.3 Diretrizes para Movimentação e ambiente

A diretriz mais importante para uma aplicação em RV é sempre rastrear a cabeça do usuário, de acordo com a Google (2018) e a Motorola Mobility LLC (a) alcançar taxa de atualização para a plataforma desejada é parte essencial para garantir que os usuários tenham uma experiência sem dores de cabeça. Uma pequena dessincronização pode acarretar em mal estar para o usuário.

Em conjunto com o rastreamento contínuo deve-se minimizar a latência Movimento-para-Fóton¹⁶, pois, de acordo com Meehan Michael & Razzaque (2003) baixa latência tem correspondência com maior presença num ambiente virtual e isso é necessário, pois, por conta da assincronia entre os sistemas visual e vestibular, usuários experimentam náuseas e tontura.

Outra diretriz citada pela Google (2018), que diminui o desconforto, é manter o usuário no controle na maior parte do tempo. Isso assume exceções como no caso de um aplicativo simulador de montanha russa e mesmo em uma aplicação deste tipo recomenda-se dar o poder de iniciar a simulação. Nesse aspecto, a diz Facebook Technologies, LLC. (b) que recomenda-se

¹⁶ Latência Movimento para Fóton é o tempo necessário para um movimento do usuário ser completamente refletido na tela do visualizador.

sempre permitir que o usuário pause a experiência e recomece quando quiser. Os desenvolvedores precisam estar atentos à necessidade de do usuário de ter intervalos de descanso da experiência.

Como o usuário não sente a aceleração ou desaceleração que pode ser apresentada na aplicação, utilizar velocidade constante na maior parte do tempo ajuda também a minimizar o desconforto. Ao criar experiências nas quais o usuário está fixo no virtual, é interessante que este esteja fixo na realidade. Isso pode ser feito ao posicioná-lo em uma cadeira ou colocar referências no espaço para que ele saiba que o que se move são os objetos ao seu redor e não ele. Isso pode acontecer quando um objeto grande se move próximo ao usuário e causa desconforto. Deve-se evitar mudanças bruscas de iluminação, pois os olhos precisam de um tempo para se acomodar à mudança, e como a tela fica muito próxima dos olhos isso causa desconforto.

2.4.4 Diretrizes para interatividade

“Para evitar que o usuário sinta-se apressado, certifique-se de que a experiência inicia apenas quando o usuário indicar que está pronto.”(GOOGLE, 2018) É interessante oferecer a possibilidade do usuário informar que está pronto através de clique em uma tela inicial.

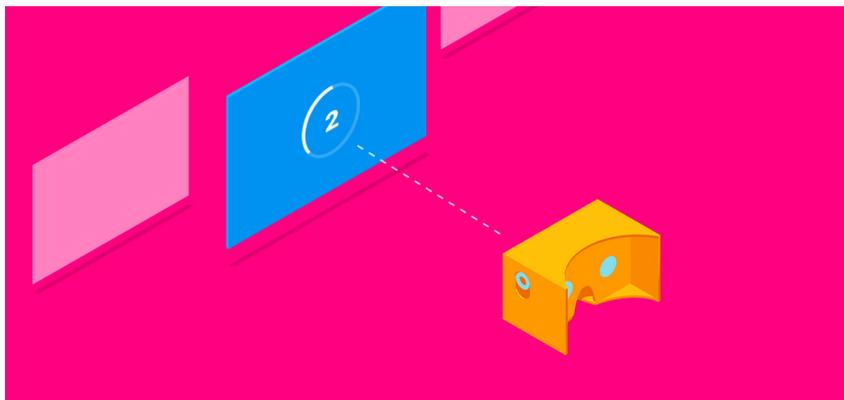
Outros pontos são:

- Opções iniciais devem estar dentro do campo de visão do usuário desde o início. Isso evita confusão que faz o usuário esperar e sentir-se perdido;
- Opções de jogo devem fazer parte de uma interface que acompanhe a localização do personagem.

Pode-se fazer uso de botões virtuais em vez do físico, presente no cardboard. Estes podem trabalhar com base em tempo, um contador, para que a ativação destes botões sejam percebidas pelo usuário e as mudanças na aplicação não pareçam ocorrer aleatoriamente por cliques não percebidos, como visto na Figura 17. Botões virtuais devem ser posicionados distantes uns dos outros e devem ser grandes para evitar cliques indesejáveis. Botões que fiquem próximos e sejam pequenos devem ser ativados apenas com cliques.

Uma boa prática é sempre manter a interação mais natural e tridimensional possível. Isto é primordial ao utilizar-se de menus e interfaces. Estes devem ser posicionados dentro do espaço tridimensional, pois, de acordo com a Unity Technologies (2018h) não é possível para o usuário focar em elementos que fiquem tão próximos e isso ocorre caso os menus sejam projetados diretamente na câmera, como em aplicações tradicionais.

Figura 17 – Imagem Ilustrando o funcionamento de um Botão fusível(*Fuse Buttons*)



Fonte: (GOOGLE, 2018)

2.4.5 Diretrizes para Áudio

"Áudio tem um papel crítico na imersão RV."(FACEBOOK TECHNOLOGIES, LLC., a) Este pode ser utilizado como fonte de informação direta, como visto nas diretrizes da Google () que recomenda o uso de clipes curtos de áudio para dar explicações em vez de texto. Pode ser utilizado para dar senso de localização quando corretamente espacializado. Para tanto a Facebook Technologies, LLC. (a) explica que o áudio deve parecer vir da mesma que a fonte.

Para aplicações que rastreiam a posição, o som deve ficar mais alto à medida que o usuário se aproxima da fonte e uniforme se o avatar está estacionário. Porém, nem todos os sons precisam ser espacializados. Sons como ruído ambiente, música de fundo e outros sons que não possuem fonte espacial óbvia podem ser executados sem espacialização.

3 DESENVOLVIMENTO DE UMA APLICAÇÃO PARA NAVEGAÇÃO VIRTUAL NO CCET

Modelar o CCET, de maneira a construir relação visual forte, requer que haja certa proximidade entre os modelos e a realidade. “Simular realismo poderia indiscutivelmente ser o objetivo geral da RV, o nome já é apropriado: “realidade virtual””(HUNTER, 2017).

3.1 AQUISIÇÃO DE MATERIAL BASE

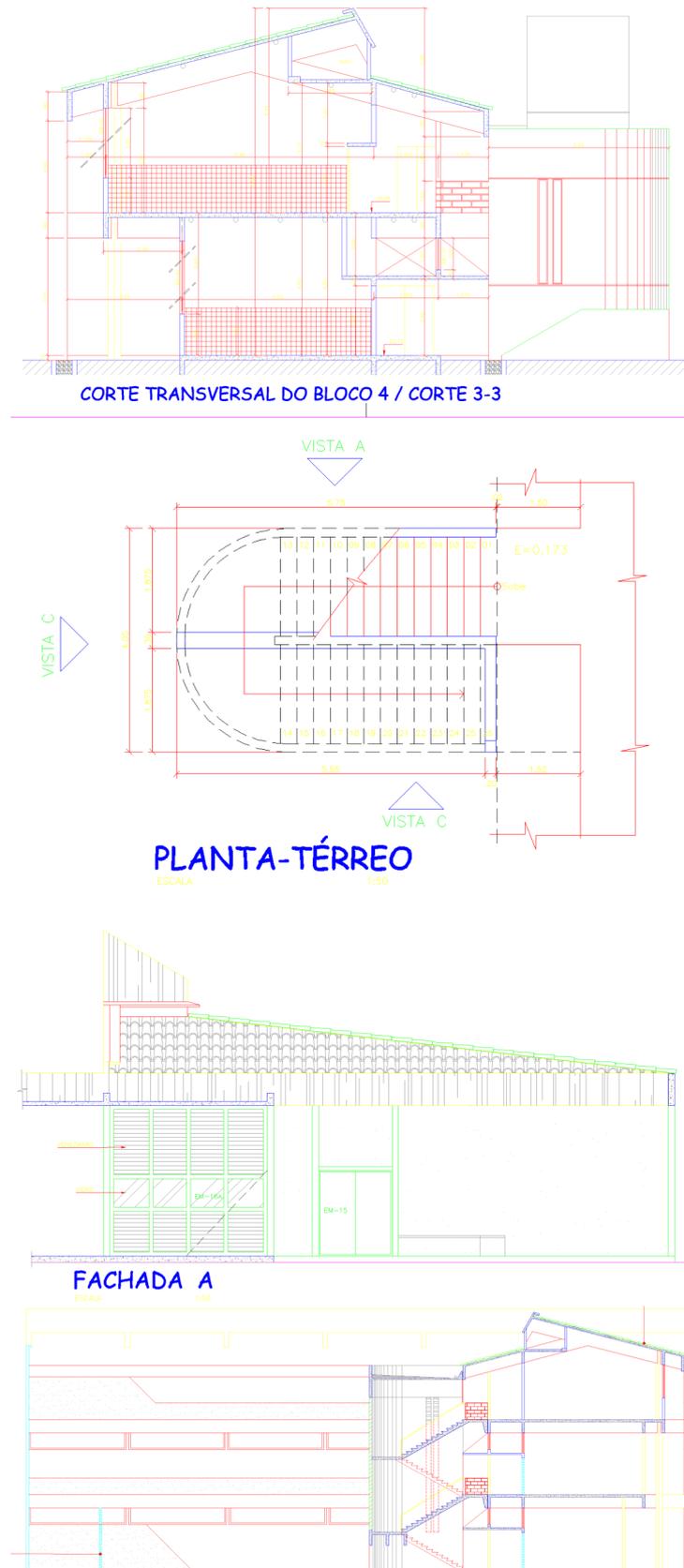
Plantas baixas servem de referência para a parte estrutural. Os arquivos foram cedidos pela Prefeitura da Cidade Universitária¹. O conjunto é composto por treze arquivos no formato DWG². Destes, onze são dados originais e dois são atualizações. Estão presentes as plantas para todos os pisos. Vistos de cima, fachadas norte, sul, leste, oeste, além de cortes que detalham o interior do prédio assim como estruturas secundárias, como auditórios e salas separadas da estrutura principal.

No conjunto de plantas baixas tem-se as medidas para micro e macro estruturas, marcas visuais e textuais sobre estilo arquitetônico e materiais utilizados. Essas informações estão distribuídas em vários cortes que evidenciam desde pisos, visão externa de várias direções, detalhamento de estruturas como escadas até coberturas. Uma amostra desse conjunto pode ser verificada na Figura 18.

¹ <http://portais.ufma.br/PortalUnidade/precam/>

² <https://www.scan2cad.com/dwg/history-future-dwg/>

Figura 18 – Amostra do conjunto de plantas baixas da UFMA



Fonte: Adaptado de arquivo da Prefeitura da Cidade Universitária- UFMA.

Recortou-se as imagens em novos conjuntos para agrupar áreas de interesse como pisos, vistas laterais e detalhes mais específicos como escadas, janelas e treliças. O resultado para a modelagem do prédio, à partir desse material, pode ser parcialmente visualizado ao comparar-se a planta baixa para o térreo, vista na Figura 19 com o modelo para o térreo visto do mesmo ângulo, visto na Figura 20, e também ao se observar a Figura 21, pela qual é possível comparar uma foto ao modelo.

Figura 19 – Imagem da UFMA, modelo tridimensional, visto de cima.

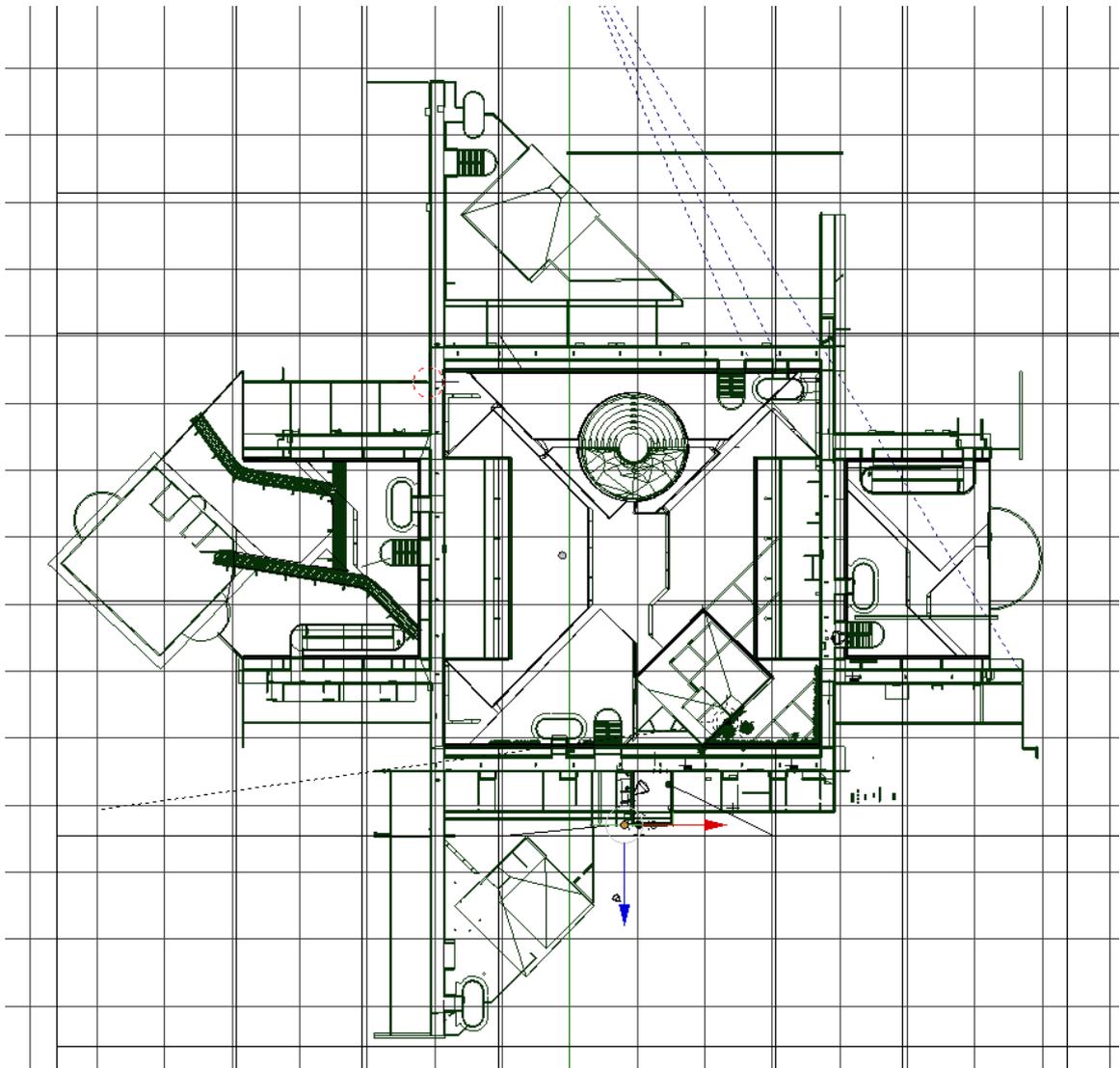
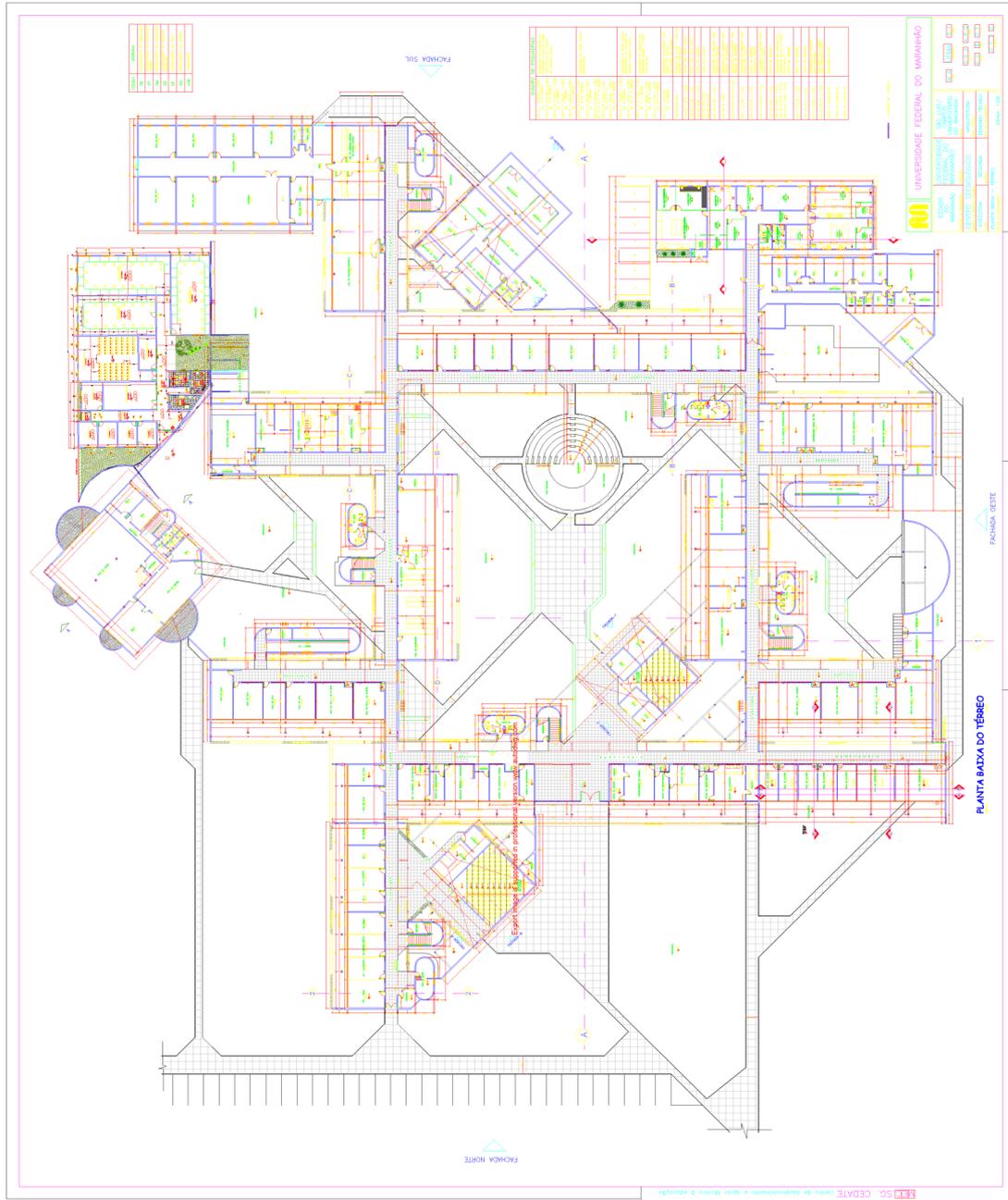


Figura 20 – Imagem da UFMA vista de cima, planta baixa.



Fonte: Adaptado de arquivo da Prefeitura da Cidade Universitária- UFMA.

Figura 21 – Foto à esquerda e imagem do modelo à direita.



Mais de quinhentas fotos foram capturadas para referência de materiais, texturas, objetos de segurança, vasos, quadros, ladrilhos, plantas, cadeiras, lixeiras, bancos, quadros de aviso, quadros de força, tubulações.

A maior parte da aquisição de fotos deu-se entre 15:00h e 17:00h, no sentido anti-horário, em cada piso, para que a separação e classificação das imagens fosse facilitada, aproximada da organização das plantas baixas e a iluminação não tivesse grande divergência entre um conjunto de imagens e outro. Na Figura 22 é possível verificar esses aspectos à partir de algumas das imagens capturadas.

Figura 22 – Fotos da UFMA



3.2 FERRAMENTAS UTILIZADAS

Utilizou-se o Unity, motor de jogos desenvolvido pela Unity Technologies, para implementar a renderização, comportamento e iteratividade da aplicação. Esta ferramenta apresenta

suporte à maioria das plataformas como Windows, Linux, Android, IOS e consoles. Apresenta também um fluxo de trabalho simplificado ao ser capaz de extrair modelos de arquivos do Blender, mantê-los atualizados, avaliar e apresentar atualização de código de maneira ágil, além de suportar a tecnologia de Realidade Virtual do Cardboard da Google através do plugin Google VR SDK³ para Unity.

Uma engine de jogo é um *framework* de desenvolvimento de jogos que suporta e reúne várias áreas importantes. Você pode importar gráficos e recursos 2D e 3D de outros *softwares*, como Maya, 3s Max ou Photoshop, montá-los em cenas e ambientes, adicionar iluminação, áudio, efeitos especiais, física e animação, interatividade e lógica de jogo ; e editar, depurar e otimizar o conteúdo para suas plataformas de destino.(UNITY TECHNOLOGIES, 2018b)

Para a criação e organização dos modelos tridimensionais no espaço, texturas, mapeamento de texturas e estudo da iluminação, utilizou-se o Blender. Esta é uma suíte para criação 3D, de código aberto, desenvolvida pela Fundação Blender. Suporta tarefas de criação, edição e texturização de modelos tridimensionais bem como outros aspectos ligados à criação e manipulação de imagem, vídeo e efeitos especiais.

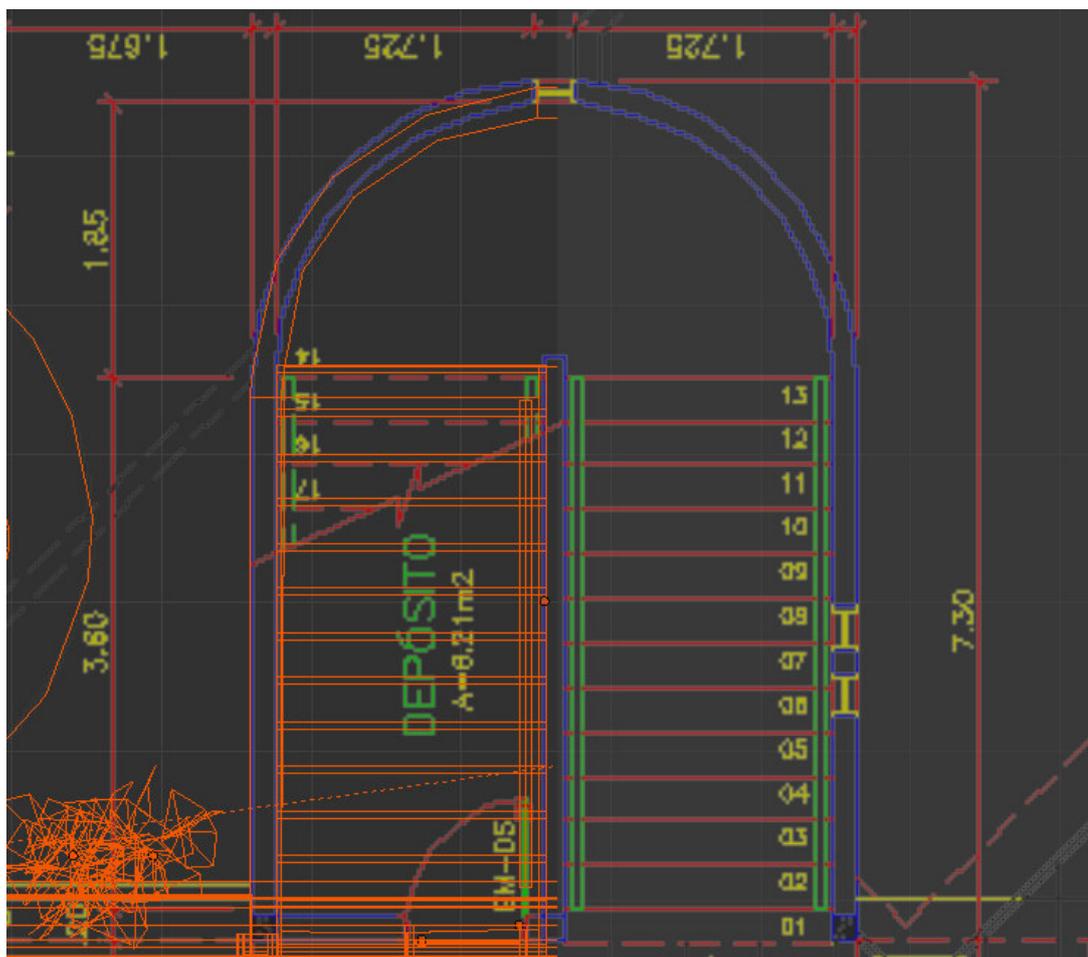
Ele suporta todo o fluxo de modelagem, armação, animação, simulação, renderização, composição e acompanhamento de movimento, até mesmo edição de vídeo e criação de games.(BLENDER FOUNDATION, 2018, traduzido pelo autor)

No Blender, é possível importar arquivos de imagem para a janela de visualização como plano de fundo, como visto na Figura 23. Isso facilita a comparação entre os modelos e a referência. Os arquivos com as plantas baixas foram originalmente gerados por softwares da AutoDesk com extensão de arquivo DWG, que não são suportados pelo Blender. Para converter tais arquivos em imagens com extensão PNG⁴, compatível com o Blender, a função de exportação do software DWGSee foi utilizada.

³ <https://github.com/googlevr/gvr-unity-sdk>

⁴ <http://www.libpng.org/pub/png/pnghist.html>

Figura 23 – Planta baixa do CCET ao fundo e arestas do modelo tridimensional

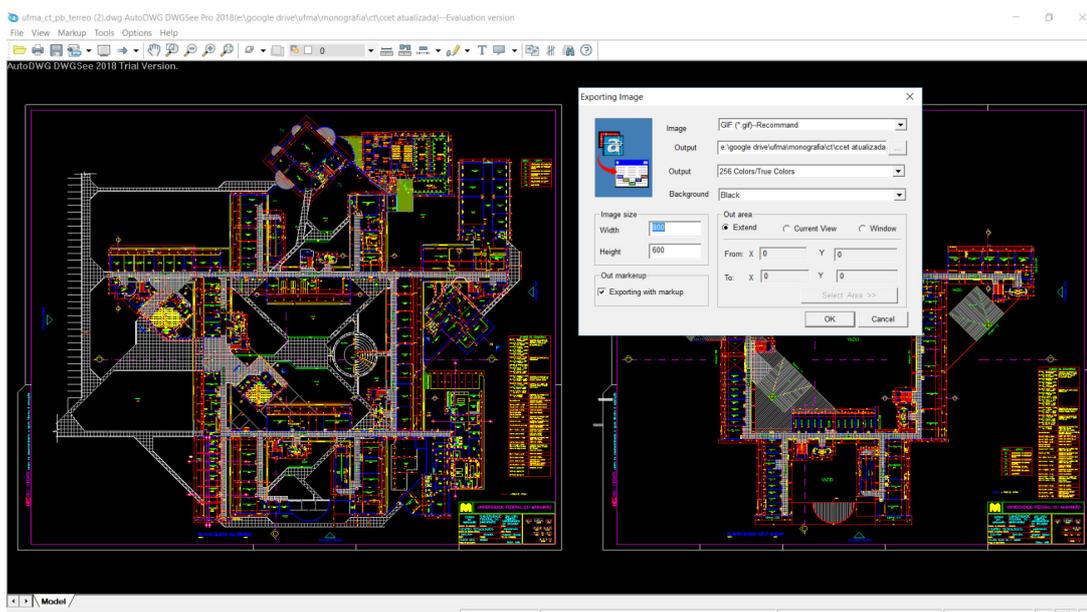


O DWGSee⁵ é um visualizador para arquivos DWG/DXF/DWF desenvolvido pela AutoDWG⁶. Esse software suporta navegação, medições, impressão e exportação desses formatos de arquivo como visto na Figura 24.

⁵ <https://dwgsee.com/>

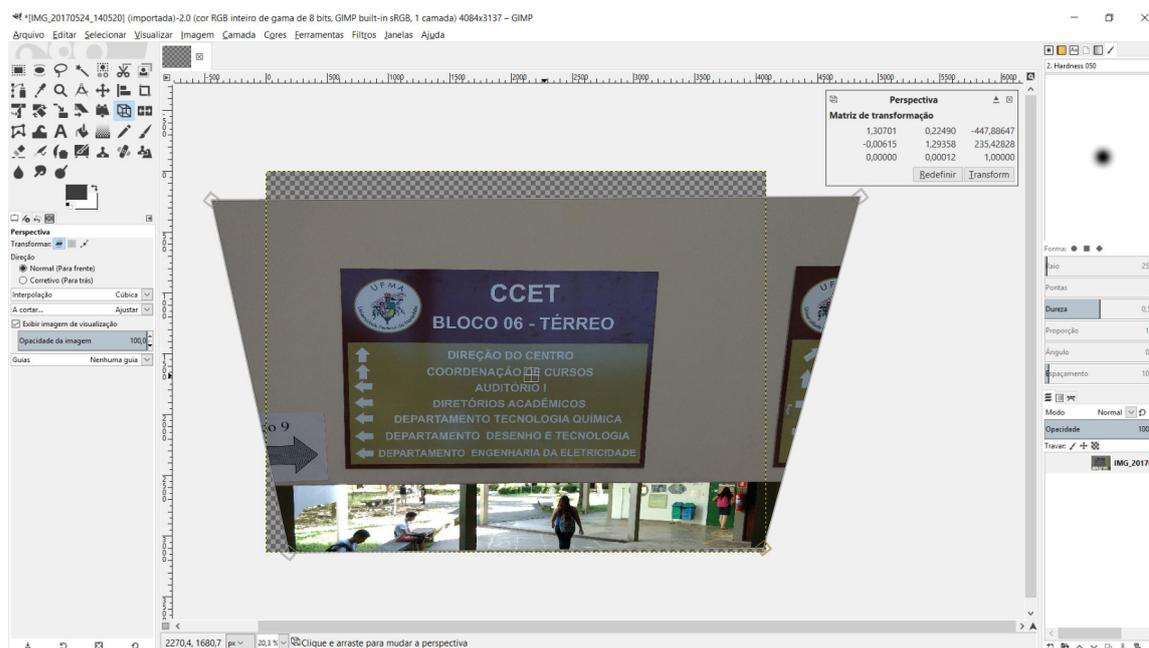
⁶ <https://www.autodwg.com/default.aspx>

Figura 24 – Captura de Tela da interface de exportação do DWGSee



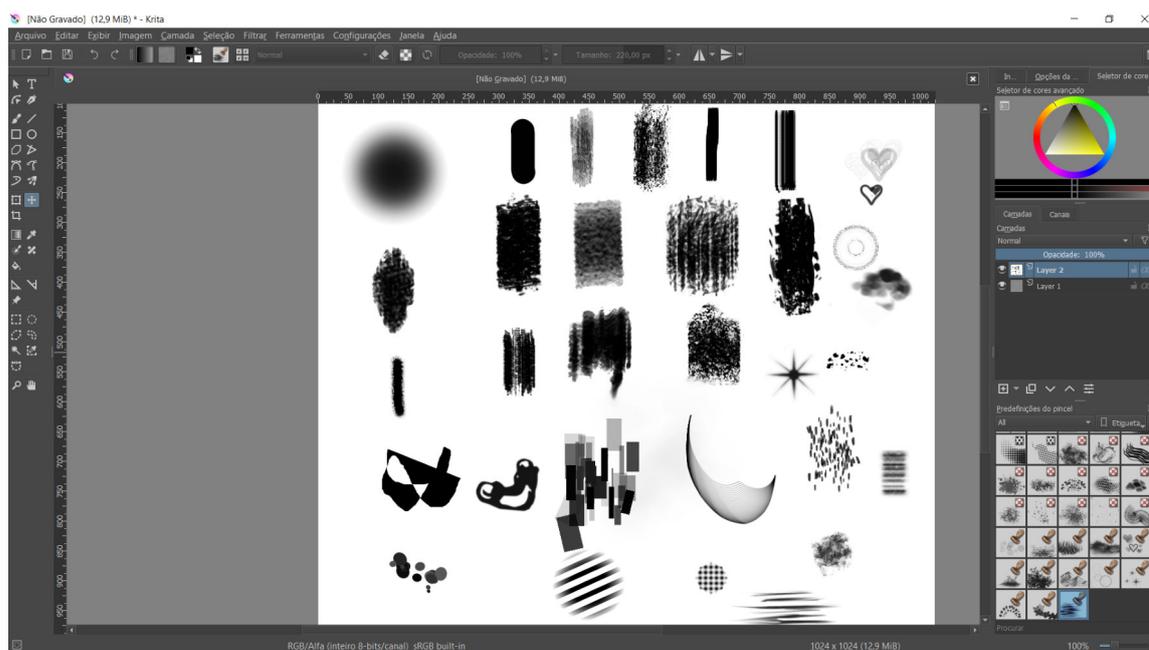
Vários dos arquivos das plantas baixas apresentam mais de uma vista ou pisos por imagem. Para facilitar o uso estas foram divididas em múltiplas imagens menores que apresentam apenas uma parte, vista ou piso do prédio. Em alguns casos foi necessário ajustar a perspectiva de algumas fotos de referência para serem utilizadas como texturas, como visto na Figura 25. Para ambos utilizou-se o GIMP(GNU Image Manipulation Program). Este é um software para manipulação de imagens, de código aberto, mantido pelo time GIMP, capaz de realizar processos básicos como recortes até a aplicação de filtros e edições complexas. Utilizou-se o GIMP, assim como o Krita para criar e editar texturas.

Figura 25 – Ajuste de Perspectiva no Gimp



Krita é um programa profissional, gratuito e de código aberto, para pintura digital. Este oferece ferramentas que facilitam a criação de imagens encaixáveis e pincéis que imitam texturas como madeira e carvão como observado na Figura 26. Esta ferramenta auxilia na criação de padrões que se assemelham às variações em texturas vistas na natureza ou manufaturadas.

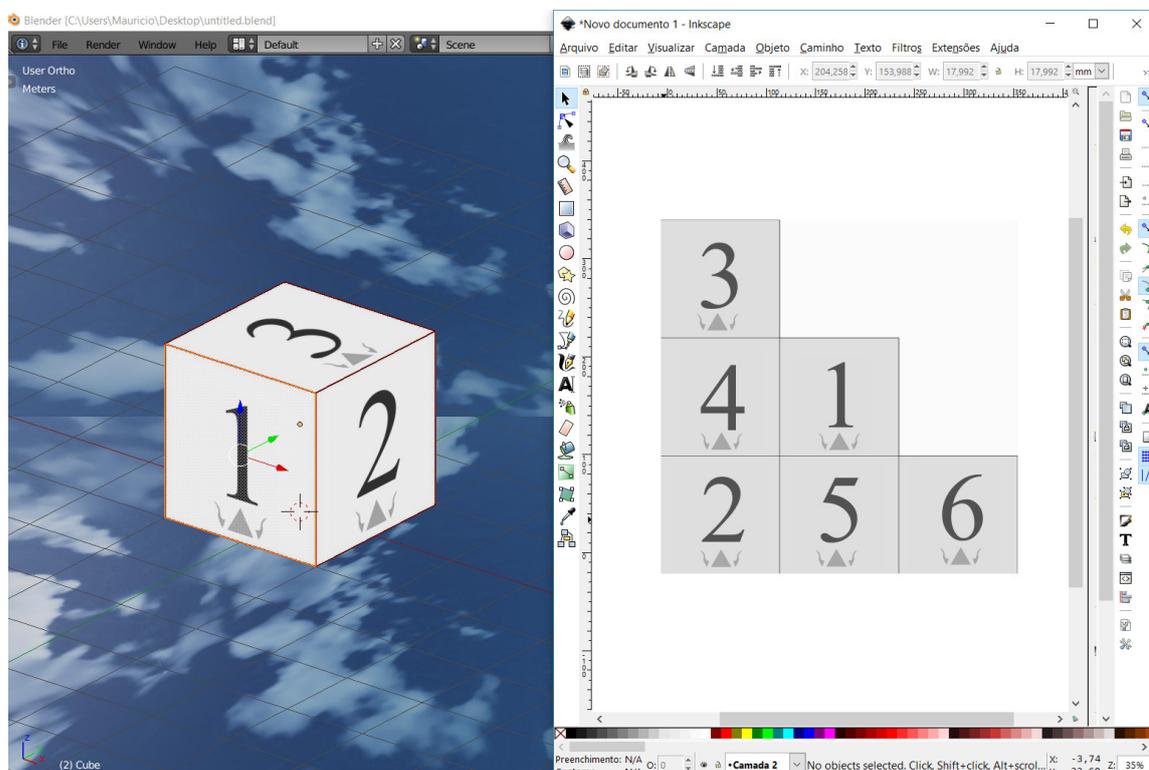
Figura 26 – Amostra de pincéis do Krita



Para criar padrões com precisão que assemelham-se a texturas criadas mecanicamente ou imprimir detalhes, como nomes e logomarcas, sobre outras texturas utilizou-se o Inkscape. Esta

ferramenta é usada para trabalhar com Gráficos Vetoriais Escalonáveis(SVG). Suporta criação, edição e exportação de gráficos vetoriais. Na Figura27 é possível ver um exemplo de gráfico projetado no objeto à partir da textura modificada pelo Inkscape.

Figura 27 – Imagem mapeada em um cubo à esquerda e imagem aberta no Inkscape à direita



Várias das texturas, geradas ou adquiridas, foram importadas no *Awesome Bump*⁷. Este é um software livre, projetado para gerar texturas com informação sobre a normal, altura, reflexão especular ou oclusão ambiente à partir de uma imagem. Foi utilizado para criar versões diferentes de materiais para as texturas básicas.

Adquiriu-se alguns modelos tridimensionais, para referência e material de base, de sites que os oferecem gratuitamente como *turbosquid*⁸ e *archive3d*⁹. Parte destes modelos estavam no formato 3DS. Para converter estes arquivos, para um formato compatível com o Blender, utilizou-se a função de exportação para OBJ da suíte 3DSMAX. Um exemplo disso é o modelo de caixa eletrônico visto na Figura 28, obtido à partir do site *cadnav*¹⁰.

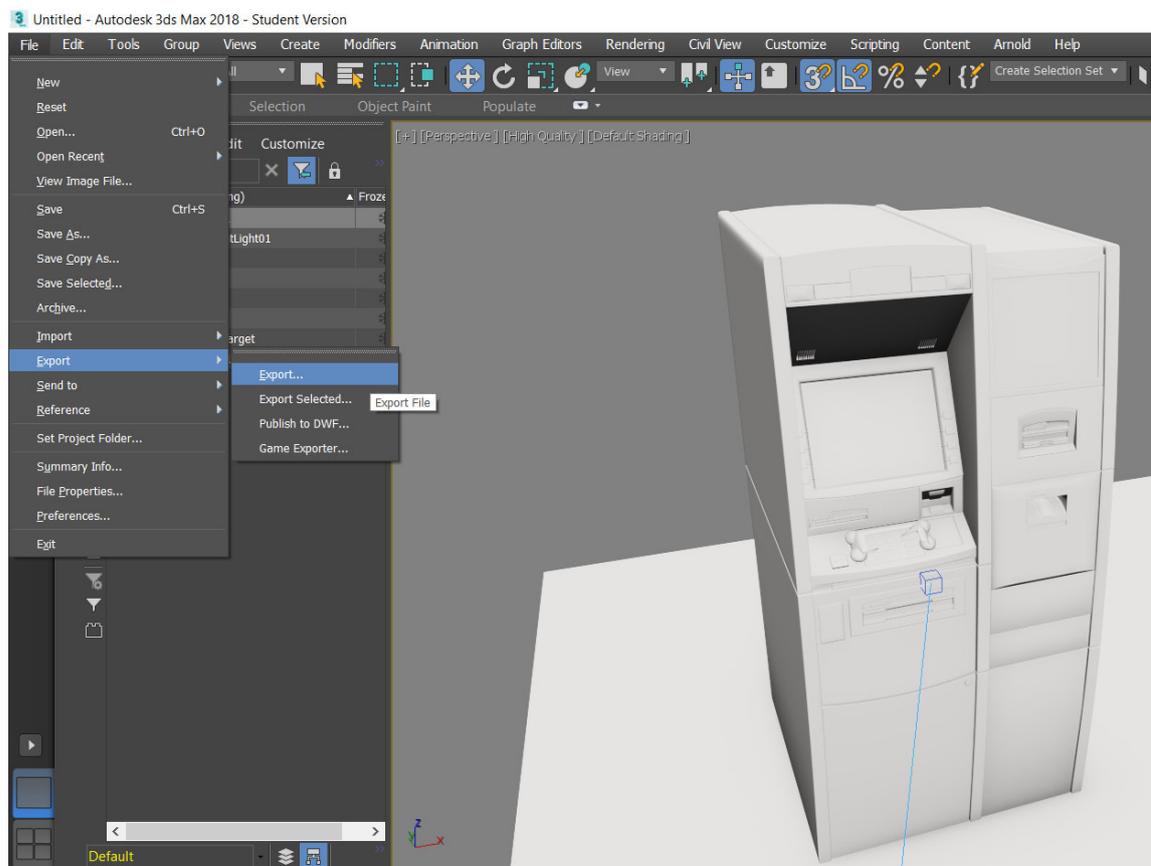
⁷ <https://github.com/kmkolasinski/AwesomeBump>

⁸ <https://www.turbosquid.com>

⁹ <https://archive3d.net/>

¹⁰ <http://www.cadnav.com>

Figura 28 – Interface do 3DSMAX, localização da função de exportação



3.3 MODELAGEM

À partir do conjunto de imagens de referência, modelou-se o térreo, o primeiro e o segundo piso, excluindo-se coberturas para estruturas inacessíveis e para o terceiro piso. Este processo seguiu os mesmos padrões estabelecidos para a captura de imagens em termos de divisões e direções.

Criar um modelo tridimensional, na computação gráfica, significa criar uma representação matemática que pode ser interpretada para gerar um conjunto de superfícies. Para facilitar esta tarefa, softwares de modelagem são utilizados em conjunto com técnicas que facilitam a criação de diferentes tipos de modelos.

A escolha da técnica mais adequada para cada aplicação e objeto a ser modelado determina a rapidez, com a qual o modelo é criado, e a qualidade mais próxima do nível necessário. Algumas dessas técnicas de modelagem são apresentadas no Quadro 3.

Na computação gráfica, modelagem 3D(ou modelagem tridimensional) é o processo de desenvolver uma representação matemática de qualquer superfície tridimensional de um objeto (inanimado ou vivo) via software especializado. O produto é chamado de modelo 3D. Modelos podem ser criados automaticamente ou manualmente.(ARCHITECTURAL CGI, 2018, traduzido pelo autor)

Quadro 3 – Técnicas de Modelagem

Nome	Descrição
Modelagem de Caixa/Subdivisão	Tipo de modelagem poligonal na qual o artista usa forma geométrica, como um cubo , cilindro ou esfera e o molda até alcançar a aparência pretendida. Inicia-se com uma malha com baixo número de polígonos.Então subdivide-se a malha, garantindo que os cantos sejam suavizados e detalhes necessários sejam adicionados. Repete-se o processo até que haja detalhe suficiente presente na malha para transmitir o conceito desejado.
Modelagem de Contorno/Aresta	Nesse processo os modelos são constuídos parte por parte em vez de refinados à partir de uma forma primitiva. Isto é feito ao se dispor cadeias de polígonos ao longo de contornos e preenchendo os espaços entre eles.
Modelagem com curvas/NURBS(Non-Uniform Rational B-Splines) - Curvas de Bezier Racionais não Uniformes	Modelagem à partir de curvas e superfícies determinadas por parâmetros como polinômios, combinadas até que se aproxime a forma desejada. Polígonos são dispostos entre as curvas e gerando a superfície.
Modelagem por Escultura Digital	Processo no qual formas básicas são esculpidas e combinadas repetidamente até que seja alcançada a forma desejada.
Modelagem Procedural	Processo de modelagem no qual as formas são geradas por algoritmos que recebem parâmetros. Como exemplo uma árvore pode ter sua altura, número de galhos, folhas etc determinados por parâmetros e ser construída através de algoritmos de geração de árvores.
Escaneamento 3D	Processo de derivar modelos à partir do escaneamento 3D de formas do mundo real. Captura-se e combina diversas imagens para calcular parâmetros de um objeto como profundidade e detalhes de superfície através de algoritmos.

Pode-se iniciar a modelagem criando um objeto geométrico básico, no *Blender* conhecido como primitiva, como uma face, um cubo, uma esfera e até mesmo um vértice. Para o *Blender* algumas primitivas são plano, cubo, cilindro e esfera como visto na Figura 29.

Figura 29 – Imagem das primitivas do Blender



À partir das primitivas, técnicas de modelagem poligonal como a de caixa e contorno foram utilizadas para criar os objetos mais próximos de formas geométricas como paredes, bancadas, bancos e peitoris. Ao utilizar essa técnica, modelos são criados ao modificando-se diretamente os polígonos de uma malha ao mover seus vértices, arestas e faces, permitindo assim um controle direto sobre a densidade da malha, ou seja, quantidade de vértices, faces e arestas.

Pode-se aplicar várias operações na malha do objeto para que sua forma se aproxime da desejada. Operações são aplicadas de maneira não editável, ou seja, não podem ter seu resultado alterado por parâmetros. Algumas das operações são rotação, translação, extensão e corte como exemplificado na Figura 30. A lista de operações pode ser vista na Figura 31

Figura 30 – Exemplos de operações sobre a malhas no Blender

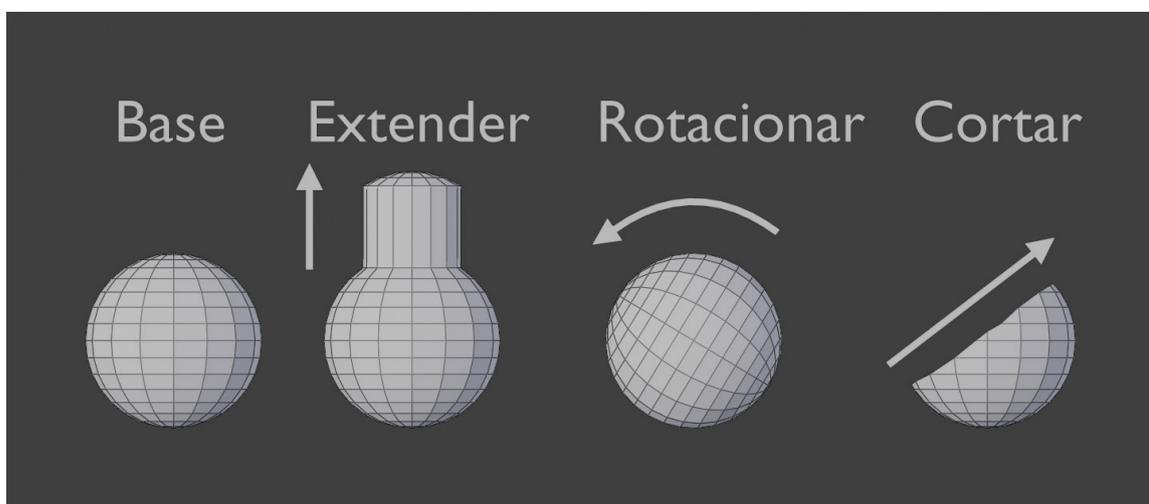
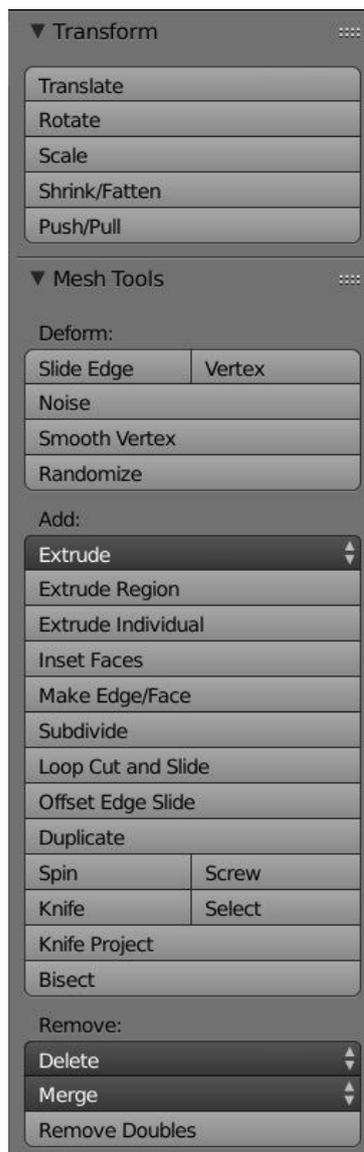


Figura 31 – Captura de tela do menu de operações para malhas no Blender.



Fonte: Captura de tela da aplicação no sistema operacional Windows 10(2018)

Utilizou-se modificadores para gerar objetos detalhados à partir de malhas simples e parâmetros. Na Figura 32 é possível observar os modificadores pele, subdivisão e estrutura em arames habilitados que, respectivamente, cria uma malha tridimensional em torno de arestas, subdivide a malha gerada e gera uma nova malha à partir das arestas da última malha gerada. Isso é possível, pois os modificadores podem ser aplicados em sequência uns sobre os outros. Uma lista de modificadores é vista na Figura 33.

Figura 32 – Modelo com modificadores desabilitados à esquerda e habilitados à direita

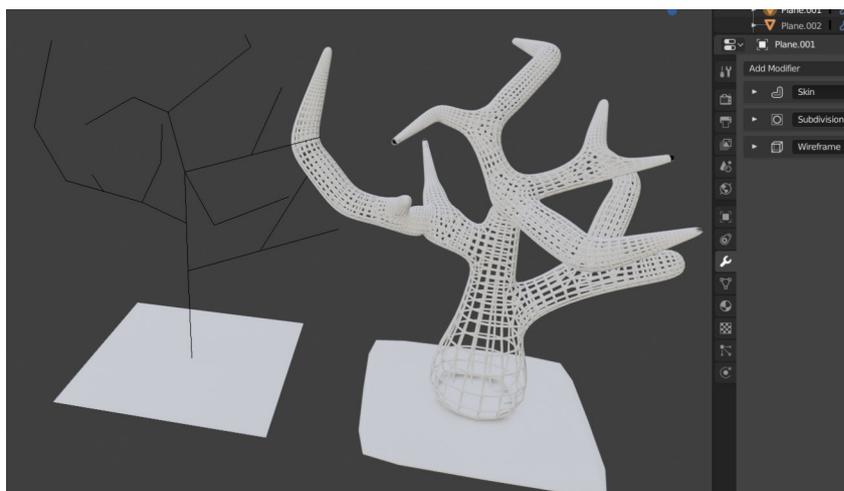
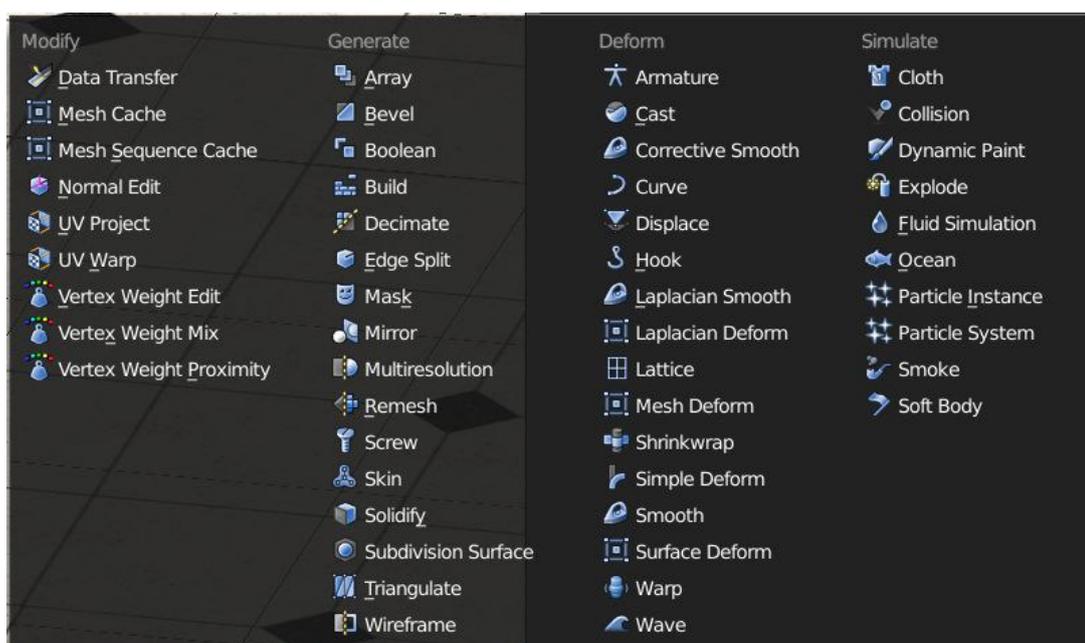


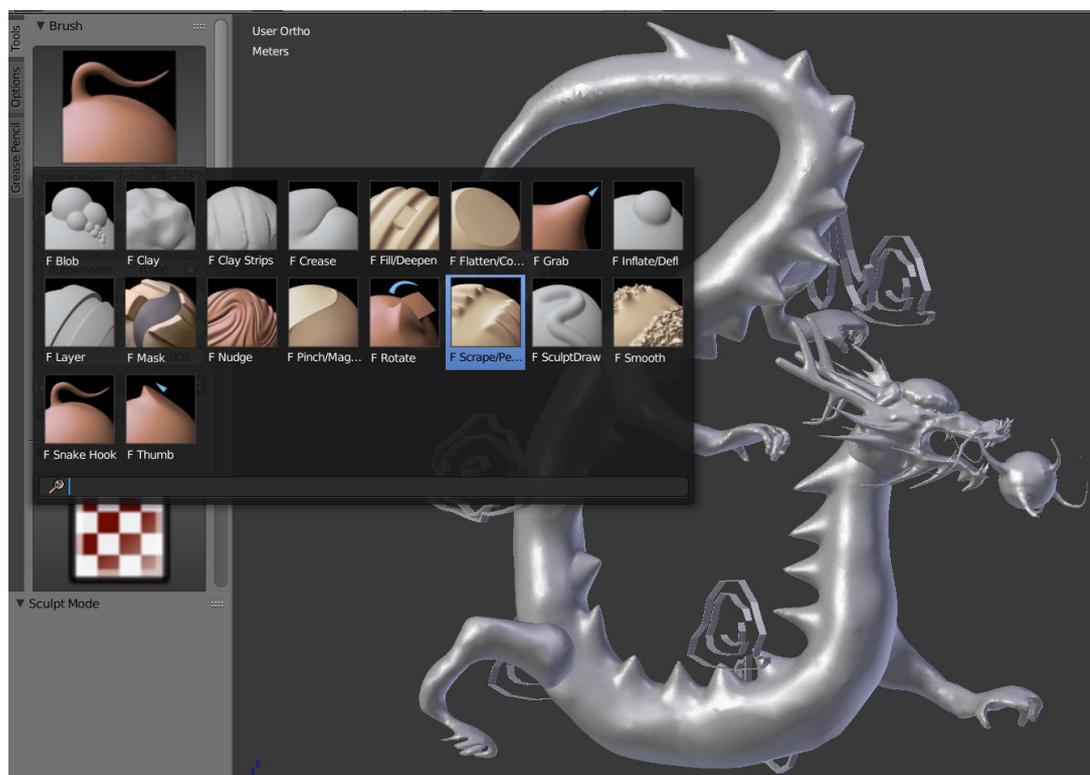
Figura 33 – Captura de tela do menu de modificadores para objetos no *Blender*



Fonte: Captura de tela da aplicação no sistema operacional Windows 10(2018)

Criou-se objetos com formas mais orgânicas e detalhadas como jarros e plantas à partir da escultura digital. Nesta técnica não há edição direta de estruturas básicas como vértices, arestas ou faces, mas a indicação de operações que se deseja aplicar sobre a malha utilizando pincéis virtuais. Estes pincéis utilizam configurações que indicam como ele irá interagir com uma malha. Estas interações podem criar bolhas, picos, montes, buracos, cortes e ranhuras. Um objeto esculpido, assim como o conjunto de pincéis disponíveis podem ser vistos na Figura 34.

Figura 34 – Captura de tela do menu de pincéis para escultura no blender e dragão chinês esculpido



Fonte: Captura de tela da aplicação no sistema operacional Windows 10(2018)

3.3.1 Otimização de Modelos

Para que a utilização de aplicações de Realidade Virtual seja confortável é necessário garantir uma alta taxa de quadros. Para isto, vários meios de otimização são aplicados sobre os a malha dos objetos. Fazer o balanço entre fidelidade visual e desempenho é um dos grandes desafios enfrentados pelos desenvolvedores desse tipo de aplicação.

Simplifique suas malhas o máximo possível. Dependendo de sua plataforma pretendida, você pode querer adicionar detalhes via texturas, mapeamento parallaxe e tesselação, apesar disso poder impactar desempenho e não ser adequado ou estar disponível para a plataforma pretendida.(UNITY TECHNOLOGIES, 2018e)

Simplificou-se ou remodelou-se objetos importados como o caixa eletrônico anteriormente citado. Para isso utilizou-se o modificador "dizimar"(*decimate*), "remendar"(*remesh*) e operações. Um exemplo do procedimento pode ser visto na Figura 35. Na Figura 36 é possível observar a diferença entre e o modelo original de caixa eletrônico e o modelo otimizado. Para objetos estáticos, que não são animados, além destas técnicas, deletou-se faces que não são visualizadas na aplicação.

Figura 35 – Exemplos de otimização



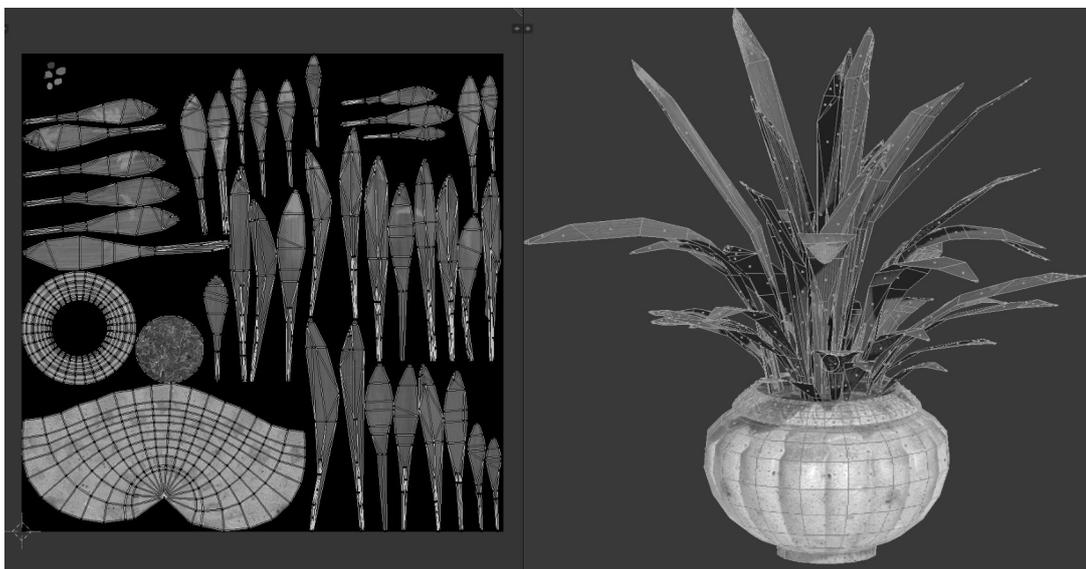
Figura 36 – Modelo de caixa eletrônico otimizado à esquerda e original à direita



Para os objetos esculpidos foram necessários mais passos que envolvem fazer pré-computação de alguns parâmetros e guardá-los em imagens chamadas de texturas. Para mapear as texturas, após a modelagem, o objeto ainda pode passar por outros processos como a criação de mapa UV. Um mapa UV é criado ao se projetar as faces de um objeto tridimensional em um plano UV, bidimensional, como na Figura 37. Isso permite o mapeamento de informações guardadas em um plano bidimensional para a superfície de um objeto tridimensional. Esse processo é necessário para que o alto nível de detalhes não seja totalmente perdido na etapa de

simplificação da malha. No caso de objetos esculpidos a otimização se dá pela criação de um novo objeto que se aproxime da forma mais básica do original. Este recebe, posteriormente, os detalhes guardados em texturas ou mapas.

Figura 37 – imagem do Mapa UV à direita e do modelo tridimensional à esquerda



3.4 MATERIAL E TEXTURA

Texturas podem conter vários tipos de informações como cor, fator de refletividade, fator de dureza, fator metálico, direção, profundidade etc. Uma textura guarda as informações para a representação de um material. No entanto texturas não são estritamente necessárias para um material.

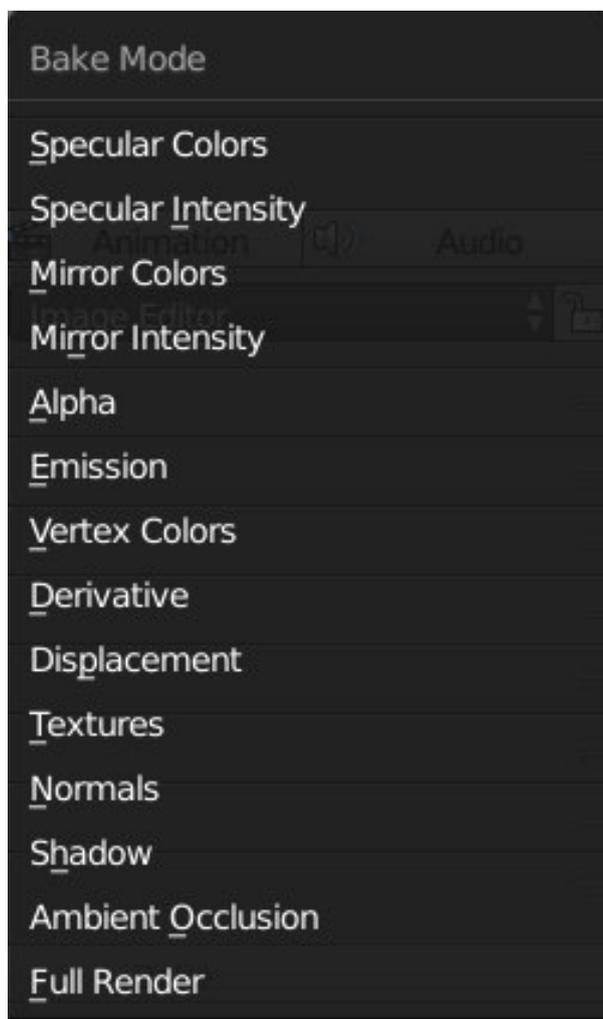
Um material é a representação da maneira que a luz interage com um objeto. Na realidade observa-se vários materiais como plástico, ouro, alumínio, virtualmente esses materiais são modelados dentro do software que irá gerar uma aproximação de seu comportamento, em relação à luz, no mundo real.

Um material define as qualidades artísticas da substância da qual um objeto é feito. Em sua forma mais simples, você pode utilizar materiais para mostrar a substância da qual é feito um objeto, ou pintar o objeto com cores diferentes. Geralmente, a substância é representada pelas características de sua superfície (cor, brilho, refletividade etc.), mas também pode exibir efeitos mais complicados como transparência, difração e dispersão de subsuperfície (subsurface scattering). Materiais típicos podem ser latão, pele, vidro ou linho. (BLENDER DOCUMENTATION TEAM, 2017)

Para evitar cálculos excessivos e garantir que os valores de cada propriedade fossem guardados de maneira portátil, acessível e editável, estes valores são guardados em texturas

utilizando mapas UV. Dentro do *blender* isso é feito num processo chamado *bake*(cozinhar). Essas texturas foram exportadas e reprojctados no objeto dentro da *Unity* utilizando os mapas UV criados para os objetos. A lista de texturas possíveis pelo *bake* é vista na Figura 38.

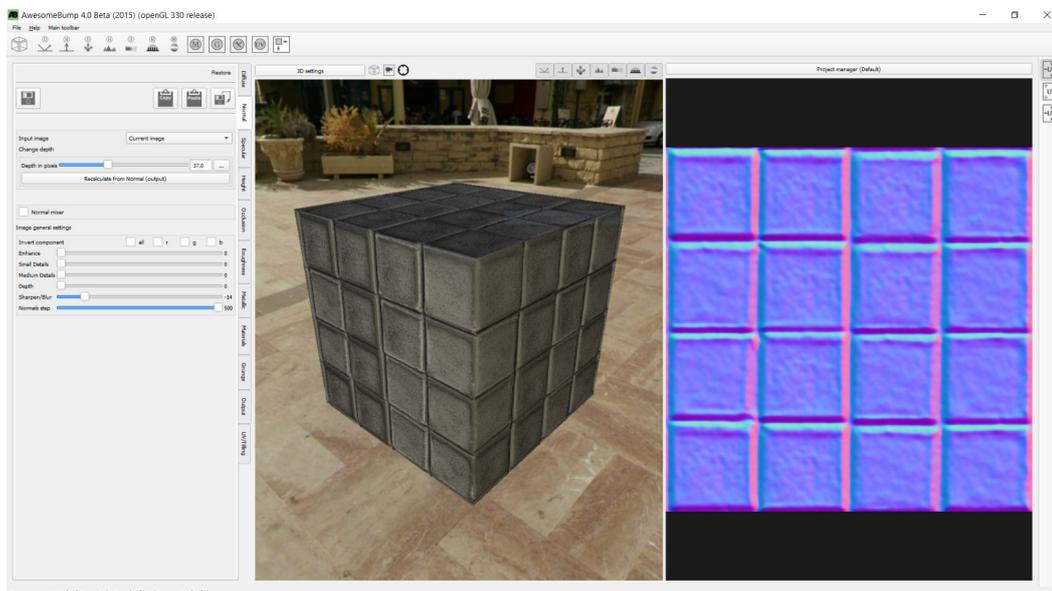
Figura 38 – Captura de tela do menu de opções de baking no blender com renderizador interno na versão 2.79b.



Fonte: Captura de tela da aplicação no sistema operacional Windows 10(2018)

Além do método anterior, foram utilizados algoritmos do *Awesome Bump* sobre fotos para a criação de texturas encaixáveis. Este programa possibilitou a produção de diversas texturas com informações como direção dos vetores normais(*normal map*), cor de difusão(*difuse map*), intensidade da reflectividade(*specular map*), aspereza(*roughness map*), fator metálico(*metallic map*) e outros. Na Figura 39 é possível verificar um mapa de vetores normais, gerado à partir de uma foto, aplicado sobre um cubo.

Figura 39 – Interface do *Awesome Bump*. Opções à esquerda, previsualização ao centro e mapa normal à direita



Fonte: Captura de tela da aplicação no sistema operacional Windows 10(2018)

Criou-se ainda imagens em blocos encaixáveis utilizando-se o *Krita*. Para isto utilizou-se sua coleção de pincéis "orgânicos" em combinação. Com esses recursos foram geradas algumas texturas para materiais como cimento e granito. Resultados como os da Figura 40 e 41 podem ser alcançados dessa maneira.

Figura 40 – Imagem encaixável criada no krita.

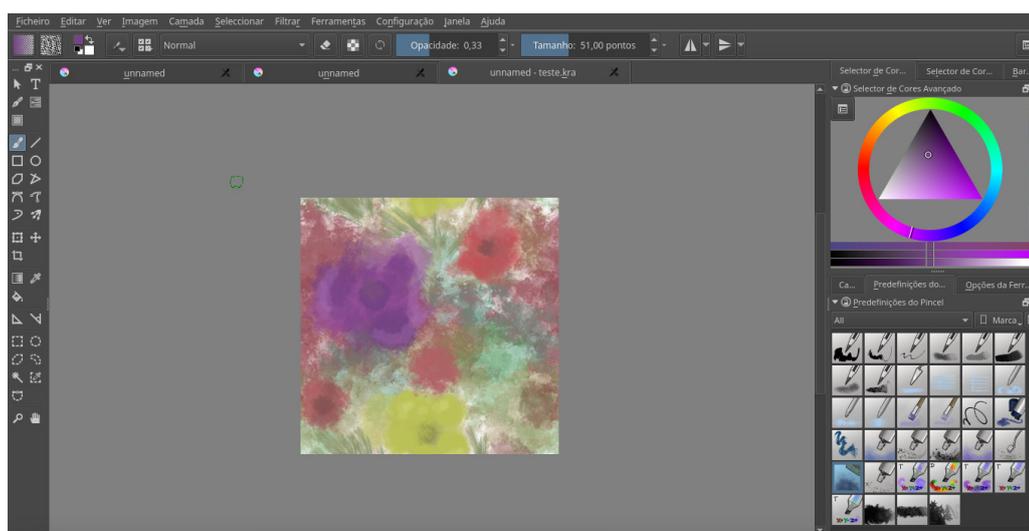
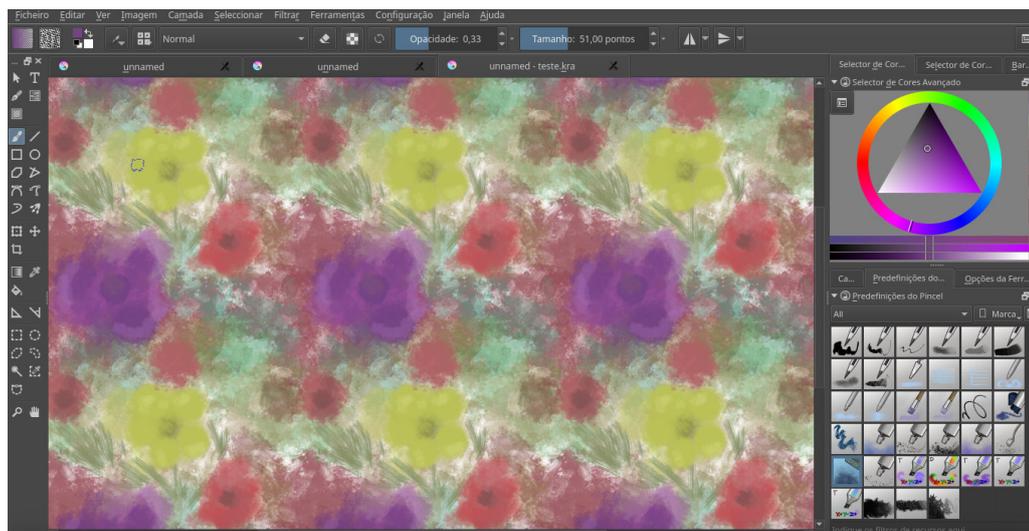
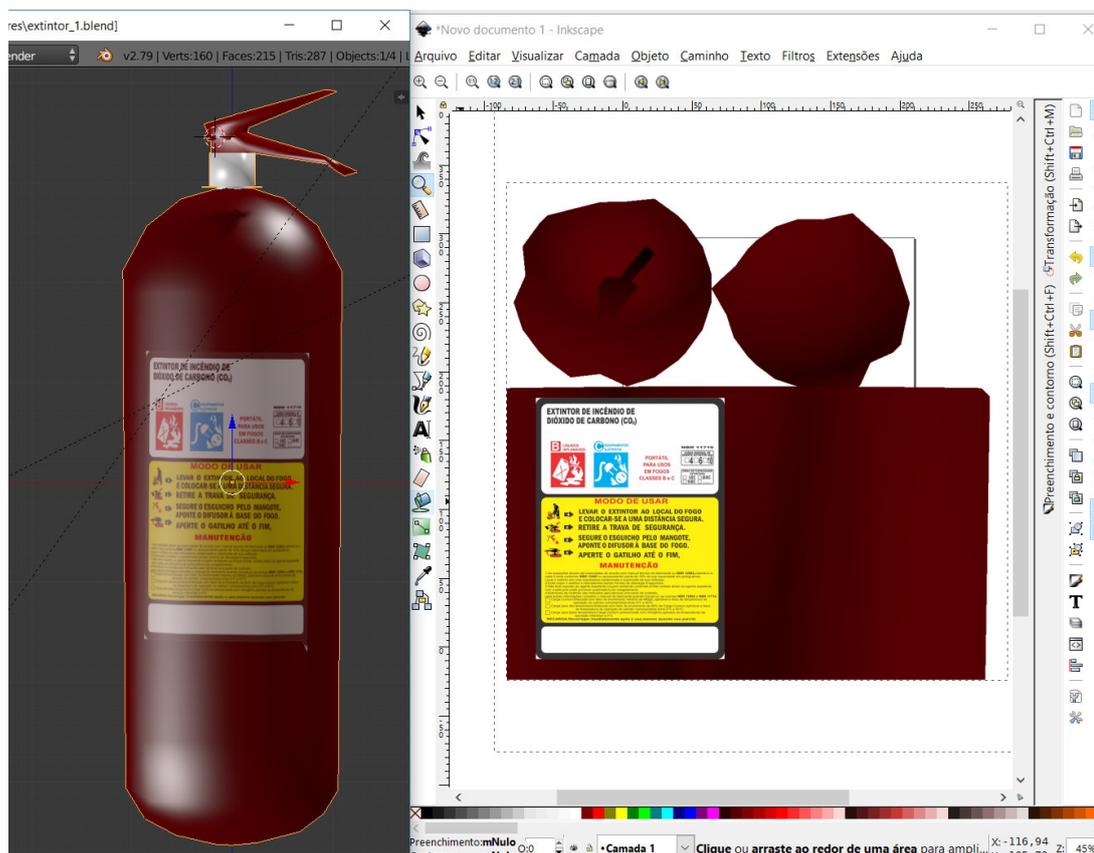


Figura 41 – Visualização do encaixe da imagem criada no krita.



Utilizou-se o software *inkscape* essencialmente para imprimir, sobre texturas com cores básicas, textos e logos, como feito no vidro da caixa de incêndio. Também foi utilizado para prototipagem e criação de *layouts* de apresentação de conteúdo dentro da aplicação como o plano de fundo do elemento de informação. A interface e exemplo de produção podem ser vistos na Figura 42

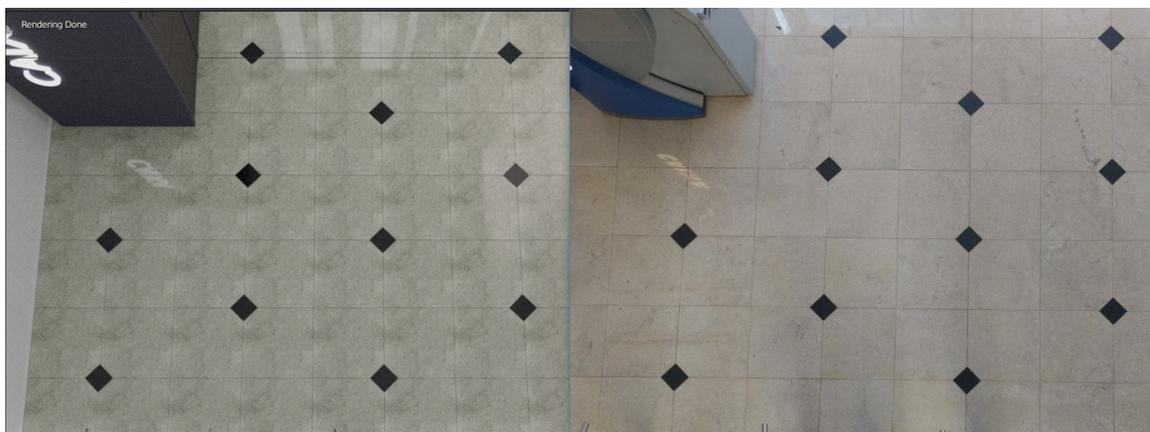
Figura 42 – Exemplo de uso do *Inkscape* para aplicação de detalhes em texturas.



As texturas para as lajotas da maior parte do prédio foram criadas à partir de fotos utilizando a ferramenta de clonagem do *Gimp*, que copia parte de uma imagem para outra posição da mesma imagem, em conjunto com o *Awesome Bump*. No *Gimp*, transformou-se uma foto não encaixável em um padrão encaixável e no *Awesome Bump* gerou-se as texturas sobre o padrão construído.

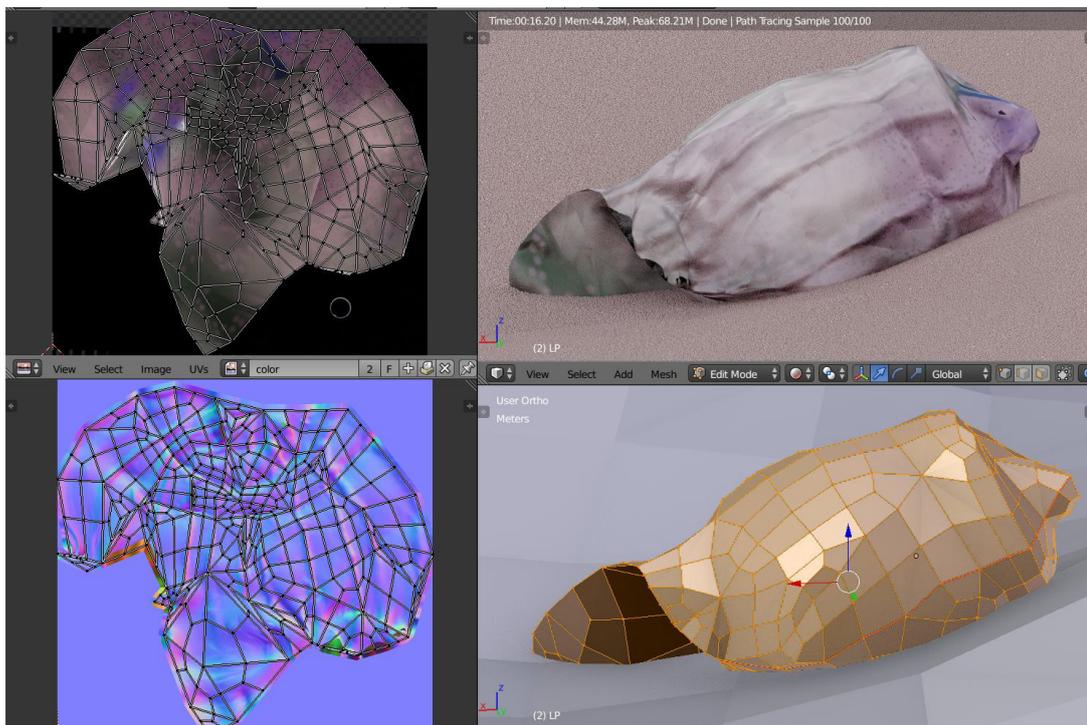
O padrão do piso da entrada norte foi totalmente recriado dentro do *Blender* para melhor aproximar o posicionamento dos elementos e a textura do material. Para recriá-lo foram criados dois modelos básicos que, repetidos com o modificador "vetor" e combinados por sobreposição, formam o padrão visto na Figura 43

Figura 43 – Renderização do piso da entrada norte à esquerda e foto da entrada norte à direita



As texturas são utilizadas para determinar propriedades como cor, índice de transparência em cada parte do objeto utilizando um mapa UV. Como resultado, tem-se um objeto com diferentes propriedades de material utilizando apenas um material como visto na Figura 44.

Figura 44 – Imagem de exemplo da propriedade cor e normal sendo determinadas por texturas mapeadas por mapas uv.

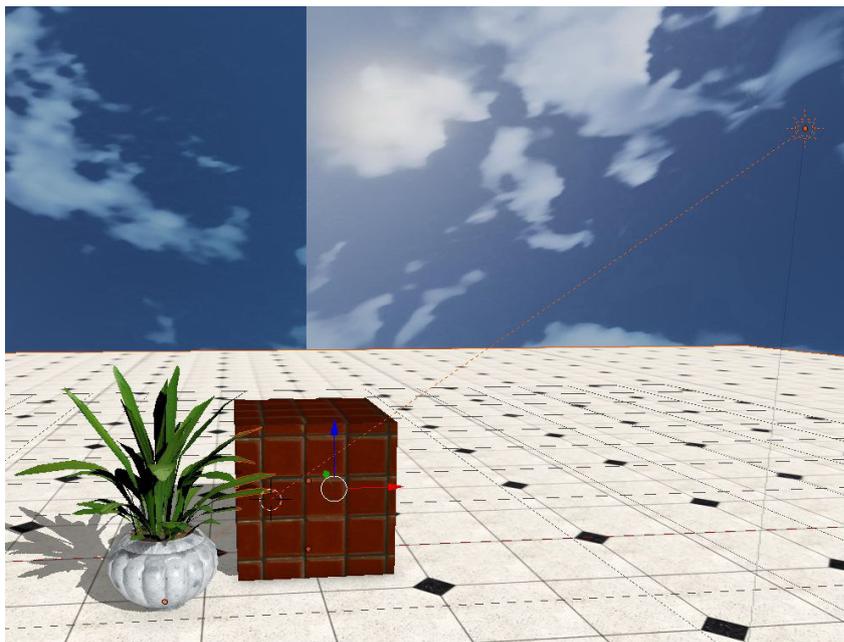


3.5 ILUMINAÇÃO E RENDERIZAÇÃO

Um dos aspectos mais importantes quando fala-se de realismo em gráficos feitos por computador é a iluminação. “Até mesmo uma iluminação muito simples pode fazer gráficos 3D parecerem mais realistas.”(ECK, 2018, p.273, traduzido pelo autor)

Para preparar os modelos e texturas foi criado um “estúdio virtual”, visto na Figura 45. A iluminação foi implementada utilizando-se uma lâmpada que funciona como o Sol e uma imagem projetada ao fundo que determina a cor ambiente em cada direção. Os modelos foram validados com shaders em GLSL¹¹, utilizado na janela de exibição do *Blender*. Dos aspectos de iluminação criados no *Blender* foram exportados para o *Unity*, apenas a direção do sol e a imagem de fundo.

Figura 45 – Imagem do Estúdio virtual para criação de objetos com iluminação básica.



Na Unity a iluminação indireta foi montada à partir de mapas de luz (*lightmaps*) de iluminação global. Estes mapas são texturas criados usando mapas UV dos modelos e o cálculo da iluminação indireta em cada ponto do mapa. Para a iluminação direta foi utilizado-se uma luz, representando o sol, utilizando direção aproximada à da presente na foto de referência apresentada na seção seguinte.

As informações de **oclusão ambiente**¹² também foram guardadas em imagens. O procedimento de guardar essas informações em mapas é necessário para que elas não sejam calculadas em tempo real e, assim, o processamento exigido pela aplicação seja muito menor. A desvantagem ao se utilizar esse procedimento é que a qualidade da imagem final é inferior para tamanhos médios de mapas e ao aplicar isso para todo o ambiente, e objetos, esses se tornam estáticos, pois sua iluminação, sombras, e reflexos não se movem ao mover-se os objetos.

No passo seguinte determinou-se o caminho de renderização utilizado pela aplicação. A Unity possui vários Caminhos de renderização (*rendering paths*). “Caminhos de renderização diferentes têm características de desempenho distintas, que afetam, na maior parte, luzes e sombras.” (UNITY TECHNOLOGIES, 2018f, traduzido pelo autor). Para escolher um desses caminhos é necessário verificar a plataforma de execução.

¹¹ “A linguagem de sombreado do OpenGL (GLSL) é a principal linguagem de sombreado para OpenGL.” (OPENGL. . . , 2017, p1, traduzido pelo autor)

¹² Na Unity Technologies (2018a), oclusão ambiente escurece dobras, buracos, intersecções e superfícies quando esses estão próximos uns dos outros. Esse efeito acontece na realidade, pois a luz ambiente é, em grande parte ou totalmente, bloqueada por essas estruturas.

Os Caminhos de renderização para a UnityUnity Technologies (2018f) pode ser vistos no Quadro 4.

Quadro 4 – Quadro com os Caminhos de Renderização e Descrições

Caminho de Renderização	Descrição
Sombreamento Postergado	Têm a melhor fidelidade de luz e sombra e é mais indicado se há muitas luzes de tempo real. Necessita de um certo nível de suporte de hardware.
Renderização Progressiva	É o caminho tradicional. Suporta todas as características típicas do Unity(mapas normais, luzes por pixel, sombras etc). Entretanto, nas configurações padrão, somente um pequeno número de das luzes mais brilhantes são renderizadas no modo de luz por pixel. O resto das luzes são calculadas por vértices do objeto ou por objeto.
Iluminação Posterior Legada	É similar ao Sombreamento Postergado. Apenas utiliza uma técnica diferente com diferentes comprometimentos. Ele não suporta o sombreamento padrão da Unity baseado em física.
Iluminação por Vértice Legada	Menor fidelidade de iluminação e sem suporte a sombras em tempo real. É um subconjunto da Renderização progressiva.

Fonte: Documentação da Unity Technologies (2018f, traduzido pelo autor)

Nesse trabalho o tipo de caminho utilizado foi a renderização progressiva, pois é a que tem melhor desempenho, possibilita o uso de MSAA (Multi-sample anti-aliasing) por hardware, lida com a maioria das características como transparência além de grande variedade de shaders customizados.

Na renderização progressiva cada objeto é renderizado em um passo para cada luz que o afeta. As luzes mais brilhantes são renderizadas no modo de iluminação por pixel completo para então até 4 pontos de luz serem calculados por vértice. As outras luzes são calculadas como “harmônicos esféricos”¹³.(UNITY TECHNOLOGIES, 2018c, traduzido pelo autor) .

3.5.1 Luz Real e Virtual

Para medir a fidelidade da iluminação, em relação à intensidade e direção, foi utilizada como referência a foto vista na Figura 46. O resultado pode ser visto na Figura 47

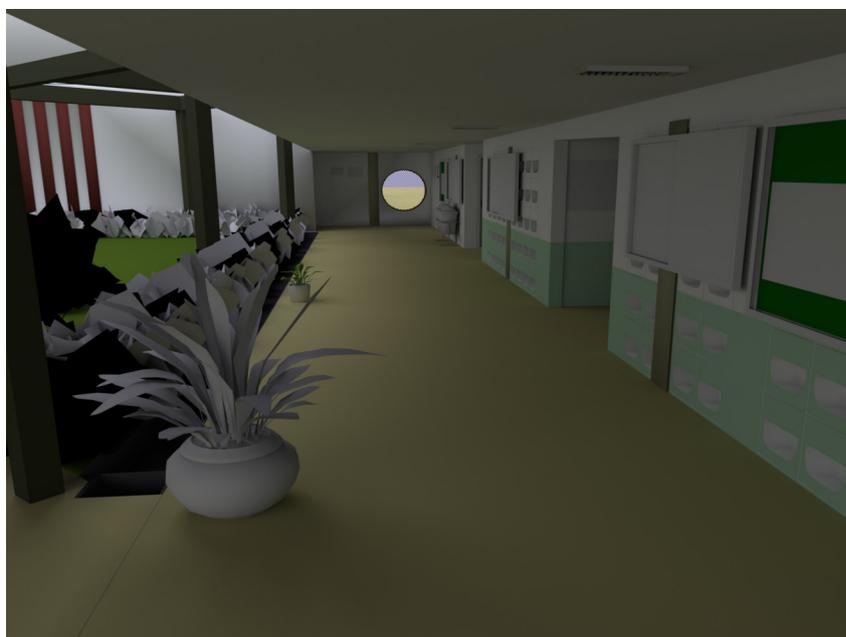
¹³ "Harmônicos esféricos(SH) representam uma função ou indicação sobre direções, e são geralmente utilizados em gráficos de computador para avaliar luz suave."(UNITY TECHNOLOGIES, 2018g, traduzido pelo autor) O Unity os utiliza para *lightprobes* e luz ambiente.

Figura 46 – Foto de referência de iluminação



Fonte: O autor(2018)

Figura 47 – Imagem renderizada da iluminação resultante

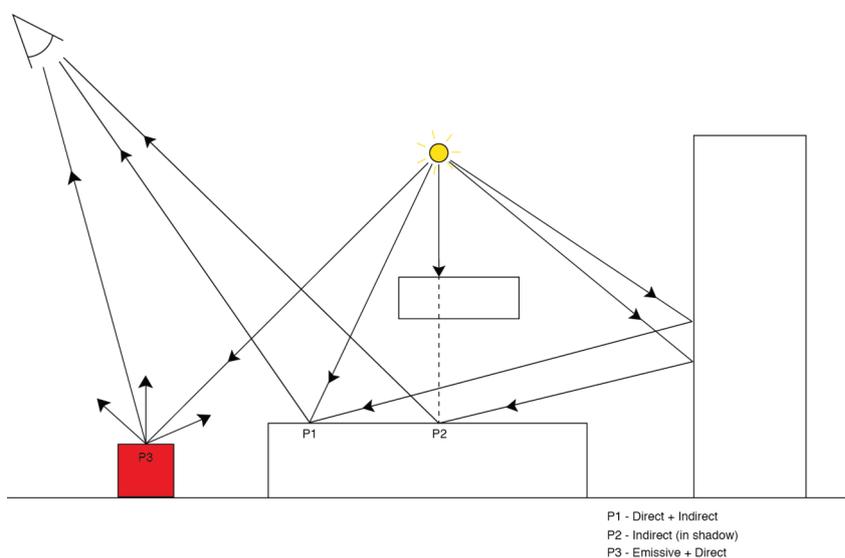


Fonte: O autor(2018)

A luz visível é uma forma de energia radiante, composta de fótons que irradiam-se pelo espaço e são refletidos ou absorvidos por superfícies de vários tipos. Para simular a luz em uma aplicação tridimensional é necessário conhecer como ela se comporta e fazer modelos computacionais sobre esses dados.

A interação da luz no ambiente 3D do Unity é calculada a partir da modelagem dos padrões na Figura 48.

Figura 48 – Imagem demonstrando o esquema da cálculo da luz na *Unity*



Fonte: Manual da Unity Technologies (2018d)

Uma das técnicas utilizada para uma iluminação mais próxima da real é o traçado de raios (*ray tracing*). Nessa técnica, raios são disparados de cada pixel do sensor de uma câmera virtual e traçam seu caminho até uma fonte de luz. Essa técnica demanda muito processamento e não é adequada para um game comum, que almeja desenhar 30 quadros por segundo e mais inadequada ainda para RV onde mais quadros são necessários para uma melhor experiência. Encaixando-se mais nesses requisitos, existe a técnica de rasterização.

De acordo com Computer Hope (2017), rasterizar é o processo de converter uma imagem vetorial(matematicamente definida) para uma imagem composta por pixels discretos. Em outras palavras, tornar um objeto geométrico da cena em pixels para exibição ou impressão.

Scratchpixel (2018) define rasterização como um método para resolver o problema da visibilidade¹⁴. Neste método triângulos são projetados na tela através de projeção por perspectiva. Depois uma técnica é utilizada para processar todos os pixels contidos no espaço delimitado por estes triângulos bidimensionais. As duas partes do processo são relativamente simples para implementar,executar e os resultados são previsíveis. Por isso, a rasterização é a técnica escolhida para gerar imagens de objetos 3D em tempo real por GPUs.

¹⁴ Visibilidade consiste em conseguir determinar que partes dos objetos 3D são observáveis pela câmera.

Para a aplicação foi utilizada rasterização com iluminação global impressa em textura (*baked*), para luz indireta. As sombras indiretas foram implementadas por mapeamento de oclusão. Já para iluminação direta utilizou-se luz direta e mapas de sombra impressa em texturas.

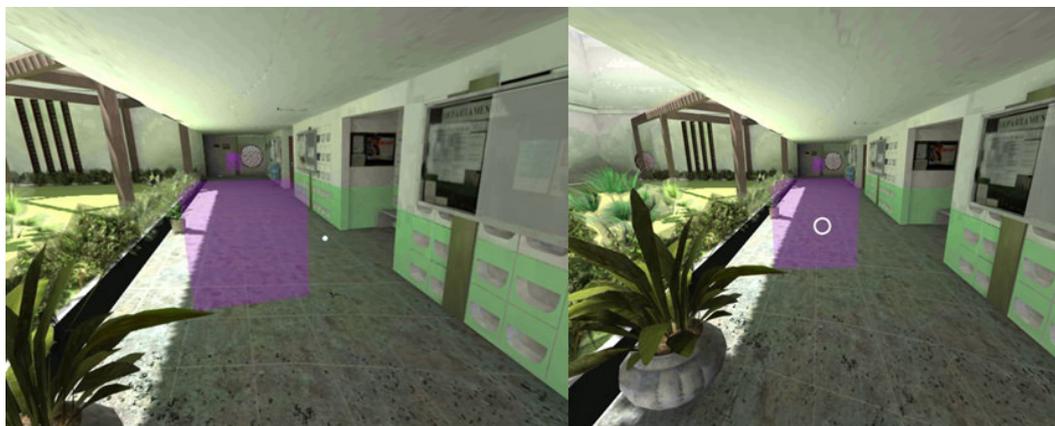
3.6 INTERAÇÃO

Para navegar no ambiente, utilizou-se o rastreamento de movimento de cabeça e um conjunto de cubos espalhados pelo espaço. Utilizou-se o rastreamento de movimento para determinar a direção observada pela câmera e cada ponto para habilitar o movimento. Segundo a Unity Technologies (2018h), uma aplicação em RV precisa ter interações simples, como por clique, poucos elementos bidimensionais para manter a sensação de imersão. Além disso precisa que os elementos de interação estejam inseridos no ambiente, como um conjunto de cubos.

O movimento da cabeça é rastreado em 360° utilizando o sensor giroscópio do aparelho móvel. Um retículo, círculo vazado inicialmente representado por um ponto, é acoplado ao centro da câmera. Este acompanha os movimentos da cabeça e auxilia na interação ao reagir, transformando-se no círculo vazado para indicar que é possível interagir com o objeto sobre o qual ele se encontra.

Cada cubo lilás, ao ser ativado por clique, faz a câmera mover-se em sua direção. Ao apontar para o cubo lilás o retículo se abre, como na Figura 49 e ao pressionar o botão de ação o jogador se move em direção do cubo. Com cubos espalhados por todo o ambiente é possível navegar e observar os pontos de interesse com direcionamento. Além do controle, esse esquema de navegação dispensa colisões nas paredes e solo, além de manter a performance da aplicação.

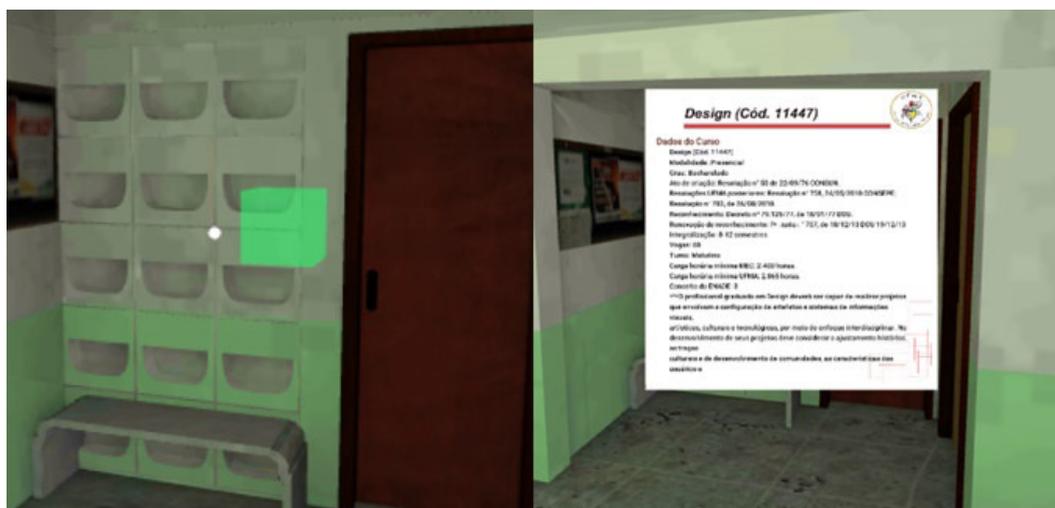
Figura 49 – Captura de tela demonstrando a ativação da mira para o cubo lilás.



A segunda forma de interação é o acesso ao conteúdo sobre os cursos. Esta é possível a partir de cubos verdes dispostos no espaço. Ao colocar o retículo sobre um destes cubos, uma

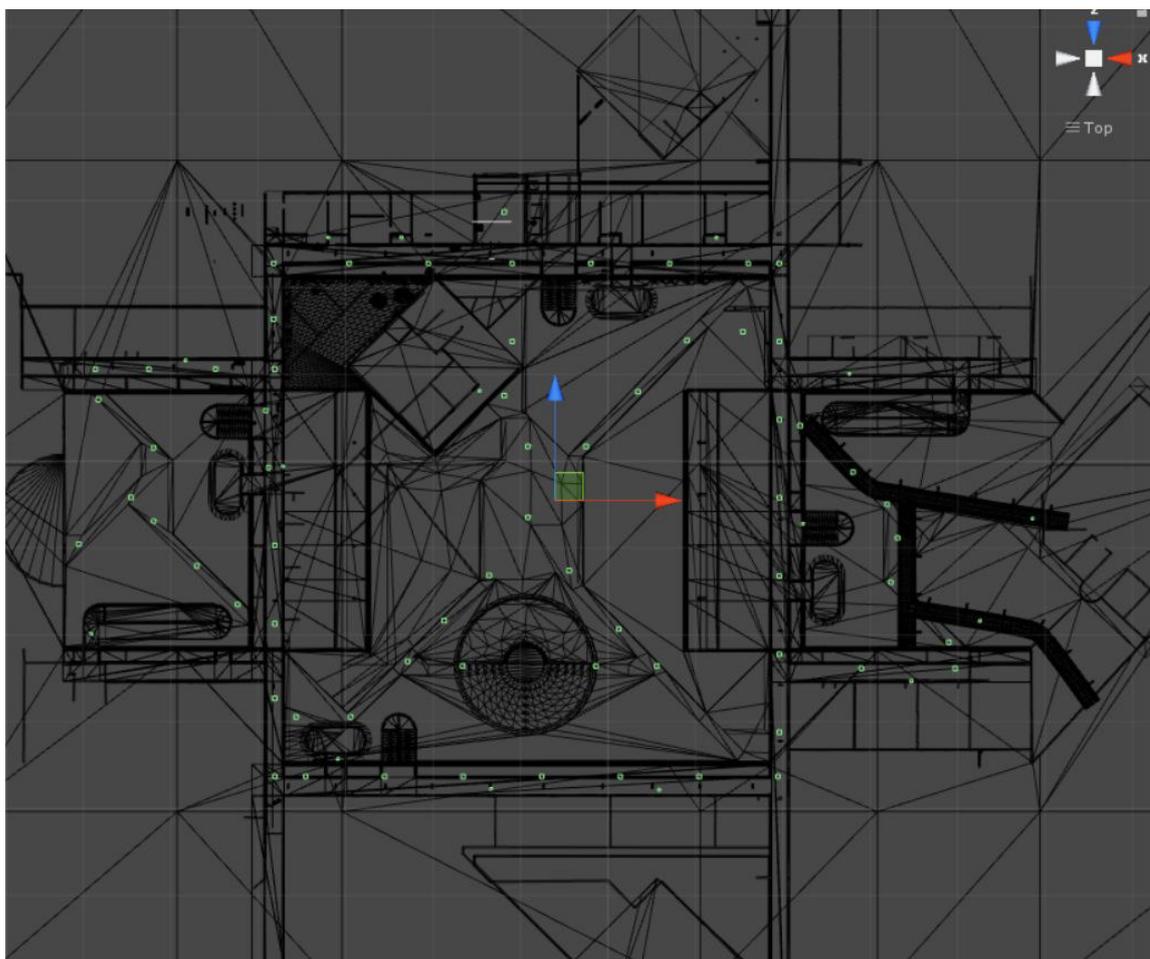
página surge e apresenta a informação sobre um dos cursos. Quando esta página está visível, o clique aciona a rolagem e permite a leitura do restante do conteúdo. A ativação da tela é retratada na Figura 50.

Figura 50 – Capturas de tela combinadas mostrando a sequencia de ativação da tela de informação.



A distribuição dos cubos de navegação juntamente com os de informação podem ser observados na Figura 51. Os cubos de navegação e informação foram dispostos manualmente, de maneira que a navegação possa ser realizada no maior número de direções, sem perpassar paredes e informação possa ser encontrada em todas as direções, estimulando a exploração.

Figura 51 – Captura de tela demonstrando a distribuição dos cubos verdes.



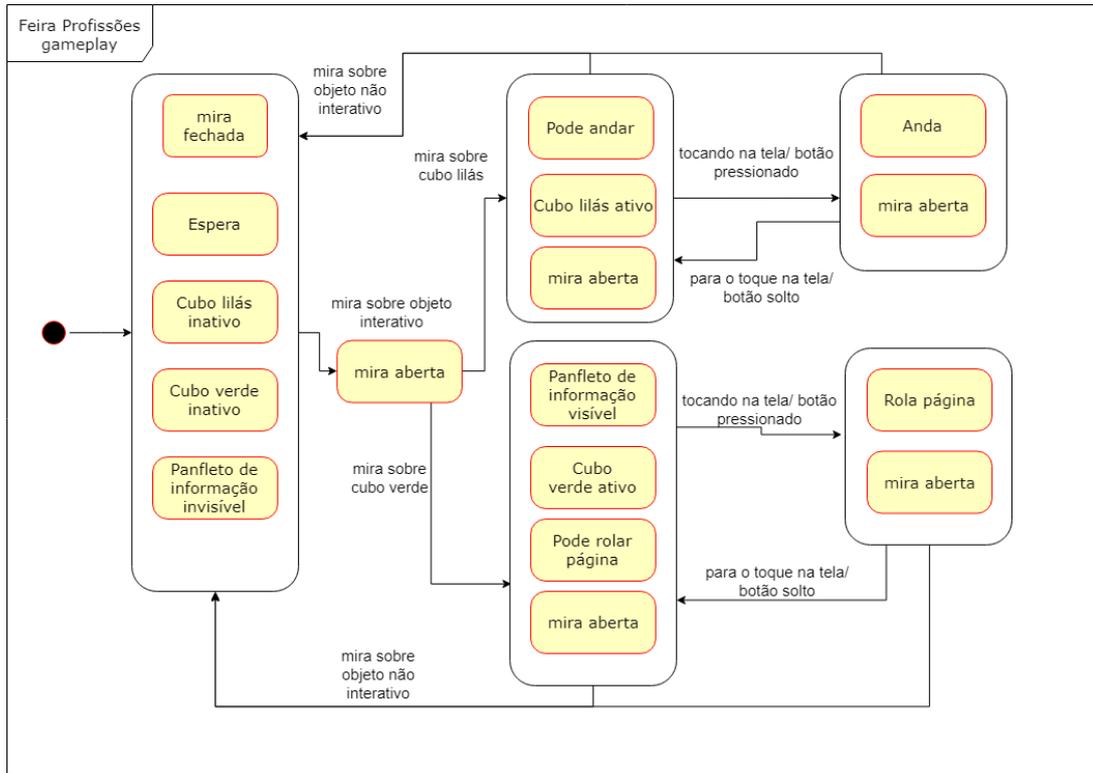
Fonte:O autor(2018)

Cada cubo tem atrelado a si um código que determina qual curso ele deverá apresentar na página de apresentação. Esta página recebe o código do curso que deve ser apresentado, lê as informações deste curso de uma lista que, para cada curso, contém o nome e a descrição. As informações sobre os cursos ficam gravadas em um arquivo de texto à partir do qual são extraídas e organizadas.

A estrutura do arquivo de cursos apresenta o nome do curso na primeira linha seguido de sua descrição nas linhas subsequentes. O final das informações sobre um curso é determinado por uma linha em branco. A lista de cursos e informações foram obtidos online nos sites da Universidade Federal do Maranhão (2018). Também, dos sites online da UFMA (2018), foi retirada o PPP (projeto político pedagógico) para o curso de engenharia ambiental e sanitária.

Na Figura 52 é possível visualizar a interatividade e os estados possíveis para a aplicação.

Figura 52 – Diagrama de estados que ilustra a jogabilidade para a aplicação.



Fonte: O autor (2018)

Inicialmente, a interatividade da aplicação seria possível apenas por toque na tela o que seria compatível com a segunda versão do Cardboard. Porém, para aumentar o número de visores compatíveis, o suporte a um controle foi adicionado. O suporte é limitado apenas a um botão que simula o toque na tela, suficiente para a experiência.

4 EXPERIMENTOS

4.1 DISPOSITIVOS UTILIZADOS

O desenvolvimento da aplicação aconteceu em duas máquinas e os testes em apenas um celular. As especificações de todas as máquinas podem ser verificadas no Quadro 5.

Quadro 5 – Quadro com a descrição do hardware utilizado

	Máquina 1	Máquina 2	Celular
Processador	i5 3337U 1.80GHz base/ 2.70GHz turbo	i7 7700HQ 2.80GHz base/ 3.80GHz turbo	snapdragon 617 1.5GHz
Placa de Vídeo	GT730M 2gb	GTX1060 6gb	Adreno 405
Memoria RAM	6gb DDR3 1600MHz	8gb DDR4 2400MHz	2gb LPDDR3 933MHz
Armazenamento	1 terabyte	2 terabytes	32 interno /64 cartão micro SD
Sistema Operacional	Windows 10	Windows 10	Android 7.0

Como complemento à lista de hardware, o óculos de realidade virtual 3D V-BOX da marca MyMax foi utilizado. As especificações são:

- Suporte a celulares com tamanho entre 3,5 e 6 polegadas;
- Campo de visão de 85 a 95 graus.

Cada máquina foi utilizada em um ou mais estágios da produção da aplicação. A máquina 1 foi utilizada para modelar e aplicar texturas a grande parte dos modelos. Também foi nela que o projeto da Unity foi criado. Já a máquina 2 foi utilizada para terminar a modelagem e a otimização dos modelos. Nela também foram criados os mapas de iluminação e todos os *bakes* foram executados, pois demandam muito do processador no caso da *Unity*. Por fim, o celular foi o local definitivo execução e testes.

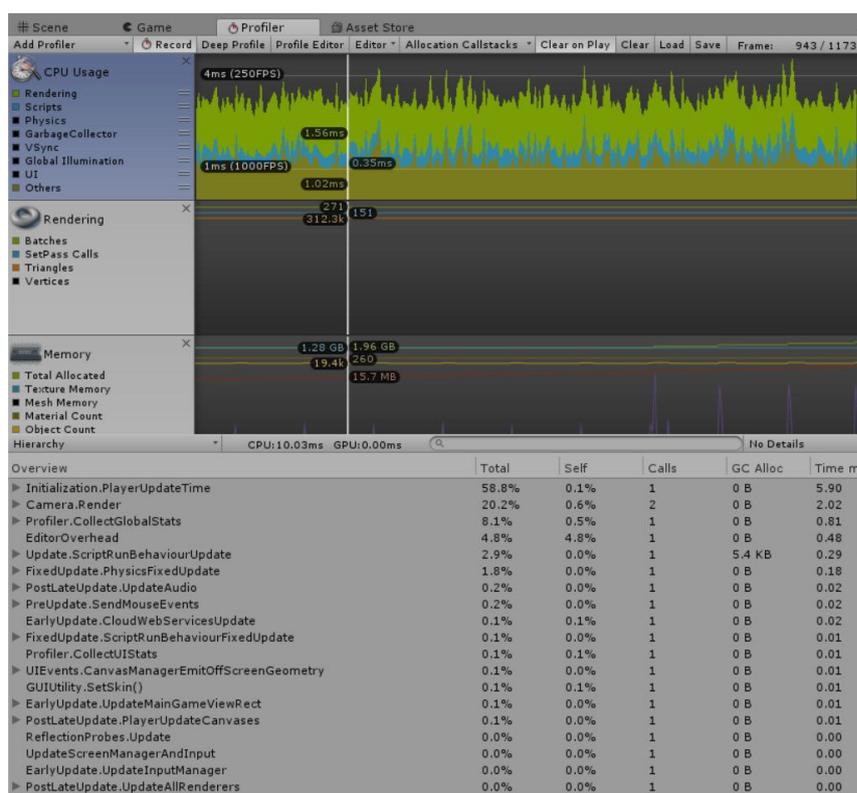
Os testes de aplicação foram realizados em duas etapas. A primeira etapa foi executada somente dentro do *Unity* e a segunda num conjunto *Unity* para acompanhamento de estatísticas e o Smartphone para visualização e execução.

A primeira etapa considerou o funcionamento somente dentro da *Unity*. Nessa etapa os scripts de comportamento, os materiais e áudio foram testados.

A segunda etapa considerou o comportamento no smartphone com o óculos 3D. Nessa etapa foram observadas diferenças de projeção das informações na tela, foi analisado o desempenho

utilizando o *profiler*, a ferramenta de análise de desempenho, do *Unity*, que pode ser visto na Figura 53 e, a partir disso, foram feitos ajustes em resoluções de textura e qualidade de shadders até alcançar trinta quadros por segundo. Levando-se em conta que o *profiler* usa recursos, que acarretam em perda de desempenho e o dispositivo tem desempenho intermediário, novos aparelhos conseguirão um desempenho igual ou superior apresentando especificações aproximadas ou superiores.

Figura 53 – Captura de tela do *profiler* da *unity*



Alguns aspectos de otimização não puderam ser utilizados no dispositivo móvel em questão por falta de suporte. Como exemplo, a tecnologia *Single-Pass Stereo rendering* (Renderização Estéreo em um Único Passo) que basicamente renderiza em apenas um passo a imagem para ambos os olhos. Ao fazer isso, o tempo de processamento é reduzido e mais quadros são gerados por segundo.

4.2 AVALIAÇÃO DA APLICAÇÃO

A aplicação foi utilizada por 21 estudantes do ensino médio. Todos utilizaram o mesmo hardware e versão do software. Cada sessão de uso durou uma média de 5 minutos. A avaliação da aplicação ocorreu em duas etapas.

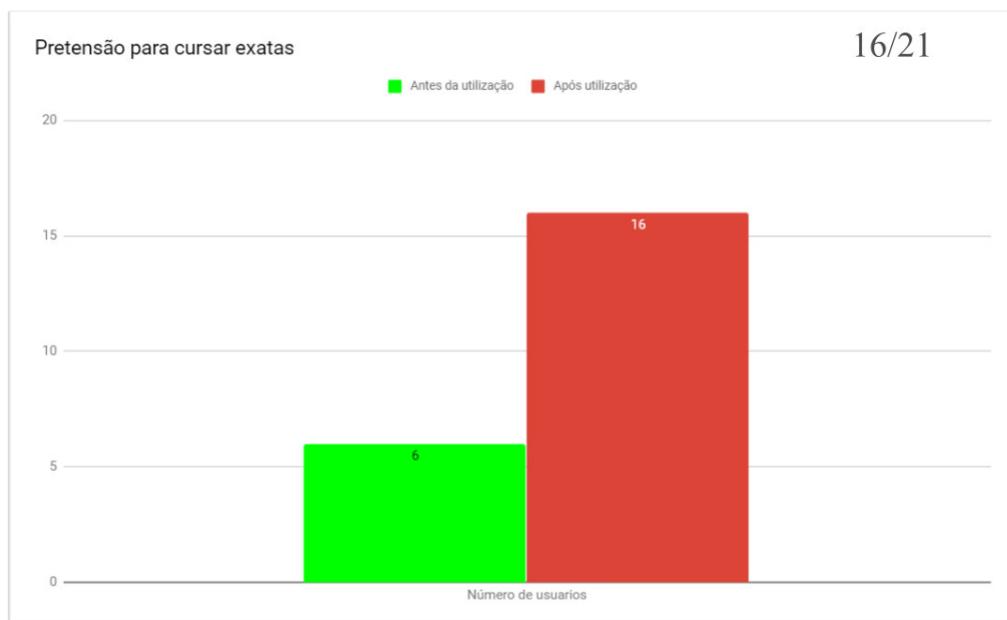
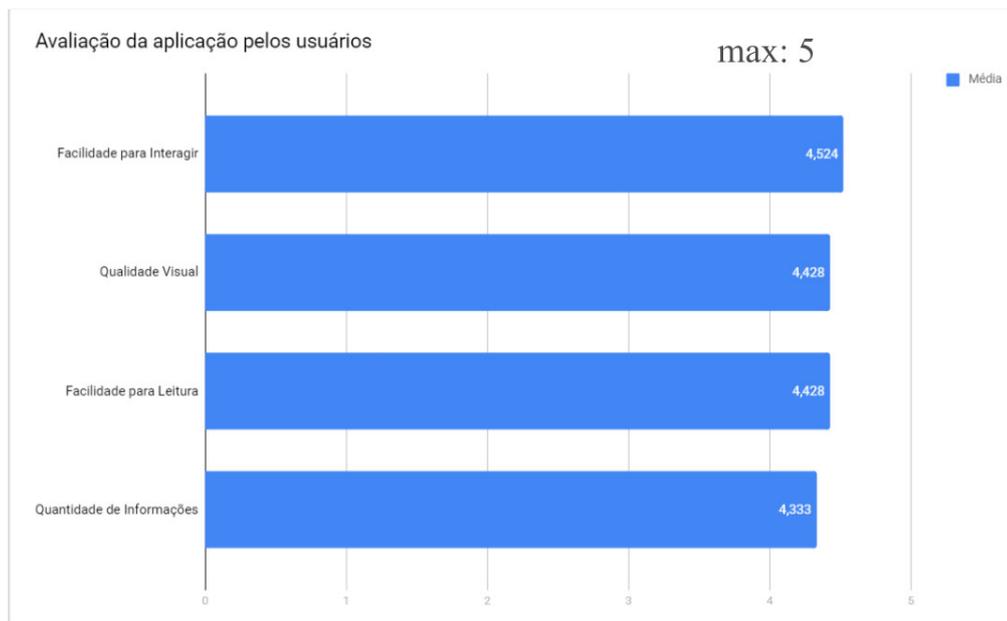
Na primeira etapa, aplicou-se questionário com itens sobre a escolha atual de área do curso do usuário antes e após a utilização da aplicação, assim como este avalia aspectos técnicos como:

- Facilidade para Interação
- Qualidade Visual
- Facilidade para Leitura
- Quantidade de informações

O questionário encerra-se com um quesito que pede para relacionar conceitos de áreas como matemática e física a aspectos utilizados para o funcionamento da aplicação como matrizes e análise de movimento. O questionário apresentou os resultados presentes na Figura 54.

Como pode ser verificado, houve impacto em 47,61% dos participantes, excluindo-se os seis que inicialmente cursariam algum curso na área de exatas, suficiente para aumentar a chances de optarem por um curso na área de exatas. A aplicação apresentou nota acima de 4 para os aspectos técnicos e o nível de compreensão dos conceitos para a capacidade de relacioná-los à uma disciplina apresentou uma taxa de 71,43% de sucesso.

Figura 54 – Resultados do questionário



Acertos ao relacionar conhecimentos necessários ao funcionamento da aplicação e suas respectivas disciplinas	60/84
Aumento do interesse nas áreas exatas	17/21
Aprendeu algo novo	19/21

5 CONCLUSÃO

O trabalho apresentou os diversos aspectos necessários para o desenvolvimento de uma aplicação de realidade virtual com base em modelos tridimensionais ao mesmo tempo que apresenta fluxos alternativos e as justificativas para os métodos escolhidos. Também apresenta uma pequena pesquisa sobre os impactos de uma aplicação dessa natureza na opinião de jovens sobre as áreas exatas do conhecimento.

A abordagem proposta utiliza majoritariamente os softwares Blender, Unity e Google VR SDK para recriar os aspectos mais substanciais do prédio do CCET, simular a presença nesse ambiente e proporcionar, ao usuário, interatividade.

O fluxo de trabalho escolhido mostrou-se custoso e inadequado para produções grandes com times pequenos e prazos definidos. Neste caso, é preferível utilizar-se de modelos já disponíveis com médio nível de detalhe, fotogrametria(escaneamento 3D de objetos) ou simples projeção de imagens. Entretanto, o fluxo é ideal para compreender a criação de objetos e materiais, assim como oferece maior capacidade de adaptação, pois a iluminação e outros aspectos podem ser modificados e retrabalhados para vários objetivos sem que os objetos não se encaixem por apresentarem iluminação fixa, como acontece com a projeção simples de imagens.

O desempenho, em torno de 30 quadros por segundo, apesar de abaixo do recomendado para esse tipo de aplicação, não causou desconforto para o usuário nos intervalos observados.

Em termos de avaliação, apesar de uma boa receptividade, bons índices de melhorias em impressões sobre a tecnologia e as disciplinas "exatas", entende-se que um maior alcance é necessário para resultados mais relevantes.

Para trabalhos futuros é possível melhorar a aplicação em termos de renderização, modelos e texturas, ampliar o estudo no âmbito educacional com minigames educativos sobre as diversas disciplinas "exatas" e elevar o nível de interatividade ao adicionar suporte para locomoção e elevação por detecção de movimento. É possível ampliar o alcance do estudo sobre o impacto da aplicação ao portá-la para a web de maneira que possa ser visualizada em um navegador, além de disponibilizá-la na loja do Google.

REFERÊNCIAS

- ARAUJO, R. B. d.; KIRNER, C. **Especificacao e analise de um sistema distribuido de realidade virtual**. Dissertação (Mestrado) — Universidade de São Paulo, 1996.
- ARCHITECTURAL CGI. **WHAT IS A 3D MODELING? THINGS YOU'VE GOT TO KNOW NOWADAYS**. 2018. Disponível em: <<https://archicgi.com/3d-modeling-things-youve-got-know/>>. Acesso em: 5.11.2018.
- AUKSTAKALNIS; BLATNER; STEVE; DAVID. Silicon mirage: The art and science of virtual reality. 01 1992.
- BLENDER DOCUMENTATION TEAM. **The Blender 2.79 Manual**. [S.l.], 2017. Disponível em: <https://docs.blender.org/manual/en/dev/render/blender_render/materials/introduction.html>. Acesso em: 15.5.2018.
- BLENDER FOUNDATION. **The software**. 2018. Disponível em: <<https://www.blender.org/about/>>. Acesso em: 15.5.2018.
- COMPUTER HOPE. **Rasterize**. 2017. Disponível em: <<https://www.computerhope.com/jargon/r/rasterize.htm>>. Acesso em: 10.11.2018.
- DAM, A. van. Post-wimp user interfaces. **Commun. ACM**, ACM, New York, NY, USA, v. 40, n. 2, p. 63–67, feb 1997. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/253671.253708>>.
- DOUG ENGELBART INSTITUTE. **The Dawn of Interactive Computing**. 2018. Disponível em: <<http://www.doungengelbart.org/content/view/159/92/>>. Acesso em: 10.11.2018.
- DUFFY, B. **The Creative Pipeline for VR Developers and Artists: An Introduction**. 2018. Disponível em: <<https://software.intel.com/en-us/blogs/2018/01/08/introduction-to-the-creative-pipeline-for-vr-developers-and-artists>>. Acesso em: 10.11.2018.
- DUNN, F.; PARBERRY, I. **3D Math Primer for Graphics and Game Development**. [S.l.: s.n.], 2002.
- ECK, D. J. **Introduction to Computer Graphics**. [s.n.], 2018. Disponível em: <<http://math.hws.edu/graphicsbook/>>. Acesso em: 3.5.2018.
- ENCYCLOPEDIA.COM. **Interactive Systems**. 2018. Disponível em: <<https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/interactive-systems>>. Acesso em: 6.11.2018.
- FACEBOOK TECHNOLOGIES, LLC. **Audio**. Disponível em: <<https://developer.oculus.com/design/latest/concepts/bp-audio/>>. Acesso em: 10.11.2018.
- FACEBOOK TECHNOLOGIES, LLC. **General User Experience**. Disponível em: <<https://developer.oculus.com/design/latest/concepts/bp-generalux/>>. Acesso em: 10.11.2018.
- GOOGLE. **Feedback**. Disponível em: <<https://designguidelines.withgoogle.com/cardboard/interactive-patterns/feedback.html>>. Acesso em: 10.11.2018.
- GOOGLE. **Designing for Google Cardboard**. 2018. Disponível em: <<https://designguidelines.withgoogle.com/cardboard/>>. Acesso em: 10.11.2018.

GUEDES, G. **Interface Humano Computador: práticas pedagógicas para ambientes virtuais**. [S.l.]: Editora Gráfica EDUFPI, 2010. (Voices That Matter). ISBN 9788574632551.

HOCK-CHUAN, C. **3D Graphics with OpenGL Basic Theory**. 2012. Disponível em: <https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html>. Acesso em: 11.5.2018.

HORNBAEK, K.; OULASVIRTA, A. What is interaction? In: **Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2017. (CHI '17), p. 5040–5052. ISBN 978-1-4503-4655-9. Disponível em: <<http://doi.acm.org/10.1145/3025453.3025765>>.

HUNTER, A. Realism, expectations, and the fidelity contract in vr design. 2017. Disponível em: <<https://virtualrealitypop.com/realism-expectations-and-the-fidelity-contract-in-vr-design-3782485bd054>>. Acesso em: 1.5.2018.

JARNO, L. **OpenGL rendering pipeline**. [S.l.: s.n.], 2017.

LATTA, J. N.; OBERG, D. J. A conceptual virtual reality model. **IEEE Computer Graphics and Applications**, v. 14, n. 1, p. 23–29, Jan 1994. ISSN 0272-1716.

LOWOOD, H. E. Virtual reality. 2015. Acesso em: 11.3.2018.

MCHENRY, K.; BAJCSY, P. Technical Report, **An Overview of 3D Data Content, File Formats and Viewers**. 2008.

MEEHAN MICHAEL & RAZZAQUE, S. . W. M. . B. J. F. Effect of latency on presence in stressful virtual environments. In: **Virtual Reality Annual International Symposium**. [S.l.: s.n.], 2003. p. 141– 148.

MOTOROLA MOBILITY LLC. **Introduction to VR Best Practice**. Disponível em: <<https://unity3d.com/pt/learn/tutorials/topics/xr/introduction-vr-best-practice?playlist=22946>>. Acesso em: 10.11.2018.

MOTOROLA MOBILITY LLC. **Moto Actions**. Disponível em: <<https://support.motorola.com/in/en/solution/MS108582>>. Acesso em: 10.11.2018.

MYERS, B. A brief history of human computer interaction technology. **Interactions**, v. 5, 09 2001.

NETTO, A. V.; MACHADO, L. dos S.; OLIVEIRA, M. C. F. de. Realidade virtual - definições, dispositivos e aplicações. In: . [S.l.: s.n.], 2002.

NITECKI, Z. H. **Calculus in 3D Geometry, Vectors, and Multivariate Calculus**. [S.l.: s.n.], 2012.

OPENGL Shading Language. 2017. Disponível em: <https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language>. Acesso em: 13.5.2018.

PIMENTEL, K.; TEIXEIRA, K. Book; Book/Illustrated. **Virtual reality : through the new looking glass**. 2nd ed. ed. [S.l.]: New York : Intel/McGraw-Hill, 1994. Includes bibliographical references (p. 418-427) and index. ISBN 007050167X.

SAFFER, D. **Designing for Interaction: Creating Innovative Applications and Devices**. Pearson Education, 2009. (Voices That Matter). ISBN 9780321679437. Disponível em: <<https://books.google.com.br/books?id=Dd3Hcs9jeoUC>>.

SANTOS, R. de A. **CAVE MÓVEL DE BAIXO CUSTO PARA AUXILIAR NA EDUCAÇÃO**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2015.

SCRATCHPIXEL. **Rasterization: a Practical Implementation**. 2018. Disponível em: <<https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation>>. Acesso em: 10.11.2018.

STEREOGRAPHICS CORPORATION. Book; Book/Illustrated. **STEREO RAPHICS G DEVELOPERS' HANDBOOK**. 2nd ed. ed. [S.l.], 1997.

UNITY TECHNOLOGIES. **Ambient Occlusion**. 2018. Disponível em: <<https://docs.unity3d.com/Manual/PostProcessing-AmbientOcclusion.html>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **Engines de jogo - como funcionam?** 2018. Disponível em: <<https://unity3d.com/pt/what-is-a-game-engine>>. Acesso em: 6.12.2018.

UNITY TECHNOLOGIES. **Forward Rendering Path Details**. 2018. Disponível em: <<https://docs.unity3d.com/Manual/RenderTech-ForwardRendering.html>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **Lighting: Technical information and terminology**. 2018. Disponível em: <<https://docs.unity3d.com/Manual/LightModes-TechnicalInformation.html>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **Optimisation-vr-unity**. 2018. Disponível em: <<https://unity3d.com/pt/learn/tutorials/topics/virtual-reality/optimisation-vr-unity>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **RenderingPaths**. 2018. Disponível em: <<https://docs.unity3d.com/Manual/RenderingPaths.html>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **SphericalHarmonicsL2**. 2018. Disponível em: <<https://docs.unity3d.com/ScriptReference/Rendering.SphericalHarmonicsL2.html>>. Acesso em: 15.5.2018.

UNITY TECHNOLOGIES. **User Interfaces for VR**. 2018. Disponível em: <<https://unity3d.com/pt/learn/tutorials/topics/virtual-reality/user-interfaces-vr>>. Acesso em: 11.5.2018.

UNIVERSIDADE FEDERAL DO MARANHÃO. **Centro de Ciências Exatas e Tecnologias - CCET**. 2018. Disponível em: <http://portais.ufma.br/PortalProReitoria/proen/paginas/pagina_estatica.jsf?id=22>. Acesso em: 1.3.2018.

UNIVERSIDADE FEDERAL DO MARANHÃO. **Projeto Pedagógico do Curso de Engenharia Ambiental e Sanitária**. 2018. Disponível em: <https://sigaa.ufma.br/sigaa/public/curso/documentos.jsf?lc=pt_BR&id=17150698&idTipo=2>. Acesso em: 1.3.2018.

APÊNDICE A – QUESTIONÁRIO

Questionário Feira de Profissões 2018

1º Levando em consideração o nível superior, qual a sua área de interesse?

Humanas Exatas Sociais Biológicas

2º Em qual curso superior você pretende ingressar?

-----À partir daqui, responder apenas após utilizar a aplicação-----

3º Numa escala de 1 a 5(sendo 5 a nota máxima), como você julga a aplicação nos seguintes aspectos(comente, se desejar):

1- Facilidade de interação

1 <input type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input type="checkbox"/>	5 <input type="checkbox"/>
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------

2- Qualidade visual

1 <input type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input type="checkbox"/>	5 <input type="checkbox"/>
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------

3- Facilidade de leitura

1 <input type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input type="checkbox"/>	5 <input type="checkbox"/>
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------

4- Quantidade informações

1 <input type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input type="checkbox"/>	5 <input type="checkbox"/>
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------

4º Após passar por esta experiência marque sim ou não para os seguintes aspectos:

1- Houve aumento de sua curiosidade em relação aos cursos da área de exatas?

<input type="checkbox"/> sim	<input type="checkbox"/> não
------------------------------	------------------------------

2- Você aprendeu algo novo enquanto utilizando a aplicação?

<input type="checkbox"/> sim	<input type="checkbox"/> não
------------------------------	------------------------------

3- Após utilizar a aplicação a probabilidade de você ingressar em um curso de exatas aumentou?

<input type="checkbox"/> sim	<input type="checkbox"/> não
------------------------------	------------------------------

5º Relacione os conceitos e as disciplinas

<input type="checkbox"/> 1 Detecção de movimento da cabeça	<input type="checkbox"/> Matemática
<input type="checkbox"/> 2 Codificação de comportamento	<input type="checkbox"/> Física
<input type="checkbox"/> 3 Propriedades e estudo de novos materiais	<input type="checkbox"/> Química
<input type="checkbox"/> 4 Testes lógicos, matrizes	<input type="checkbox"/> Computação