



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE COMUNICAÇÃO E EXPRESSÃO  
DEPARTAMENTO DE EXPRESSÃO GRÁFICA  
BACHARELADO EM ANIMAÇÃO

Maria Carolina Lucena de Melo Souto

**“Sala de Espera” - Adaptação de pintura à óleo para interação com Realidade  
Virtual**

Florianópolis

2023

Maria Carolina Lucena de Melo Souto

**Sala de Espera - Adaptação de pintura à óleo para interação em Realidade  
Virtual**

Trabalho de Conclusão de Curso submetido ao curso de Bacharelado em Animação do Centro de Comunicação e Expressão da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharela em Animação.

Orientador: Prof. Gabriel de Souza Prim, Dr.

Florianópolis

2023

Souto, Maria Carolina Lucena de Melo  
"Sala de Espera" - Adaptação de Pintura à Óleo em uma cena virtual 3D para interação em Realidade Virtual / Maria Carolina Lucena de Melo Souto ; orientador, Gabriel de Souza Prim, 2023.  
41 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro de Comunicação e Expressão, Graduação em Animação, Florianópolis, 2023.

Inclui referências.

1. Animação. 2. Realidade Virtual. 3. Cena virtual. 4. Pintura. 5. Arte. I. Prim, Gabriel de Souza. II. Universidade Federal de Santa Catarina. Graduação em Animação. III. Título.

Maria Carolina Lucena de Melo Souto

**Sala de Espera - Adaptação de pintura à óleo em uma cena virtual 3D para interação em Realidade Virtual**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Animação e aprovado em sua forma final pelo Curso de Animação.

Local Florianópolis, 28 de Novembro de 2023.

---

Coordenação do Curso

**Banca examinadora**

---

Prof. Gabriel de Souza Prim Dr.

Orientador

Profa. Mônica Stein, Dra.

Instituição Universidade Federal de Florianópolis

Prof. Nicholas Bruggner Grassi, Dr.

Instituição Universidade Federal de Florianópolis

Florianópolis, 2023.

Dedico este trabalho à minha família e aos meus amigos, que me apoiaram ao longo desta jornada. Se cheguei até aqui, foi graças à presença deles em minha vida.

## **AGRADECIMENTOS**

Quero agradecer a todos que me apoiaram incondicionalmente durante essa jornada e na elaboração deste trabalho. Agradeço a minha família, em especial aos meus pais que desde pequena me incentivaram e fizeram de tudo em seu alcance para me ajudar a seguir em frente durante minha trajetória, apesar de ter sido difícil lidar com a distância, eles sempre se fizeram presentes. Agradeço aos meus amigos que tornaram esse período mais leve e divertido e me deram motivação para continuar. Agradeço à minha irmã, com quem eu sempre pude e posso contar para conversar e me escutar nos momentos mais difíceis. Agradeço ao meu professor e orientador Gabriel Prim que me guiou na condução deste projeto e foi uma referência intelectual e profissional, seu encorajamento e presença foram essenciais para que eu chegasse até o fim. Este trabalho de conclusão de curso é apenas um dos resultados de muitos aprendizados adquiridos ao longo desses anos de estudo.

“Outros viram o que é e perguntaram por quê.

Eu vi o que poderia ser e perguntei: por que não?”

Pablo Picasso

## RESUMO

O desenvolvimento dos jogos e processadores gráficos popularizou e expandiu a tecnologia da Realidade Virtual. A venda de óculos VR tornou possível simular experiências, ambientes e situações de qualquer tipo em diversas áreas, sejam pesquisas, simulações ou entretenimento. Uma dessas áreas é a arte, que sempre esteve presente como forma de expressão da criatividade humana e pode ser realizada em meios variados, sendo extremamente versátil para novas possibilidades de interação com a mesma. Este trabalho de conclusão de curso tem como objetivo simular a exploração de uma pintura a óleo por meio da sua recriação e adaptação em um ambiente tridimensional, utilizando tecnologias de Realidade Virtual para interagir com a obra. O projeto foi realizado por meio da seleção do quadro “Sala de Espera” de Rodrigo Yudi e da recriação do mesmo em uma cena virtual tridimensional por meio da modelagem de seus elementos, levando em consideração suas formas, cores e texturas para garantir fidelidade ao estilo do artista. O resultado é uma experiência interativa de exploração da pintura que proporciona diferentes interpretações da obra original.

**Palavras-chave:** Animação; 3D; Realidade Virtual; cena virtual; pintura.



## **ABSTRACT**

The development of games and graphic processors popularized and expanded Virtual Reality technology. The sale of VR glasses has made it possible to simulate experiences, environments and situations of any type in different areas, whether it is research, simulations or entertainment. One of these areas is art, which has always been present as a form of expression of human creativity and can be carried out in a variety of ways, being extremely versatile for new possibilities of interaction with it. This course completion work aims to simulate the exploration of an oil painting through its recreation and adaptation in a three-dimensional environment, using Virtual Reality technologies to interact with it. The project was made by selecting the painting "Waiting Room" by Rodrigo Yudi and recreating it in a three-dimensional virtual scene through the modeling of its elements, taking into account their shapes, colors and textures to ensure fidelity to the style of the artist. The result is an interactive experience of exploration of the painting that provides different interpretations of the original work.

**Keywords:** Animation; 3D; Virtual Reality; virtual scene; painting.

## LISTA DE FIGURAS

Figura 1 – Localização do vértice, aresta, polígono e face em um cubo 3D	20
Figura 2 – Demonstração de como a curvatura de um modelo é influenciada pela quantidade de polígonos em sua malha	20
Figura 3 – Modelo coerente tridimensional formado por malha poligonal de triângulos de um golfinho	21
Figura 4 – Modificador <i>unwrap</i> aplicado ao modelo 3D de um cubo	23
Figura 5 – Integração dos componentes de uma Realidade Virtual por meio de uma engine gráfica	24
Figura 6 – Interface da Unreal Engine	25
Figura 7 – Exemplo de Blueprint da Unreal Engine	26
Figura 8 – Pinturas “Quarta de Cinzas”, “Happy Hour” e “Guaraná, Chocolate em Pó e Caixas de Fósforo”	28
Figura 9 – “Sala de espera”, pintura óleo sobre tela	29
Figura 10 – Cronograma de execução do projeto	31
Figura 11 – Plano no 3Ds Max com a imagem da pintura	33
Figura 12 – Demonstração de como foi feita a correspondência entre modelo tridimensional e referência bidimensional	34
Figura 13 – Comparação de proporções entre espaço 2D e 3D	34
Figura 14 – Comparação entre o modelo da garrafa de café e a referência da pintura	34
Figura 15 – Comparação entre o modelo do ventilador e a referência da pintura	35
Figura 16 – Referência da torneira do bebedouro e do galão de água	36
Figura 17 – Referência da torneira do bebedouro e do galão de água	36
Figura 18 – Referência dos bancos de espera	36
Figura 19 – Início da modelagem do copo como apenas um cone	37
Figura 20 – Adição de loops ao longo da superfície vertical do copo	37
Figura 21 – Complementação do modelo com detalhes no topo e na base do copo	37
Figura 22 – Início da modelagem do suporte da mesa como apenas um cilindro	38
Figura 23 – Cilindro modificado, agora semelhante a um dos suportes da mesa da pintura	38
Figura 24 – Modelo final do suporte da mesa	39
Figura 25 – Fio em espiral do telefone e galhos da planta por spline	39

Figura 26 – Fio em espiral do telefone e galhos da planta por spline	39
Figura 27 – Fio que liga o telefone ao <i>plug</i> na parede	40
Figura 28 – Modelo do porta copos com dois suportes que o prendem à “parede”	40
Figura 29 – Modelos do teclado, mouse e CPU destacados	41
Figura 30 – Modelo da mesa da recepcionista criado por referências externas	41
Figura 31 – Modelo das cadeiras de espera e da cortina	42
Figura 32 – Modelo das cadeiras de espera e da cortina	42
Figura 33 – Modelos dos copos plásticos	42
Figura 34 – Resultado final da representação da pintura em um ambiente tridimensional	43
Figura 35 – Exemplo de Unwrap automático de um objeto	44
Figura 36 – Modelo 3D com cortes em loop na malha	44
Figura 37 – Unwrap do modelo feito a partir de cortes manuais na malha	45
Figura 38 – Imagem final do mapa de textura do modelo da porta	45
Figura 39 – Mapa de textura da porta no Photoshop	46
Figura 40 – Pincéis utilizados no processo de texturização	46
Figura 41 – Pintura da porta feita manualmente	47
Figura 42 – Modelo tridimensional da porta com textura de imagem como cor aplicada	47
Figura 43 – Representação das revistas e do sachê de açúcar na pintura original	48
Figura 44 – Representação das revistas e do sachê de açúcar na pintura original	48
Figura 45 – Modelo do sachê de açúcar das revistas com mapa de textura aplicado	48
Figura 46 – Modelo do sachê de açúcar das revistas com mapa de textura aplicado	48
Figura 47 – Modelo da cadeira da recepcionista e do mouse com mapa de textura aplicado	49
Figura 48 – Modelo da cadeira da recepcionista e do mouse com mapa de textura aplicado	49
Figura 49 – Cena virtual do quadro montada	49
Figura 50 – Modelos da cena virtual selecionados e sendo exportados	50
Figura 51 – Material do pôster do consultório	51
Figura 52 – Material da poltrona	52
Figura 53 – Material do computador	52

Figura 54 – Material da cortina	53
Figura 55 – Modelo da cortina em resposta à luz direta	53
Figura 56 – Modelo do computador com a textura de vídeo aplicada na tela	54
Figura 57 – Lightmass Portal na janela	55
Figura 58 – Spheres Reflection Capture da cena virtual	55
Figura 59 – Resultado final da cena iluminada	56
Figura 60 – Modelo do computador com a textura de vídeo aplicada na tela	56
Figura 61 – Blueprint do relógio	57
Figura 62 – Event Graph do componente “Inspecionar Objeto”	58
Figura 63 – Função “Setar Detecção” do “Inspecionar Objeto”	58
Figura 64 – Função “Agarrar” do “Inspecionar Objeto”	58
Figura 65 – Blueprint dos objetos. As configurações de física e de peso estão sublinhadas	59
Figura 66 – Demonstração de como funciona a função de agarrar um objeto à distância	60
Figura 67 – Demonstração de como funciona segurar um objeto	60
Figura 68 – Cápsulas projetadas a partir do modelo <i>low poly</i> da garrafa no modelo <i>high poly</i>	61
Figura 69 – Cápsulas projetadas a partir do modelo <i>low poly</i> da garrafa no modelo <i>high poly</i>	61
Figura 70 – Mapas de normais “ <i>object space</i> ” do modelo da garrafa de café	62
Figura 71 – Mapas de normais “ <i>object space</i> ” do modelo da garrafa de café	62
Figura 72 – Mapa de normal “ <i>object space</i> ” do modelo da garrafa pintada de forma estilizada	62
Figura 73 – Comparação entre a garrafa de café sem e com as normais estilizadas	63
Figura 74 – Comparação entre a garrafa de café sem e com as normais estilizadas	63
Figura 75 - Comparação entre parte da cena sem e com estilização de normais	63
Figura 76 - Comparação entre parte da cena sem e com estilização de normais	63

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>18</b>
<b>2.1</b>	<b>Realidade Virtual</b>	<b>18</b>
<b>2.2</b>	<b>Criação de mundos virtuais</b>	<b>19</b>
2.2.1	Modelagem 3D	19
2.2.2	Texturas e mapeamento 3D	21
2.2.3.	Programa de Realidade Virtual	23
2.2.4	Unreal Engine	24
2.2.5	Iluminação e Renderização	26
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>27</b>
<b>3.1</b>	<b>Pré-Produção</b>	<b>27</b>
3.1.1	Escolha do artista e da pintura	28
3.1.2	Definir o que será executado no projeto	30
<b>3.2</b>	<b>Produção</b>	<b>30</b>
3.2.1	Modelagem	31
3.2.2	Mapeamento	43
3.2.3	Integração da cena na Unreal Engine	50
3.2.4	Criação de materiais	51
3.2.5	Iluminação	54
3.2.6	Animação	56
3.2.7	Inspeção de objeto	57
3.2.8	Mapeamento das normais	60
<b>4</b>	<b>CONCLUSÃO</b>	<b>65</b>
	<b>REFERÊNCIAS</b>	<b>67</b>

## 1 INTRODUÇÃO

Os avanços tecnológicos modificaram a forma como interagimos com o mundo. Graças à evolução dos computadores e das placas de vídeo é possível representar imagens, ambientes e simulações com alta resolução e imersividade de forma totalmente digital (OLIVEIRA, 2016). Nesse contexto, a Realidade Virtual (RV) ressurgiu após anos de inviabilidade monetária e tecnológica, atualmente se encontrando em constante expansão devido à acessibilidade a ferramentas de desenvolvimento de jogos. O lançamento do Oculus Rift em 2012 no mercado tornou a tecnologia da RV acessível para o público e trouxe a possibilidade de criar jogos digitais e ser transportado para um mundo próprio, com regras próprias, sem precisar sair do lugar (GOBIA, MOZELLI, 2016).

A ascensão da Realidade Virtual possibilitou sua aplicação em diversas áreas, como pesquisa psicológica, reabilitação, marketing, saúde, educação e entretenimento (LONG LIN, CHEN, RUNGTAI LIN, 2020). Isso ocorre pois ela permite simular ambientes e situações, tanto reais quanto imaginários, com um alto grau de imersão, proporcionando novas possibilidades de interação e, conseqüentemente, de sensações com campos conhecidos. O cotidiano é expandido e experimentamos novas formas de enxergar e sentir além do habitual, e por se tratar de um cenário virtual, é possível simular qualquer ambiente interativo, desde um teste de direção até a experiência de visitar uma galáxia sideral distante (NETTO; MACHADO; OLIVEIRA, 2006).

Nesse sentido, um dos diversos campos que há grande potencial de aplicação da Realidade Virtual é a arte. Representações da realidade ou da imaginação sempre fizeram parte da história do ser humano, desde desenhos primitivos, figuras, pinturas e teatro até o cinema, jogos e ilusionismo. A arte é tão antiga quanto nossa capacidade de se expressar e permanece relevante e presente até os dias de hoje (TORI; KIRNER, 2006). Dessa forma, uma das principais motivações para o uso da Realidade Virtual no meio artístico é que ela permite a interação com mundos virtuais impossíveis de se viver no mundo real. Os usuários percebem sua presença nesse mundo de forma semelhante à que percebem na realidade, tornando a experiência imersiva. O sentido da presença é crucial para incentivar o interesse e a exploração no processo de aprendizagem e de conhecimento de um meio (HO; SUN; TSAI, 2019).

Há uma forte tendência da simulação do real nas aplicações da Realidade Virtual, mas a realização do imaginário é fundamental em função de comunicar conceitos e ideias abstratas e criativas (TORI; KIRNER, 2006). Por intermédio dessa ferramenta tecnológica, a representação interativa e imersiva do imaginário artístico torna-se possível (MORIE, 1994). Aplicada em pinturas, a oportunidade de controlar uma câmera virtual e poder movimentar-se virtualmente possibilitam perceber e entender a arte de forma totalmente diferente. A partir dessa interação, o observador passa a não apenas olhar de forma passiva para a pintura, mas passa a fazer parte dela ativamente (OLIVEIRA, 2016).

Em um estudo realizado pelos estudantes de psicologia Antonietti e Cantoia (2000), foi mostrado para dois grupos de pessoas a mesma pintura, porém, para um grupo foi apresentada a pintura como um quadro, e para o outro em Realidade Virtual, onde eles podiam interagir com a mesma rotacionando-a, escalonando-a e movimentando-a. O resultado mostrou que, comparado com formas tradicionais de observação, a Realidade Virtual providenciou novos contextos e permitiu a exploração de perspectivas diferentes. Os usuários da Realidade Virtual também se sentiram estimulados a fazerem ligações associativas com outras áreas e a evocar experiências anteriores de suas vidas (ANTONIETTI; CANTOIA, 2000).

Observa-se que uma produção artística pode ser interpretada de ainda mais formas e analisada por perspectivas mais complexas. O nível de envolvimento que já existe é desenvolvido pela dimensão estética da Realidade Virtual e seduz o observador. Com uma nova possibilidade de interação, surgem novas sensações (MORIE, 1994).

O objetivo deste trabalho é propor uma nova forma de interagir com a pintura tradicional por meio da Realidade Virtual. Pretende-se explorar as propriedades dos objetos e eventos, perceber as implicações dessas propriedades e olhar para elementos que não podem ser vistos na imagem bidimensional. Para isso, uma pintura a óleo de um artista brasileiro foi escolhida e utilizada como referência para criar um ambiente virtual tridimensional interativo. Foram criados modelos 3D e os mesmos foram pintados digitalmente, levando em consideração o estilo do artista. Depois, os modelos foram exportados e organizados no motor de jogos Unreal Engine 4, onde foi feita a etapa de animação de alguns dos modelos e programação para tornar os objetos 3D interativos.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Realidade Virtual

O termo “Realidade Virtual” (VR) foi originalmente utilizado por Jaron Lanier, fundador da VPL Research Inc, no início dos anos 80, para diferenciar as simulações tradicionais feitas por computadores de simulações envolvendo múltiplos usuários em um ambiente compartilhado (ARAÚJO, 1996). Atualmente, esse termo é muito genérico, e estudiosos na área tendem a utilizá-lo de acordo com suas próprias experiências, fazendo com que existam diversas definições na literatura (NETTO; MACHADO; OLIVEIRA, 2007).

A definição proposta LaValle (2017) inclui toda a variedade de sistemas e tipos de realidade virtual: realidade virtual é a indução de uma situação ou comportamento em um organismo a partir de uma estimulação artificial sensorial, gerando uma experiência, de forma que o organismo tem pouca ou nenhuma sensação dessa interferência (LA VALLE, 2017). Para que uma simulação VR seja convincente, ela precisa ter uma taxa rápida de atualização das imagens, ou seja, ter um tempo de resposta mínimo no universo virtual e pouco ou nenhum atraso entre uma ação efetuada pelo usuário e a visualização e/ou resultado dessa ação virtual (ARAÚJO, 1996).

Em seu uso mais popular, a estimulação e percepção artificial geralmente dão-se por meio da criação de um ambiente tridimensional interativo gerado por computador. O usuário, quando interage com esse ambiente por meio de algum dispositivo (controles, Oculus Rift), é submetido à uma experiência programada, e tem a liberdade de navegar, observar e interagir com esse mundo digital em tempo real com seis graus de liberdade (6DOF). Ou seja, o *software* deve definir, e o *hardware* deve reconhecer, seis tipos de movimento: para frente/para trás, acima/abaixo, esquerda/direita, inclinação para cima/para baixo, angulação à esquerda/à direita e rotação à esquerda/à direita.

O indivíduo existe em três dimensões, têm a sensação do tempo real e a capacidade de interagir com o ambiente ao seu redor em um cenário totalmente digital. Esse cenário é um mundo virtual, uma cena simulada por computador, no qual o usuário pode navegar e explorar em tempo real com seis graus de liberdade 6DOF (NETTO; MACHADO; OLIVEIRA, 2006).



Assim, a RV se torna uma imitação da realidade física: o usuário existe em três dimensões, têm a mesma sensação de tempo contínuo e a capacidade de interagir com o mundo ao seu redor. A vantagem dessa experiência é que o conhecimento intuitivo do indivíduo é aplicado no espaço virtual (SCHWEBER, 1995).

## **2.2 Criação de mundos virtuais**

Um mundo virtual imersivo é um cenário tridimensional dinâmico, processado por um computador e exibido através de uma tela, projetor ou qualquer outro dispositivo com um display gráfico, em tempo real, de forma que o usuário se sinta parte do ambiente (PINHO; REBELO, 2006).

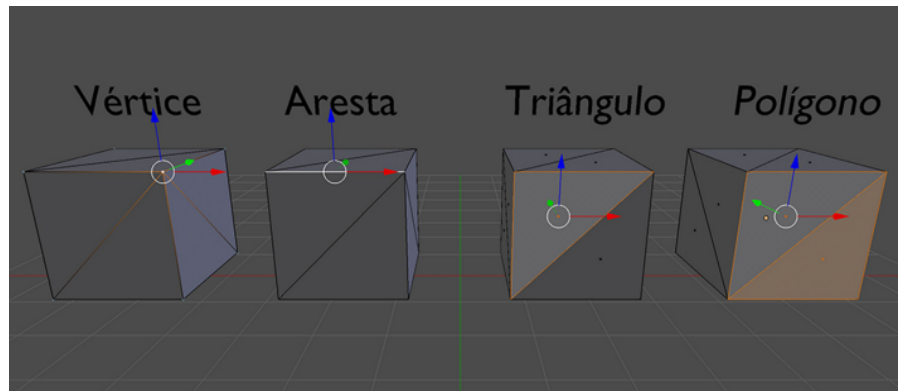
No processo de aplicação de uma RV, é necessário que um ambiente virtual seja construído, pois ele compreende tudo que o usuário irá visualizar e controlar. O mundo virtual envolve um determinado espaço e uma situação delimitada, incluindo todos os componentes nele inseridos, como os objetos e condições perceptíveis com as quais é possível interagir (KIRNER; SALVADOR, 2007).

A criação do ambiente engloba cinco etapas básicas: modelagem 3D, texturização, salvar e exportar os modelos, integração no software da RV e iluminação do ambiente. Essas etapas podem ocorrer de forma diferente dependendo do projeto.

### **2.2.1 Modelagem 3D**

Os modelos 3D de um mundo virtual, em termos gráficos, tem a função de representar o que o ambiente fornece no sentido de informação visual (FERREIRA; PELLEGRINO, 2007). Eles estão contidos em um ambiente virtual, formado por um espaço 3D euclidiano com coordenadas cartesianas. Dessa forma, todo objeto 3D modelado é formado por pontos que são representados por valores localizados em coordenadas específicas (x,y,z), como é possível observar na Figura 1 abaixo, em que os pontos de um de seus vértices localizam-se, respectivamente, em  $x=3$ ,  $y=5$  e  $z=4$ . Os modelos 3D são compostos por vértices, arestas, polígonos e faces, como mostrado na Figura 1, e as unidades desses dados são a referência para as dimensões do objeto modelado.

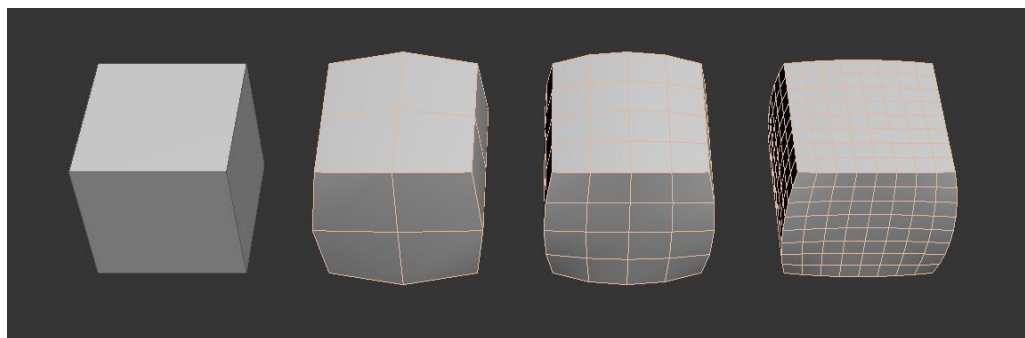
Figura 1 - Localização do vértice, aresta, polígono e face em um cubo 3D.



Fonte: Gilberto Martini de Oliveira (2016).

Modelos tridimensionais virtuais são formados por polígonos primitivos e fáceis de serem processados. O polígono primitivo mais simples e mais versátil é o triângulo, pois basta a união de três pontos para se formar um plano. Para fazer um objeto complexo, diversos triângulos podem ser organizados em uma malha. No caso da modelagem de curvas, quantos mais polígonos forem alinhados na malha, mais curvo será o objeto naquela seção, como mostrado na Figura 2 abaixo.

Figura 2 - Demonstração de como a curvatura de um modelo é influenciada pela quantidade de polígonos em sua malha.

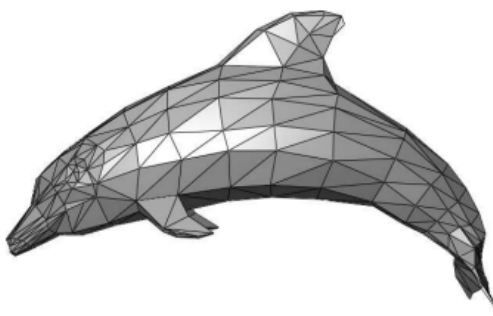


Fonte: Unreal Engine 5.3 Documentation. Disponível em: <https://docs.unrealengine.com/>

O processamento dos dados deve ser eficiente e econômico, no sentido de desempenho, portanto várias modelagens de objetos são formadas apenas pela sua "casca", ou seja, um modelo em que, na sua malha, todos os triângulos estão conectados entre si e não existe interseção entre nenhum deles, formando uma barreira entre o lado de fora e o lado de dentro do objeto. Assim, tem-se a formação

de um modelo coerente com uma superfície poligonal, como é mostrado na Figura 3 abaixo, utilizando como exemplo o modelo 3D de um golfinho (LA VALLE, 2017).

Figura 3 - Modelo coerente tridimensional formado por malha poligonal de triângulos de um golfinho.



Fonte: Steven M. LaValle (2017).

Outras primitivas podem ser utilizadas para modelar a malha de um objeto, como quadrados, splines e superfícies semi-algébricas. No entanto, essas aplicações implicam em uma maior exigência do computador para processar cada primitiva e conseqüente mais custo do projeto. Triângulos são os mais usados e mais recomendados pois são a forma mais simples que o algoritmo do programa pode processar, especialmente se o projeto envolver a aplicação de hardware (LA VALLE, 2017).

Os modelos coerentes, ou seja, aqueles formados apenas por sua "casca" e ela é uma malha conectada em si mesma, encerram todas as propriedades que um sólido pode contar de forma simples e processável. É possível alterar qualquer parte do objeto apenas manipulando seu aspecto externo ou gerar outras formas a partir de um conjunto de pontos espacialmente distribuídos (FERREIRA; PELLEGRINO, 2007).

### **2.2.2 Texturas e mapeamento 3D**

Depois de realizada a modelagem dos objetos da cena, inicia-se o processo de texturização e mapeamento dos mesmos. Em um projeto que busca uma experiência realista, esta etapa é tão importante quanto a modelagem 3D, pois são as texturas que informam aos objetos como eles se comportam quando são expostos à luz ou, de forma prática, quando o usuário os visualizar dentro do mundo

virtual. Texturas são imagens compostas por pixels, unidades digitais básicas que também formam a tela de visualização (BIANCHINI; et al, 2006).

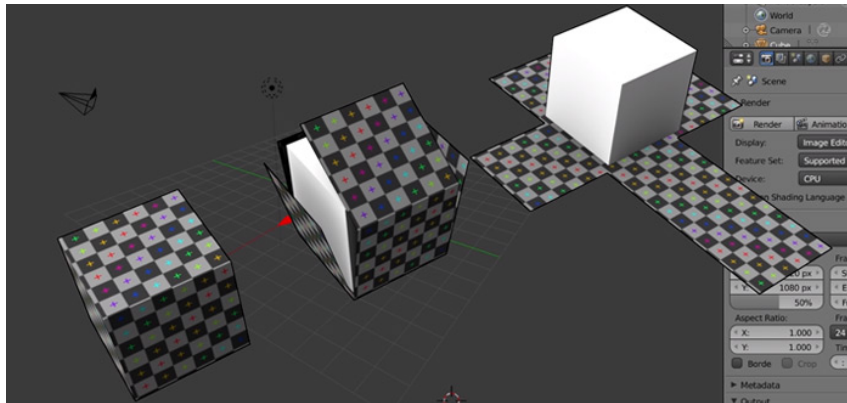
As texturas podem ser obtidas por meio de imagens bidimensionais geradas por programas, fotografias ou scanners de objetos reais ou podem ser pintadas do zero em softwares de criação de imagens digitais (RUIZ. 2011). Em todas as formas, elas são compreendidas como vetores bidimensionais que são mapeados por todos os polígonos da malha do modelo 3D. Qualquer imagem digital pode ser mapeada na malha de um objeto tridimensional, como um envoltório que o cobre. Quando é aplicada como textura, ela atribui um material ao modelo e suas informações gráficas são “impressas” na malha, incluindo a forma como a luz é refletida nesse material (LA VALLE, 2017). O processo de aplicar uma textura em um modelo 3D é chamado de mapeamento (RUIZ. 2011).

A texturização carrega diversos tipos de informações, sendo três as mais básicas:

- Informação de cor, que quando aplicadas à superfície de um modelo agem de maneira semelhante à um adesivo;
- Informação de propriedades como cor, brilho ou transparência, que permitem a simulação de diferentes materiais do mundo real (metal, madeira, plástico) e informam como a superfície deve se comportar quando iluminada;
- Informações que alteram a própria superfície, contendo coeficientes de refletividade e deslocamento. Essas texturas modificam as propriedades da malha, gerando novos relevos na mesma (SOUZA; ARAÚJO; LEE, 2011).

Há casos em que é necessário ser feito um mapeamento mais detalhado nos polígonos da malha, para colocar informações em locais específicos do objeto. Nesses casos, é utilizado o modificador *unwrap* (traduzido como “desembrulhar” ou “desempacotar”). O *unwrap* assimila todos os polígonos que compõem a malha do objeto e constrói uma superfície plana a partir de sua geometria, gerando uma espécie de envoltório com as faces do modelo 3D. A Figura 4 abaixo mostra como esse processo funciona na prática. A superfície pode ser usada como uma tela e nela pode ser criada a textura desejada em programas de tratamento e criação de imagens digitais, como Adobe Photoshop.

Figura 4 - Modificador *unwrap* aplicado ao modelo 3D de um cubo



Fonte: Thomas Denham. Disponível em: <https://conceptartempire.com/uv-mapping-unwrapping/>

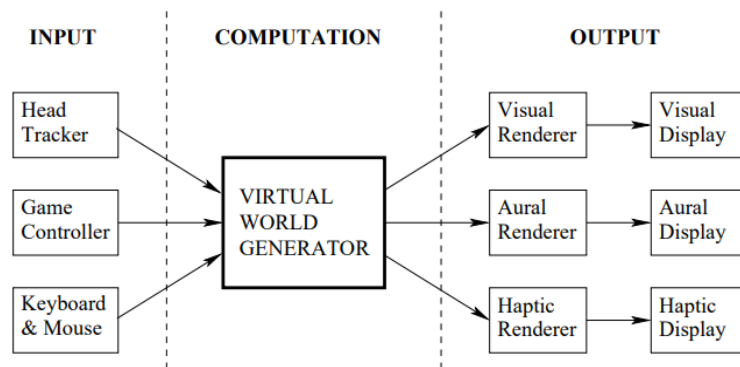
Com a textura pintada sob a superfície, esta é salva em um formato de arquivo de imagem (jpeg, png ou bmp) e o modelo é automaticamente atualizado com as novas informações (RUIZ. 2011).

### 2.2.3 Programa de Realidade Virtual

A Realidade Virtual possui quatro partes básicas: um ambiente físico (onde o usuário está), um sistema computacional de processamento (*hardware*), um sistema de visualização (a saída de vídeo da RV, onde o mundo virtual pode ser visualizado pelo usuário, incluindo áudio ou retorno tátil) e um sistema de interação (o meio pelo qual o usuário controla a RV, como *joysticks* e controles de teclado e mouse). Para a integração das últimas três partes, é preciso um software que una esses sistemas. Esse software é o programa onde o aplicativo da Realidade Virtual é propriamente sintetizado e coordenado, sendo geralmente uma engine gráfica (GNECCO; GUIMARÃES; DAMAZIO, 2007).

Na Figura 5 abaixo é possível observar um esquema que mostra como a integração acontece. O “*Virtual World Generator*” (Gerador de Mundo Virtual) corresponde ao *software* de programação da RV, que mantém o mundo virtual e recebe os comandos do usuário, por meio de um rastreador de movimento, controle ou mouse e teclado. Com os comandos recebidos no “*Input*”, a engine gera uma resposta, correspondendo ao “*Output*” que pode ser visual, auditiva ou sensitiva.

Figura 5 - Integração dos componentes de uma Realidade Virtual por meio de uma *engine* gráfica.



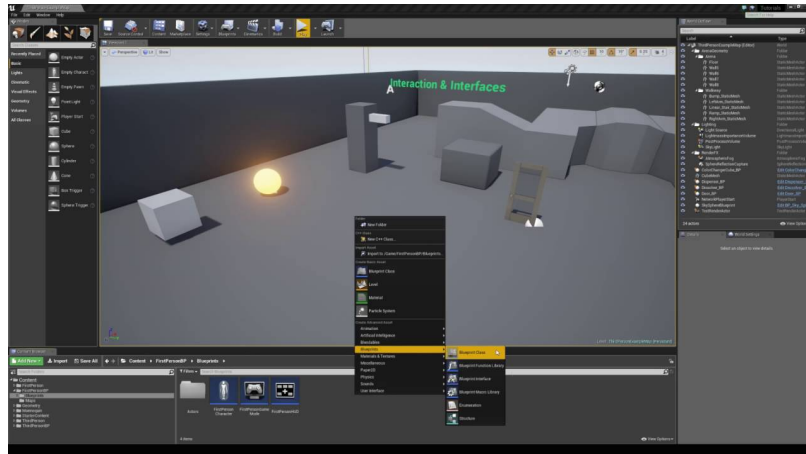
Fonte: Steven M. LaValle (2017).

A opção mais viável para criar uma Realidade Virtual é utilizar uma *engine* de jogos adaptada para desenvolver sistemas RV (LAVALLE, 2017). *Softwares* de programação têm a vantagem de possuir recursos essenciais na sintetização de qualquer RV, como iluminação, controle de som, interface gráfica com ícones de interação para o usuário, programação de controles e atualização automática do programa (GNECCO; GUIMARÃES; DAMAZIO, 2007). Alguns exemplos das engines para jogos adaptadas para RV mais utilizadas são o Open Simulator, Vizard, Unity 3D e Unreal Engine (LA VALLE, 2017).

#### 2.2.4 Unreal Engine

*Game Engines* ou motores de jogos são responsáveis por integrar elementos de desenvolvimento de aplicações gráficas, simplificando o processo, por meio de ferramentas como motores gráficos, físicos e de conexão (MACEDO; SERPA; RODRIGUES, 2015). Como dito anteriormente, a Unreal Engine é uma engine para jogos digitais que foi adaptada para a implementação de Realidades Virtuais. Ela foi desenvolvida pela empresa Epic Games e atualmente é o *software* líder na área de jogos no quesito de visualização, vegetação e criação de terreno realista, sendo utilizada mais comumente para grandes projetos (SMID, 2017). A Figura 6 abaixo mostra a interface inicial da Unreal Engine no começo de um projeto.

Figura 6 - Interface da Unreal Engine.



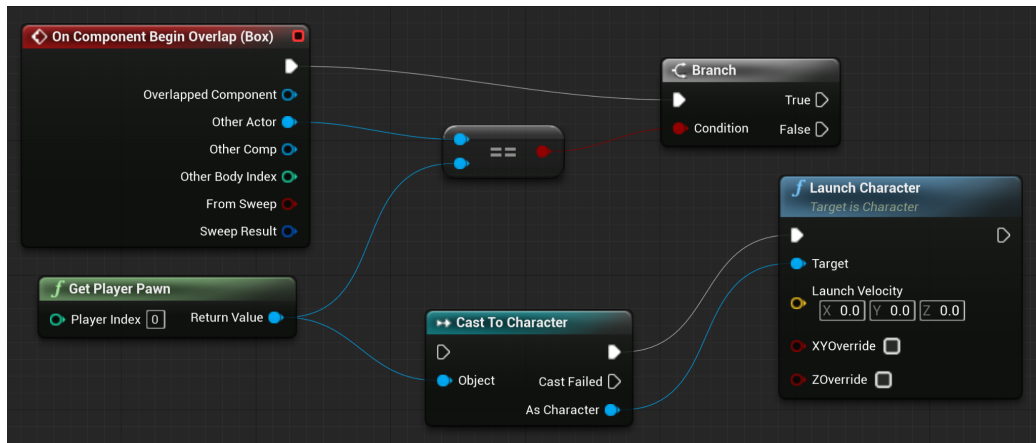
Fonte: Tesla Dev (2016). Disponível em:

<https://www.youtube.com/watch?app=desktop&v=xNKwsJGv--k>

A Unreal Engine oferece desde ferramentas para o desenvolvimento de jogos 2D para dispositivos móveis até simuladores de realismo. O *software* pode ser implementado em diferentes plataformas, como Windows, MacOS, Linux, Xbox, Playstation, Steam OS, Android e iOS. Dentre suas funcionalidades, pode-se citar efeito de partículas, programação e criação de novos materiais, gerenciamento de animações, produção de cutscenes e filmes e modelagem de terrenos e plantas. Como a maioria das engines, a Unreal possui uma biblioteca de conteúdo onde guarda todos os elementos do jogo ou aplicação, organizado por pastas, assets e códigos responsáveis pela aplicação (MACEDO; SERPA; RODRIGUES, 2015).

O diferencial do *software*, é que ele possui um sistema de Blueprints para programação visual. Blueprints são gráficos em formato de blocos que se conectam por “*nodes*” como pode ser visualizado na Figura 7 abaixo. As conexões geram uma lógica entre os elementos da aplicação, de forma que não é necessário entender de linguagem de programação para programar um jogo mais simples, tornando a engine mais acessível.

Figura 7 - Exemplo de Blueprint da Unreal Engine.



Fonte: Unreal 4.27 Documentation. Disponível em:

<https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/QuickStart/>

### 2.2.5 Iluminação e Renderização

Uma vez que os objetos da cena forem modelados, texturizados, salvos e exportados na *engine* de Realidade Virtual e a cena estiver montada, começa a fase final da iluminação e renderização. Nessa etapa, é determinado o que aparecerá para o usuário e é feita a iluminação da cena de acordo com as físicas do mundo virtual.

A iluminação do mundo virtual é vital para transmitir sentimentos e emoções, portanto sua intensidade e cores devem ser pensados de acordo com a experiência que se deseja passar para o usuário. As diferentes configurações das características da luz (luminosidade, direção, cor) e a forma como elas se mantêm para afetar o usuário são chamadas de padrões de luz. Os padrões de luz são utilizados em diversas mídias, como cinema, performances teatrais ou experiências imersivas e interativas (NIEDENTHAL, 2005). Em contexto de animação e computação gráfica, a luz tem cinco objetivos: direcionar a visão do espectador, criar profundidade, informar o período do dia e do ano, definir a atmosfera da cena e revelar a personalidade de um personagem ou de uma situação (CALAHAN, 2010).

Dentro dos *softwares* de modelagem 3D e de RV há diversos recursos para iluminar uma cena, como tipos de luzes com funções específicas, filtros e refletores, que buscam simular as ferramentas de uma iluminação de estúdio ou realista. Há também a possibilidade de alterar configurações mais específicas, como controle de exposição e correção de gama, como se faz em máquinas fotográficas. Em uma



Realidade Virtual, as luzes geralmente são utilizadas para destacar os objetos e modelos da cena, seja ela externa ou interna (BERNARDO, 2010).

Com as luzes do mundo virtual definidas, inicia-se a renderização, a última etapa do processo de criação de um mundo virtual. Há duas formas de se renderizar um projeto 3D: *off-line* e em tempo real. A renderização *off-line* é utilizada quando o computador não possui a performance necessária para fazer os cálculos de luz e textura em tempo real, sendo feitos com o auxílio da memória e do processamento da placa gráfica e levam certo tempo para serem completados (INSOL, 2022). A renderização em tempo real é utilizada principalmente em jogos interativos e calculam a luz e textura em um ritmo bem mais rápido, contando com um *hardware* gráfico específico para essa função. A principal vantagem é que os usuários podem interagir com a renderização enquanto ela é desenvolvida (AUTODESK, 2021).

No processo de renderização de uma RV, o computador calcula como a luz atinge os objetos 3D em tempo real, fazendo apenas o cálculo dos triângulos da malha que entram no campo de visão do usuário e os atualizando à medida que o mesmo interage com os modelos. Dessa forma, os valores dos pixels exibidos são calculados com base nas condições de iluminação e propriedades materiais definidas pela texturização, gerando informações de sombras, reflexões e refrações da luz.

Como essas condições são todas artificiais, é possível produzir qualquer efeito por meio da simulação das físicas do mundo virtual programado, sejam elas reais ou fantásticas (LA VALLE, 2017).

### **3 DESENVOLVIMENTO**

Este capítulo relata o desenvolvimento prático do projeto, discorrendo as etapas de planejamento do que será realizado e a descrição do que foi feito durante a produção: as etapas de modelagem, mapeamento de texturas dos modelos, integração da cena com a Unreal, criação de materiais, iluminação e programação.

#### **3.1 Pré-Produção**

Neste projeto, a fase da pré-produção refere-se à parte de pesquisa, escolha do artista e da obra e, a partir disso, o direcionamento da execução do projeto: o que será feito e quais programas serão utilizados para alcançar os objetivos.

### 3.1.1 Escolha do artista e da pintura

O artista escolhido para ser homenageado no projeto foi Rodrigo Yudi Honda, natural e residente da cidade de São Bernardo do Campo, em São Paulo. Seu trabalho envolve principalmente o uso de pintura a óleo sobre tela e foca suas obras no cotidiano suburbano do Brasil, geralmente retratando espaços com prédios, salas e ruas das cidades em um dia comum. Alguns exemplos de suas pinturas podem ser vistos abaixo na Figura 8.

Figura 8 - Pinturas “Quarta de Cinzas”, “Happy Hour” e “Guaraná, Chocolate em Pó e Caixas de Fósforo”



Fonte: Rodrigo Yudi.

Como o próprio afirma, suas pinturas não retratam lugares específicos, mas são inspiradas em locais reais do centro das grandes cidades e buscam retratar o estereótipo arquitetônico e o dia a dia das pessoas que vivem ali.

A sensação que suas pinturas passam são de cenas tão comuns no dia a dia do brasileiro que são difíceis de registrar como algo importante e belo, mas quando se vê em um quadro, tem-se a sensação de familiaridade e nostalgia. O resultado final da Realidade Virtual deve ser atrativo para usuários que não tenham conhecimento da área e das implicações tecnológicas, mas se sintam compreendidos pelo objeto de pesquisa, no caso, uma pintura que represente seus

cotidianos, instigando a curiosidade de conhecer um meio familiar de uma nova forma.

Outra razão que definiu a escolha final é o interesse de representar uma pintura real em meios digitais de um artista contemporâneo dos usuários da futura Realidade Virtual e ativo nas redes sociais. Há muitos artistas que realizam pinturas 2D digitais e muitos artistas brasileiros influentes de outras épocas, porém, Yudi desenvolve pinturas de forma tradicional e mostra seu processo, inspirações e trabalhos ativamente em plataformas de rede social, como Twitter, Instagram e Facebook, unindo tecnologia e pinturas clássicas.

A pintura escolhida foi “Sala de espera”, uma pintura em óleo sobre tela que retrata a sala de espera de um consultório odontológico, como visto na Figura 9 abaixo.

Figura 9 - “Sala de espera”, pintura óleo sobre tela.



Fonte: Rodrigo Yudi.

O quadro foi escolhido também por uma questão de otimização do processamento do hardware e experiência do usuário. Por se tratar da pintura de uma sala fechada, não há preocupação com a infinidade de uma cidade ou de limitar a exploração da Realidade Virtual com colisões invisíveis. O limite é o próprio espaço representado na pintura. A imersão do usuário assim não fica comprometida e tem-se um pequeno espaço definido com quantidade processável de objetos que serão modelados.

### 3.1.2 Definir o que será executado no projeto

Com a escolha da pintura feita, tornou-se possível traçar as próximas etapas práticas que se seguirão: quais objetos serão modelados, quais serão animados e onde a cena virtual interativa será programada.

Neste trabalho, foi decidido que todos os objetos serão modelados no 3Ds Max, programa da Autodesk, e pintados manualmente por textura no Photoshop. Essa escolha foi feita com base na experiência prévia e mais familiaridade da autora com os dois *softwares*.

Para facilitar o processo de modelagem, os objetos da pintura foram identificados e listados separadamente, permitindo-se ter uma noção de quantos modelos seriam necessários e a complexidade individual de cada um.

Depois disso, optou-se que alguns objetos na cena seriam animados. É possível identificar algumas representações de itens mecânicos e digitais que não ficam estáticos na vida real, como o relógio, o ventilador e o jogo de computador Paciência na tela da figura identificada como a Recepcionista. Na pintura original eles são estáticos, mas para explorar as possibilidades do meio em que será integrado, optou-se por animá-los, na busca pela imersão do usuário e de ultrapassar as limitações do meio tradicional.

Optou-se por não incluir as figuras humanas na modelagem: o Paciente e a Recepcionista, devido ao limite de prazo que o projeto precisa ser entregue.

Como engine de Realidade Virtual, onde será feita a etapa da programação, iluminação e renderização, optou-se pelo *software* Unreal Engine, detalhado na Revisão Bibliográfica acima. O programa foi escolhido pela experiência de professores e parceiros deste projeto, e podem auxiliar mais facilmente. Além disso, o *software* tem o recurso da programação em tabelas de nodes, permitindo que um usuário iniciante na área consiga programar um jogo executável sem necessidade de conhecer linhas de código complexas.

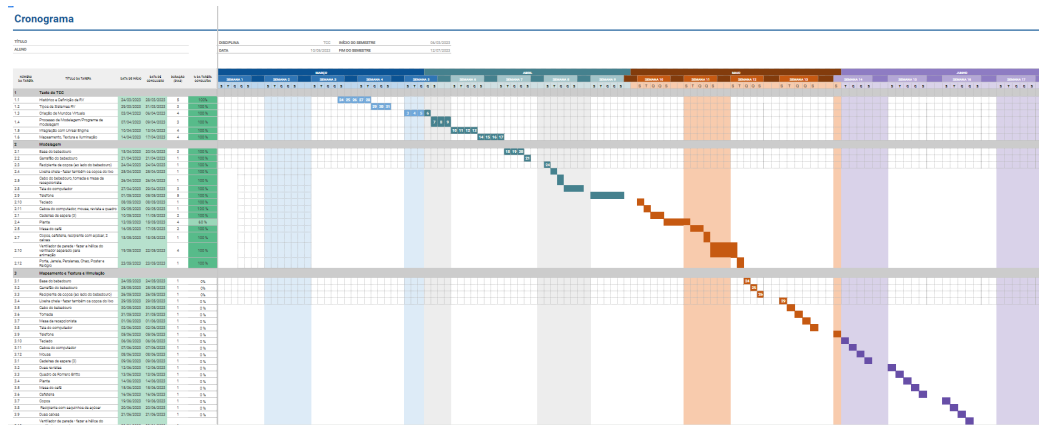
Para o *hardware*, utilizou-se o Oculus Meta Quest 2, desenvolvido pela Facebook Reality Labs.

## 3.2 Produção

Na fase da produção, iniciou-se a execução prática do projeto. Foram feitas as modelagens tridimensionais dos objetos da pintura escolhida, a montagem da

cena, o uso do modificador *unwrap* nos modelos 3D, a pintura da textura manualmente em cada modelo e a integração desses modelos na Unreal Engine. A partir dessa etapa, seguiu-se o cronograma mostrado na Figura 10 abaixo, com todas as partes do projeto descritas e datadas em quanto tempo deveriam ser executadas.

Figura 10 - Cronograma de execução do projeto.



Fonte: elaborada pela autora (2023).

### 3.2.1 Modelagem

Como especificado anteriormente, toda a modelagem dos objetos da cena foi executada no 3Ds Max. Antes de dar início à essa etapa, todos os objetos da pintura foram identificados e listados, para se ter uma ideia de quanto tempo cada um levaria para ser concluído.

Quadro 1 - Lista de objetos para serem modelados.

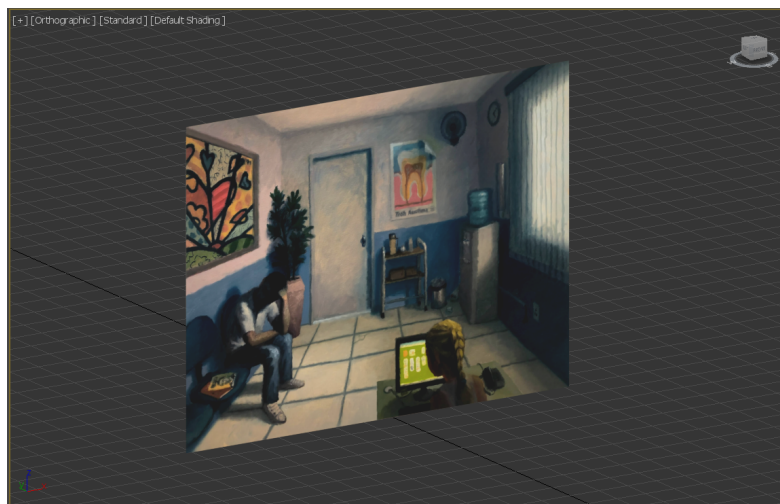
Número	Nome do objeto
1	Base do bebedouro
2	Garrafão do bebedouro
3	Recipiente de copos
4	Lixeira cheia de copos
5	Cabo do bebedouro
6	Tomada

7	3 cadeiras de espera
8	2 revistas
9	Quadro de Romero Britto
10	Planta
11	Mesa do café
12	Cafeteira
13	Copos
14	Cabo do bebedouro
15	Cabos do computador
16	Mouse
17	2 caixas
18	Recipiente com adoçantes
19	Ventilador de parede
20	Relógio
21	Porta
22	Persianas da janela
23	Janela
24	Pôster na parede
25	Mesa da recepcionista
26	Tela do computador
27	Telefone

Fonte: elaborado pela autora (2023).

O primeiro passo foi criar um plano no espaço tridimensional com a mesma proporção da imagem da pintura e carregar o arquivo desta imagem no plano, como mostra a Figura 11 abaixo, para ser usado como referência e facilitar a visualização.

Figura 11 - Plano no 3Ds Max com a imagem da pintura.



Fonte: elaborada pela autora (2023).

As dimensões da sala e dos objetos não são definidas por valores exatos, apenas pela forma como são representadas na pintura. Portanto, para modelar os objetos em um tamanho realista, o plano da Figura 11 foi dimensionado para representar uma das paredes da sala e assim servir como referência de tamanho para os outros objetos. Na prática, interpreta-se que uma parede da sala de espera tenha 4 metros de altura, largura e comprimento. A partir dessa proporção, os outros objetos são modelados, levando essa escala e seus correspondentes na imagem de referência em consideração.

Com o plano definido, foi criado um cubo para corresponder às dimensões do espaço fechado da sala de consultório, onde toda a cena foi montada. Para manter as proporções aproximadas, o objeto do cubo teve sua opacidade diminuída, enquanto a imagem de referência foi posicionada por trás, tentando corresponder às dimensões do cubo com as da sala, como demonstrado na Figura 12.

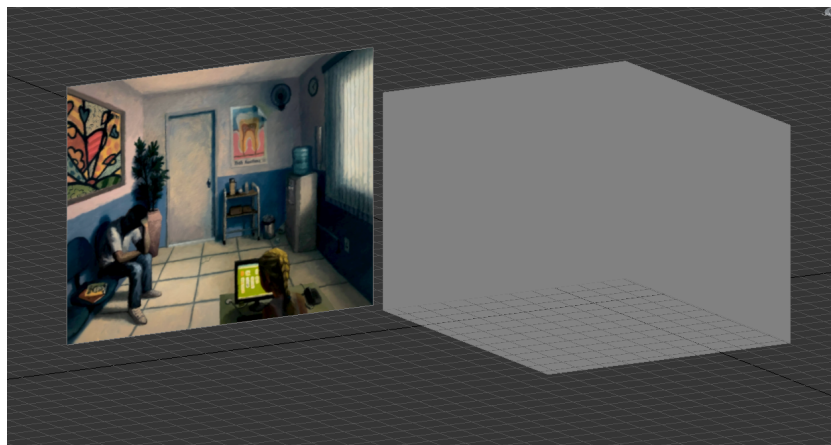
Figura 12 - Demonstração de como foi feita a correspondência entre modelo tridimensional e referência bidimensional.



Fonte: elaborada pela autora (2023).

A Figura 13 mostra como ficou a dimensão da sala tridimensional em comparação com a da sala bidimensional da pintura.

Figura 13 - Comparação de proporções entre espaço 2D e 3D.



Fonte: elaborada pela autora (2023).

Nas Figuras 14 e 15 abaixo compara-se a representação da garrafa de café e do ventilador na pintura com seus respectivos correspondentes tridimensionais. A proporção de alguns dos modelos tridimensionais não correspondem à referência bidimensional, mas seguem sua proposta de formato.

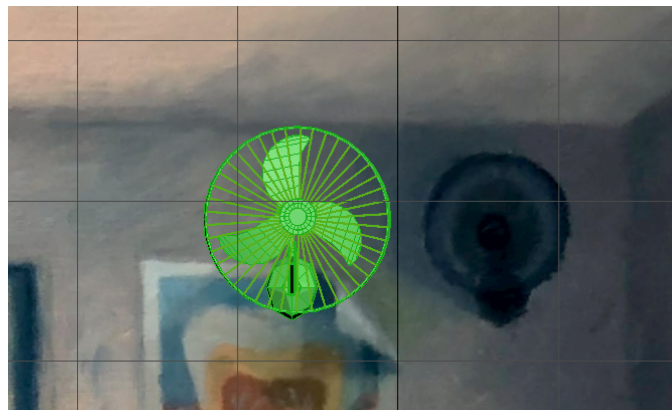
Figura 14 - Comparação entre o modelo da garrafa de café e a referência da pintura.





Fonte: elaborada pela autora (2023).

Figura 15 - Comparação entre o modelo do ventilador e a referência da pintura.



Fonte: elaborada pela autora (2023).

Além do uso das representações dos objetos na pintura, também foram utilizadas referências de imagens reais dos objetos, com marcas e modelos semelhantes aos da pintura, para facilitar a modelagem dos mesmos no ambiente 3D tendo referência outros pontos de vista.

As Figuras 16, 17 e 18 abaixo mostram exemplos de referências dos objetos reais que foram usadas no processo.

Figura 16 e 17 - Referência da torneira do bebedouro e do galão de água.



Fonte: site comercial da empresa Gasprati. Disponível em:  
<https://gasprati.loji.com.br/pagina/entrega?fora=1>

Figura 18 - Referência dos bancos de espera.

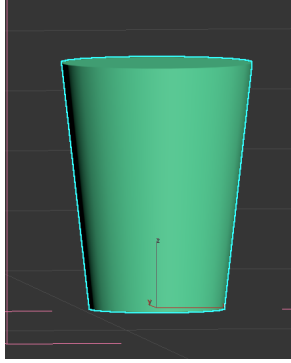


Fonte: site comercial da empresa Shop Cadeiras. Disponível em:  
<https://www.shopcadeiras.com.br/vvj4jbi57-longarina-secretaria-azul-com-preto-2-lugares-shop-cadeiras>

Para o processo de modelagem optou-se pelo *box modelling*, em que todos os objetos foram modelados a partir de um objeto primitivo inicial e informação e detalhes foram sendo adicionadas, até que representassem de forma mais acurada o objeto desejado.

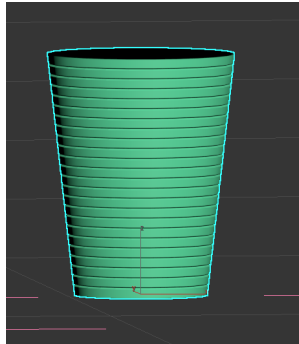
O copo plástico, por exemplo, começou como um cone, como mostrado na Figura 19. O modificador Edit Poly foi aplicado e, por meio dele, adicionados loops na malha para imitar a superfície irregular de um copo plástico, mostrado na Figura 20. Por fim, foram adicionados detalhes nas partes superior e inferior do copo, visualizados em seu formato final na Figura 21.

Figura 19 - Início da modelagem do copo como apenas um cone.



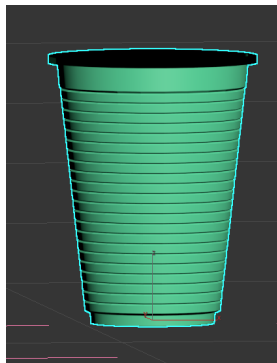
Fonte: elaborada pela autora (2023).

Figura 20 - Adição de *loops* ao longo da superfície vertical do copo.



Fonte: elaborada pela autora (2023).

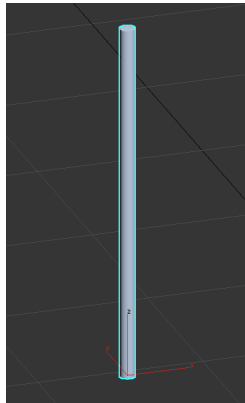
Figura 21 - Complementação do modelo com detalhes no topo e na base do copo.



Fonte: elaborada pela autora (2023).

Um exemplo de modelagem mais complexa é a do suporte da mesa de café. Ao invés de um cone, a primitiva mais apropriada para o começo da modelagem desse objeto foi um cilindro, mostrado na Figura 22 abaixo.

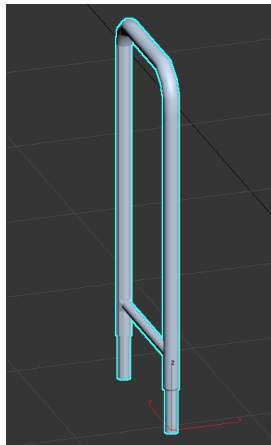
Figura 22 - Início da modelagem do suporte da mesa como apenas um cilindro.



Fonte: elaborada pela autora (2023).

Com o modificador Edit Poly e Symmetry, a forma do cilindro foi estendida e mais faces foram adicionadas, até formar o suporte que se tem na pintura, mostrado na Figura 23 abaixo.

Figura 23 - Cilindro modificado, agora semelhante a um dos suportes da mesa da pintura.

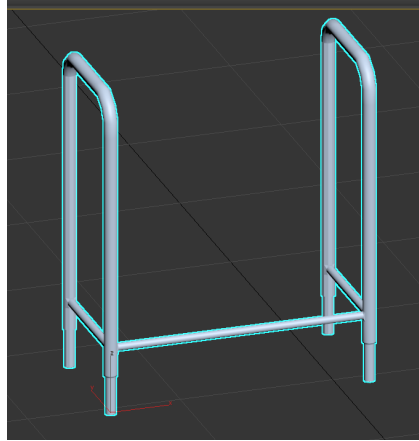


Fonte: elaborada pela autora (2023).

Por fim, o modelo foi duplicado e foi aplicado o modificador ProBoolean, que serve para interseccionar modelos de diversas formas. Nesse caso, a ferramenta foi utilizada para unir a cópia do modelo original ao mesmo, fazendo que fizessem parte do mesmo objeto. Para unir as duas superfícies, foi criada uma intersecção no modelo e criada uma face, que foi estendida e conectada à outra superfície, como

mostra a Figura 24, tendo um modelo fechado e conectado e a forma acurada do suporte.

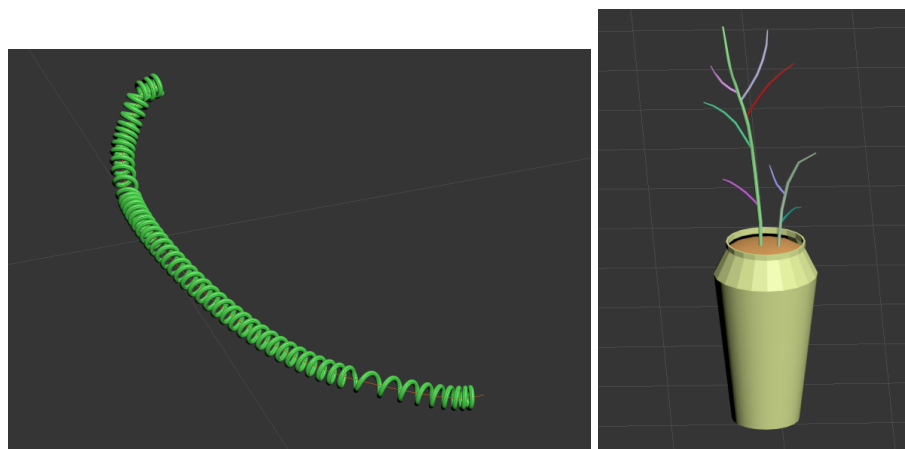
Figura 24 - Modelo final do suporte da mesa.



Fonte: elaborada pela autora (2023).

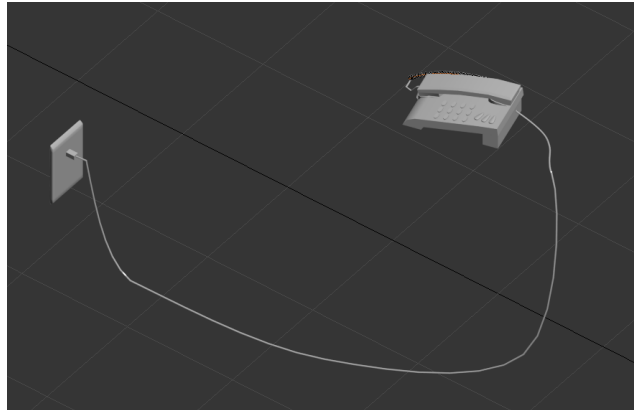
Além dos objetos primitivos, também foram utilizados *splines* para modelar alguns objetos, como o fio em espiral do telefone, os galhos da planta e os fios que conectam o telefone e o computador e seus componentes à tomada, vistos, respectivamente, nas Figuras 25, 26 e 27 abaixo. Optou-se por *splines* nesses casos pois esses objetos traçam um formato específico, que seria mais difícil de obter por um objeto primitivo com várias faces. Assim, representar fios e galhos foi mais fácil a partir de uma única linha.

Figura 25 e 26 - Fio em espiral do telefone e galhos da planta por *spline*.



Fonte: elaborada pela autora (2023).

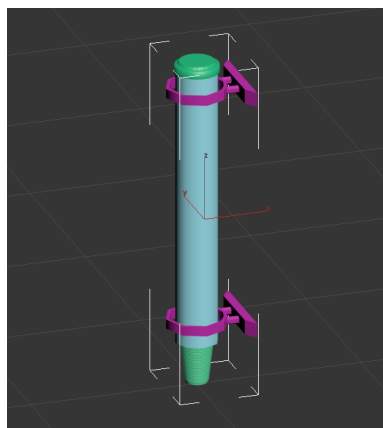
Figura 27 - Fio que liga o telefone ao *plug* na parede.



Fonte: elaborada pela autora (2023).

Uma questão que surgiu durante o processo de modelagem foi a necessidade de criar mais modelos do que os inicialmente planejados e listados. Isso porque alguns objetos quando representados em um ambiente 3D faltavam componentes que justificassem a natureza do seu funcionamento. Por exemplo, o porta copos do lado do bebedouro não possui suportes visíveis que o prendem à parede na pintura. No entanto, durante a modelagem, foi necessário fazer suportes para justificar a presença do porta copos e para que não parecesse que ele está flutuando no ar. A Figura 28 mostra os dois suportes que foram adicionados, não inicialmente planejados na pré-produção.

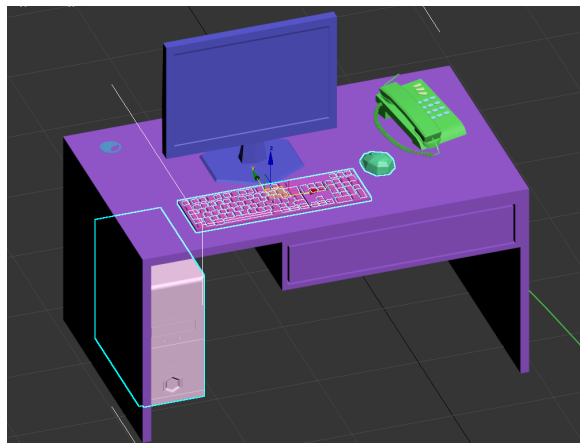
Figura 28 - Modelo do porta copos com dois suportes que o prendem à “parede”.



Fonte: elaborada pela autora (2023).

Um modelo de teclado, mouse e CPU, representados na Figura 29 abaixo, também foram adicionados à cena, para complementar o funcionamento do computador da recepcionista e dar maior sentido ao papel dele na construção da cena.

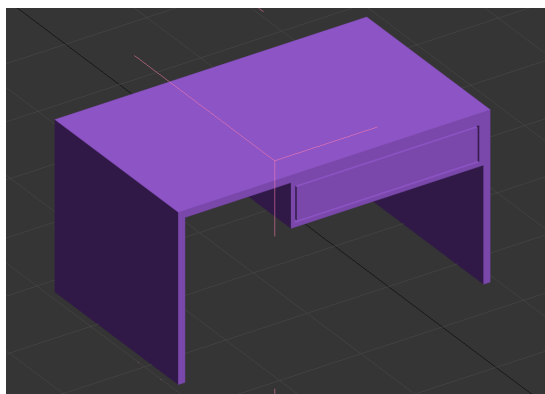
Figura 29 - Modelos do teclado, mouse e CPU destacados.



Fonte: elaborada pela autora (2023).

Outra adaptação necessária foi complementar de forma criativa os modelos que não aparecem totalmente na pintura. A mesa da recepcionista, por exemplo, não é mostrada integralmente, limitada a apenas um retângulo que é suporte para os objetos sobre ela. Em um ambiente virtual de exploração, é necessário que todos os objetos sejam representados em sua forma total para tornar a experiência imersiva. Assim, a mesa foi feita a partir de referências externas e em seu resultado final não vai ser totalmente fiel à pintura, adaptação essencial na tradução de um ambiente 2D para um ambiente 3D. A interpretação do modelo da mesa pode ser visualizado na Figura 30 abaixo.

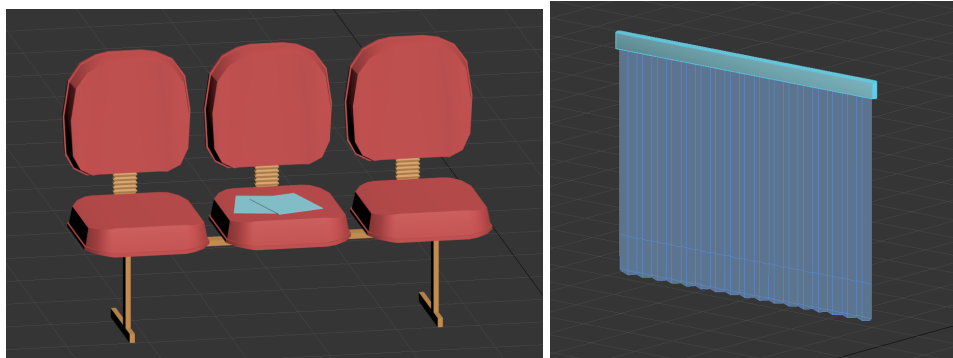
Figura 30 - Modelo da mesa da recepcionista criado por referências externas.



Fonte: elaborada pela autora (2023).

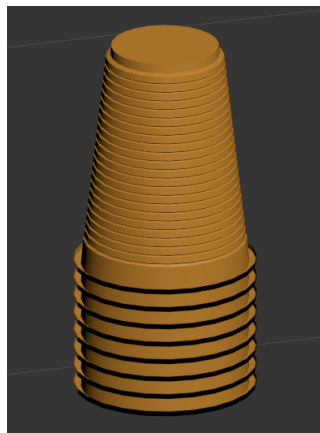
Para objetos repetidos, apenas um modelo foi feito, copiado e redistribuído na cena, para otimizar a modelagem e o processamento do arquivo pelo computador. Alguns objetos em que esse recurso foi utilizado foram as cadeiras de espera, as cortinas e os copos plásticos, mostrados respectivamente nas Figuras 31, 32 e 33 abaixo.

Figura 31 e 32 - Modelo das cadeiras de espera e da cortina.



Fonte: elaborada pela autora (2023).

Figura 33 - Modelos dos copos plásticos.



Fonte: elaborada pela autora (2023).

Durante a modelagem, foi criada uma câmera, posicionada para simular a perspectiva da pintura e facilitar posicionar os objetos o mais aproximado possível com a pintura original. Com todos os objetos modelados e devidamente posicionados, foi montada a cena do ambiente virtual, que pode ser visualizada abaixo na Figura 34.



Figura 34 - Resultado final da representação da pintura em um ambiente tridimensional.



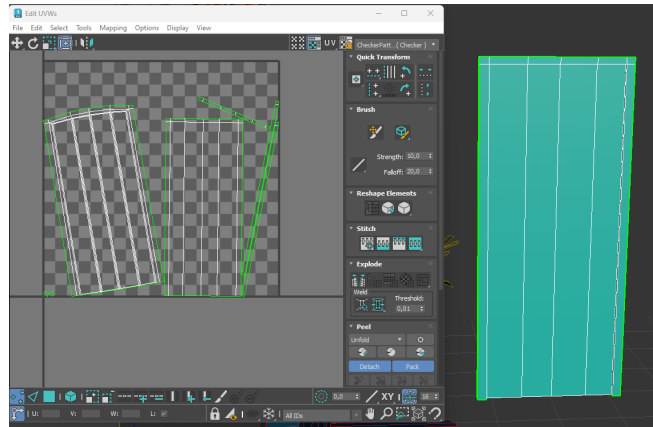
Fonte: elaborada pela autora (2023).

### 3.2.2 Mapeamento

Com a modelagem de todos os objetos e a cena virtual montada no 3Ds Max, a próxima etapa foi a aplicação do modificador Unwrap nos modelos tridimensionais e a pintura manual da cor de cada um deles. Essa etapa foi uma das mais importantes, pois nela foram estudados os traços do artista para que fossem reproduzidos o mais fielmente possível nos objetos correspondentes.

O primeiro passo foi utilizar o modificador Unwrap em cada objeto da cena. Como explicado anteriormente, essa ferramenta constrói uma superfície plana a partir da geometria do modelo 3D com objetivo de permitir a pintura digital em *software* de pintura e edição de imagens. O *software* normalmente fornece uma superfície plana para o modelo que é automaticamente feita quando o modificador é aplicado, porém, na maioria dos casos, o resultado não é satisfatório. Na Figura 35 abaixo, vê-se um exemplo de Unwrap do modelo da porta feito automaticamente pelo 3Ds Max. As partes que compõem a porta são claras, porém, é uma malha desorganizada e desalinhada, além de que pode ser simplificada em menos separações.

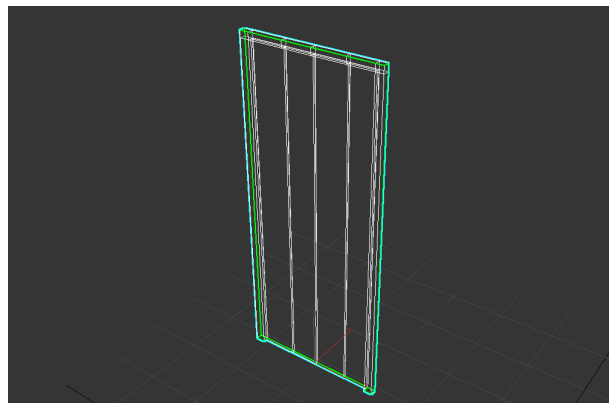
Figura 35 - Exemplo de Unwrap automático de um objeto.



Fonte: elaborada pela autora (2023).

Por essa razão, o Unwrap automático foi pouco utilizado, e a maioria dos objetos teve sua malha "cortada" manualmente. Foram feitos *loops*, representados pelas linhas verdes, que marcam onde a malha do objeto foi separada, como pode ser visto na Figura 36 abaixo.

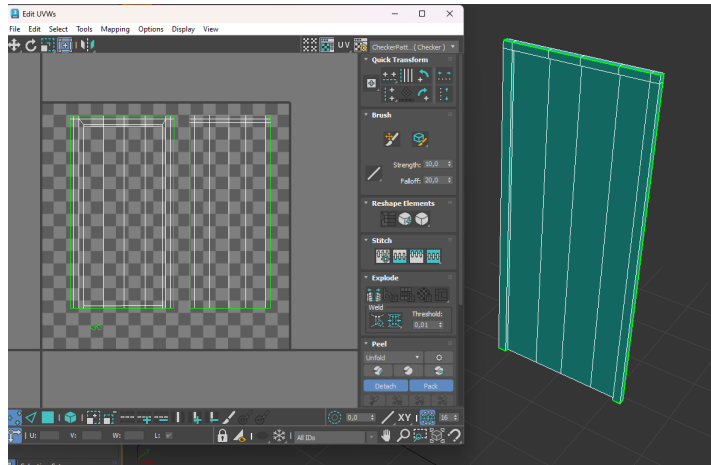
Figura 36 - Modelo 3D com cortes em loop na malha.



Fonte: elaborada pela autora (2023).

Feito isso, foi utilizada a ferramenta “Quick Peel”, que fornece o Unwrap do objeto a partir dos cortes feitos na malha. O resultado final pode ser visto na Figura 37 abaixo. A superfície plana fornecida do objeto ficou mais clara e é possível entender suas partes separadamente.

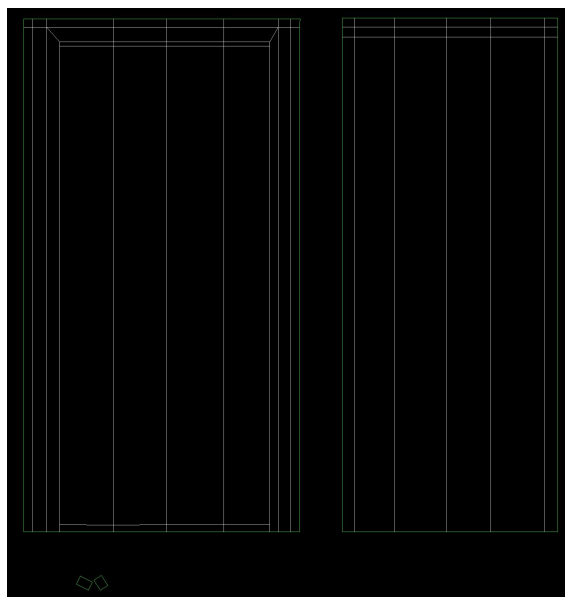
Figura 37 - Unwrap do modelo feito a partir de cortes manuais na malha.



Fonte: elaborada pela autora (2023).

Finalmente, a superfície gerada é renderizada em um formato de imagem. Nesse projeto, o formato de imagem escolhido foi o PNG, sempre em formato quadrado e dimensionados com números que sejam potências de 2 como 512x512 e 1024x1024. Esse padrão deve ser mantido por questão de otimização e desempenho do cálculo da luz feito na Unreal Engine. A imagem final gerada do modelo da porta pode ser vista na Figura 38 abaixo. Com esse processo aplicado para todos os modelos, obteve-se o mapa de textura de cada um deles.

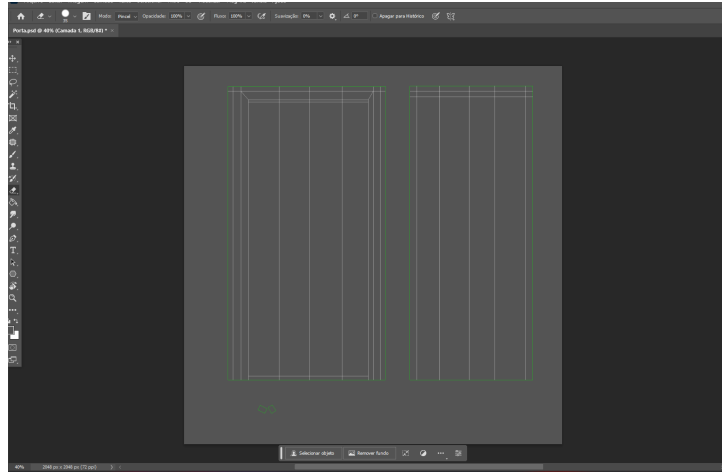
Figura 38 - Imagem final do mapa de textura do modelo da porta.



Fonte: elaborada pela autora (2023).

Com essa etapa concluída, as imagens dos mapas de textura foram abertos no Adobe Photoshop. O mapa de textura do modelo da porta no Photoshop pode ser visto na Figura 39 abaixo.

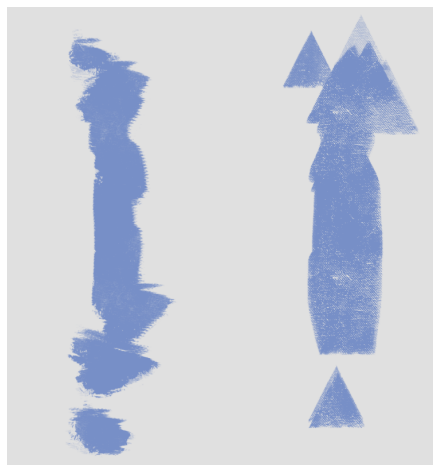
Figura 39 - Mapa de textura da porta no Photoshop.



Fonte: elaborada pela autora (2023).

No Photoshop, os mapas de textura foram utilizados como telas, para direcionar onde a pintura ficaria em qual parte do objeto dependendo da face pintada digitalmente. Optou-se por pincéis digitais que imitassem o traçado de um pincel real em uma tela para reproduzir as pinceladas e traços da pintura original. Os dois pincéis mais usados durante o processo e seus traços podem ser visualizados na Figura 40 abaixo.

Figura 40 - Pincéis utilizados no processo de texturização.



Fonte: elaborada pela autora (2023).

Com os pincéis escolhidos e o mapa de textura pronto, iniciou-se a pintura. A imagem da obra original foi usada constantemente como referência de cor, forma e

direção dos traçados. No caso do modelo da porta, o resultado final da pintura pode ser visto na Figura 41 abaixo, em comparação com a porta original ao lado. Apenas algumas faces foram pintadas, pois são as únicas que serão visualizadas pelo usuário no mundo virtual. Esse método visa otimizar o processamento da cena nas etapas finais.

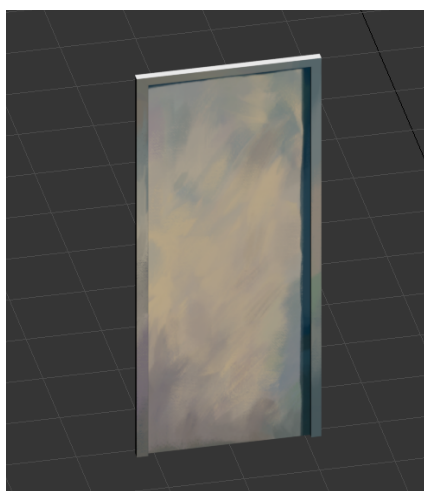
Figuras 41 - Pintura da porta feita manualmente.



Fonte: elaborada pela autora (2023).

Com a pintura finalizada, o arquivo salvo em formato PNG e esta imagem é aplicada como mapa de cor no objeto da porta no 3Ds Max, selecionando a opção Bitmap. O modelo com a pintura aplicada pode ser visto na Figura 42 abaixo.

Figura 42 - Modelo tridimensional da porta com textura de imagem como cor aplicada.

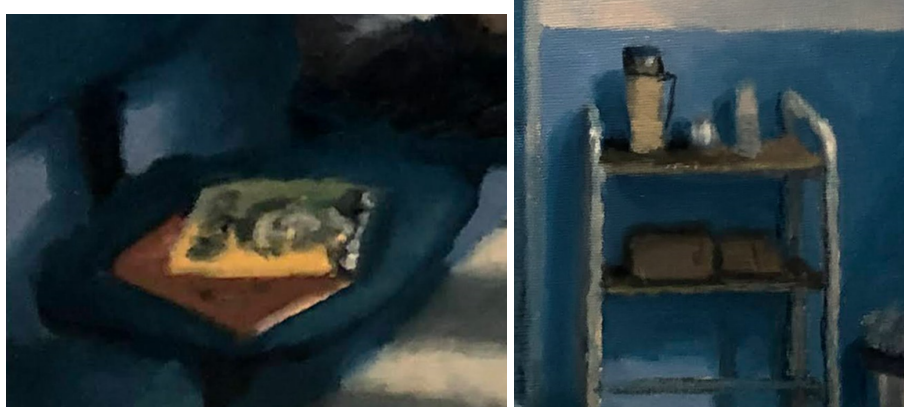


Fonte: elaborada pela autora (2023).

Qualquer alteração feita no arquivo da imagem PNG é atualizada automaticamente no modelo no programa.

Uma adaptação necessária foi a pintura das revistas em cima das cadeiras de espera e do sachê de açúcar na mesa de café. Na pintura original, a forma como eles foram pintados não permite identificar com clareza as marcas, formas e cores, como pode ser visto nas Figura 43 e 44 abaixo.

Figura 43 e 44 - Representação das revistas e do sachê de açúcar na pintura original.



Fonte: Rodrigo Yudi.

Como o usuário terá mais liberdade de interação e visualização desses objetos no jogo, foram utilizadas referências externas para representá-los com mais clareza no mundo virtual. Apesar da representação não corresponder exatamente à pintura, optou-se em oferecer ao usuário a sensação de imersão e de ser parte do mundo do quadro, podendo enxergar seus detalhes com clareza. As Figuras 45 e 46 abaixo mostram o resultado final das revistas e do sachê.

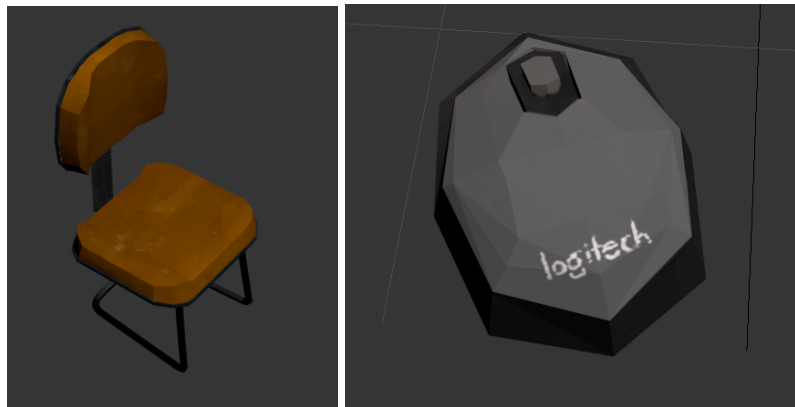
Figuras 45 e 46 - Modelo do sachê de açúcar das revistas com mapa de textura aplicado.



Fonte: elaborada pela autora (2023).

Também se utilizou referências externas para a pintura dos objetos que foram adicionados por não serem representados no quadro, como a cadeira da recepcionista e o mouse. A pintura original foi usada como referência dos traços e do estilo, para que os objetos fizessem parte do mesmo cenário, seguindo a mesma direção de arte. As Figuras 47 e 48 mostram, respectivamente, os modelos de cada um com a textura aplicada.

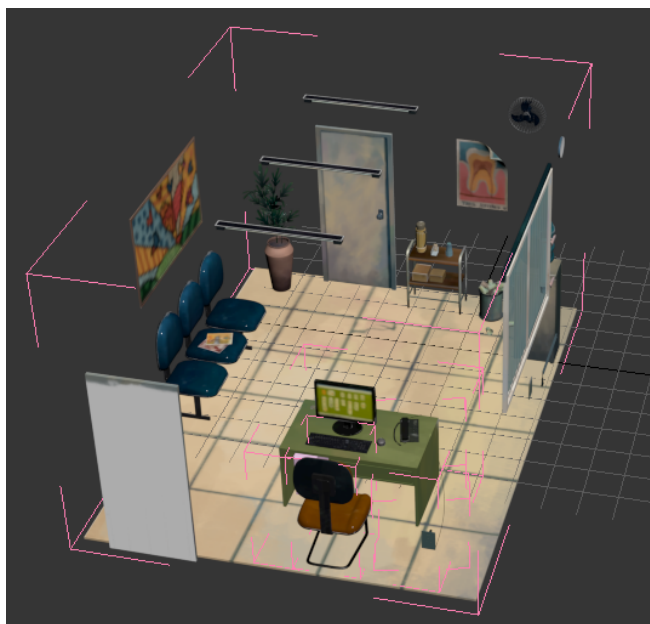
Figura 47 e 48 - Modelo da cadeira da recepcionista e do mouse com mapa de textura aplicado.



Fonte: elaborada pela autora (2023).

A partir de todas as adaptações e utilizando-se a pintura original como referência, a cena virtual foi colorida como um quadro. O resultado da pintura é mostrado na Figura 49 abaixo.

Figura 49 - Cena virtual do quadro montada.



Fonte: elaborada pela autora (2023).

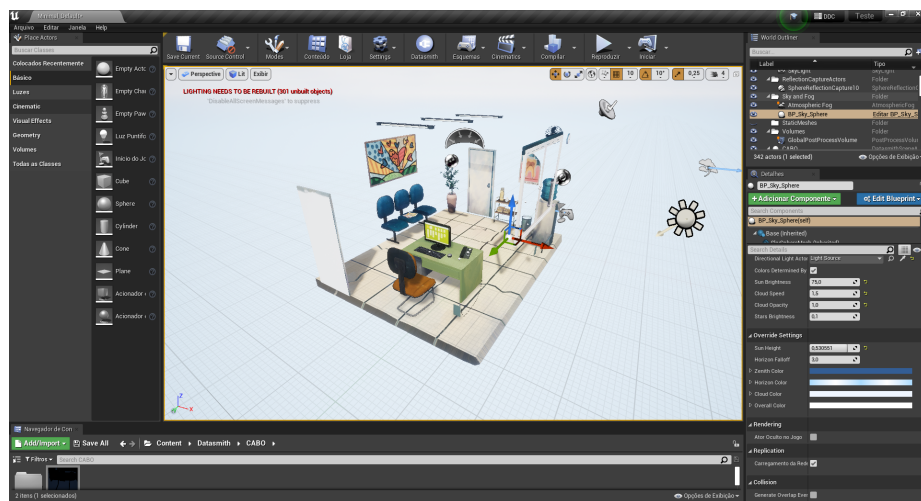
### 3.2.3 Integração da cena na Unreal Engine

Com a etapa da modelagem, mapeamento de texturas e colorização dos objetos no 3Ds Max, inicia-se a integração da cena virtual construída na Unreal Engine, ou seja, enviar os modelos para o programa de jogos. A cena precisa ser enviada para a Unreal pois é nela que o ambiente virtual se tornará interativo em Realidade Virtual. O *software* implementa códigos que tornam essa programação possível.

Para importar os modelos tridimensionais do 3Ds Max para um novo projeto na Unreal Engine, foi utilizado o plug-in Datasmith, desenvolvido pela própria desenvolvedora de jogos para importar arquivos externos para dentro da engine com facilidade. Toda a cena foi selecionada e depois exportada como arquivo Unreal Datasmith, como visto na Figura 56 abaixo.

Começando um novo projeto de jogo na Unreal, o arquivo Datasmith foi importado no programa de forma que todos os objetos da cena mantivessem sua posição e formatos originais. Além dos objetos, também foram importados os mapas de colorização feitos na etapa anterior, ligados aos seus respectivos modelos, advindos do 3Ds Max. A Figura 50 mostra a cena virtual implementada na Unreal Engine. Por motivos de desempenho do *hardware*, optou-se por utilizar a versão 4.27 do *software* de jogos.

Figura 50 - Modelos da cena virtual selecionados e sendo exportados.



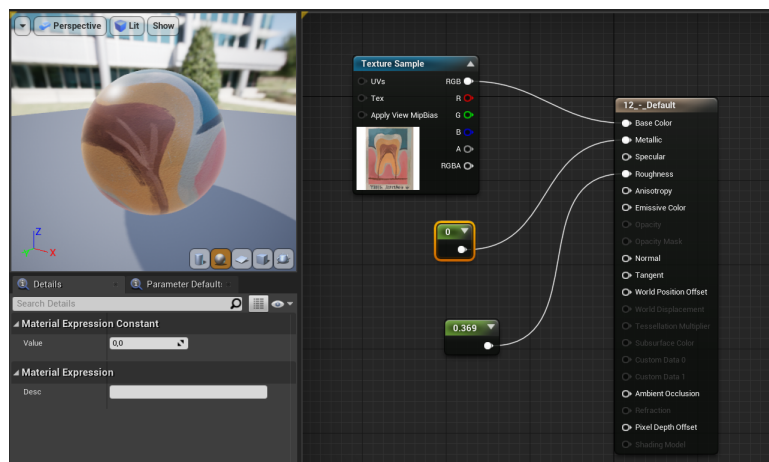
Fonte: elaborada pela autora (2023).



### 3.2.4 Criação de materiais

A próxima etapa que se seguiu foi a complementação dos materiais dos modelos. O Datasmith exporta os materiais de cada objeto, que no 3Ds Max e na Unreal são do tipo Physical Material. Na Unreal, esses materiais são acessíveis e editáveis, todos já possuindo uma informação de cor que foi feita no Photoshop e atribuída aos objetos no 3Ds Max, como descrito nas etapas anteriores. Os objetos da cena, dessa forma, foram integrados na Unreal junto de seus respectivos materiais. Porém, eles possuem apenas a informação de cor e precisam ser refinados para que os modelos representem características de seus referentes na vida real. Os principais valores alterados nessa etapa foram os de Roughness e Metalness<sup>1</sup>, variando de 0 a 1, que são expressões constantes. As constantes foram suficientes para definir o comportamento de muitos modelos da cena em relação à luz, como é o caso do material do pôster de dentista, mostrado na Figura 51 abaixo.

Figura 51 - Material do pôster do consultório.



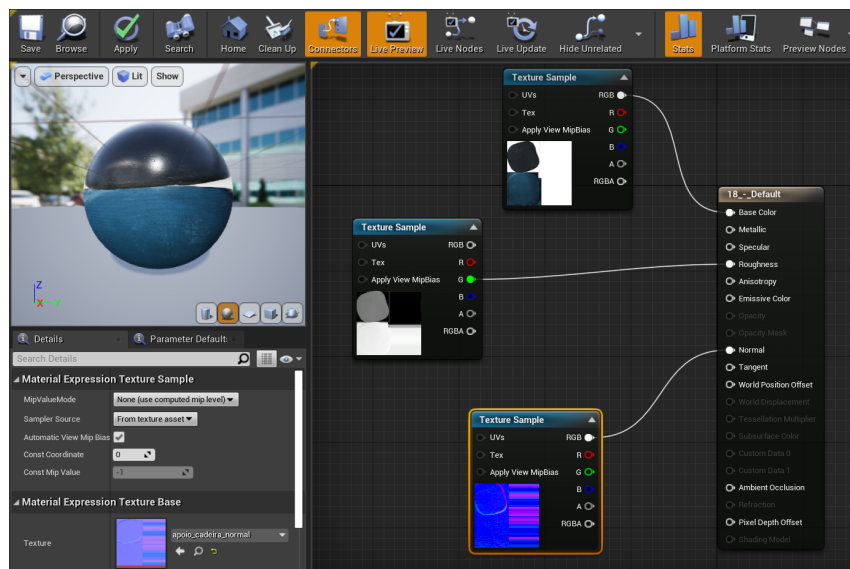
Fonte: elaborada pela autora (2023).

Para alguns materiais, atribuiu-se informações com imagens, pois correspondiam a objetos mais complexos e com mais de um comportamento em relação à luz, como é o caso das cadeiras. Para esses modelos, utiliza-se dos mesmos mapas de textura que os utilizados para a cor. Assim, as informações de Roughness, Metallic e Normal foram desenhadas, informando a Unreal quais partes

<sup>1</sup> O mapa de Roughness (ou mapa de rugosidade) define como a luz se espalha pela superfície do modelo 3D. O valor 0 resulta em uma superfície brilhante e o valor 1 resulta em uma superfície fosca. Já o mapa de Metalness (ou mapa de metal) define se o material é mais ou menos metálico na mesma escala de 0 para 1.

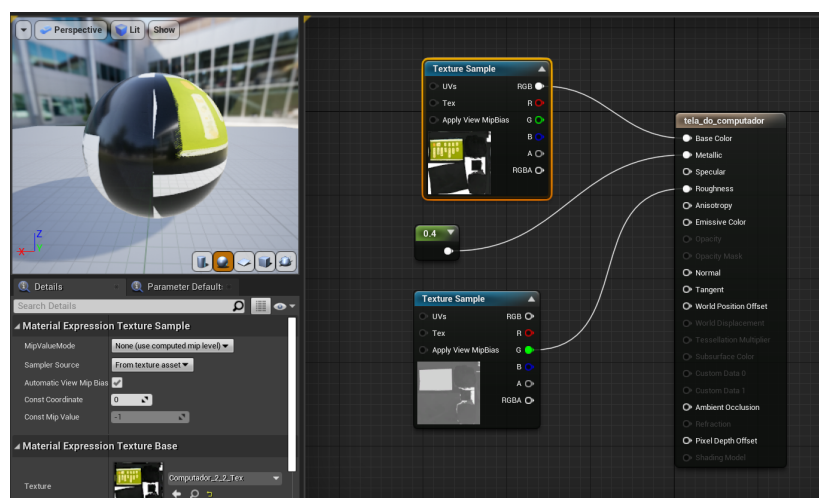
do modelo se comportam de maneira diferente. É o caso do material das cadeiras, mostrado na Figura 52 abaixo, e do computador, mostrado na Figura 53 abaixo. Para as cadeiras, foi feito um mapa de Normal e Roughness procedural no Photoshop, fazendo com que a parte estofada e a parte de plástico possuísem diferentes texturas sobressaltadas.

Figura 52 - Material da poltrona.



Fonte: elaborada pela autora (2023).

Figura 53 - Material do computador.

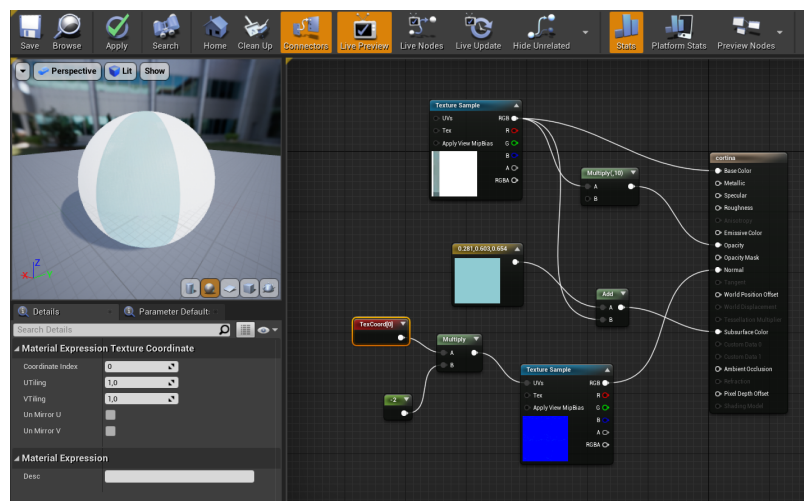


Fonte: elaborada pela autora (2023).

Para o material do computador, foi feito um mapa de Roughness procedural, informando que a parte metálica é mais reflexiva que a tela.

As cortinas têm um comportamento específico em relação à luz, então seu material foi modificado para o tipo Masked ao invés de opaco, como a maioria dos modelos é. Dessa forma, eles mudam de cor e se tornam mais emissivos quando atingidos pela luz. A Figura 54 mostra como seu material foi construído e a Figura 55 como seus modelos respondem à iluminação. Optou-se por fazer materiais do tipo Masked para os objetos translúcidos, como os copos plásticos, pois o tipo de material Translucent exige muito do processamento do computador.

Figura 54 - Material da cortina.



Fonte: elaborada pela autora (2023).

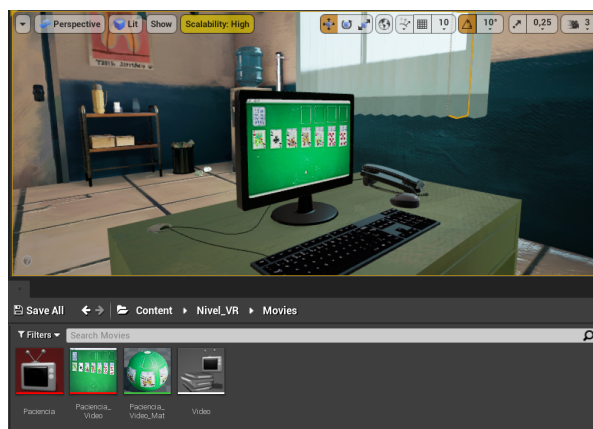
Figura 55 - Modelo da cortina em resposta à luz direta.



Fonte: elaborada pela autora (2023).

Outro material específico é a tela do computador. Foi criado um plano do mesmo tamanho da tela e aplicado nele uma textura de vídeo, para simular que alguém está jogando Paciência no computador. O vídeo foi estilizado para parecer mais uma pintura e corresponder melhor à direção artística da cena. A Figura 56 mostra o resultado final do material aplicado na tela do computador e os componentes utilizados para que o vídeo funcione como uma textura. Quando o jogo começa, o vídeo se inicia e reinicia quando termina.

Figura 56 - Modelo do computador com a textura de vídeo aplicada na tela.



Fonte: elaborada pela autora (2023).

### 3.2.5 Iluminação

A iluminação de uma cena de Realidade Virtual é feita em tempo real, o que consome muito do processamento. Por causa disso, optou-se por utilizar luzes estacionárias na Unreal. Esse modo permite que as luzes projetem sombras dinâmicas em objetos móveis e que sua iluminação direta pode ser modificada durante o jogo, porém, sua iluminação indireta e sombras em objetos estáticos também são estáticas, e permanecem as mesmas uma vez que a luz é carregada. Utilizando a pintura como referência, percebe-se que a iluminação vem do sol e entra pela janela de forma difusa.

Por isso, optou-se por evitar usar luzes dentro das paredes da sala e iluminar a cena de forma semelhante: utilizando apenas a luz do sol e a luz do céu, respectivamente, uma Directional Light e uma Sky Light. Ambas as luzes tiveram suas intensidades aumentadas e configurou-se que suas luzes indiretas deveriam bater nos objetos e serem redirecionadas diversas vezes, utilizando a função Indirect

Lighting Intensity, para que fossem suficientes para iluminar o espaço fechado. Também foi adicionado um Post Process Volume, que permite alterar a exposição, temperatura e colorização geral da cena virtual, independente da luz, funcionando como uma ferramenta de pós-produção de edição.

A Unreal precisa ser informada que há uma janela na cena, então foi colocado um Lightmass Portal com as mesmas dimensões da janela por onde entra a luz, como pode ser visualizado na Figura 57 abaixo.

Figura 57 - Lightmass Portal na janela.



Fonte: elaborada pela autora (2023).

Por fim, foram colocadas cinco Sphere Reflection Captures na cena e espalhadas pela sala, vistas na Figura 58 abaixo. A função dessas esferas é refletir a cena virtual, dentro do raio que elas contemplam, nos objetos próximos. Elas são essenciais para que objetos metálicos e reflexivos espelhem o que está ao redor deles.

Figura 58 - Spheres Reflection Capture da cena virtual.



Fonte: elaborada pela autora (2023).

Para os testes de luz, a iluminação foi carregada apenas como Preview de visualização, pois é a forma mais rápida de processar as mudanças que são feitas na cena. Quando a iluminação final foi decidida, a luz foi carregada e inserida na cena no modo de Produção. O resultado final da cena iluminada pode ser visto na Figura 59 abaixo.

Figura 59 - Resultado final da cena iluminada.

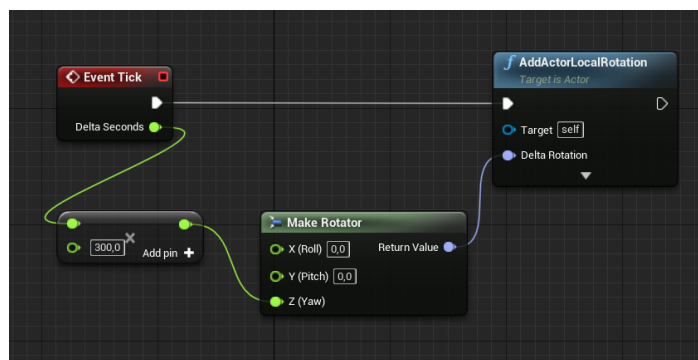


Fonte: elaborada pela autora (2023).

### 3.2.6 Animação

Na Unreal, a etapa de animação foi feita de forma mecânica, através de Blueprints. No caso do ventilador, cada uma das três pás teve uma Blueprint implementada nela, informando que o objeto deve girar em torno do *pivot*. Esse movimento é definido em graus de rotação por segundo, como mostra a Figura 60, em que é possível ver quais nodes foram utilizados para obter o resultado final. A mesma configuração foi aplicada nas três pás para que elas girassem em sincronia a partir de suas respectivas posições iniciais.

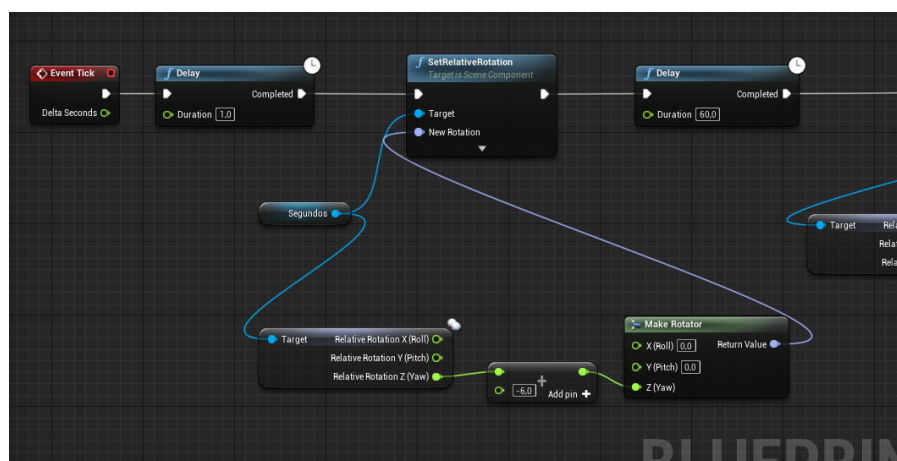
Figura 60 - Modelo do computador com a textura de vídeo aplicada na tela.



Fonte: elaborada pela autora (2023).

Para o relógio, a rotação programada foi mais complexa para que cada ponteiro se comportasse como um mecanismo de verdade. Os ponteiros foram informados que deveriam girar com um certo atraso, que corresponde à contagem de segundos, minutos e horas, em uma rotação definida. Assim, os segundos foram ditos para girar com o atraso de um segundo, os minutos com um atraso de sessenta segundos e as horas com um atraso de 3600 segundos. O começo da Blueprint do relógio, que foi repetida ao longo da mesma, apenas alterando os valores, pode ser vista na Figura 61 abaixo.

Figura 61 - Blueprint do relógio.



Fonte: elaborada pela autora (2023).

### 3.2.7 Inspeção de objeto

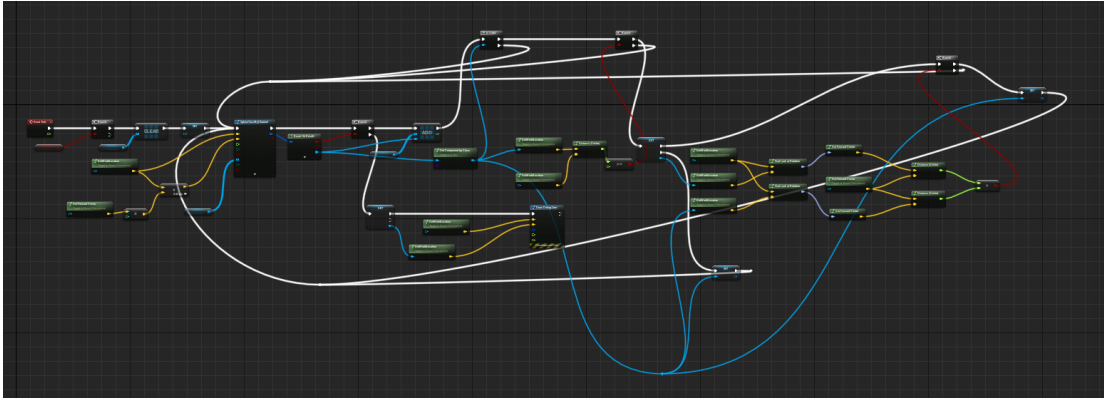
Com os materiais, iluminação e animações prontos, deu-se início a programação do componente “Inspeccionar Objeto”, uma Blueprint personalizada feita para interagir com os modelos da cena. Para montar o componente, um tutorial do YouTube<sup>2</sup> foi seguido passo a passo. A Blueprint final tem três funções principais. A primeira e principal é o Event Graph, visto na Figura 62 abaixo, que inicia-se com um node de Event Tick, montada para permitir que os controles do jogador localizem objetos programados com o componente “Grab Component”, padrão da Unreal, a uma certa distância. Ela também estabelece as condições do que deve acontecer

<sup>2</sup> Disponível em:

[https://www.youtube.com/watch?v=qAHVNN6CHH4&list=PLN6amzd018SI4pZkioMZQvXUX10bHy\\_Sc&index=90&t=802s](https://www.youtube.com/watch?v=qAHVNN6CHH4&list=PLN6amzd018SI4pZkioMZQvXUX10bHy_Sc&index=90&t=802s)

quando esses objetos forem localizados, que no caso é apontar uma linha até o objeto.

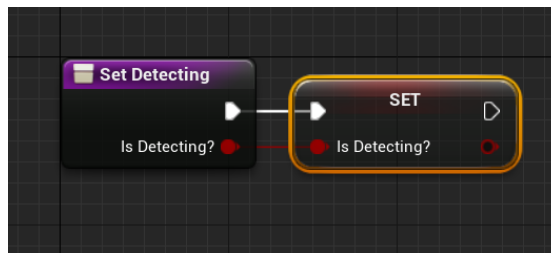
Figura 62 - Event Graph do componente “Inspecionar Objeto”.



Fonte: elaborada pela autora (2023).

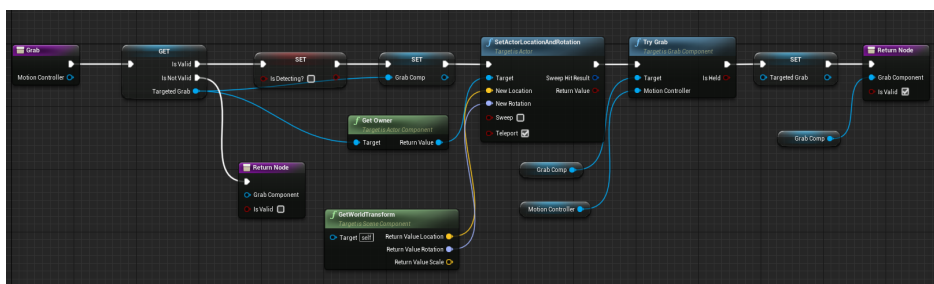
A segunda função, vista na Figura 63, é a “Setar Detecção”, que consiste na criação de uma variável de detecção de objeto, utilizada nas outras duas funções para dizer ao programa o que fazer se a condição de pegar um objeto for verdadeira ou falsa. A terceira e última função é “Agarrar”, vista na Figura 64 abaixo, e ela é responsável por ligar o componente com as funções de movimento e interação dos controles, tornando a Blueprint interativa durante o jogo e permitindo que ela funcione.

Figura 63 - Função “Setar Detecção” do “Inspecionar Objeto”.



Fonte: elaborada pela autora (2023).

Figura 64 - Função “Agarrar” do “Inspecionar Objeto”.



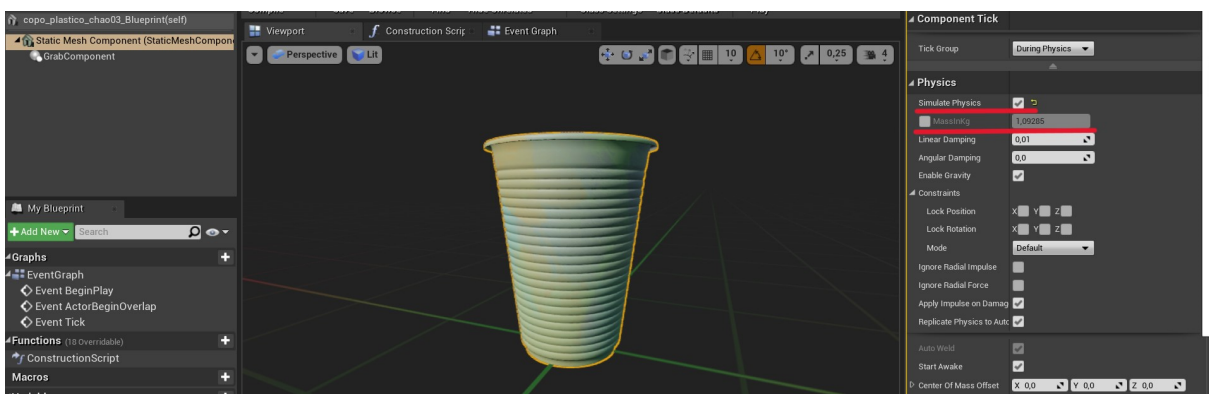
Fonte: elaborada pela autora (2023).



Para finalizar a montagem do componente, os controles do Quest 2 foram programados. Para isso, o “Inspeccionar Objeto” foi filiado aos controles direito e esquerdo do VR Pawn, que é o equivalente ao jogador. Depois, as funções padrões do VR Pawn de agarrar objeto com o controle esquerdo e direito foram modificadas para incluir a função do componente criado, de forma que ao apertar o botão de gatilho dos controles quando a linha aparecer, o objeto seja teletransportado para as “mãos” do jogador. Ao soltar o gatilho, a função é dada como falsa e o objeto cai no chão.

A última etapa dessa programação é programar os objetos para que sejam interativos. Para isso, todos os objetos que escolheu-se que poderiam ser pegos e inspeccionados foram modificados de “Estáticos” para “Móveis”. Depois, para cada um foi criada uma Blueprint e filiados às suas malhas o componente “Inspeccionar Objeto”. Por fim, ligou-se a física das malhas dos modelos 3D e informado o peso de cada um ao programa, para que respondessem apropriadamente à gravidade do jogo quando foram soltos pelo jogador. A Figura 65 abaixo mostra como a Blueprint de cada objeto interativo foi montada.

Figura 65 - Blueprint dos objetos. As configurações de física e de peso estão sublinhadas.



Fonte: elaborada pela autora (2023).

O resultado final é que o jogador, ao chegar próximo o suficiente de um objeto interativo, consegue ver uma linha branca que começa no controle esquerdo, direito ou ambos (se estiverem na distância necessária do modelo) e termina no próprio objeto, como mostra a Figura 66. Se apertar o gatilho e o mantiver apertado, o jogador consegue segurar o objeto, como mostra a Figura 67 abaixo, e pode

movê-lo livremente entre as duas “mãos” ou soltá-lo e vê-lo cair de acordo com sua física programada.

Figura 66 - Demonstração de como funciona a função de agarrar um objeto à distância.



Fonte: elaborada pela autora (2023).

Figura 67 - Demonstração de como funciona segurar um objeto.



Fonte: elaborada pela autora (2023).

### 3.2.8 Mapeamento de normais

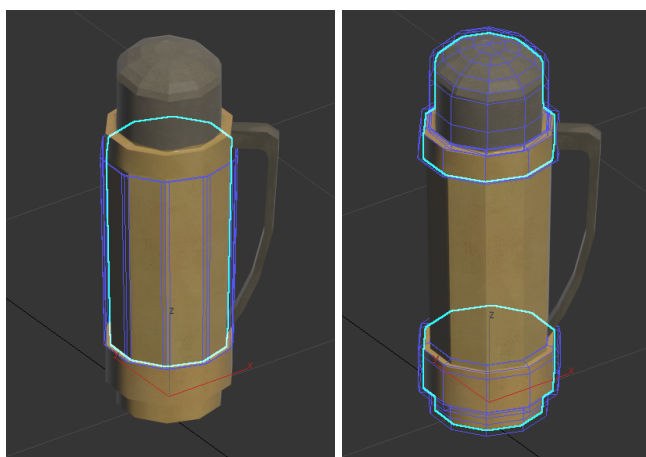
A última etapa do projeto foi feita para obter uma pintura e design mais característicos de alguns dos objetos da cena. Os mapas normais dos modelos com materiais reflexivos foram pintados a partir dos mapas originais, para que suas superfícies ficassem facetadas e texturizadas, como se tivessem sido pintadas manualmente por um pincel.

Para fazer isso, o primeiro passo foi obter os mapas normais dos modelos no programa 3Ds Max, a partir das UVW's feitas anteriormente no projeto. O *software* obtém as normais por meio de duas versões do modelo 3D: a versão *low poly* e a versão *high poly*. A versão *low poly* corresponde ao resultado da modelagem por *box modeling*, com baixo número de polígonos arranjados de forma a representar o formato do objeto. A versão *high poly*, no 3Ds Max, é criada utilizando-se o modificador “TurboSmooth”, que subdivide a malha em mais polígonos, suavizando a superfície do objeto.

Assim, foram feitas cópias dos modelos originais em *low poly* e nelas foram aplicadas o modificador TurboSmooth, obtendo duas versões do mesmo modelo.

Por meio do processo de renderização “Bake to Texture”, uma cápsula do modelo *low poly* é projetada e contém a versão *high poly* do mesmo, como pode ser visto nas Figura 68 e 69 abaixo, utilizando o modelo da garrafa de café como exemplo.

Figuras 68 e 69 - Cápsulas projetadas a partir do modelo *low poly* da garrafa no modelo *high poly*.



Fonte: elaborada pela autora (2023).

Ao selecionar a propriedade que se deseja projetar e renderizar, obtém-se o mapa normal de como a luz atinge o objeto *high poly* no mesmo mapa de textura feito no *unwrap* do modelo, como pode ser visto nas Figura 70 e 71 abaixo. Para realizar a pintura das normais, foi obtido as normais do tipo “*object space*”.

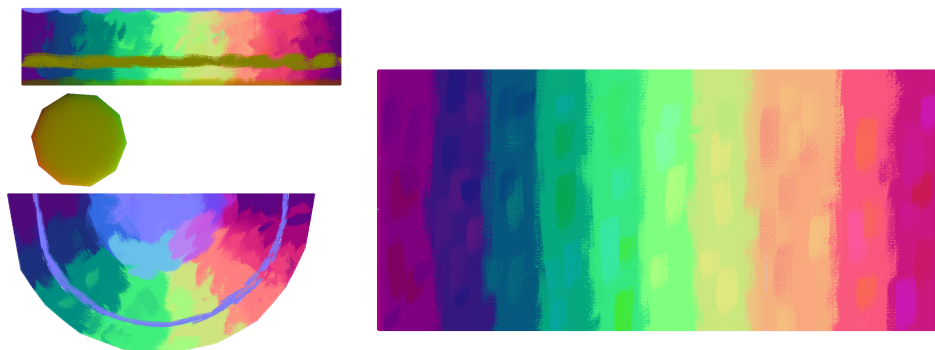
Figuras 70 e 71 - Mapas de normais “*object space*” do modelo da garrafa de café.



Fonte: elaborada pela autora (2023).

Em seguida, os mapas foram abertos no Photoshop e por cima deles, utilizando a ferramenta Color Picker, as áreas coloridas foram pintadas utilizando um brush que simula um pincel, criando facetas texturizadas, como pode ser visto na Figura 72 abaixo.

Figura 72 - Mapa de normal “*object space*” do modelo da garrafa pintada de forma estilizada.



Fonte: elaborada pela autora (2023).

Com as normais pintadas, a imagem é colocada no material do modelo para definir sua normal. A partir daí, o modelo fica multifacetado pois a luz interage de forma estilizada com o mesmo. O modelo da garrafa de café sem a normal aplicada, em comparação com o resultado final do mesmo modelo, com a normal aplicada, pode ser visto nas Figuras 73 e 74 abaixo.

Figura 73 e 74 - Comparação entre a garrafa de café sem e com as normais estilizadas.



Fonte: elaborada pela autora (2023).

Essa estilização pode ser feita em todos os modelos, porém é mais perceptível nos modelos que refletem bastante luz, pois é possível observar melhor as facetas pinceladas. As imagens 75 e 76 abaixo mostram uma comparação entre duas partes do ambiente virtual, a imagem da esquerda com os objetos reflexivos sem as normais aplicadas aos seus materiais e a imagem da direita com as normais aplicadas.

Figura 75 - Comparação entre parte da cena sem e com estilização de normais.



Fonte: elaborada pela autora (2023).

Figura 76 - Comparação entre parte da cena sem e com estilização de normais.



Fonte: elaborada pela autora (2023).

## 4 CONCLUSÃO

O processo de produção do projeto seguiu, no geral, alinhado com o planejamento feito no início de seu desenvolvimento. Foi possível aprender e entender melhor sobre a construção e programação de uma Realidade Virtual e seus requisitos de desempenho em relação à modelos 3D e iluminação. Também foram observadas as novas possibilidades de interações com a pintura em uma nova perspectiva e como isso muda a relação do observador com o quadro.

Algumas adaptações tiveram de ser feitas durante o seguimento do projeto para obter resultados satisfatórios. A primeira delas foi em relação à modelagem dos objetos da cena, em que alguns tiveram que ter sua escala aumentada para que fizesse sentido com as proporções propostas. Como dito no desenvolvimento, também fez-se necessário buscar referências externas para objetos que não podem ser vistos em sua totalidade na pintura, como a mesa da recepcionista e sua cadeira. Outra adaptação necessária foi em relação à pintura dos modelos. Por terem de ter sido pintados manualmente no Photoshop, utilizando a pintura original como referência para cores e estilo de pinceladas, os objetos não ficaram exatamente fiéis ao quadro. Por essas razões, não foi possível recriar a pintura com precisa exatidão em uma cena virtual tridimensional. A representação do quadro “Sala de Espera” em um ambiente digital 3D pode ser vista mais como uma adaptação e uma versão alternativa do mesmo. Sendo mídias diferentes de cenas distintas, haverá diferentes interpretações e níveis de imersibilidade.

O maior desafio do projeto se iniciou com a exportação dos modelos na Unreal Engine, pois não havia experiência prévia por parte da autora no programa. Apesar disso, a parte de criação de materiais e programação, com o auxílio de tutoriais e vídeo aulas, fluiu melhor que o esperado e não houve grandes dificuldades nessas etapas, tendo resultados satisfatórios. Isso também se deve à boa parte da programação da cena já estar pronta dentro do template de Realidade Virtual na Unreal, assim muitos dos controles essenciais praticamente não precisaram de ajustes. A *engine* de jogos foi, dessa forma, bastante intuitiva para uma nova usuária sem experiência.

No entanto, a etapa de iluminação foi muito exaustiva. O controle de luz da Unreal é bastante complexo, portanto foi difícil entender quais parâmetros precisavam ser alterados e quais as melhores luzes para colocar no ambiente para obter a iluminação da pintura. Foi necessário realizar vários testes e mudar

manualmente a densidade de luz de todos os objetos para conseguir soluções com um desempenho leve. No fim, a iluminação não pôde ser a mesma da sugerida no quadro, pois o resultado não era esteticamente agradável e, assim, decidiu-se seguir outra direção para que o ambiente ficasse satisfatório.

A mecânica de pegar objetos a distância também não teve um resultado final condizente com os objetivos do trabalho, pois a linha branca que se formava entre o controle e o modelo 3D era bastante distrativa e dificultava a apreciação do ambiente virtual. Por essa razão, ela foi descartada e é possível interagir com os objetos da cena apenas se o jogador entrar em contato direto com suas malhas. Essa opção foi a mais viável, pois, apesar de mais trabalhosa, incentiva a exploração e torna a experiência de estar na pintura mais convincente, já que imita a que o usuário da RV teria em uma sala de verdade.

Em síntese, o resultado final do projeto mostrou-se uma alternativa viável para apreciação de pinturas. A cena virtual 3D não substitui sua inspiração original, mas incentiva a ressignificação do quadro “Sala da Espera”. A proposta de exploração do ambiente, observação das propriedades dos objetos em tempo real, contemplação dos objetos em perspectivas que não podem ser vistas na imagem bidimensional e interação com os mesmos utilizando os controles manuais é imersiva e diverte o usuário como um jogo. A apreciação da obra original permite um olhar mais atento ao estilo do artista e a sensação de nostalgia e familiaridade brasileira que a pintura expõe, enquanto que a apreciação da cena tridimensional sugere uma imersão mais dinamizada e tátil, ambas perspectivas válidas para admirar e refletir sobre a arte.



## REFERÊNCIAS

ANDALÓ, F.; VIEIRA, M. L. H.; MERINO, E.. Iluminando Objetos 3D: Iluminação Tradicional versus Iluminação Realista. **Design & Tecnologia**, Florianópolis, v. 5, n. 02, p. 44-54, fev. 2010. Disponível em: [https://www.researchgate.net/publication/313547249\\_Iluminando\\_objetos\\_3D\\_iluminacao\\_tradicional\\_versus\\_iluminacao\\_realista](https://www.researchgate.net/publication/313547249_Iluminando_objetos_3D_iluminacao_tradicional_versus_iluminacao_realista). Acesso em: 12 abr. 2023.

ANTONIETTI, Alessandro; CANTOIA, Manuela; TSAI, Tsun-Hung. To see a painting versus to walk in a painting: an experiment on sense-making through virtual reality. **Computers & Education**, Milão, v. 3, n. 8, p. 213-223, abr. 2000. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0360131599000469>. Acesso em: 07 out. 2023.

ALVES, Francielle da Cruz Medeiros. **Uma proposta para utilização dos óculos de realidade virtual no processo de ensino-aprendizagem**. 2017. 92 f. TCC (Graduação) - Curso de Tecnologia da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2017. Disponível em: <https://repositorio.ufsc.br/handle/123456789/181875>. Acesso em: 27 mar. 2023.

BERNARDO, Gustavo de Almeida. **Iluminação exterior de ambientes frios em animação 3D**. 2010. 54 f. Dissertação (Mestrado) - Curso de Som e Imagem, Escola das Artes da Universidade Católica Portuguesa, Porto, 2010. Disponível em: <https://ciencia.ucp.pt/pt/studentTheses/ilumina%C3%A7%C3%A3o-exterior-de-ambientes-frios-em-anima%C3%A7%C3%A3o-3d>. Acesso em: 12 abr. 2023.

FERNANDÉZ, M. (2011). **Modelado, texturizado y ajuste de malla**. Madrid: E-Archivos Universidad Carlos III de Madrid.

FREITAS, Igor Flores. **Criação de Iluminação para Jogo 3D com Foco em Narrativa**. 2020. 101 f. TCC (Graduação) - Curso de Design, Universidade Federal de Santa

Catarina, Florianópolis, 2020. Disponível em: <https://repositorio.ufsc.br/handle/123456789/218802>. Acesso em: 12 abr. 2023.

GOBIRA, Pablo; MOZELLI, Antônio. As Interfaces da Realidade Virtual no Século XXI. **Z Cultural**, Belo Horizonte, v. 3, n. 8, p. 13-16, mar. 2016. Disponível em: <http://revistazcultural.pacc.ufrj.br/as-interfaces-de-realidade-virtual-no-seculo-xxi-tecnologia-jogos-digitais-e-industria-1/#:~:text=Os%20avan%C3%A7os%20tecnol%C3%B3gicos%20das%20interfaces,para%20o%20p%C3%ABlico%20em%20geral..> Acesso em: 23 mar. 2023.

GRASNICK, Armin. **Basics of Virtual Reality**: from the discovery of perspective to vr glasses. Berlin: Springer, 2021.

LAVALLE, Steven M.. **Virtual Reality**. Cambridge: Cambridge University Press, 2017.

HO, Li-Hsing; SUN, Hung; TSAI, Tsun-Hung. Research on 3D Painting in Virtual Reality to Improve Students' Motivation of 3D Animation Learning. **Sustainability**, Taipei, v. 11, n. 6, p. 11-24, 16 mar. 2019. MDPI AG. <http://dx.doi.org/10.3390/su11061605>. Disponível em: <https://www.mdpi.com/2071-1050/11/6/1605>. Acesso em: 05 out. 2023.

VILAR, Elisângela; FILGUEIRAS, Ernesto; REBELO, Francisco. **Virtual and Augmented Reality for Architecture and Design**. Lisboa: Crc Press, 2022.

KIRNER, Claudio; SISCOOTTO, Robson; REBELO, Francisco. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações**. Petrópolis: Pré-Simpósio, 2007.

LIN, Chih-Long; CHEN, Si-Jing; LIN, Rungtai. Efficacy of Virtual Reality in Painting Art Exhibitions Appreciation. **Applied Sciences**, New Taipei, v. 3, n. 8, p. 1-15, mar. 2020. Disponível em: <https://www.mdpi.com/2076-3417/10/9/3012>. Acesso em: 05 out. 2023.

MACEDO, Daniel Valente de; SERPA, Yvens Rebouças; RODRIGUES, Maria Andréia Formico. Desenvolvimento de Aplicações Gráficas Interativas com a Unreal Engine 4. **Rita**, Belo Horizonte, v. 2, n. 22, p. 181-202, out. 2015. Disponível em:

MORIE, Jacquelyn Ford. Inspiring the future. **Acm Siggraph Computer Graphics**, [S.L.], v. 28, n. 2, p. 135-138, maio 1994. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/178951.178973>. Disponível em: <https://dl.acm.org/doi/10.1145/178951.178973>. Acesso em: 28 mar. 2023.

TORI, Romero; KIRNER, Claudio; SISCOUTTO, Robson. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém: Pré-Simpósio, 2006.

NETTO, Antônio Valério; MACHADO, Liliane dos Santos; OLIVEIRA, Maria Cristina Ferreira de. **Realidade Virtual - Definições, Dispositivos e Aplicações**. 2010. 33 f. TCC (Graduação) - Curso de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Paulo, 2010. Disponível em: [http://www.de.ufpb.br/~labteve/publi/2002\\_reic.pdf](http://www.de.ufpb.br/~labteve/publi/2002_reic.pdf). Acesso em: 31 mar. 2023.

NUNO MATEUS COELHO. **Realidade Virtual "Estado da Arte"**. 2016. 12 f. Tese (Doutorado) - Curso de Informática, Universidade de Trás-Os-Montes e Alto Douro, Vila Real, 2016. Disponível em: [https://www.researchgate.net/publication/307010600\\_Realidade\\_Virtual\\_Estado\\_da\\_Arte\\_Resumo\\_Vila\\_Real\\_-\\_Portugal](https://www.researchgate.net/publication/307010600_Realidade_Virtual_Estado_da_Arte_Resumo_Vila_Real_-_Portugal). Acesso em: 28 mar. 2023.

OLIVEIRA, Gilberto Martini de; MACHADO, Liliane dos Santos; OLIVEIRA, Maria Cristina Ferreira de. **Realidade virtual em apresentação de projeto arquitetônico**. 2016. 82 f. TCC (Doutorado) - Curso de Design, Universidade Federal de Santa Catarina, Florianópolis, 2016. Disponível em: <https://repositorio.ufsc.br/handle/123456789/173189>. Acesso em: 15 abr. 2023.

SMÍD, Antonín. **Comparison of Unity and Unreal Engine**. 2017. 77 f. TCC (Doutorado) - Curso de Computação Gráfica, Czech Technical University In Prague, Praga, 2017. <https://seer.ufrgs.br/index.php/rita/article/view/RITA-VOL22-NR2-181>. Acesso em: 05 abr. 2023.

SOUZA, Carla Florentino de; ARAUJO, Edson Leite; LEE, Jennifer Chuin. **Mapeamento de Textura Uma Introdução**. 2011. 26 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 2011. Disponível em: <https://www.dca.fee.unicamp.br/courses/IA725/1s2011/projetos/lee-araujo-souza/downloads/Monografia.pdf>. Acesso em: 11 abr. 2023.