

UNIVERSIDADE FEDERAL DO PAMPA

Mariana Toledo Costa

**Implementação de uma ferramenta para  
modelagem 3D simplificada**

Alegrete  
2023

Mariana Toledo Costa

# Implementação de uma ferramenta para modelagem 3D simplificada

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

Alegrete  
2023

**Mariana Toledo Costa**

## **Implementação de uma ferramenta para modelagem 3D simplificada**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 06 de julho de 2023.

Banca examinadora:

---

Prof. Dr. Marcelo Resende Thielo  
Orientador  
UNIPAMPA

---

Profa. Dra. Aline Vieira de Mello  
UNIPAMPA

---

Prof. Me. Jean Felipe Patikowski Cheiran  
UNIPAMPA



Assinado eletronicamente por **ALINE VIEIRA DE MELLO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 06/07/2023, às 20:46, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **MARCELO RESENDE THIELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 03:01, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **JEAN FELIPE PATIKOWSKI CHEIRAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 09:41, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1172058** e o código CRC **F40A485F**.

---

Este trabalho é dedicado aos meus pais,  
pelo incentivo e apoio que recebi  
por todos esses anos.

## **Agradecimentos**

Primeiramente, agradeço meus pais, mãe Thelma Toledo de Souza e ao meu pai, Antonio Costa, que sempre estiveram ao meu lado, apoiando e incentivando minhas decisões. Aos colegas que me acompanharam durante estes anos de formação acadêmica, devo meus agradecimentos. Se um deles não estivesse presente, acredito que alguma parte minha estaria faltando hoje. Portanto, meus agradecimentos especiais a Rafael de Oliveira Jarczewski, Douglas Vinícius Ledur Dullius, Gustavo Lopes Tamiosso, Uéslei Bervanger Brandt, Mágino Antero Corrêa Leguissamo Filho e Vitor Xavier Siqueira. Que, para além dos estudos, trouxeram conhecimento que moldou minha visão sobre o mundo e somou ao meu crescimento pessoal, presente esse que levarei para onde eu for. Agradeço também ao professor Marcelo Resende Thielo que, mesmo pelo curto período em que foi meu orientador, acrescentou muito aos meus conhecimentos técnicos e, em tempos de reclusão, sempre trouxe bom humor para as aulas, algo que sempre me ajudou muito.

“A morte de um ser humano me empobrece  
porque é um universo único e irrepetível  
que deixei de travar contato“  
(Marco Aurélio)

## Resumo

A modelagem tridimensional, essencial para uma ampla gama de setores, desde a engenharia e arquitetura até a animação e jogos, depende fortemente do uso de softwares especializados. No entanto, a utilização destes programas apresenta desafios significativos tanto para profissionais quanto para estudantes na área. Estes desafios geralmente surgem de duas restrições principais: a necessidade de alto poder de processamento para operar estes softwares e o custo associado à aquisição e manutenção das licenças de software. Essas barreiras podem impedir o acesso e limitar a experimentação e inovação na modelagem 3D, especialmente para estudantes. Nesse contexto, os ambientes *web* surgem como uma solução promissora para essas restrições. A proposta deste trabalho é introduzir uma ferramenta que auxilie na modelagem de objetos em ambiente *web*, de forma que seja acessível e portátil, sem depender de nenhuma plataforma específica para gerenciar e modelar objetos. Como principal resultado obtido nesse trabalho, temos uma ferramenta de modelagem que contém algumas funcionalidades, tais como importar e exportar, assim como a funcionalidade de mover vértices a partir de um ponto definido pelo usuário.

**Palavras-chave:** Modelagem 3D. Computação gráfica. Web.



## Abstract

Three-dimensional modeling, essential for a wide range of sectors, from engineering and architecture to animation and gaming, heavily relies on the use of specialized software. However, the use of these programs presents significant challenges for both professionals and students in the field. These challenges typically arise from two main constraints: the need for high processing power to operate these softwares and the cost associated with acquiring and maintaining software licenses. These barriers can limit access experimentation and innovation in 3D modeling, especially for students. In this context, web environments emerge as a promising solution to these constraints. Web technology offers the possibility to circumvent these barriers. The proposal of this work is to introduce a tool that assists in the modeling of objects in a web environment, in a way that is accessible and portable, without depending on any specific platform to manage and model objects. As the main result of this work, we have a modeling tool that includes some functionalities, such as importing and exporting, as well as the feature of moving vertices from a point defined by the user.

**Key-words:** 3D modeling. Computer graphics. Web.

## Lista de figuras

Figura 1 – Interface de usuário Blender . . . . .	16
Figura 2 – Interface de usuário ZBrush . . . . .	17
Figura 3 – Interface de usuário Sculptris . . . . .	18
Figura 4 – Operações de CSG . . . . .	19
Figura 5 – Varredura Translacional . . . . .	20
Figura 6 – Varredura Rotacional . . . . .	20
Figura 7 – Interface do protótipo . . . . .	25
Figura 8 – Interface do protótipo . . . . .	25
Figura 9 – Interface do menu de modificação de parâmetros . . . . .	26
Figura 10 – Exemplo da utilização da seta . . . . .	27
Figura 11 – Função de criar a seta . . . . .	29
Figura 12 – Resultado do pincel ao ser aplicado em toda a área do objeto . . . . .	30
Figura 13 – Resultado do pincel ao ser aplicado em toda a área do objeto com uma força inversa . . . . .	31
Figura 14 – Resultado do pincel aplicado com um força inversa localmente em uma área do objeto . . . . .	31
Figura 15 – Resultado do pincel ao ser aplicado parcialmente . . . . .	31
Figura 16 – Resultado do redimensionamento no modelo . . . . .	32
Figura 17 – Resultado do pincel ao ser aplicado em apenas um vértice . . . . .	32

## Lista de abreviaturas

**2D** Bidimensional

**3D** Tridimensional

**API** *Application Programming Interface*

**CSG** *Constructive Solid Geometry*

**GUI** *Graphic User Interface*

**VR** *Virtual Reality*

## Sumário

1	INTRODUÇÃO . . . . .	12
1.1	Organização do trabalho . . . . .	13
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	14
2.1	Computação gráfica . . . . .	14
2.2	Softwares de modelagem tridimensional . . . . .	14
2.3	Aprendizado na modelagem digital tridimensional . . . . .	15
2.4	Portabilidade . . . . .	15
2.5	Análise de interfaces principais em softwares de modelagem 3D	16
2.6	Componentes comuns em softwares de modelagem tridimen- sional . . . . .	18
2.6.1	Técnicas de modelagem . . . . .	18
2.6.1.1	CSG . . . . .	18
2.6.1.2	Varredura . . . . .	18
2.6.1.3	Instanciação de primitivas . . . . .	19
2.6.2	Ferramentas para modelagem digital . . . . .	20
2.6.2.1	Pincéis de modelagem . . . . .	20
2.6.3	Visão de câmera . . . . .	21
3	TRABALHOS RELACIONADOS . . . . .	22
4	DESENVOLVIMENTO . . . . .	24
4.1	Definição do funcionamento da ferramenta . . . . .	24
4.1.1	Abordagem inicial . . . . .	24
4.1.2	Configuração da cena . . . . .	24
4.1.2.1	Funcionamento da câmera . . . . .	24
4.1.3	Interface do ambiente . . . . .	25
4.1.4	Funcionalidades da interface . . . . .	25
4.1.5	Funcionalidades e alterações básicas do objeto . . . . .	26
4.1.6	Parâmetros de ajuste da ferramenta de modelagem . . . . .	27
4.1.7	Ferramenta de modelagem . . . . .	27
4.1.7.1	Seta auxiliar . . . . .	27
4.1.7.2	Pincel de modelagem . . . . .	28
5	RESULTADOS . . . . .	30
5.1	Pincel de modelagem . . . . .	30
5.2	Aplicações utilizando toda a área do objeto . . . . .	30
5.3	Aplicações utilizando apenas uma área específica do objeto .	31
5.4	Aplicações de ferramentas para redimensionamento do objeto	32

5.5	Aplicações utilizando apenas um vértice do objeto . . . . .	33
6	CONSIDERAÇÕES FINAIS . . . . .	34
	REFERÊNCIAS . . . . .	35

## 1 Introdução

Em 1972, Edwin Catmull e Fred Parke deram início à computação gráfica tridimensional ao produzir a curta-metragem “A Mão Animada por Computado” (CATMULL; PARKE, 1972). Esse marco histórico apresentou a animação computadorizada ao público daquela época, sendo possível graças aos polígonos e triângulos desenhados por Edwin em sua própria mão, que foram digitalizados e utilizados como base para a animação.

Somente em 1976, o primeiro filme incorporou uma parte da curta-metragem, exibindo a animação da mão desenvolvida por Edwin. Este filme, intitulado “*Futureworld*” (HEFFRON et al., 1976), tornou-se o primeiro longa-metragem a usar gráficos 3D computadorizados, não apenas incorporando a animação da mão de Catmull, mas também incluindo um modelo de cabeça criado com as mesmas metodologias de Catmull. Este uso inovador da tecnologia de animação computadorizada marcou um ponto de virada importante na indústria cinematográfica, abrindo caminho para futuros avanços no campo da animação e efeitos visuais.

Após esses avanços, a modelagem tridimensional passou a desempenhar um papel significativo em diversas áreas do mercado, incluindo a indústria cinematográfica e publicitária, permitindo que profissionais criassem personagens e cenários detalhados e realistas para uma variedade de meios. Além disso, revolucionou outras indústrias, como a de jogos, onde a modelagem 3D e a animação computadorizada contribuíram para a criação de ambientes de jogo imersivos e personagens visualmente complexos.

Com uma ampla variedade de aplicações, a modelagem tridimensional é utilizada em simulações científicas e visualizações arquitetônicas, permitindo a cientistas e arquitetos compreender melhor e comunicar suas ideias. Com um leque abrangente de possibilidades, a aplicação da modelagem tridimensional no desenvolvimento *web* ganha destaque, permitindo o *design* de sites com elementos tridimensionais para melhorar a experiência do usuário.

A inauguração da modelagem 3D para aplicações *web* ocorreu em 2011 com a introdução do *WebGL*, uma API baseada em *OpenGL* (GAVANE, 2019). O *WebGL* impulsionou a utilização de elementos tridimensionais, especialmente no *design* de sites, inaugurando uma nova era de *design UI/UX* em 3D.

Neste trabalho, nosso foco está no desenvolvimento de uma ferramenta de modelagem tridimensional simplificada utilizando o *Three.js* (CABELLO, 2021), uma biblioteca *JavaScript* baseada em *WebGL*. O objetivo do trabalho é fornecer uma ferramenta de modelagem tridimensional que envolve conceitos de portabilidade, permitindo que a ferramenta seja acessível ao usuário de qualquer plataforma desde que o usuário tenha um navegador atualizado.

A portabilidade e versatilidade de softwares de modelagem 3D são importantes, especialmente em casos em que os artistas não têm acesso a sistemas operacionais que suportam softwares exclusivos.

Uma das inspirações para o desenvolvimento deste trabalho é o software Sculptris, que por muito tempo foi um software de modelagem 3D gratuito fornecido pela Pixologic. Embora tenha algumas limitações em comparação com o ZBrush (um software de modelagem 3D comercial também fornecido pela Pixologic), o Sculptris era uma ótima introdução à modelagem. No entanto, após quase uma década sem atualizações, o Sculptris foi descontinuado e incorporado a um pacote de licença comercial, sendo renomeado como Sculptris Pro (MARCUS, 2018)<sup>1</sup>.

Com base nessa referência, o objetivo principal deste trabalho é introduzir uma ferramenta que combine características de portabilidade e facilidade de uso em todas as plataformas. Um dos propósitos deste trabalho é proporcionar uma ferramenta totalmente portátil e acessível. O público-alvo consiste em artistas 3D e entusiastas da área, incluindo estudantes que tenham interesse em aprender computação gráfica.

## 1.1 Organização do trabalho

O trabalho está organizado como segue:

- **Capítulo 2 Fundamentação teórica:** introduz os principais fundamentos e conceitos utilizados neste trabalho;
- **Capítulo 3 Trabalhos relacionados:** são apresentados alguns conceitos importantes para o desenvolvimento deste trabalho;
- **Capítulo 4 Desenvolvimento:** apresenta mostra o as etapas que foram necessárias para desenvolvimento do nosso trabalho;
- **Capítulo 5 Resultados:** apresenta os resultados que conseguimos com o nosso protótipo atual;
- **Capítulo 6 Considerações finais:** é apresentado algumas conclusões sobre o nosso trabalho e definido os trabalhos futuros.

---

<sup>1</sup> <<http://docs.pixologic.com/user-guide/3d-modeling/modeling-basics/creating-meshes/sculptris-pro/>>

## 2 Fundamentação teórica

Neste capítulo serão apresentados os principais conceitos e técnicas utilizados para o desenvolvimento deste trabalho.

### 2.1 Computação gráfica

A Computação Gráfica é a subárea da Ciência da Computação que trata da produção e representação de informação através de imagens e vídeos, utilizando-se de recursos computacionais. A Computação Gráfica emprega várias técnicas que permitem a geração de imagens de objetos, objetos imaginários ou modelos computacionais (MANSSOUR; COHEN, 2006).

A modelagem tridimensional é um dos ramos mais importantes da Computação Gráfica. Dentre as técnicas mais comuns, encontram-se a modelagem baseada em *Nurbs*, modelagem baseada em sólidos e modelagem baseada em malhas. *Nurbs* é a técnica amplamente utilizada para representar curvas e superfícies suaves, permitindo a criação de formas complexas e precisas através da manipulação de pontos de controle e funções matemáticas. Sua versatilidade é empregada em áreas como design industrial, animação por computador, modelagem 3D e CAD. A modelagem baseada em sólidos é conhecida como modelagem booleana, na qual geralmente os objetos são primitivos, como, por exemplo, cubos e esferas que são combinados através de operações booleanas (união, diferença e interseção) para criar formas complexas e realistas. Já a modelagem baseada em malhas utiliza polígonos interconectados para representar as superfícies de um objeto. Nesta abordagem, os modelos 3D são criados a partir da manipulação direta dos vértices.

### 2.2 Softwares de modelagem tridimensional

Alguns softwares oferecem um nível de complexidade maior quando se trata de modelagem 3D, impactando na curva de aprendizado do usuário, tornando-a mais desafiadora. Ao mesmo tempo, existem opções mais simples que oferecem uma curva de aprendizado menor. Porém, em alguns desses casos, podem haver limitações para trabalhar com aspectos mais avançados, o que pode afetar diretamente a qualidade do modelo, como é apresentado por Sarstedt (2012). As várias ferramentas de modelagem disponíveis no mercado, as mais populares incluem:

**ZBrush:** De acordo com Pixologic (2023), o *ZBrush* é um software de modelagem tridimensional que é licenciado tanto academicamente quanto comercialmente. Ele é bastante completo em relação à documentação e possui uma ampla variedade de ferramentas de modelagem. É destacado como a melhor opção para fotorrealismo, fornecendo alta qualidade em termos de renderização, iluminação e sombreamento. Considerado o melhor software pago do mercado, é mais utilizado por empresas de grande porte. O *ZBrush*, no entanto, não possui suporte para todas as plataformas, sendo compatível apenas com



*Windows* e *MacOS*. Além disso, há requisitos de hardware que podem ser considerados mais exigentes para que seja possível utilizá-lo sem gargalos de recursos. Isso acaba tornando o *ZBrush* inacessível para uma parcela dos usuários.

**Blender:** De acordo com Community (2023a), o *Blender* é um software de licença livre e de código aberto que possui uma grande variedade de ferramentas e recursos. Ao contrário do *ZBrush*, que possui recursos focados apenas na modelagem, o *Blender* também conta com texturização, animação, renderização gráfica e composição, oferecendo um ambiente completo para empresas de pequeno porte. Ele suporta a maioria das plataformas, sendo compatível com *MacOS*, *Windows* e distribuições Linux.

O diferencial da nossa ferramenta em relação ao *ZBrush* e ao *Blender* reside na portabilidade que nossa solução oferece, além de não exigir muito poder computacional. Um dos objetivos da nossa ferramenta é ter uma curva de aprendizado menor em relação às ferramentas citadas acima.

### 2.3 Aprendizado na modelagem digital tridimensional

A utilização de softwares para a modelagem 3D tornou-se amplamente difundida no desenvolvimento de jogos e animações. A modelagem 3D proporciona uma estética visualmente atrativa e versátil, permitindo a criação de ambientes virtuais imersivos e personagens realistas. Assim como o estilo clássico de *pixel art*, que possui uma identidade própria, o uso de modelos 3D tem conquistado um público-alvo cada vez maior, graças à sua capacidade de criar experiências mais dinâmicas.

Essas ferramentas apresentam níveis de complexidade diferentes, o que implica que alguns *designers* precisam ter um maior conhecimento e prática, dependendo do software de modelagem tridimensional no qual optaram por se especializar.

De acordo com o estudo apresentado por Sarstedt (2012), há uma comparação entre dois softwares distintos, levando em consideração o tempo de modelagem e a qualidade do resultado. Os resultados sugerem que a qualidade do modelo e o tempo utilizado para a modelagem são proporcionais entre os dois softwares.

Algumas abordagens incluem o uso de óculos de realidade virtual (*Virtual Reality (VR)*) para tornar o ambiente mais envolvente e atrativo para o usuário.

Contudo, independentemente do método utilizado, uma boa curva de aprendizado na modelagem beneficia tanto o interesse de quem quer ingressar na área como alunos em áreas de engenharia e matemática, bem como a quem gostaria de utilizar os conhecimentos em *hobbies*, como sugere o estudo de Bailey e Law (2014).

### 2.4 Portabilidade

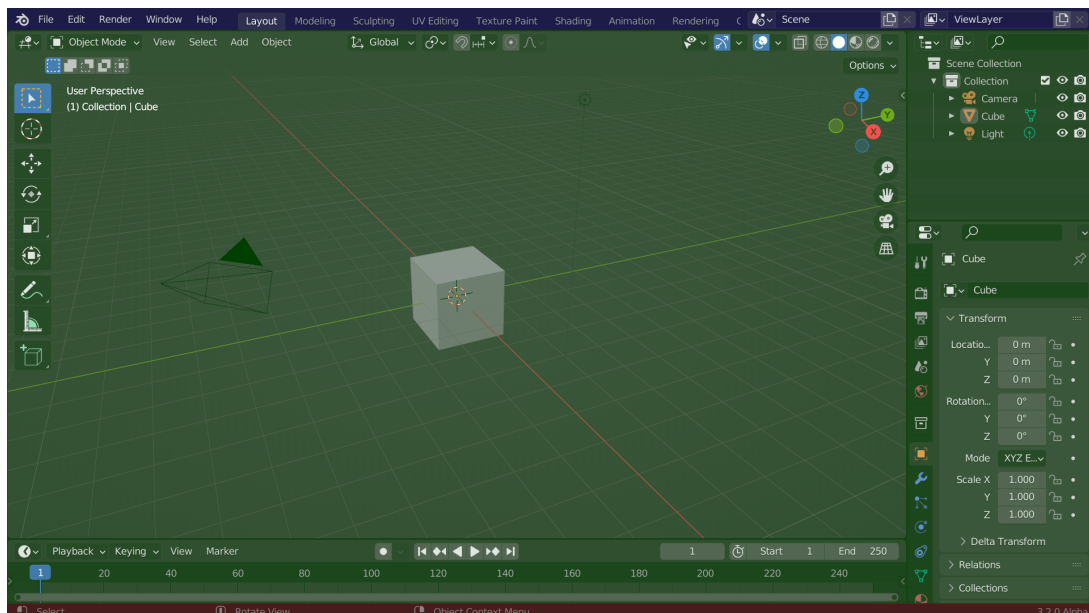
A portabilidade (International Organization for Standardization, 2011) cada vez mais se mostra essencial nas ferramentas de modelagem tridimensional, permitindo que os

artistas acessem e trabalhem em seus projetos de qualquer lugar. O *Blender* e o *ZBrush* são exemplos conhecidos nesse campo. Enquanto o *Blender* é um software gratuito e de código aberto, o *ZBrush* é especializado em escultura digital. Embora o *Blender* seja bastante portátil, estando disponível para diferentes sistemas operacionais, o *ZBrush* é mais restrito, funcionando apenas em *Windows* e *macOS*. No entanto, a modelagem tridimensional na web oferece uma solução de portabilidade, permitindo que os artistas trabalhem em projetos 3D diretamente no navegador, independentemente do dispositivo. A modelagem em ambiente web se mostrou revolucionária, oferecendo várias oportunidades e possibilidades para *designers* e artistas. Com essa abordagem, os usuários não sofrem restrições quanto à plataforma utilizada, desta forma, necessitando apenas de um navegador mais atualizado para realizar as interações.

## 2.5 Análise de interfaces principais em softwares de modelagem 3D

A interface gráfica ou *Graphic User Interface* (GUI) é o ponto-chave para a comunicação com o usuário. Em outras palavras, a interface é responsável por proporcionar um ambiente organizado para o conforto do usuário por meio de diversos fatores. Nesta seção, serão discutidas e apresentadas algumas interfaces de ferramentas de modelagem 3D. Dentre as interfaces disponíveis no mercado selecionamos as interfaces dos softwares *Blender*, *ZBrush* e *Sculptris*.

Figura 1 – Interface de usuário Blender

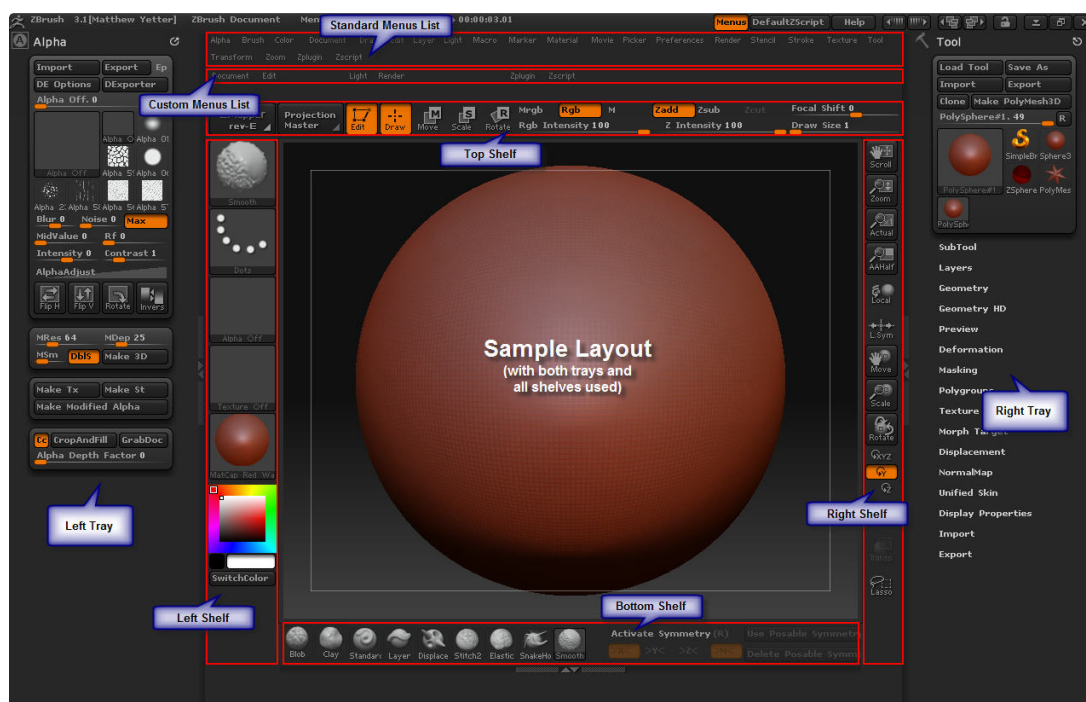


Fonte: Community (2023b)

- **Blender:** A interface do *Blender* pode ser observada através da Figura 1, onde a interface é separada em três partes principais: a barra superior ou *Topbar* (representada pela área demarcada de azul), onde temos o menu principal. Esse menu

abriga opções de salvar, importar, exportar, definições de configuração, renderização, dentre outras funções integradas ao *Blender*, como *shading* e animação. Além disso, temos a área do meio ou *Areas* (marcada em verde), que designa o espaço de trabalho, englobando áreas para os editores, como o Tridimensional (3D) *Viewport* e *Outliner*. Cada um destes editores desempenha uma funcionalidade diferente. Por último, temos a barra de status ou *Status Bar* (demarcada em vermelho), que exhibe sugestões de atalho, mensagens e estatísticas relevantes.

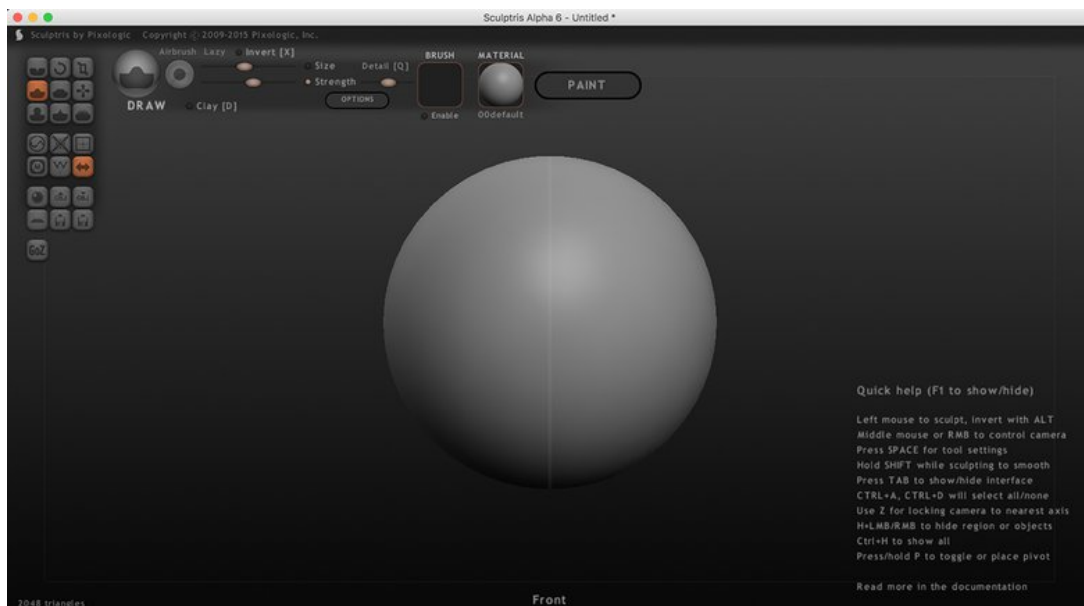
Figura 2 – Interface de usuário ZBrush



Fonte: GMBH (2023)

- **ZBrush:** O *ZBrush* apresenta uma interface mais complexa se comparada à do *Blender*, como pode ser observado na Figura 2. Nesta interface, existem diversas áreas, sendo algumas delas: *Trays* – espaço que contém os menus, os quais podem ser movidos entre as *trays*; *Shelves* – áreas expansíveis que estão ao redor do *canvas* e contêm configurações das ferramentas de modelagem que também podem ser movidas; e o *Custom menu list* ou menu personalizável, localizado acima da barra de status. Nele, podem-se observar algumas opções gerais, como, por exemplo, opções de ferramentas de modelagem e de cor.
- **Sculptris:** A interface do *Sculptris*, como pode ser observada na Figura 3, possui um menu simples, com algumas ferramentas de modelagem básicas que podem ser visualizadas no canto superior esquerdo da Figura. Há também um menu de customização da ferramenta e do modelo, localizado no canto superior. Outros detalhes, como a quantidade de triângulos presentes no modelo, podem ser observados no canto inferior esquerdo.

Figura 3 – Interface de usuário Sculptris



Fonte: Hatton (2019)

## 2.6 Componentes comuns em softwares de modelagem tridimensional

Nesta seção, serão apresentados alguns conceitos comuns entre softwares de modelagem tridimensional que são de suma importância para o desenvolvimento deste trabalho.

### 2.6.1 Técnicas de modelagem

Nesta seção, serão apresentadas brevemente algumas técnicas de modelagem geométrica amplamente utilizadas.

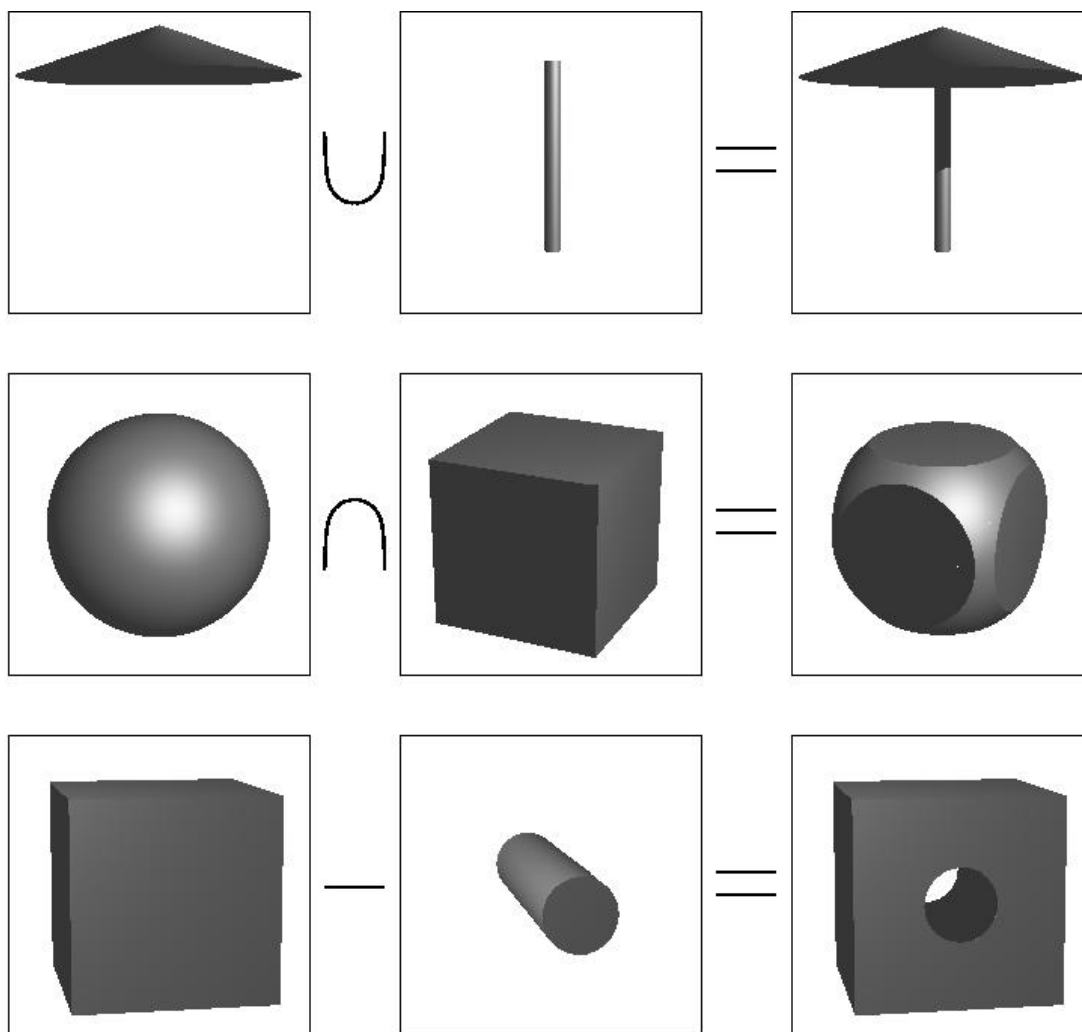
#### 2.6.1.1 CSG

Uma das técnicas mais conhecidas para a modelagem é a *Constructive Solid Geometry (CSG)*, ou Geometria Sólida Construtiva, que pode ser observada na Figura 4. Essa técnica utiliza sólidos mais simples para a composição de sólidos mais complexos. A composição destes objetos é feita através de operadores *booleanos*: união, diferença e intersecção. Temos, por exemplo, na primeira linha, a operação de união, na qual o objeto resultante se torna a junção dos dois objetos que sofreram a operação. Na segunda linha, o objeto resultante é a intersecção da esfera e do cubo, enquanto na terceira linha, subtrai-se o segundo objeto do primeiro, resultando em uma abertura no primeiro objeto.

#### 2.6.1.2 Varredura

A varredura translacional consiste em um objeto (geralmente bidimensional) e uma trajetória definida; desta forma, o sólido é gerado pela varredura da trajetória. A varredura pode ser dividida em dois tipos: extrusão e varredura rotacional.

Figura 4 – Operações de CSG



Fonte: Endo (2007)

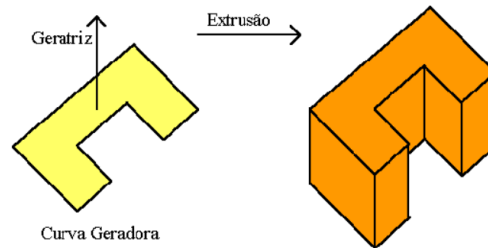
A extrusão ou varredura translacional é representada pela Figura 5, onde vemos a geratriz, que indica a direção da extrusão, e o resultado pode ser visto na segunda imagem desta Figura.

Há ainda a varredura rotacional, que pode ser observada na Figura 6, onde uma face (face A) é o ponto de partida e, diferentemente da extrusão, a trajetória é circular.

### 2.6.1.3 Instanciação de primitivas

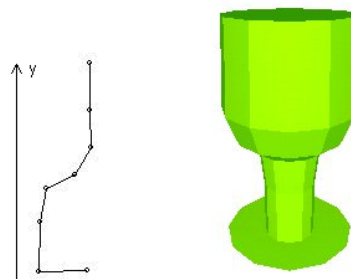
Através de formas primitivas, como, por exemplo, cubo, esfera e cone, são gerados modelos mais complexos. A vantagem de se utilizar essa técnica é a facilidade e precisão na criação de modelos mais complexos utilizando formas mais simples. Há ainda técnicas mais avançadas de instanciamento de primitivas, como, por exemplo, a geração procedural que, através de regras pré-definidas, permite gerar um modelo de forma automatizada por meio de um algoritmo.

Figura 5 – Varredura Translacional



Fonte: Pinho (2023b)

Figura 6 – Varredura Rotacional



Fonte: Pinho (2023a)

## 2.6.2 Ferramentas para modelagem digital

Nesta seção, serão apresentadas algumas tecnologias atuais que são utilizadas no desenvolvimento do projeto.

### 2.6.2.1 Pincéis de modelagem

Nos pincéis de modelagem padrão, da maioria dos softwares de modelagem, pode ser necessária a utilização de técnicas (DUNN; PARBERRY, 2011), como combinação linear, para realizar alguns dos seguintes procedimentos:

- **Orientação do pincel:** Posicionar e orientar o pincel no espaço tridimensional, utilizando técnicas de escala, rotação e translação;
- **Efeito do pincel sob o modelo:** Operação que pode definir a direção e a força aplicadas ao modelo. É necessário fazer adição de vetores, produtos escalares e produtos vetoriais;
- **Interpolação das alterações feitas pelo pincel:** Utilizado principalmente para suavização das alterações feitas pelo pincel na malha. Pode-se utilizar técnicas como interpolação linear ou interpolação baricêntrica, permitindo que as alterações feitas possam ser misturadas suavemente na superfície em que for aplicada no modelo.

### 2.6.3 Visão de câmera

De acordo com Dunn e Parberry (2011), na maioria dos softwares de modelagem, o efeito de câmera 3D é implementado utilizando-se de combinação linear para transformar as coordenadas de um ponto na cena em coordenadas na tela. Para isso, utiliza-se de rotação de câmera, a distância da câmera para o ponto e a projeção. A distorção perspectiva é aplicada através da matriz de projeção da câmera, enquanto a matriz de rotação é utilizada para girar a câmera.

O *WebGL* é uma *Application Programming Interface (API)* baseada em *OpenGL* que permite a renderização de gráficos tridimensionais diretamente nos navegadores web utilizando *JavaScript*. Esta API foi concebida com o objetivo de trazer gráficos tridimensionais nativos para a web, sem a necessidade de *plugins* ou *add-ons*. Isso é possível graças à capacidade do *WebGL* de usar o acelerador gráfico do computador por meio do *JavaScript*, permitindo assim criar gráficos 3D complexos e de alta qualidade.

Por ser uma *API* baseada em *OpenGL ES 2.0*, o *WebGL* oferece um bom nível de compatibilidade com os recursos gráficos mais modernos. Isso possibilita a utilização de *shaders* para a manipulação de gráficos 3D. Ele é suportado por todas as principais plataformas e navegadores modernos, o que aumenta ainda mais a sua versatilidade.

### 3 Trabalhos relacionados

Neste capítulo serão analisados alguns artigos e discutido a relevância dentro do contexto do nosso trabalho, levando em consideração a proximidade de tema. Foi utilizado com critério as seguintes palavras chaves: modelagem 3D, intuitividade, *Blender*, *ZBrush*, *bootcamp*.

Bailey e Law (2014) mostraram que o *Blender*, ao ser aplicado em escolas de ensino médio, pode ajudar alunos a desenvolverem interesse em algumas áreas como engenharia e matemática, assim como mostrou que em certa parte do *bootcamp* alguns alunos encontravam as ferramentas e as praticavam individualmente, mostrando algumas evidências que a *Engine* possui um certo grau de intuitividade.

Sarstedt (2012) compara as ferramentas de modelagem *Mudbox* e *ZBrush* com base no modelo “qualidade versus tempo”, onde o objetivo é encontrar a ferramenta mais adequada para modelar com mais tempo. A estimativa foi dada a partir de uma modelagem por um iniciante de uma mesma arte conceitual. Foi mostrado que a ferramenta *MudBox* é mais intuitiva, mas possui uma quantidade de ferramentas limitantes enquanto o *ZBrush* possui uma interface de difícil entendimento, mas com uma quantidade de ferramentas extensas que proporciona mais detalhes aos modelos.

Com o objetivo de tornar o ambiente de modelagem mais intuitivo, Kim e Park (2014) apresentam um sistema que permite que o usuário desenvolva as habilidades de modelagem com mais facilidade. Desta forma, é apresentado o ambiente de modelagem de espaços bidimensionais o qual demanda treino e habilidade do usuário, é implementado uma interface intuitiva com o *VR* trazendo um ambiente que simula a modelagem de argila para torna-lá mais natural.

Assim como Kim e Park (2014), Takala, Mäkäräinen e Hämäläinen (2013) apresenta a ideia de trazer a modelagem de ambientes Bidimensional (2D) para ambientes 3D utilizando-se de hardware de prateleira, com o objetivo de deixar a modelagem mais intuitiva e de fácil manuseio. Para tal ele apresenta uma solução que, diferentemente do apresentado por Kim e Park (2014), traz o *Blender* como base do software de modelagem, realizando também uma pesquisa de usuário que conduziu usuários experientes e não experientes a utilizarem o ambiente com o objetivo de entender o quanto o ambiente que lhes foi proporcionado ajudaria no desempenho nas tarefas de modelagem.

Bærentzen (2001) apresenta um sistema intuitivo de modelagem volumétrica com uma linha de aprendizado linear utilizando-se de alguns tipos de formas simples onde a modelagem baseia-se em operações de construção, deformação, fusão de formas assim como a diferença entre elas. O sistema apresenta operações de geometria sólida construtiva, isto é, operações *booleanas* para criar um objeto novo a partir de objetos já existentes. Estas operações também fazem parte da teoria dos conjuntos tendo como suas principais operações: intersecção, união e diferença.

Perng et al. (2001) apresentam uma ferramenta de modelagem em tempo real,



baseada em algoritmos de *marching cubes*, ao qual possibilita a criação de modelos de alta qualidade de superfícies com suavidade aplicada. Além disso é aplicado um método de otimização, onde é utilizada uma estrutura de dados bem planejada.

Devaux, Brédif e Papanoditis (2012) utiliza *WebGL* propõem um mapa interativo do ponto de vista de um pedestre, similar ao *Google Maps*. Ao longo do artigo o autor explora a criação de um otimizador de visualizador geográfico para *GPU*. Ao mesmo tempo é explorado as alternativas para navegadores e placas de vídeo que não possuem compatibilidade com o *WebGL*, onde alternativamente é possível fazer a utilização do Flash.

Nossa abordagem utiliza um método de avaliação semelhante ao apresentado por Sarstedt (2012), onde a eficiência do protótipo é testada utilizando métricas de tempo. A comparação é feita entre o tempo utilizado na nossa abordagem e em uma ferramenta já conhecida no mercado. Seguindo o proposto por Bailey e Law (2014), nosso projeto tem como um dos objetivos ser utilizado como meio de aprendizado e incentivo.

Bærentzen (2001) introduziu o conceito de *CSG*, que serviu como inspiração e abriu a possibilidade para a implementação de técnicas de modelagem menos complexas. Assim como Perng et al. (2001), nossa ferramenta é uma solução que utiliza uma biblioteca do *Javascript* baseada em *WebGL*. Diferentemente dessas abordagens, nosso projeto é mais direcionado para a modelagem tridimensional.

## 4 Desenvolvimento

Neste capítulo, é apresentado o processo percorrido para o desenvolvimento do projeto deste trabalho. Serão demonstradas as ferramentas utilizadas para o desenvolvimento do protótipo, as funcionalidades implementadas e os artifícios utilizados na construção da ferramenta.

### 4.1 Definição do funcionamento da ferramenta

Para o desenvolvimento desse trabalho foi necessário a compreensão de alguns conceitos de álgebra linear e de ferramentas de modelagem em geral. Utilizamos a linguagem *JavaScript* e a biblioteca *Three.js*, que faz uso da *API WebGL* para construir o ambiente da aplicação.

Selecionamos algumas técnicas que atendem ao escopo do trabalho, e a primeira definição foi a implementação dos pincéis de modelagem. Esses pincéis oferecem maior flexibilidade em comparação com outras alternativas apresentadas na seção 2.

Além disso, incluímos funções essenciais, como rotação e movimento da câmera. Devido à complexidade da versão mais recente do *Three.js*, algumas técnicas foram deixadas para serem implementadas em trabalhos futuros.

#### 4.1.1 Abordagem inicial

Para resolução deste trabalho, algumas etapas do desenvolvimento foram previstas para nosso protótipo:

- Desenvolver metodologia da ferramenta de modelagem
- Desenvolver a interface do ambiente
- Definir e desenvolver funcionalidades básicas do objeto

#### 4.1.2 Configuração da cena

Antes de iniciar o desenvolvimento das funcionalidades da ferramenta, foram definidos e aplicados alguns conceitos a respeito da cena.

##### 4.1.2.1 Funcionamento da câmera

Para a projeção perspectiva de uma câmera 3D, são utilizadas matrizes de transformação. A projeção perspectiva leva em consideração a posição da câmera, a relação de aspecto e a distância do plano de visão. A formula geral é dada por:

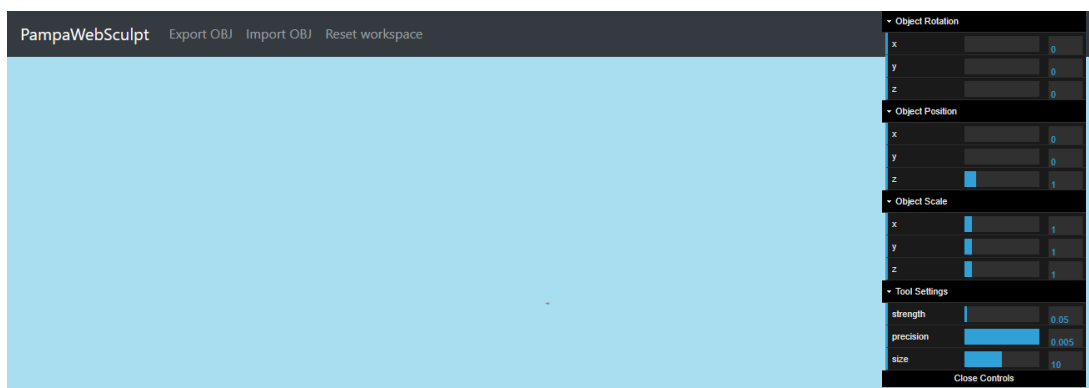
$$P' = MP$$

Onde  $\mathbf{P}'$  é o ponto resultante;  $\mathbf{M}$  é a matriz de projeção de perspectiva, que engloba funções de rotação, translação e projeção;  $\mathbf{P}$  é o ponto a ser projetado. Para implementação das funcionalidades da câmera foi utilizado a biblioteca *OrbitControls* Cabello (2021).

#### 4.1.3 Interface do ambiente

Como parte essencial da ferramenta, foi criada uma interface com várias funcionalidades, conforme pode ser observado na Figura 7. No menu superior, foram incluídas algumas opções, tais como: importar objeto para a cena, exportar objeto com as alterações realizadas no *workspace* e, por último, a opção de excluir os objetos da cena (*reset workspace*). Essas opções foram escolhidas para permanecerem no menu superior, pois são ações internas ou que têm um grande impacto na cena, diferentemente do menu menor, localizado à direita. O menu cascata à direita possui modificações internas, com altera-

Figura 7 – Interface do protótipo

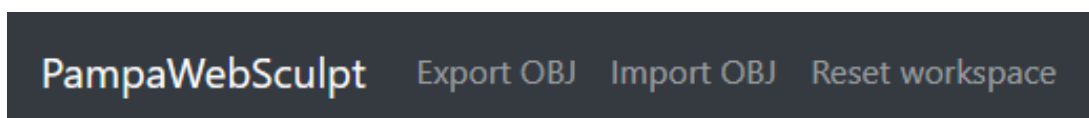


Fonte: A autora

ções de posição e rotação do objeto na cena. Para esse menu, foi utilizada a biblioteca *Dat.GUI*. Nesse menu, existem opções de rotação, posição e configurações da ferramenta de modelagem, como o tamanho da área da ferramenta, a intensidade do movimento da rolagem do *mouse* e, conseqüentemente, a força com que as alterações são aplicadas no modelo. Por fim, temos a opção de ajustar a intensidade da ferramenta.

#### 4.1.4 Funcionalidades da interface

Figura 8 – Interface do protótipo



Fonte: A autora

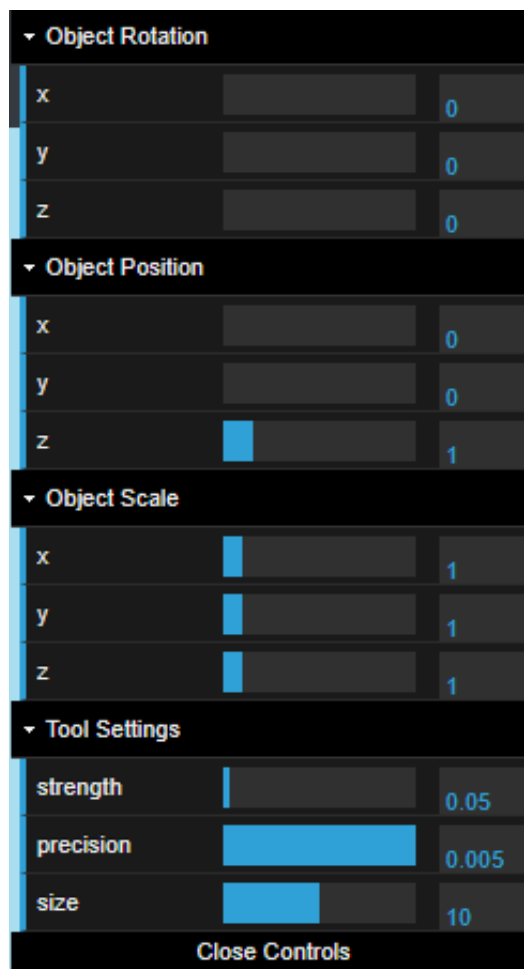
Referenciando a Figura 8, no menu superior, temos algumas funções importantes para o funcionamento do protótipo. A função de exportar um objeto salva o objeto

atualizado da cena, ou seja, todas as mudanças feitas na cena são transferidas também para o objeto a ser salvo. A opção de importar objeto carrega um novo objeto para a cena. Se já houver um objeto presente na cena, então o novo modelo carregado substituirá o modelo antigo. Por último, temos a opção de reiniciar a área de trabalho, que exclui o modelo presente na cena, se houver. A opção de importar utiliza-se naturalmente da opção de reiniciar a área de trabalho em seu processo.

#### 4.1.5 Funcionalidades e alterações básicas do objeto

Antes de definir o pincel de modelagem, foram adicionadas algumas funcionalidades mais básicas ao software. Isso inclui a capacidade de redimensionar e rotacionar o modelo. Essas operações simples permitem que o usuário faça alterações básicas no objeto, assim como pode se perceber na Figura 9.

Figura 9 – Interface do menu de modificação de parâmetros



Fonte: A autora

Existem três funcionalidades principais na ferramenta: **Rotação**, que permite ao usuário trabalhar em qualquer ângulo do objeto sem precisar mover a câmera; **Reposicionamento**, que permite ao usuário alterar a posição do objeto na cena sem ter que mover

a câmera; e **Redimensionamento**, que permite ao usuário ajustar o tamanho do objeto na cena. Além dessas funcionalidades, também foi implementada a opção de desfazer e refazer ações no objeto. Para refazer uma ação, basta pressionar Ctrl+Shift+Z, e para desfazer uma ação, pressione Ctrl+Z.

#### 4.1.6 Parâmetros de ajuste da ferramenta de modelagem

Para a ferramenta de modelagem foi adicionado três parâmetros ajustáveis, que assim como mostrado na figura 9, na aba de *Tool settings*. Nessa aba temos **Strenght**: É a força que o pincel é aplicado ao modelo, determinando o quanto a rolagem do mouse vai influenciar na alteração do modelo; **Precision**: Que determina a quantidade vértices vão sofrer as alterações aplicadas; **Size**: Determina o tamanho do diâmetro da aplicação das alterações a partir do ponto de interseção do clique do mouse para o objeto na cena. É o fator que implica na direção da média das normais, ou seja, quanto maior o tamanho da área, mais a direção que a média das normais aponta tende a variar.

#### 4.1.7 Ferramenta de modelagem

Como foco deste trabalho, foi criada uma das ferramentas descritas no capítulo 2. O desenvolvimento dessa ferramenta pode ser dividido em algumas etapas:

##### 4.1.7.1 Seta auxiliar

Figura 10 – Exemplo da utilização da seta



Fonte: A autora

Para auxiliar o usuário ao utilizar a ferramenta, definiu-se a criação de uma seta que utiliza a média da normal dos vértices para determinar a direção da ponta da seta, e

consequentemente a direção que os vértices seguirão para realizar o evento.

A área em que o cálculo ocorre é baseada majoritariamente pela posição da câmera e pelo tamanho pré-definido do pincel. Isso sugere que a área de atuação do pincel sobre o modelo pode ser flexível, dependendo dessas duas variáveis.

Podem ocorrer alguns casos possíveis. Em proximidade extrema do modelo, o cálculo das normais é reduzido ao mínimo, resultando em uma alteração aplicada apenas a um vértice. Em contrapartida, pode ocorrer o oposto, onde a câmera é posicionada longe do modelo e a alteração pode afetar todo o modelo, mesmo com o tamanho do pincel sendo mínimo.

Além de indicar a direção da modificação, a seta auxiliar também pré-define o local da modificação. Para mudar a área do modelo que o usuário deseja modelar, será necessário ativar o evento de clique em outra área do objeto. Este pré-processamento é vantajoso para o desempenho da ferramenta, pois ao calcular a área de atuação, não será necessário refazer os cálculos posteriormente na etapa de modificar o modelo. No entanto, essa abordagem torna a etapa de modelagem inflexível. Um exemplo de utilização pode ser visto na Figura 10, onde temos a atuação da seta na parte do nariz do modelo da vaca.

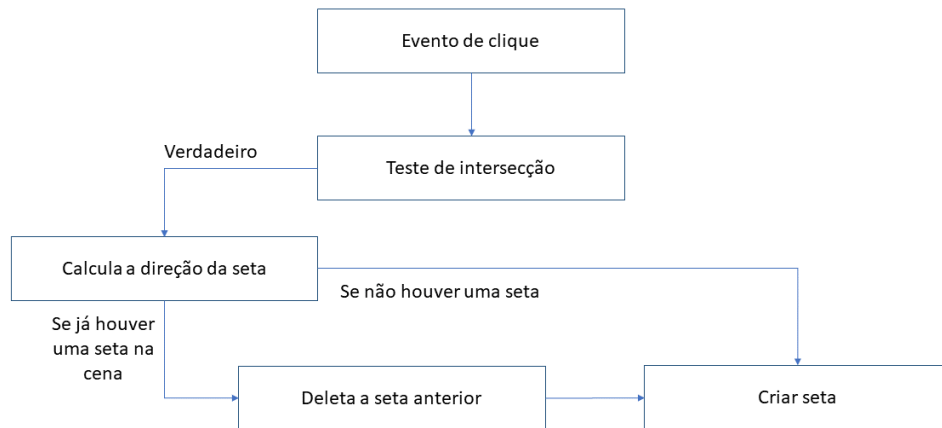
Na Figura 11, podemos observar o processo que ocorre na função do evento de adição da seta auxiliar.

- **Evento de clique:** Quando o usuário clica em qualquer parte da cena, o evento de clique é acionado;
- **Teste de Intersecção:** Este teste verifica se houve alguma intersecção entre a posição do clique e o objeto presente na cena. Se não houver intersecção, a seta não será adicionada. Se houver intersecção, então a direção da seta é calculada a partir do ponto de intersecção;
- **Cálculo da direção da seta:** Utiliza-se a média das normais dos vértices presentes dentro de uma distância  $N$  definida pelo usuário para demarcar o tamanho da área de atuação do pincel. Se já existir uma seta de uma interação anterior, então a seta é excluída da cena;
- **Criação da seta:** Se não existir nenhuma seta na cena ou se a seta já tiver sido excluída previamente, uma nova seta é adicionada com as coordenadas obtidas através do cálculo da média das normais.

#### 4.1.7.2 Pincel de modelagem

Após a definição da seta auxiliar, introduzimos a função do nosso pincel de modelagem, a qual é ativada se a seta auxiliar existir e quando a combinação da rolagem do *mouse* com a tecla “Alt” é utilizada.

Figura 11 – Função de criar a seta



Fonte: A autora

Para o desenvolvimento do pincel, é utilizada uma função gaussiana tridimensional. Neste trabalho, essa função é empregada principalmente para suavização e criação de um formato de sino na modelagem. A formula da gaussiana pode ser definida como:

$$f(x, y, z) = Ae^{-\frac{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}{2\cdot\sigma^2}}$$

Onde  $f(x, y, z)$  representa o valor da função gaussiana nos pontos  $x, y$  e  $z$ ;  $A$  representa a amplitude máxima da função gaussiana;  $f(x_c, y_c, z_c)$  São as coordenadas do centro da função gaussiana;  $\sigma$  é o desvio padrão que controla a largura da curva gaussiana.  $e$  é o número de Euler, que dará a forma característica de sino.

## 5 Resultados

Neste capítulo, serão apresentados os resultados obtidos durante o desenvolvimento e a prototipação da ferramenta, além de uma análise dos pontos alcançados e dos próximos passos a serem seguidos.

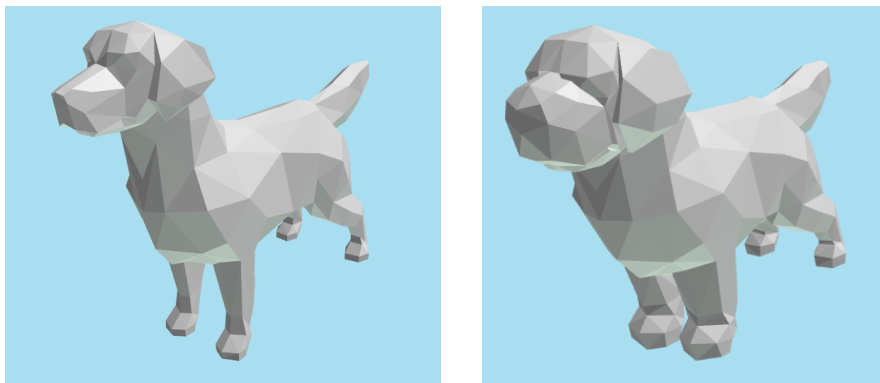
### 5.1 Pincel de modelagem

Para introduzir nossos resultados obtidos pela nossa ferramenta, vamos apresentar algumas aplicações que nossa ferramenta pode ter: Aplicação parcial do pincel, aplicação total do pincel, aplicação do pincel com uma força inversa e aplicação em apenas um vértice do objeto. O pincel pode ser aplicado de duas formas, empurrando a rolagem do *mouse* para frente ou para trás. quando aplicado uma força com a rolagem do *mouse* para trás então os vértices do objeto a serem alterados serão enviados para dentro do objeto, ou seja, para a direção oposta a média das normais previamente calculadas.

### 5.2 Aplicações utilizando toda a área do objeto

Uma das aplicações possíveis da ferramenta é utilizar um alto *range* de forma que o pincel seja aplicado no objeto por completo. Um exemplo pode ser visto na figura 12, onde temos um objeto em sua forma original à esquerda e o objeto modificado à direita. Curiosamente, o objeto modificado adquiriu uma aparência de filhote. Ao aplicar essas alterações, o objeto modificado ganhou uma aparência mais arredondada.

Figura 12 – Resultado do pincel ao ser aplicado em toda a área do objeto

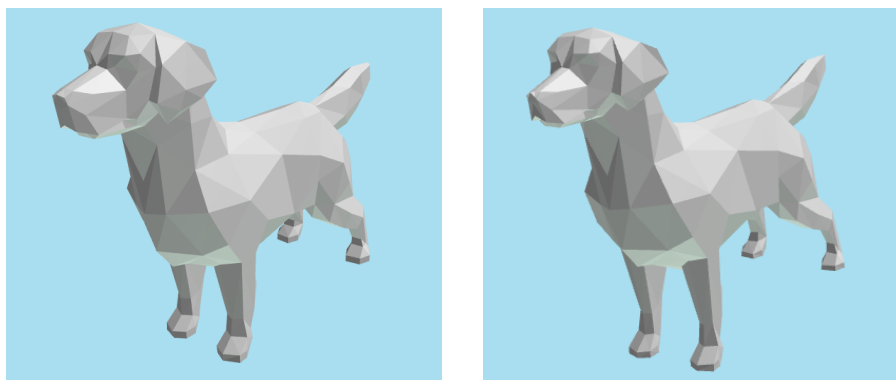


Fonte: A autora

O mesmo efeito pode ser observado quando aplicado com uma força inversa. Esse efeito pode ser visualizado na figura 13. Embora os resultados sejam mais sutis do que na figura 12, ainda é possível perceber, principalmente pela cabeça do modelo do cachorro, que o corpo ficou relativamente maior, tornando o objeto mais harmônico. Isso representa uma utilidade para os usuários que desejam aplicar uma simples modificação no modelo.



Figura 13 – Resultado do pincel ao ser aplicado em toda a área do objeto com uma força inversa



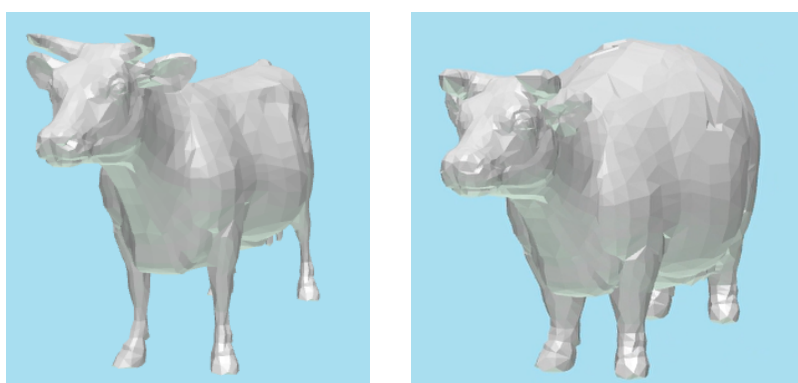
Fonte: A autora

Figura 14 – Resultado do pincel aplicado com um força inversa localmente em uma área do objeto



Fonte: A autora

Figura 15 – Resultado do pincel ao ser aplicado parcialmente



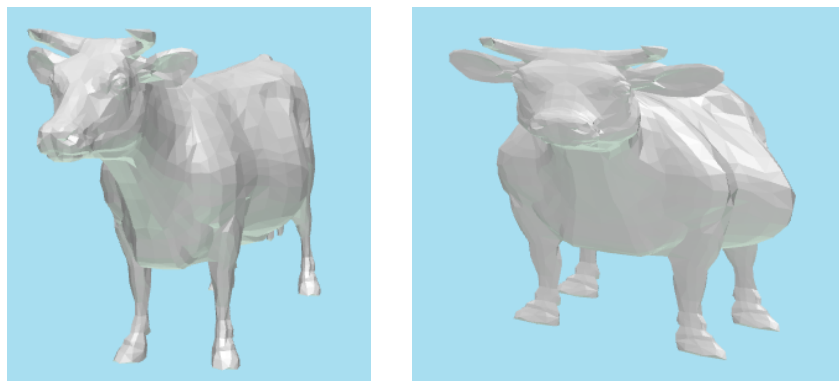
Fonte: A autora

### 5.3 Aplicações utilizando apenas uma área específica do objeto

A aplicação parcial também é possível nesta ferramenta, levando em conta o *range* ao ser aplicado no modelo. Um *range* menor pode resultar em um impacto maior na área selecionada pelo usuário e um impacto menor em áreas mais distantes do ponto de seleção. Por exemplo, na imagem 15, temos à esquerda o modelo original da vaca, e

à direita, a vaca com a modificação. No caso da vaca à direita, o ponto de aplicação foi na área das costas da vaca, efeito que pode ser observado principalmente nas patas, que engrossaram no processo. Em contraste com esse resultado temos o efeito do pincel com uma força inversa a do apresentado na Figura 15. Na Figura 14, temos o efeito da ferramenta aplicado apenas a uma parte específica do objeto da vaca, onde apenas o chifre sofreu alterações.

Figura 16 – Resultado do redimensionamento no modelo

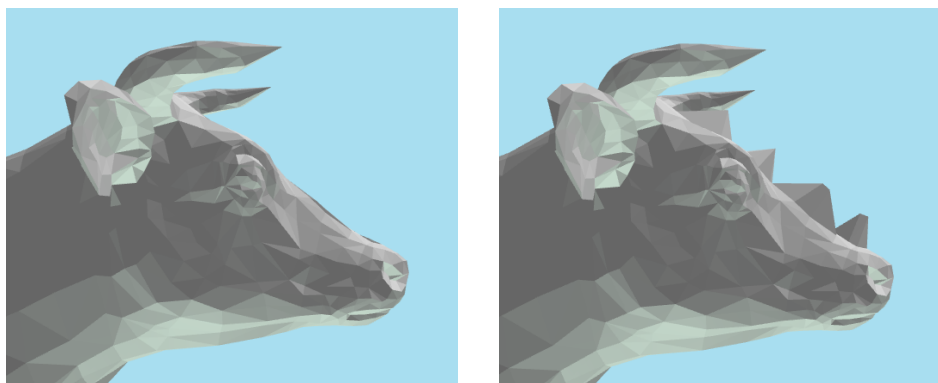


Fonte: A autora

#### 5.4 Aplicações de ferramentas para redimensionamento do objeto

Como mencionado no capítulo 4, há uma opção relevante no contexto do nosso trabalho que se refere ao redimensionamento do modelo no cenário. Na nossa solução, o redimensionamento é feito individualmente em cada eixo. Na Figura 16, temos à esquerda o modelo original, enquanto à direita, é possível visualizar o modelo da vaca com modificações nos três eixos.

Figura 17 – Resultado do pincel ao ser aplicado em apenas um vértice



Fonte: A autora

## 5.5 Aplicações utilizando apenas um vértice do objeto

Temos ainda a opção de aplicar as alterações apenas a um vértice, como exemplificado na Figura 17. Nesta figura, foram realizadas alterações no rosto do modelo da vaca, selecionando quatro vértices do objeto para que se destacassem. O efeito contrário é possível através da aplicação de uma força no sentido oposto.

## 6 Considerações Finais

Neste trabalho, apresentamos softwares de modelagem 3D com o objetivo de introduzir conceitos básicos de modelagem tridimensional e destacar algumas diferenças entre esses softwares, bem como discutir questões sobre interfaces de usuário. Estudamos características comuns entre os softwares de modelagem tridimensional citados, considerando a portabilidade dessas ferramentas. Analisamos alguns trabalhos relacionados que abordam a questão da intuitividade e aprendizado, assim como o impacto destas características nos usuários.

Na primeira etapa, discutimos e definimos algumas técnicas a serem aplicadas em nossa ferramenta, e apresentamos um protótipo. Este nos trouxe algumas funcionalidades úteis; no entanto, como discutido no capítulo 5, não recomendamos seu uso em objetos mais complexos, já que nossa ferramenta não possui todas as otimizações necessárias para suportar modelos detalhados.

Embora o protótipo não tenha sido validado, o projeto possui potencial e espaço para melhorias, tanto em relação às ferramentas aplicadas quanto à interface. Em todos os casos, é necessária uma pesquisa mais aprofundada sobre estes tópicos.

Para trabalhos futuros, planejamos aprofundar as pesquisas sobre padrões de interface, otimizar a ferramenta e adicionar mais pincéis e opções de personalização dos modelos.

## Referências

- BÆRENTZEN, J. A. **Volume Sculpting: Intuitive, Interactive 3D Shape Modelling**. 2001. Disponível em: <<http://www2.compute.dtu.dk/pubdb/pubs/712-full.html>>. Citado 2 vezes nas páginas 22 e 23.
- BAILEY, M.; LAW, C. A summer blender camp: Modeling, rendering, and animation for high school students. **IEEE Computer Graphics and Applications**, v. 34, n. 1, p. 65–67, 2014. Citado 3 vezes nas páginas 15, 22 e 23.
- CABELLO, R. **Three.js**. 2021. Disponível em: <<https://threejs.org/>>. Citado 2 vezes nas páginas 12 e 25.
- CATMULL, E.; PARKE, F. **A Computer Animated Hand**. 1972. Film. University of Utah. Diretor: Edwin Catmull, Produtor: Fred Parke. Citado na página 12.
- COMMUNITY, B. **Blender 3.4 Reference Manual**. [S.l.], 2023. Disponível em: <<https://docs.blender.org/manual/en/latest/index.html>>. Citado na página 15.
- COMMUNITY, B. **Window System Introduction**. 2023. Disponível em: <[https://docs.blender.org/manual/en/latest/interface/window\\_system/introduction.html](https://docs.blender.org/manual/en/latest/interface/window_system/introduction.html)>. Acesso em: 5 de Jan. 2023. Citado na página 16.
- DEVAUX, A.; BRÉDIF, M.; PAPARODITIS, N. A web-based 3d mapping application using webgl allowing interaction with images, point clouds and models. In: **Proceedings of the 20th International Conference on Advances in Geographic Information Systems**. New York, NY, USA: Association for Computing Machinery, 2012. (SIGSPATIAL '12), p. 586–588. ISBN 9781450316910. Disponível em: <<https://doi.org/10.1145/2424321.2424422>>. Citado na página 23.
- DUNN, F.; PARBERRY, I. **3D Math Primer for Graphics and Game Development**. 2nd. ed. [S.l.]: A K Peters/CRC Press, 2011. Citado 2 vezes nas páginas 20 e 21.
- ENDO, L. C. Y. **Constructive Solid Geometry - CSG**. 2007. Disponível em: <<http://www.ime.usp.br:80/~lye/hp/sg3d/objetos.html>>. Acesso em: 5 de Jan. 2023. Citado na página 19.
- GAVANE, A. **My First WebGL Program**. 2019. <<https://medium.com/@ajinkyagavane/a-brief-introduction-to-webgl-8530749154b1>>. Citado na página 12.
- GMBH, M. C. **Window System Introduction**. 2023. Disponível em: <<http://docs.pixologic.com/user-guide/customizing-zbrush/interface-layout>>. Acesso em: 5 de Jan. 2023. Citado na página 17.
- HATTON, P. **A beginner's guide to Sculpttris**. 2019. Disponível em: <<https://3dtotal.com/tutorials/t/a-beginners-guide-to-sculptris>>. Acesso em: 5 de Jan. 2023. Citado na página 18.
- HEFFRON, R. T. et al. **Futureworld**. 1976. Film. American International Pictures. Diretor: Richard T. Heffron, Produtor: James T. Aubrey and Paul N. Lazarus III. Citado na página 12.

International Organization for Standardization. **Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models**. [S.l.]: ISO, 2011. ISO/IEC 25010:2011(E). Citado na página 15.

KIM, H.; PARK, C. Sound clay: An immersive art form by sculpting clay and sound. In: STEPHANIDIS, C. (Ed.). **HCI International 2014 - Posters' Extended Abstracts**. Cham: Springer International Publishing, 2014. p. 602–606. ISBN 978-3-319-07857-1. Citado na página 22.

MANSSOUR, I.; COHEN, M. Introdução à computação gráfica. **RITA**, v. 13, p. 43–68, 01 2006. Citado na página 14.

MARCUS. **Sculptris Pro**. 2018. <<http://docs.pixologic.com/user-guide/3d-modeling/modeling-basics/creating-meshes/sculptris-pro/>>. Citado na página 13.

PERNG, K.-L. et al. A real-time 3d virtual sculpting tool based on modified marching cubes. p. 9, 01 2001. Citado 2 vezes nas páginas 22 e 23.

PINHO, P. M. S. **Modelagem por Varredura(sweep)**. 2023. Disponível em: <<https://www.inf.pucrs.br/pinho/CG/Aulas/Modelagem/Modelagem3D.htm>>. Acesso em: 2 de Jan. 2023. Citado na página 20.

PINHO, P. M. S. **Varredura Rotacional**. 2023. Disponível em: <<https://www.inf.pucrs.br/pinho/CG/Aulas/Modelagem/Modelagem3D.htm>>. Acesso em: 2 de Jan. 2023. Citado na página 20.

PIXOLOGIC. **ZBrush Reference Guide**. 2023. Citado na página 14.

SARSTEDT, I. **Comparisons of Sculpting Efficiency Between Mudbox and ZBrush**. Tese (Doutorado) — Dissertation, 2012. Disponível em: <<http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-12231>>. Citado 4 vezes nas páginas 14, 15, 22 e 23.

TAKALA, T. M.; MÄKÄRÄINEN, M.; HÄMÄLÄINEN, P. Immersive 3d modeling with blender and off-the-shelf hardware. In: **2013 IEEE Symposium on 3D User Interfaces (3DUI)**. [S.l.: s.n.], 2013. p. 191–192. Citado na página 22.