



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO DE EXPREÇÃO GRÁFICA
CURSO DE ANIMAÇÃO

Eduardo Correa Machado

**Processo de criação de um modelo de Vtuber para utilização em plataformas
de streaming**

Florianópolis
2023

Eduardo Correa Machado

**Processo de criação de um modelo de Vtuber
para utilização em plataformas de streaming**

Trabalho de Conclusão de Curso submetido ao curso de animação do Campus Florianópolis da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em animação.

Orientador: Prof. Gabriel de Souza Prim, Dr.

Florianópolis
2023

Machado, Eduardo Correa

Processo de criação de um modelo de Vtuber para utilização em plataformas de streaming / Eduardo Correa Machado ; orientador, Gabriel de Sousa Prim, 2023.

72 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. Vtuber. 3. Design de personagem. 4. Rigging. 5. Live2D. I. Prim, Gabriel de Sousa. II. Universidade Federal de Santa Catarina. Graduação em Animação. III. Título.

Eduardo Corrêa Machado

Título: Processo de criação de um modelo de Vtuber para utilização em plataformas de streaming

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

Local: Sala Hassis - térreo do bloco B – CCE, 27 de novembro de 2023.



Coordenação do Curso

Banca examinadora



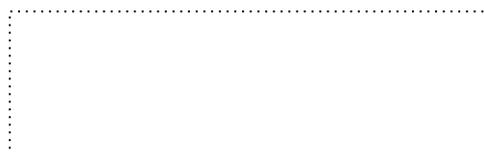
Prof. Gabriel de Souza Prim, Dr.

Orientador



Prof. Flávio Andaló, Dr

Instituição UFSC.



Prof. Nicholas Bruggner Grassi, Dr.

Instituição UFSC.

Florianópolis, 2023.

RESUMO

Este trabalho tem como objetivo introduzir o conceito de Vtuber assim como demonstrar o processo de criação de um modelo para ser utilizado em transmissões ao vivo em plataformas de streaming. O trabalho aborda a criação do *design* do personagem para uma cliente, desde seu conceito até a ilustração final, assim como introduz conceitos do programa Live2D além de técnicas utilizadas no processo de *rigging* do personagem. E por fim explica sobre a exportação para um programa de captura de movimento facial e como utilizar essa captura em um *software* próprio para streaming.

Palavras-chave: *Vtuber*; *YouTuber* virtual; *Rigging*; *Streaming*; Design de personagem; Live2D.

ABSTRACT

This work aims to introduce the concept of a Vtuber and demonstrate the process of creating a model to be used in live broadcasts on streaming platforms. The work covers the creation of a character design for a client, from concept to final illustration, as well as introducing concepts from the Live2D program and techniques used in the character rigging process. Finally, it explains how to export the character to a facial motion capture program and how to use this capture in streaming software.

Keywords: Vtuber; virtual YouTuber; *Rigging*; Streaming; character design; Live2D.

LISTA DE FIGURAS

Figura 1 - Kizuna AI	18
Figura 2 - Gráfico ilustrativo do crescimento de visualizações de conteúdo Vtuber no YouTube entre janeiro e outubro de 2020	19
Figura 3 - Processo de criação de um Vtuber.....	22
Figura 4 - Compilação de referencias para o design da personagem	23
Figura 5 - Processo de concepção de conceito de Vtuber.....	24
Figura 6 - Gawr Gura e Artemis of the Blue.....	25
Figura 7 - Kobo Kanaeru, uma xamã da chuva que não gosta de fazer seu trabalho	25
Figura 8 - Natsuiro Matsuri, uma animadora de torcida muito energética. .	26
Figura 9 - Kson, uma ex líder de uma gang japonesa que se tornou um lobo solitário.....	26
Figura 10 - Porcelain Maid, é um homem adulto encarnado em uma boneca de porcelana.....	27
Figura 11- HOLOLIVE COUNCIL . Em sequência, Ceres Fauna, Tsukumo Sana, Hakos Baelsz, Ouro Kronii e Naashi Mumei.	27
Figura 12- Exemplos de roupas do estilo <i>Lolita fashion</i>	28
Figura 13 - <i>Model Sheets</i> para as personagens, Shirogane Noel, Takane Lui,	29
Figura 14 - Teste de diferentes tipos de roupas.	30
Figura 15 - Teste de diferentes esquemas de cores.....	30
Figura 16 - Resultado final da ilustração.	31
Figura 17 - Exemplo de separação e organização das peças da cabeça.....	33
Figura 18 - Exemplo de separação e organização das peças do olho.	34
Figura 19 - Exemplo de separação e organização das peças da boca.	34
Figura 20- Exemplo de separação e organização das peças do nariz.	35
Figura 21 - Exemplo de organização das sombras da cabeça.	35
Figura 22 - Exemplo de separação e organização das peças do cabelo.....	36
Figura 23 - Exemplo de ilustração da Franja.	37
Figura 24 - Exemplo de separação e organização das peças de uma parte do cabelo.....	37
Figura 25 - Exemplo de organização das camadas do corpo.....	38

Figura 26 - Exemplo de separação e organização das peças do colar.	39
Figura 27 - Exemplo de separação e organização das peças da camisa.	39
Figura 28 - Exemplo de separação e organização das peças do colete.....	40
Figura 29 - Exemplo de separação e organização das peças da saia.....	40
Figura 30 - Exemplo de separação e organização das peças da perna.	41
Figura 31 - Exemplo de separação e organização das peças do sapato.	41
Figura 32 - Exemplo de separação e organização das peças do laço.....	42
Figura 33 - Exemplo do <i>Atlas texture</i> da modelo.....	43
Figura 34 - Demonstração de como funciona a interpolação linear do <i>warp deformer</i>	44
Figura 35 Exemplo de <i>mesh</i> sem as divisões <i>bezier</i> e com as divisões <i>bezier</i>	45
Figura 36 - Exemplo de como funciona o <i>rotation deformer</i>	45
Figura 37 - Sequência de imagens demonstrando como funciona o <i>deform path</i>	46
Figura 38 – Demonstração do <i>clipping mask</i> da íris no olho.	47
Figura 39 - Demonstração do movimento da cabeça no ângulo X.	48
Figura 40 - Demonstração do movimento da cabeça no ângulo Y.	49
Figura 41 - Demonstração do movimento da cabeça com o ângulo X e Y linkados.	50
Figura 42 - Demonstração dos <i>keys</i> do parâmetro “ <i>EyeL open</i> ”.	51
Figura 43 - Demonstração dos <i>keys</i> do parâmetro “ <i>EyeL smile</i> ”.	51
Figura 44 – Demonstração da janela de configurações da física.....	53
Figura 45 – Exemplo do resultado final do rig dos olhos.	55
Figura 46 - Exemplo do resultado final do rig das sobrancelhas.	56
Figura 47 - Montagem com a referência usada e a versão final boca.	57
Figura 48 – Demonstração da orelha “rigada” de forma incorreta.	58
Figura 49 - Demonstração da máscara aplicada a orelha para corrigir o problema anterior.	58
Figura 50 - Demonstração da deformação do <i>deformer</i> da franja.	59
Figura 51 - Demonstração da deformação do <i>deformer</i> de cabelos médios.	60
Figura 52 - Demonstração da deformação do <i>deformer</i> de cabelos longos.	60
Figura 53 - Exemplo do resultado final dos parâmetros da física do cabelo.	61

Figura 54 - Demonstração da separação dos <i>deformers</i> para movimentar o corpo.	62
Figura 55 - Menu do programa VTube Studio.....	64
Figura 56 - Demonstração do menu de configuração de câmera.	66
Figura 57 - Exemplo de configuração de parâmetro.	67
Figura 58 - Botão para edição de expressões.	68
Figura 59 Exemplo de uma expressão de raiva configurada no menu de expressoes.	68
Figura 60 - Demonstração do modelo em frente a um jogo dentro do software de transmissão.	69

LISTA DE TABELAS

Tabela 1 - comparação entre programas de tracking facial	21
--	----

SUMÁRIO

1	INTRODUÇÃO	16
2	FUNDAMENTAÇÃO TEÓRICA: O QUE É V-TUBER	16
	PANDEMIA, CONSUMO E ENGAJAMENTO	17
	MERCADO DE ENTRETENIMENTO COM VTUBERS.....	18
	SOFTWARES PARA VTUBERS	20
3	PROCEDIMENTOS METODOLÓGICOS	22
	BRIEFING	23
	CONCEITO	24
	DESIGN DE PERSONAGEM	28
3.1.1	Evolução do design	30
	ILUSTRAÇÃO E CUTTING	31
3.1.2	Cabeça	32
3.1.3	Cabelo	36
3.1.4	Corpo e roupas	38
	LIVE 2D.....	42
3.1.5	Parts	42
3.1.6	Textura	42
3.1.7	Artemesh e Polígonos	43
3.1.8	Parâmetros	43
3.1.9	Keyform	44
3.1.10	Deformer	44
	WORKFLOW LIVE 2D.....	46
3.1.11	Cabeça	48
3.1.12	Cabelo	58
3.1.13	Corpo	61
3.1.14	Respiração	62
3.1.15	Ângulo Z	63
	EXPORTAÇÃO	63
	VTUBE STUDIO	64
3.1.16	Interface	64
3.1.17	Configuração	65
3.1.17.1	<i>Configurações gerais:</i>	<i>65</i>

3.1.17.2	<i>Configurações de câmera:</i>	65
3.1.17.3	<i>Configurações do modelo:</i>	66
3.1.17.4	<i>Configuração de parâmetros</i>	66
3.1.17.5	<i>Configurações de Hotkeys/Expressões</i>	68
	INTEGRAÇÃO COM SOFTWARE DE STREAMING	69
4	CONCLUSÃO	70

1 INTRODUÇÃO

A pandemia global da COVID-19 trouxe mudanças significativas no comportamento das pessoas, especialmente no que diz respeito ao consumo de conteúdo online. Com o distanciamento social e o aumento do tempo livre em casa, indivíduos de diferentes faixas etárias se tornaram mais suscetíveis a explorar plataformas de vídeo e streaming, como *youtube* e *Twitch*, em busca de entretenimento e conexão social virtual. Plataformas como *youtube* e *Twitch* oferecem uma ampla gama de conteúdo, desde tutoriais, *vlogs*, *gameplay*, até transmissões ao vivo e performances artísticas. Essa diversidade permitiu às pessoas encontrar formas de entretenimento personalizadas de acordo com seus interesses e preferências. No entanto, um fenômeno particular ganhou destaque nesse cenário: a popularização dos Vtubers. Vtubers são avatares animados, 2d ou 3d, que são controlados por uma pessoa em tempo real através de softwares de captura de movimento. Ao se identificar com esses avatares virtuais, os espectadores desenvolvem um senso de proximidade e afinidade, uma vez que esses personagens são capazes de expressar emoções e se envolver de forma interativa. A popularidade dos Vtubers cresceu exponencialmente durante a pandemia, impulsionada pela ampla oferta de conteúdo disponível nas plataformas de vídeo e streaming.

Este trabalho tem como objetivo introduzir o conceito de Vtuber e também aborda a criação de uma Vtuber para uma cliente, desde a idealização do conceito e design da personagem, até a entrega da animação em tempo real para que a cliente possa utilizar em uma *stream*.

2 FUNDAMENTAÇÃO TEÓRICA: O QUE É V-TUBER

O Termo “*Vtuber*” é usado para se referir a pessoas que utilizam softwares e aplicativos para criarem avatares virtuais 2D ou 3D CG controlados por programas de captura de movimento ou sincronização labial para se comunicar com o seu público através de transmissões ao vivo em plataformas de streaming como o *youtube*, *twitch* ou *bilibili*. O termo é derivado de “*youtuber*” junto com a palavra “virtual” e foi mencionado pela primeira vez em um vídeo de 2016 com a auto introdução da personagem Kizuna AI, que se apresentava como uma inteligência virtual avançada

que tem consciência da própria existência. Com o imenso sucesso que Kizuna AI fez, chegando a alcançar 1 milhão de inscritos no *youtube* em janeiro de 2018, surgiram diversos outros artistas que se apresentavam como *Vtubers* e até mesmo empresas agenciadoras de talentos que trabalham somente com *Vtubers* como é o caso das empresas HOLOLIVE, NIJISANJI e VSHOJO.

PANDEMIA, CONSUMO E ENGAJAMENTO

Como consequência da pandemia de COVID-19, o interdito aos espaços de socialização em escala global modificou sensivelmente a matriz de consumo de produtos culturais. O ecossistema configurado pelo isolamento social em decorrência de ações contra a COVID-19 proporcionou a este público em questão mais possibilidades para consumir conteúdos culturais diversos online e também uma abertura maior à novos tipos de conteúdo na internet. Sob essas circunstâncias, foca-se aqui em um fenômeno originalmente nipônico na internet e amplamente massificado através de plataformas de transmissão ao vivo de todo o mundo: os "Vtubers".

Dentre os principais motivadores psicológicos do engajamento do público com um *streamer*, que é quem faz *livestreams*, estão o entretenimento oferecido, a busca por novas informações, a oportunidade de conhecer novas pessoas, as interações sociais, o suporte social ou da comunidade, o senso de comunidade, a ansiedade social e o apoio externo. (HILVERT-BRUCE Et al., 2018). Este engajamento resultado destes fatores gera maior conexão emocional, maior tempo gasto com o conteúdo, mais tempo inscrito no canal e em mais doações, que é a forma de gratidão que o público dá ao criador de conteúdo, sem necessariamente receber conteúdo adicional por isso. (Ibidem, 2018). Pode-se dizer que a massificação da procura por este tipo de criador de conteúdo foi correlacionada ao surgimento e expansão da pandemia de COVID-19, que permitiu ao público a abertura e mais tempo para assistir coisas novas e diferentes na internet, catapultando o conteúdo Vtuber a ser um dos mais lucrativos e chamativos das plataformas de vídeos ao vivo, como *youtube* e *Twitch*, ganhando relevância e altas taxas de engajamento

MERCADO DE ENTRETENIMENTO COM VTUBERS

Criada em 2016, Kizuna AI pertence à Kizuna AI Inc., parte da Activ8, uma empresa japonesa que fornece softwares com um sistema de tecnologia em tempo real, em que o usuário consegue operar um personagem através de sensores posicionados no corpo, os quais reproduzem os movimentos e feições faciais, além de dublar o avatar. Sendo originalmente um sucesso apenas no Japão, Kizuna AI expandiu as *fronteiras* para outros países e plataformas do Sudeste Asiático. Com o sucesso da personagem, outras empresas voltadas ao mercado de *youtubers* virtuais foram surgindo, inicialmente no Japão, e, posteriormente, em outros países da Ásia e do resto do mundo. Atualmente, entre as maiores companhias do ramo estão a japonesa HOLOLIVE, criada em 2017 a partir da incubadora Tokyo VR Startups com o objetivo de desenvolvimento de softwares que tornassem a criação de uma agência de ídolos pops virtuais possível (WAKAYAMA, 2019); a também nipônica NIJISANJI, fundada em 2018 e administrada pela ANYCOLOR Inc. E com presença no Japão, Indonésia, China e Coreia do Sul (ANYCOLOR, 2022); e a estadunidense VSHOJO, fundada como uma startup em 2020 por ex-empregados da *Twitch* e com conteúdo em língua inglesa, convertendo-se na maior empresa de *Vtubers* da *Twitch* (AMOS, 2021), agenciando Ironmouse, atualmente a maior *streamer* feminina da *Twitch* com mais de 90.000 pagantes mensais.

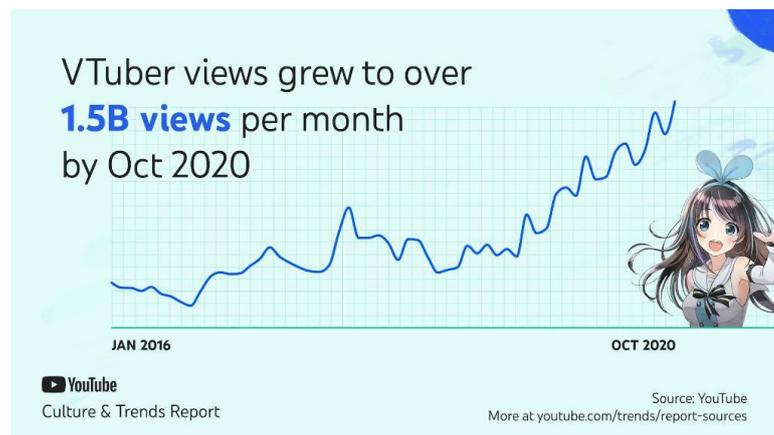
Figura 1 - Kizuna AI



Fonte: Twitter da Kizuna AI. Disponível em:
https://twitter.com/aichan_nel/status/834259517038661632

Em questão de visualizações no *youtube*, o conteúdo Vtuber alcançou a marca de crescimento de 1,5 bilhões de visualizações por mês na plataforma até outubro de 2020. (YOUTUBE, 2020). Uma outra pesquisa do *youtube* mostrou que dentre o público que acessa a plataforma, 47% está aberto a assistir conteúdo de criadores fictícios virtuais, como Vtubers. (YOUTUBE CULTURE & TRENDS, 2020). Pode haver ainda uma grande margem de mercado a ser explorada por este tipo de conteúdo, que tende a se diversificar em novos nichos e a melhorar sua própria qualidade técnica. Podemos *considerar* que os Vtubers já são parte relevante das mídias digitais, possuem vantagens próprias intrínsecas, como o anonimato, qualidade visual, liberdade criativa e conexão facilitada com o público, em especial infanto-juvenil, e que este movimento tende a continuar crescendo, tornando as pessoas imaginárias e virtuais cada vez mais relevantes e influentes no mundo real.

Figura 2 - Gráfico ilustrativo do crescimento de visualizações de conteúdo Vtuber no YouTube entre janeiro e outubro de 2020



Fonte: YouTube Trends Report de 2020. Disponível em:
<https://twitter.com/YouTubeCreators/status/1338924287504355335>

Devido à popularidade do *youtube*, a indústria do entretenimento tem demonstrado um profundo interesse em monetizá-lo (KIM, 2012). Isso permitiu que canais inicialmente projetados como passatempos, se tornassem canais profissionais. Ou simplesmente, já nascessem como canais profissionais. (LAVAVESHKUL, 2012). O impacto econômico que os Vtubers obtiveram em poucos anos de criação, gerando empresas já focadas neste tipo de conteúdo, com o seu forte engajamento representado por doações do público, somado a sua internacionalização, criaram um mercado cultural totalmente digital e em expansão, muito intensivo tecnologicamente,

que já influencia na própria realidade e nas estratégias comerciais de grandes empresas para determinados públicos.

SOFTWARES PARA VTUBERS

Considerando um personagem produzido com técnica 2D, no mercado há diversos softwares utilizados para fazer a parte de ilustração do modelo, pode-se usar qualquer um que tenha a opção de *layers* e preferencialmente alguma ferramenta de clipping e modos de camada. Dentre as opções disponíveis, os mais populares são o Clipstudio, Photoshop, Paint Tool SAI e Procreate, além de opções gratuitas como o Krita e Medibang.

O live2d é um software de animação que pode ser usado para gerar animações em tempo real baseadas em uma ilustração 2D que consegue se mover usando movimento 2.5D utilizando efeitos de *parallax* e *deformadores* para dar a impressão de tridimensionalidade para o modelo. Para o processo de *rigging* do modelo existe apenas este programa usado para a finalidade de usar o rig junto com algum software de *tracking*, por isso é o único software usado na indústria de Vtubers. O software oferece também a parte de animação na qual utiliza-se dos mesmos métodos. Para essa finalidade de animação há outros softwares como After effects, Adobe Animate, Toonboom Harmony e Animeeffects. Adicionalmente, há ainda os programas voltados para o estilo de animação discutido neste texto com a característica de serem mais adaptados para games como Spine e Dragon Bones

As animações feitas nos programas são muito utilizadas em jogos de console como *Fire Emblem Fates* (2015, 3DS), *Kancolle Kai* (2016, PS Vita) e *Sword Art Online: Alicization Lycoris* (2020; PS4, Xbox One), Mobile games como *Azur Lane* (2017; Android, ios), *Alchemy Stars* (2021; Android, ios), *Blue Archive* (2021; Android, ios), *Arknights* (2020; Android, ios) e *Disney: Twisted-Wonderland* (2020; Android, ios), além de visual novels como *Nekopara* (2014; Windows, macos), *Date A Live: Rio Reincarnation* (2017; Windows, PS4) e *Ensemble Girls!* (2012; Android, IOS).

Para que o modelo ilustrado e “rigado” se movimente, é necessário que haja um *software* de *tracking* que traduza os movimentos e expressões do usuário através de uma *webcam* ou do uso da câmera do celular. Existem vários softwares de *tracking* no mercado, os mais conhecidos e usados no mercado atualmente são Vtube Studio,

VUP, PRPR Live's e Animaze, todos são encontrados com facilidade na plataforma de distribuição de jogos digitais STEAM.

Vtube Studio é o mais utilizado na comunidade por ser o mais completo de todos, possui uma interface simples e intuitiva e com diversos recursos interessantes para alguém que irá utilizar o avatar para *lives*, como *dynamic lighting* que faz com que a luz da tela interaja com o modelo, edição e adição de parâmetros dentro do programa para se adequar a qualidade da captura, editor de expressões sem que necessite um arquivo específico para isso, tem sistema de itens no qual pode-se “colar” imagens no modelo e essas imagens irão seguir o movimento que o modelo fizer. O lado negativo deste programa é que ele pode demandar um pouco do computador do usuário caso este esteja fazendo várias coisas ao mesmo tempo como jogar, fazer *stream* e usar algum software de chamada, além também de ser um programa gratuito, porém com uma marca d'água na captura que pode ser retirada caso compre a versão paga que é vitalícia.

Quanto aos outros programas, a maioria tem coisas em comum com Vtube Studio com poucas diferenças. O programa VUP não conta com *dynamic lighting* nem com a edição de parâmetros dentro do programa, possui uma interface um pouco menos intuitiva que a do Vtube studio, demanda bastante do computador do usuário e não possui integração com câmeras de celular, o ponto positivo é que é um programa totalmente gratuito. Já o programa PRPR Live's não possui um editor de expressões, conta com o editor de parâmetros e um melhor *tracking* de movimento apenas em sua versão paga e sua interface também é um pouco menos intuitiva se comparada com o Vtube Studio. Por fim o Animaze é o que conta com a interface menos interativa entre todas as opções, não possui um editor de expressões e há apenas a opção de editar parâmetros não podendo adicionar novos e possui marca d'água na sua versão gratuita além de um limite de tempo de uso em *stream* de 90 minutos, o seu ponto positivo é que não demanda muito do computador do usuário.

Tabela 1 - comparação entre programas de tracking facial

(continua)

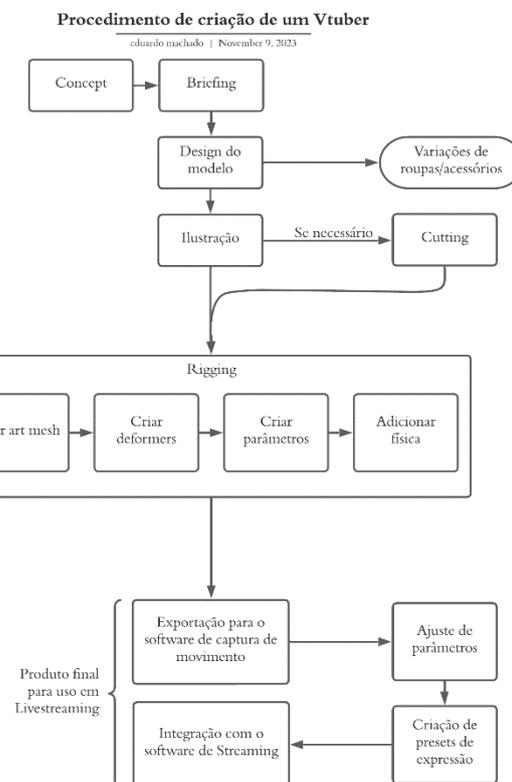
NOME	VTUBE STUDIO	VUP	PRPR LIVE	ANIMAZE
PREÇO	GRATUITO	GRATUITO	GRATUITO	GRATUITO
DYNAMIC LIGHTING	SIM	NÃO	SIM	SIM

EDIÇÃO DE PARÂMETROS	SIM	NÃO	SIM*	SIM
ADIÇÃO DE PARÂMETROS	SIM	NÃO	SIM*	NÃO
EDIÇÃO DE EXPRESSÕES	SIM	SIM	NÃO	NÃO
SISTEMA DE ITENS	SIM	SIM	SIM	SIM
MARCA D'ÁGUA (ANTES DO PAGAMENTO)	SIM	N/A	NÃO	SIM
MELHORA DE TRACKING APÓS PAGAMENTO	N/A	N/A	SIM	SIM
DEMANDA DE HARDWARE	MÉDIA	ALTA	MÉDIA	BAIXA

3 PROCEDIMENTOS METODOLÓGICOS

Este artigo explora os procedimentos metodológicos fundamentais para a criação de um modelo de Vtuber, desde a concepção do design até a implementação técnica.

Figura 3 - Processo de criação de um Vtuber



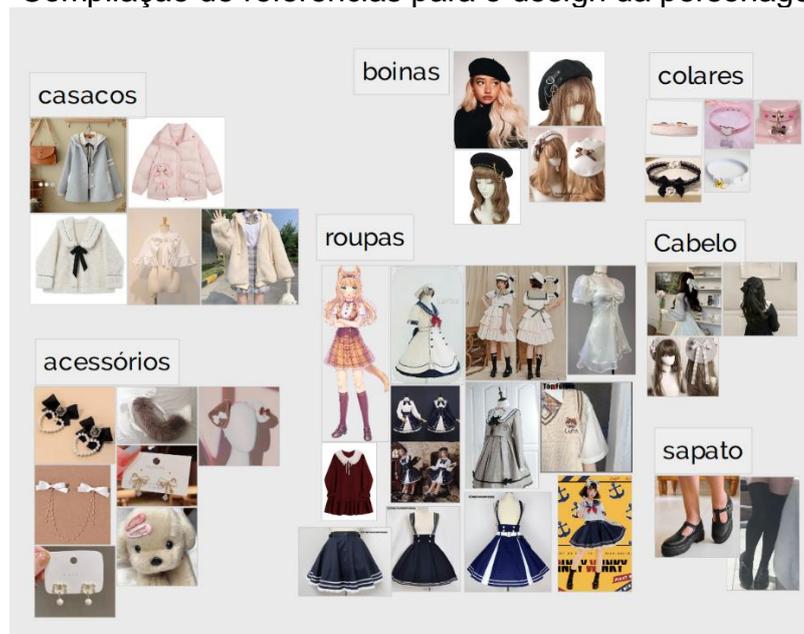
Fonte: Produção própria.

BRIEFING

O briefing é documento que serve como um guia para a execução do projeto, ele contém uma série de informações que são passadas ao artista pelo cliente. Ele é discutido no primeiro contato e é orientado ao cliente para organizar em um documento referências que auxiliem o artista a chegar a um produto final satisfatório.

O briefing foi documentado com a cliente e contém diversas imagens e descrições da visão dele sobre o produto final. Imagens essas que contém fotos de roupas, sapatos, acessórios e estilos que a personagem usa ou usaria, além de referências de outros personagens que se aproximam do que a cliente quer. Após o documento ser entregue ao artista com as imagens e descrições, ele foi organizado em um software de referências chamado Pureref e separado em seções: acessórios, boinas, cabelos, casacos, colares, roupas e sapatos.

Figura 4 - Compilação de referencias para o design da personagem



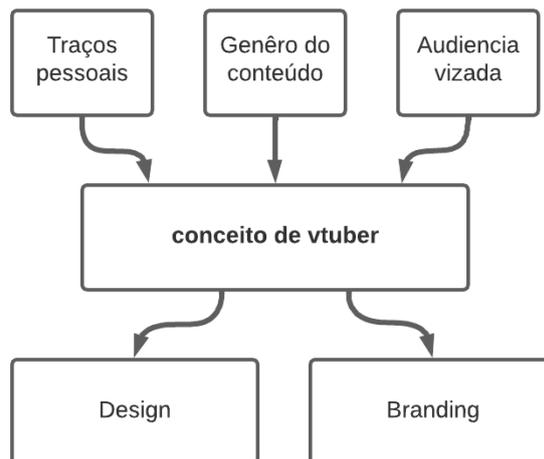
Fonte: produção Própria

O modelo se baseia em um conceito complementar que se baseia muito na aparência da cliente além de ressaltar alguns traços de personalidade. A cliente descreveu o conceito como uma estudante humana meio cachorro energética que gosta de jogar jogos. O modelo vai ser utilizado para transmitir *lives* na *twitch* com foco em games variados e na categoria “*Just chatting*” do site, categoria essa usada para *streams* em que o *streamer* fica conversando com o chat.

CONCEITO

O conceito é uma etapa crucial no processo de concepção de um personagem virtual que antecede a sua criação. O conceito de Vtuber pode ser entendido como uma espécie de introdução de um personagem e normalmente é expresso em uma única frase que descreve a identidade do personagem sem recorrer a recursos visuais. O conceito é a base para todas as características relacionadas a um Vtuber, incluindo a sua identidade e a sua ideia central. O conceito pode ser a razão pela qual um Vtuber faz sucesso ou não em um mercado saturado ou, mais importante ainda, se um Vtuber consegue se conectar com a sua personalidade e com a comunidade que busca construir.

Figura 5 - Processo de concepção de conceito de Vtuber



Fonte: “*What Many VTubers are Missing*” Disponível em:

https://youtu.be/s7OJTip0Rsw?si=jmwQagwM_sukGfla&t=206

Dois Vtubers com a mesma ideia de design podem apresentar execuções completamente diferentes, dependendo do conceito atrelado a eles. Um exemplo disso pode ser observado na grande diferença de atmosfera entre Gawr Gura e Artemis of the Blue (figura 6): ambos são designs de garotas tubarões, mas uma foi criada com um conceito bobo e fofo em mente, enquanto a outra foi criada com um conceito de uma jogadora mais madura e calma em mente. Suas personalidades, público-alvo e conteúdo também são diferentes.

Figura 6 - Gawr Gura e Artemis of the Blue



Fonte: HOLOLIVE oficial website. Disponível em: <https://hololive.hololivepro.com/en/talents/gawr-gura/> Artemis of the Blue no twitter. Disponível em: <https://twitter.com/ArtemisVtuber/status/1316823148021473280>

- **Conceitos únicos:** composto por conceitos de personagens que não são vistos com frequência e conceitos de personagens que têm uma combinação exclusiva de temas (figura 7). Garante que um Vtuber tenha uma marca interessante, uma história intrigante e um design memorável.

Figura 7 - Kobo Kanaeru, uma xamã da chuva que não gosta de fazer seu trabalho



Fonte: HOLOLIVE oficial site. Disponível em: <https://hololive.hololivepro.com/en/talents/kobo-kanaeru/>

- **Conceitos de estereótipos de anime:** alguns personagens de Vtubers têm seus conceitos baseados em estereótipos de personalidade de personagens de anime (figura 8). Isso é mais comum nos primeiros Vtubers de agências japonesas, como HOLOLIVE e NIJISANJI. Esse tipo de conceito de Vtuber

rapidamente saiu de moda à medida que o gênero de *Vtubing* se tornou mais popular e dominante.

Figura 8 - Natsuiro Matsuri, uma animadora de torcida muito energética.



Fonte: *HOLOLIVE* oficial site. Disponível em: <https://hololive.hololivepro.com/en/talents/natsuiro-matsuri/>

- **Conceitos complementares:** são conceitos de Vtuber que combinam muito com os traços de personalidade do ator ou da atriz (figura 9). Isso não precisa necessariamente incluir sua aparência física na vida real, mas pode.

Figura 9 - Kson, uma ex líder de uma gang japonesa que se tornou um lobo solitário.



Fonte: *Vshojo* website. Disponível em <https://www.vshojo.com/talent/kson>

- **Conceitos contrastantes:** são a antítese dos conceitos complementares. Um conceito de contraste pegará as características naturais de um ator ou atriz e selecionará o tema e o design que são o oposto direto, chamando a atenção pela quebra de expectativa da diferença entre o ator e o avatar (figura 10).

Figura 10 - Porcelain Maid, é um homem adulto encarnado em uma boneca de porcelana



Fonte: *Virtual youtuber wiki*. Disponível em: <https://virtualyoutuber.fandom.com/wiki/FoxyJoel>

- **Conceitos de grupo:** quando um Vtuber pertence a uma agência, geralmente é uma boa ideia fazer com que seu conceito corresponda ao restante do grupo ou da geração para que todos se encaixem tematicamente. A NIJISANJI e a HOLOLIVE também têm um tema que cada uma de suas gerações segue. Por exemplo, os membros da geração HOLOLIVE COUNCIL (figura 11) tem como base do design das personagens, retratar aspectos da criação do universo, como o tempo representado por Ouro Kronii, o espaço representado por Tsukumo Sana, o caos representado por Hakos Baelz, a natureza representada por Ceres Fauna e a civilização representada por Nanashi Mumei.

Figura 11- HOLOLIVE COUNCIL . Em sequência, Ceres Fauna, Tsukumo Sana, Hakos Baelsz, Ouro Kronii e Naashi Mumei.

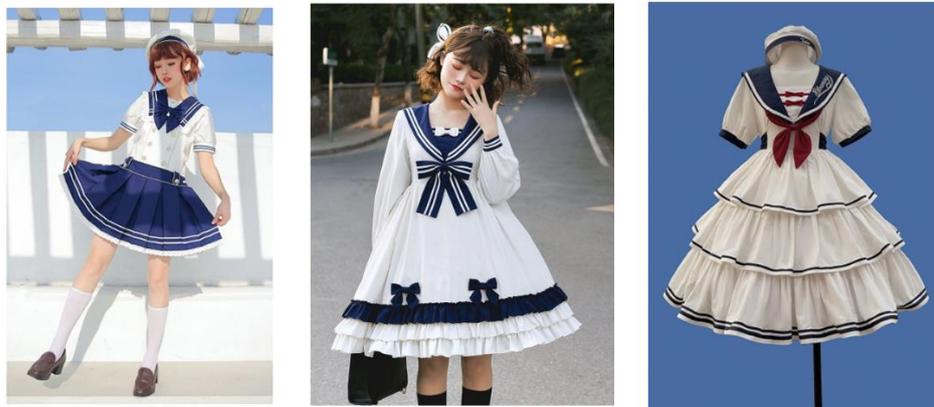


Fonte: "【Debut PV】 Council | #holoCouncil". Disponível em: <https://www.youtube.com/watch?v=SLnxCyzFgPw>

DESIGN DE PERSONAGEM

Depois de receber o briefing com todas as referências que a cliente passou é feito um estudo para entender um padrão nas ideias dele. No caso, é notado que o estilo de roupa que a cliente mostrou é muito próximo a um estilo de vestimenta muito popular no Japão, o estilo *Lolita fashion* que é um estilo japonês de moda cujas primeiras manifestações apareceram no fim da década de 70 e começo da década de 80. Inspiradas em parte na cultura '*kawaii*' japonesa e na nostalgia de outros tempos, as lolitas se dividem em vários sub-estilos e o que se achou mais adequado para o design da personagem foi o estilo "*sailor lolita*" que é inspirado em uniformes de marinheiros japoneses. Os uniformes escolares japoneses também se baseiam nessa estética, então foi definido que o design seria baseado nesse estilo de uniforme sendo os elementos mais proeminentes a saia em tom azul naval, a camiseta branca e o lenço azul no pescoço.

Figura 12- Exemplos de roupas do estilo *Lolita fashion*.



Fonte: *Sailor Lolita Salopette*. Disponível em: https://www.lolitawardrobe.com/your-gift-the-cute-sailor-sailor-lolita-salopette_p6055.html ; *Sweet Sailor Lolita Dress*. Disponível em: <https://www.lolitashow.com/Sweet-Lolita-Dress-Polyester-Long-Sleeves-Ruffles-Navy-Style-Dress-p997752.html> ; *Sailor Lolita OP Dress*. Disponível em: https://www.lolitawardrobe.com/gloaming-the-unknown-sea-sailor-lolita-op-dress_p7011.html

O resto do design das roupas foi mais baseado nas referências que a cliente sugeriu, com ênfase no colete de tricô que foi exigido que fosse para a versão final do design. O casaco é um casaco peludo de cor clara que reforça um elemento da estética *kawaii* japonesa e possui referências ao tema geral baseado em cachorro do design da personagem com os ossos pendurados no cordão do capuz. As meias $\frac{3}{4}$ e os sapatos foram a pedido da cliente. A Boina foi uma cópia exata de uma das

referências com apenas uma leve mudança de cor para se mesclar melhor com o modelo.

Para chegar ao design final da personagem foram pesquisadas algumas referências de modelos de outras Vtubers que possuem alguns elementos em comum com as referências que a cliente passou. Foi importante estudar essas personagens no processo de *Brefing*.

Figura 13 - *Model Sheets* para as personagens, Shirogane Noel, Takane Lui, Scarle Yonaguni e Inugami Korone.



Fontes: わたお no twitter. Disponível em:

https://twitter.com/wait_ar/status/1560629772161777665 ; かかげ no Twitter. Disponível em: <https://twitter.com/kakage0904/status/1464587625428185090> ; RIZ3 no Twitter. Disponível em: <https://twitter.com/ScarleYonaguni/status/1549579111550427136> ; フカヒレ no Twitter. Disponível em: https://twitter.com/fuka_hire/status/1122120707683471360

As referências contêm elementos que serão utilizados na concepção do design como os estilos das roupas, renderização dos elementos, acessórios e proporções do corpo. Com as referências reunidas passa-se para a fase de esboço do personagem. É recomendável começar pelo rosto, que é uma das partes mais importantes de uma Vtuber visto que é a parte que mais vai aparecer durante uma transmissão ou vídeo, além de ser a parte mais complexa de fazer o *rig* por conter muitos elementos complexos que se mexem como a boca e os olhos.

3.1.1 Evolução do design

Primeiro passo foi fazer uma base simétrica para o corpo da personagem. Com a base pronta se começa a criar os elementos principais do design, focando principalmente no rosto do modelo. Com o rosto pronto começa-se a testar roupas que se encaixam no briefing criado, utilizando-se de vários elementos pesquisados e também testando várias combinações desses elementos para entender melhor o que funciona e o que não funciona no design.

Figura 14 - Teste de diferentes tipos de roupas.



Fonte: produção própria.

Depois de definido um design para a roupa, é feito um teste de cores para ver quais cores se encaixam melhor no conceito da personagem.

Figura 15 - Teste de diferentes esquemas de cores.



Fonte: produção própria.

Com as cores selecionadas o design passa para o processo final de finalização, adicionando detalhes e fazendo um *model sheet* para referência no futuro caso queira ser feito alguma alteração por parte da cliente

Figura 16 - Resultado final da ilustração.



Fonte: produção própria.

ILUSTRAÇÃO E CUTTING

Para a produção da ilustração desse modelo foi utilizado o Clipse Studio produzido pela empresa japonesa de software gráfico Celsys. O motivo de ter escolhido esse aplicativo ao invés de outro se deve ao fato de possuir mais familiaridade com o programa que conta com uma loja dentro do programa no qual pode se encontrar diversas ferramentas gratuitas que auxiliam na hora de fazer detalhes na ilustração, além de contar com modelos 3D editáveis que ajudam em poses mais complexas.

Para a ilustração final foi escolhido um estilo de renderização semi-realista utilizando um *cell shading* como base e depois usando métodos de ilustração para mesclar as cores das sombras com a cor base.

Cutting é o processo de separação da ilustração em partes para que cada parte seja trabalhada individualmente para gerar uma sensação de movimento mais convincente.

A ilustração começou preparando um arquivo de bitmap dentro do software de ilustração, foi criada um *canvas* de 5000px x 5000px para que o modelo tivesse uma alta resolução. Logo após começou um processo de criação de *layers* e organização em grupos. A princípio foi criado um grupo para referências para que se fosse possível

visualizar com rapidez elementos do design presentes na personagem. Foram criadas camadas para o rosto, cabelo, tronco e pernas como base da organização de camadas. Além disso foi utilizado uma marcação em cada camada para indicar de qual lado cada parte da ilustração está, elementos que estão na direita da ilustração tem a letra “R”, da palavra *right*, no nome da *layer* enquanto elementos que estão na esquerda tem a letra “L”, de *left*, no nome. A organização em pastas e com indicativos é muito importante para quando o arquivo for exportado para o programa de *rigging* pois ajuda a localizar melhor os elementos que vão ser trabalhados.

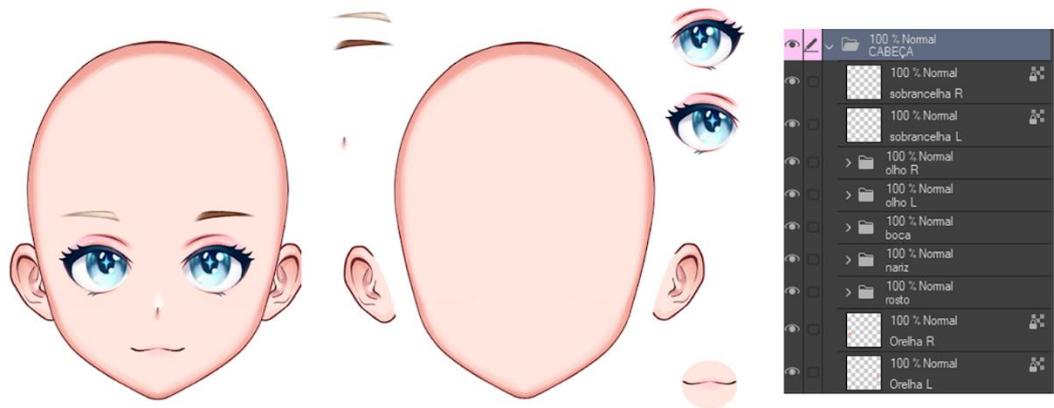
A ilustração em sua maior parte se utilizou da ferramenta de simetria do software para garantir que a personagem está proporcional dos dois lados, a única exceção a essa regra foi na hora de fazer o cabelo e a saia pois se fossem simétricos teriam um aspecto artificial.

3.1.2 Cabeça

Primeiramente começa-se pelo rosto que é a parte mais importante da ilustração, já que é a parte que mais vai se mexer e a que mais vai aparecer quando estiver em *live*. Primeiro começa-se a desenhar a base da cabeça dentro de uma pasta que irá conter os elementos da cabeça, seguindo o esboço da referência da base e separando as camadas de *lineart*, cor base e sombras da cabeça, respectivamente. A *lineart* fica acima das outras enquanto as sombras ficam acima da camada de cor base com a opção de *clipping mask* ativada para que quando chegar na parte de *rigging* seja possível mexer a sombra do rosto separadamente da base e criar um efeito de tridimensionalidade mais realista. Além dessas camadas também existem as de sombras do cabelo que serão melhor explicadas mais à frente.

São criadas então outras camadas de separação para o rosto. As orelhas são desenhadas usando simetria e depois são separadas cada uma em layers diferentes e renomeadas de acordo, o mesmo ocorre com as sobrancelhas. Para resto dos elementos como olhos, nariz e boca são necessários cada um uma separação mais complexa.

Figura 17 - Exemplo de separação e organização das peças da cabeça.

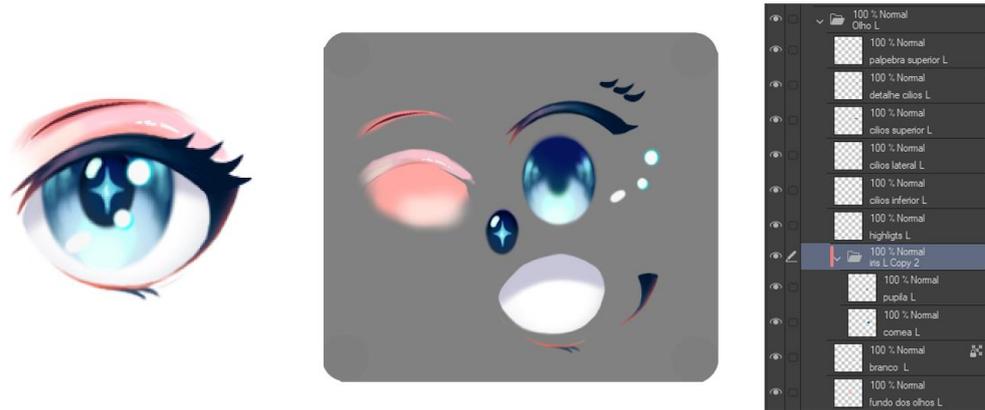


Fonte: produção própria.

Da parte da cabeça os olhos são os mais importantes, pois é a área que mais chama atenção em um personagem. Para que o olho de um Vtuber fique de boa qualidade é necessário pensar não apenas na área estética, mas também pensar no trabalho que vai ter no *rigging*. A princípio o mais importante nos olhos para fazer um rig bom são, o fundo do olho, a íris, e os cílios superiores e inferiores. Para a personagem foi adicionado alguns detalhes a mais nos olhos pensando em como ficariam eles no produto final. Foram criadas camadas de pálpebra e de fundo de olho (esclera), além de cílios laterais e como detalhe extra na parte superior. A íris foi dividida em duas camadas, a camada de pupila e a camada da córnea para que o olho possua uma tridimensionalidade melhor na hora em que a personagem olhar para os lados e virar a cabeça. Cada um dos brilhos dos olhos também foi separado individualmente em uma camada específica para que eles sejam animados individualmente na hora do *rig*, dando uma característica de um olhar mais vivo.

O olho foi separado em camadas distintas antes do processo de ilustração para facilitar na hora de fazer o desenho do mesmo. Depois de finalizar um olho, este foi copiado, colado e invertido horizontalmente no *canvas* com o ponto de ancoragem exatamente no centro da ilustração para que houvesse simetria no personagem. As únicas alterações feitas na cópia foi a renomeação do final do nome das camadas de "L" para "R" e a inversão do *highlights* para que houvesse coerência na direção da luz.

Figura 18 - Exemplo de separação e organização das peças do olho.



Fonte: produção própria.

A boca é a segunda parte mais importante e complexa do modelo. Sua importância é devido ao fato de que é necessário um bom *rig* para ter um movimento realista e convincente e para que se obtenha um bom *rig* é necessário que as partes da boca estejam separadas corretamente e que por conta disso gera a complexidade. Para uma boca simples são necessários o lábio inferior e o lábio superior, de preferência com uma linha fazendo a separação deles, e também o fundo da boca.

Para a boca da personagem as camadas foram divididas em lábio superior, lábio inferior, dente superior, dente inferior, dente de trás, língua e fundo da boca. Todos os elementos, com exceção dos lábios, estão com *clipping mask* na camada de fundo da boca para dar a sensação que os mesmos estejam dentro dela. Os dentes foram separados dessa forma para melhorar o efeito de tridimensionalidade do modelo.

Figura 19 - Exemplo de separação e organização das peças da boca.



Fonte: produção própria.

O nariz é a parte mais simples de todo o rosto, porém é a parte que melhor vende uma rotação no rosto na hora do movimento. O nariz é onde mais se variam as formas de se desenhar, desde designs mais complexos com várias formas mais sólidas até simplesmente se resumindo a uma linha. Foi escolhido o design mais simples de ser representado por uma linha para poupar tempo na produção. Ele consiste de uma linha representando a ponta do nariz, uma luz e uma sombra para ajudar a sensação de volume do nariz junto com a sombra em uma camada separada para trazer uma sensação de tridimensionalidade.

Figura 20- Exemplo de separação e organização das peças do nariz.



Fonte: produção própria.

Por fim, a última coisa a se adicionar no rosto foi a sombra do cabelo, feito copiando a forma da franja e das mechas e colocando em uma camada separada, acima da camada de cor base da pele e utilizando essa camada como uma máscara para a sombra do cabelo.

Figura 21 - Exemplo de organização das sombras da cabeça.



Fonte: produção própria.

3.1.3 Cabelo

O cabelo é muito importante para vender a ideia de profundidade em um modelo de Vtuber pois as camadas dele são bem espaçadas uma da outra, por exemplo, na imagem a seguir, a “franja” (em verde) e as “mechas laterais” (em rosa e amarelo) ficam em frente a todas as outras camadas da cabeça e do corpo. Já as “mechas atrás” (em roxo) ficam atrás das camadas da cabeça, porém ficam à frente do corpo para dar a sensação de uma mecha de cabelo que está saindo de trás da orelha e está sobre o peito. Por último o “cabelo atrás” (em vermelho escuro) fica por trás de todas as outras do modelo.

Figura 22 - Exemplo de separação e organização das peças do cabelo.



Fonte: produção própria.

A franja é feita de apenas uma única camada devido ao fato de ser uma parte do cabelo que não possui muito movimento e normalmente acompanha o movimento do rosto. Uma mudança que poderia ter sido feita seria separar cada mecha da franja para dar mais movimento, porém por se tratar de uma pequena faixa de cabelo não se aproveitaria muito do movimento individual de cada parte e seria um desperdício de tempo fazer.

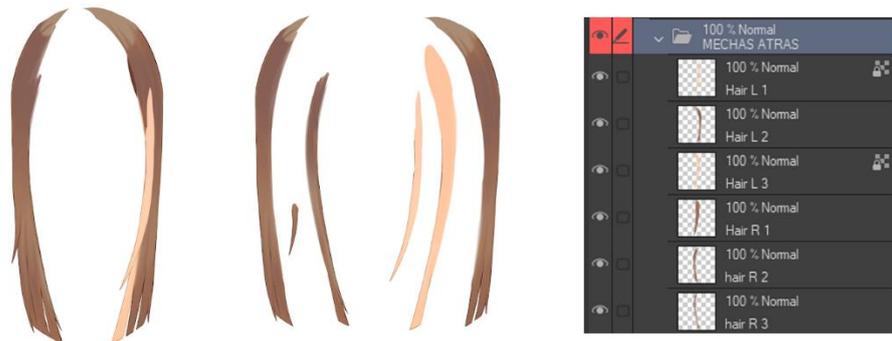
Figura 23 - Exemplo de ilustração da Franja.



Fonte: produção própria.

Por outro lado, as mechas laterais que emolduram o rosto são mais longas e são mais afetadas pelo movimento do rosto, por esse motivo é interessante fazer essa parte de maneira mais complexa. Aqui, cada lado das mechas foi separado em 3 partes que serão animadas individualmente. É importante notar que para se ter uma boa coesão na hora do movimento, é necessário que se atente na hora de separar as camadas e em como a *lineart* de cada elemento vai interagir um com o outro para que na hora de animar não fique muito aparente a separação das camadas.

Figura 24 - Exemplo de separação e organização das peças de uma parte do cabelo.



Fonte: produção própria.

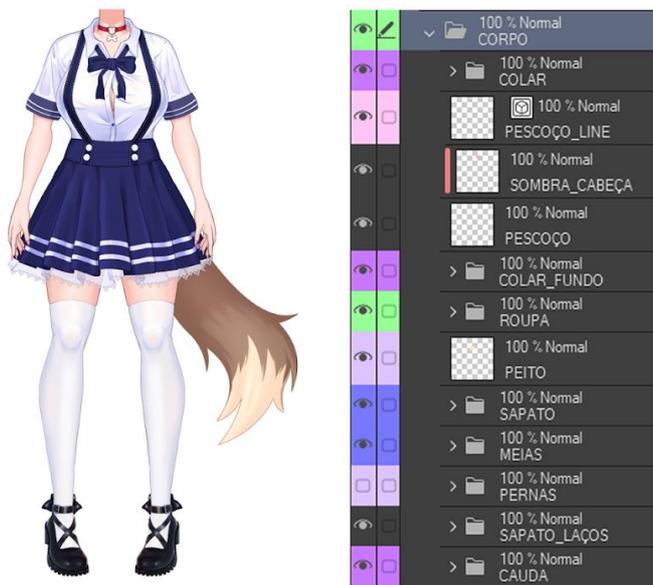
Para as mechas laterais que ficam por trás do rosto não foi feita uma separação para vender a impressão de que essa parte do cabelo esteja acomodada sob o peito da personagem. E por fim, a parte de trás do cabelo também é composta de uma única camada, pois não é uma parte com muito movimento na parte mais próxima à cabeça, o que irá vender a ideia de tridimensionalidade são as ferramentas de *deform* no

programa de *rigging*. Já a parte mais abaixo tem mais movimento, porém para vender essa ideia não é necessário que as camadas estejam separadas já que é uma peça grande no modelo, a distância entre cada mecha é o suficiente para dar movimento na peça.

3.1.4 Corpo e roupas

O corpo é uma parte mais complexa de organizar as camadas por conta de sobreposições que acontecem com camadas de roupa e acessórios, assim como partes do corpo como partes do cabelo e dos braços. Por exemplo, os braços estão divididos entre “braço” e “antebraço”, o “braço” está em uma camada junto a camiseta, que está abaixo da saia, então para que o “antebraço” aparente estar sobre a saia ele precisa estar em uma pasta separada do “braço” e algumas camadas acima.

Figura 25 - Exemplo de organização das camadas do corpo.

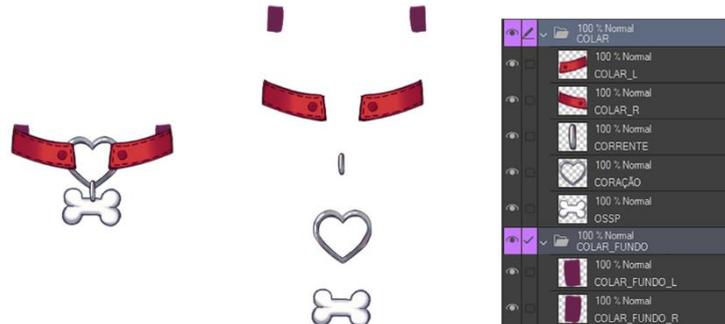


Fonte: produção própria.

O colar é outro exemplo de como as camadas precisam estar bem separadas para ajudar no *rigging*, por exemplo, o fundo do colar está separado em uma pasta abaixo pois precisa aparentar estar por trás do pescoço. Além disso o colar tem outras separações especiais, como a corrente e o osso que foram separados para que cada um se movimenta exclusivamente na hora de animar o modelo, e o coração foi

separado do colar para que ele não se esticasse na hora de *deformar* por se tratar de um material rígido.

Figura 26 - Exemplo de separação e organização das peças do colar.



Fonte: produção própria.

A camiseta foi uma das partes mais complexas de se separar por conta da variação de roupas que a personagem faz onde o modelo pode vestir um colete que por se tratar de uma peça de roupa sem mangas, alguns elementos da camiseta apareceriam também, então para que não fosse criado os braços novamente, optou-se por mantê-los separados em uma pasta própria que servisse para ambas as opções, não necessitando de um segundo rig. Além dos braços a gola também foi separada em uma pasta própria acima das outras camadas pelo mesmo motivo da pasta do braço.

Figura 27 - Exemplo de separação e organização das peças da camisa.



Fonte: produção própria.

O colete segue a mesma lógica da camisa na hora de separar suas partes e usa as pastas mencionadas anteriormente.

Figura 28 - Exemplo de separação e organização das peças do colete.



Fonte: produção própria.

A saia possui um movimento muito simples então não é necessário que haja muita separação. A única separação que existe é a separação da saia superior e da saia de baixo para que ambas tenham movimentos separados para dar mais profundidade ao modelo.

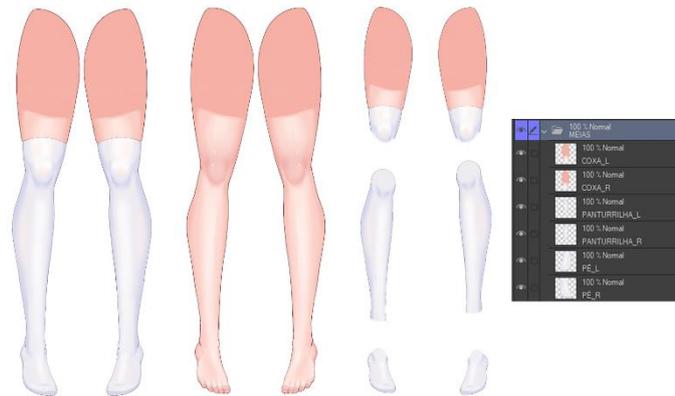
Figura 29 - Exemplo de separação e organização das peças da saia.



Fonte: produção própria.

Para a ilustração das pernas foram feitas versões com e sem a meia para trazer uma variedade de opções para a cliente. A versão sem a meia foi colorida de acordo com a mesma paleta de cores das camadas de pele feitas anteriormente dando uma atenção especial aos joelhos por ser a parte que conecta as duas partes da perna, que se não for renderizada cuidadosamente a movimentação pode mostrar algumas falhas na pintura. As pernas foram separadas de acordo com as articulações.

Figura 30 - Exemplo de separação e organização das peças da perna.



Fonte: produção própria.

A ilustração dos braços segue a mesma lógica da ilustração das pernas, porém sem se atentar ao cotovelo pois ele está virado para trás. A sua separação também segue sendo nas articulações do cotovelo e do pulso.

Para a ilustração dos sapatos foi necessário fazer muitas pesquisas de como renderizar o material que é feito de couro. O sapato foi separado apenas entre sapato e salto para que o pé fosse animado para parecer que está girando. Os laços foram separados em uma pasta separada pois ficariam por trás das pernas.

Figura 31 - Exemplo de separação e organização das peças do sapato.



Fonte: produção própria.

O laço foi uma simples ilustração de tecido com apenas dois tons. Para a separação das partes foi pensado que o suporte ficasse separado do resto para ficar debaixo da gola para dar a impressão de sobreposição. Já o resto foi separado para se mover individualmente com fluidez.

Figura 32 - Exemplo de separação e organização das peças do laço.



Fonte: produção própria.

A cauda foi renderizada de forma parecida com a do cabelo, se apropriando da mesma paleta de cores, junto com alguns detalhes para parecer mais peluda e ficar mais parecida com a cauda de um cachorro. Por se tratar de uma parte que vai se mover pendularmente não é necessário que haja uma separação de camadas.

LIVE 2D

O mercado de Vtubers é bastante recente, sendo provável que muitos profissionais na área de animação, internet e audiovisual ainda não tiveram contato com softwares para a criação de animações em Vtubers. Sendo assim, este texto apresenta aqui um detalhamento técnico minucioso de como funcionam as animações no software Live 2D.

3.1.5 Parts

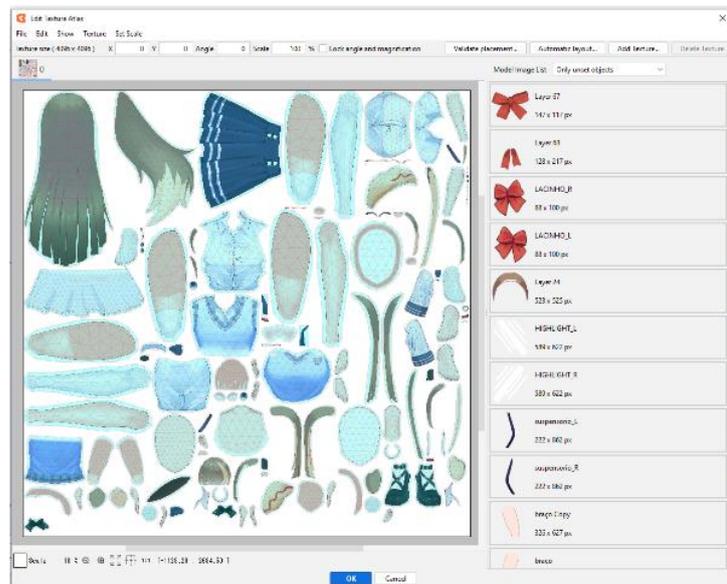
O menu “parts” é uma parte do layout que contém o agrupamento dos componentes de um personagem (por exemplo, olhos, nariz, etc.) separados em pasta igual ao arquivo PSD. Ao se criar *deformers*, *deform paths*, *rotation deformers* etc... todos esses elementos aparecem organizados nessa janela.

3.1.6 Textura

A textura refere-se a arte do modelo, ou seja, tudo que está ligado ao arquivo PSD que foi criado na hora de exportar o modelo. Os dados da textura estão dentro do arquivo PSD e o arquivo PSD está conectado com o arquivo do modelo dentro do live2d, porém algumas informações do arquivo vão ser modificadas quando importadas pelo programa, por exemplo, máscaras de camada, efeitos de camada e

transparência não vão aparecer quando abrir o arquivo e terão que ser configuradas dentro do programa. O live2d apenas suporta 3 tipos de efeitos de camada, que são “normal”, “multiply” e “additive”. Para o modelo funcionar de maneira eficiente cada layer do PSD é colocado em um mapa quadrado com fundo transparente que é chamado de “texture atlas”, que é equivalente ao “texture map” usado no 3D.

Figura 33 - Exemplo do *Atlas texture* da modelo.



Fonte: produção própria.

3.1.7 Artmesh e Polígonos

Assim que o modelo é criado no programa, cada layer de textura é atribuído a uma malha e colocada no canvas, cada uma dessas malhas é chamada de *artmesh*. O *artmesh* inicialmente é quadrado, se adequando ao tamanho da textura, mas pode ser editada para se adequar melhor a textura. Cada *artmesh* é composta de polígonos. Polígonos se referem a cada segmento triangular em cima da textura. Cada ponto no triângulo é chamado de vértice, que pode ser movido individualmente ou em grupo, alterando a forma do triângulo, e quando a forma do triângulo é alterada, a textura a qual o *artmesh* está ligado acompanha essa alteração.

3.1.8 Parâmetros

Define o estado (forma, tamanho, posição, opacidade, ordem de camadas/desenhos) da *Artmesh* & *deformadores* através de *keyforms*. Para alterar

um parâmetro é necessário um duplo clique no parâmetro que quer se modificar, fazendo isso irá abrir uma janela de diálogo que é o *keyform* editor. Para adicionar um *keyform* basta clicar em qualquer lugar na linha, e clicar novamente para remover. O *Keyform* define o estado do determinado objeto, ou seja, cada ponto criado armazena a informação do movimento.

3.1.9 Keyform

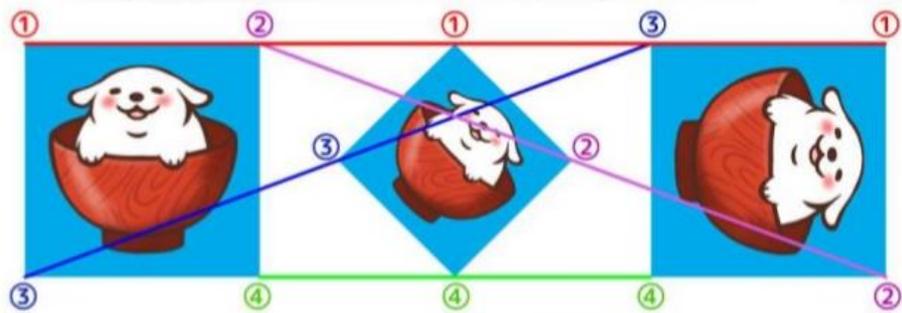
Keyform são os pontos criados dentro de um parâmetro, eles atuam como *keyframes* em uma animação.

3.1.10 Deformer

O programa permite *deformar* objetos movendo vértices individualmente na malha para alterar o formato da textura. No entanto, é muito demorado mover manualmente os vértices um por um para ajustar a forma para o jeito desejado. Para isso existe uma ferramenta chamada *deformer*. Os *deformers* são divididos em *warp deformer*, *rotation deformer* e *deform path*.

O *warp deformer* cria uma malha retangular em cima da *mesh* criada, pode-se determinar a quantidade de quadrantes dentro dessa malha. Com ela pode-se mover áreas da *artmesh* que estão dentro de cada quadrante movendo cada vértice dentro dela, além de aumentar, diminuir e rotacionar. Ao rotacionar o *warp deformer* deve-se ter em mente que a *mesh* irá fazer uma interpolação linear, ou seja, cada vértice ira do ponto A para o ponto B em linha reta, o que pode acarretar em *deformações* indesejadas na malha.

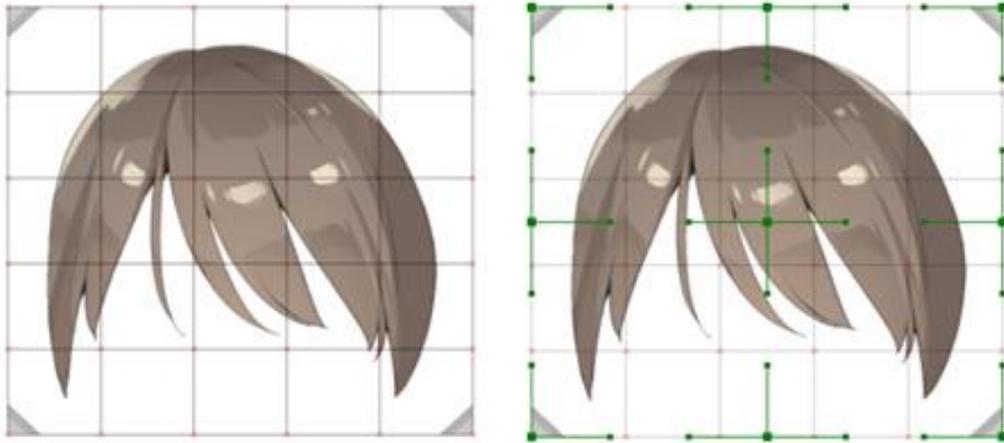
Figura 34 - Demonstração de como funciona a interpolação linear do *warp deformer*.



Fonte: Live2D Manual: Disponível em: <https://docs.live2d.com/en/cubism-editor-manual/deformer/>

Quanto maior o número de quadrantes, mais precisa será a *deformação* da *artmesh*. Além do número de quadrantes na malha, pode-se determinar quantos pontos vão ser divisões *bezier*. As divisões *bezier* são pontos com âncoras que controlam os vértices ao redor deles. Essa opção é muito utilizada pois cria uma *deformação* mais uniforme, sendo geralmente utilizadas as configurações de 2x2 ou 3x3.

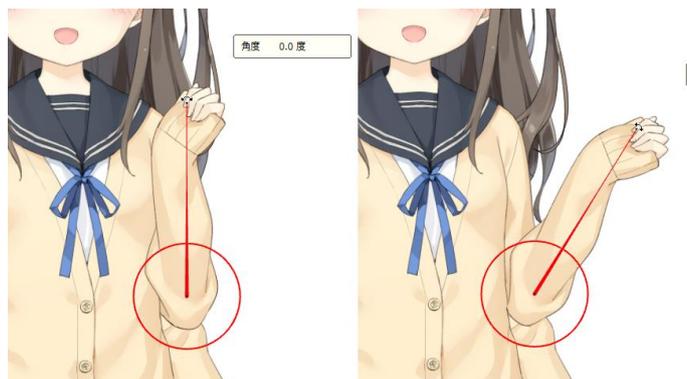
Figura 35 Exemplo de *mesh* sem as divisões *bezier* e com as divisões *bezier*.



Fonte: Live2D Manual: Disponível em: <https://docs.live2d.com/en/cubism-editor-manual/making-and-placement-of-warp-deformer/>

O *rotation deformer* segue o mesmo conceito do *warp deformer*, porém ao invés de *deformar* a malha dentro dele ele funciona como uma âncora de rotação. Ele é geralmente utilizado para o movimento de membros do corpo como os braços e as pernas. Este *deformer* irá rotacionar todos os elementos dentro dele, inclusive outros *deformers*.

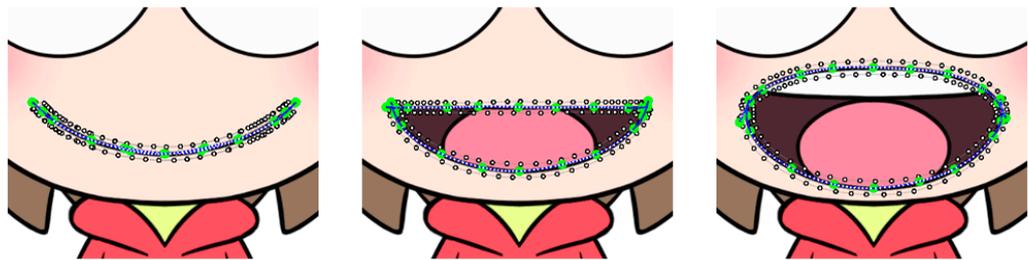
Figura 36 - Exemplo de como funciona o *rotation deformer*.



Fonte: Live2D Manual: Disponível em: <https://docs.live2d.com/en/cubism-editor-manual/making-and-rotation-of-rotationdeformer/>

Por fim, o *deform path* são pontos criados em uma malha que juntos formam uma linha contínua, ao mover estes pontos a malha irá se mover junto da linha criada. Por conta dessa funcionalidade esse *deformer* possui características que facilitam a *deformação* de algumas partes grandes do modelo, como o cabelo, além também de facilitar a *deformação* de alguns elementos finos como a boca e os olhos.

Figura 37 - Sequência de imagens demonstrando como funciona o *deform path*.



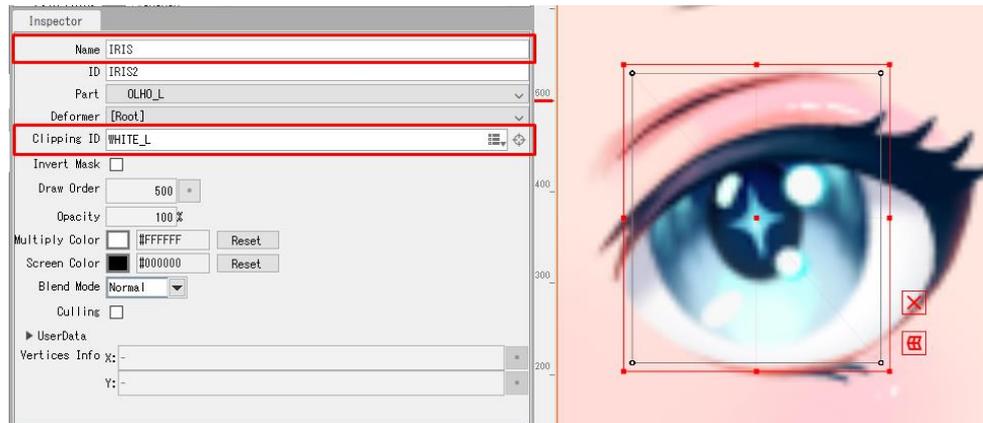
Fonte: Live2D Manual. Disponível em <https://docs.live2d.com/en/cubism-editor-manual/deformpath/>

WORKFLOW LIVE 2D

Existem diversas formas de se abordar o workflow do processo de *rigging* no live2d. O workflow a seguir foi escolhido pois se atenta primeiro aos detalhes principais da cabeça da personagem e apenas depois a outras partes do modelo que aparecerão menos em uma *livestream*.

Para começar, basta arrastar o arquivo em PSD para dentro do programa, seleciono todas as partes e crio o *texture atlas* com o atalho “ctrl+t”. Depois selecionar as camadas que no arquivo original se utilizavam de *clipping masks*, os olhos, a boca e os óculos, ajustando cada uma no programa para recriar o *clipping mask* original dentro do live2d.

Figura 38 – Demonstração do *clipping mask* da íris no olho.



Fonte: produção própria.

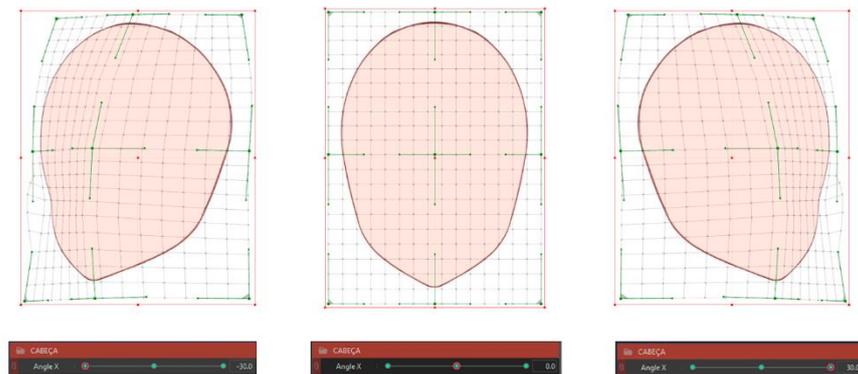
Depois de ajustar as máscaras, inicia-se a edição do *artmesh* de cada textura individualmente pois cada textura começa tendo apenas 4 vértices formando um retângulo na textura, o que não é ideal pois a *deformação* ficaria ruim. Existem 2 modos de fazer o *artmesh* em uma textura, manualmente ou com a opção de *automesh*, cada um com sua vantagem. O *auto mesh* que é a opção automática é interessante para certas ocasiões como por exemplo uma peça muito grande no modelo, como por exemplo a saia, ou uma peça que precise de muita *deformação*, por exemplo a cauda. Esses ajustes de *automesh* são configuráveis e possuem predefinições baseada na finalidade da *mesh*.

Por exemplo, a cauda é um elemento que foi utilizado o *auto mesh* para fazer a malha, com a predefinição de *heavy deformation*, e foram feitos alguns ajustes nas configurações como *vertice spacing* que é a distância média do espaço entre os vértices, fazendo aumentar ou diminuir a quantidade de triângulos existentes na malha, além de configurações como *boundary margin* que é a distância em que os vértices ficam da borda da textura, interna e externamente. Porém a *mesh* não fica perfeita, então é necessário fazer alguns ajustes para a malha ficar boa, modificar coisas como vértices desnecessários corrigir arestas para que elas formem triângulos mais padronizados etc... Depois do *artmesh* de todas as camadas do modelo forem finalizadas pode se dar início a edição dos parâmetros para dar movimento ao modelo. O programa começa com alguns parâmetros padrões, como movimentos x e y, movimento dos olhos, movimentos da sobrancelha, etc...

3.1.11 Cabeça

Primeiramente, inicia-se preparando alguns elementos da cabeça. Depois de criar a *mesh* da camada da *artline* do rosto, cria-se uma cópia desta *artmesh*, copiando na *artmesh* do rosto para que ambas tenham a mesma *artmesh*. Para que ambas as partes se movam juntas é necessário aplicar o *glue* que faz com que as camadas “coladas” se movam igualmente mesmo estando em layers diferentes. Com as duas camadas já coladas, utiliza-se um *deformer* que é um plano com vários pontos no qual eu posso modificar a *mesh* de maneira uniforme. Com o *deformer* criado na camada da face, cria-se um parâmetro para este *deformer* chamado de ângulo x que é o parâmetro que vai controlar a forma como que a cabeça deve *deformar* no eixo x. O parâmetro conta com 3 *keys*, a do meio que é a padrão, que representa estar olhando para frente, um *key* com valor de 30 e um de -30 que representam olhar para a direita e para esquerda respectivamente. Para que o rosto vire para a direita eu devo primeiro ajustar o valor do parâmetro para ficar no valor máximo de 30, e então, modifica-se o *deformer* fazendo com que a arte do rosto aparente olhar para a direita, e para ficar um movimento simétrico eu utilizo a função de *reflect motion* que reflete o movimento do *deformer* no valor oposto do parâmetro.

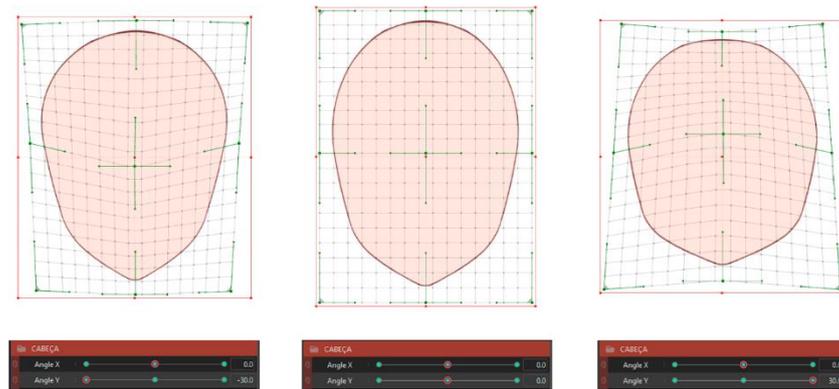
Figura 39 - Demonstração do movimento da cabeça no ângulo X.



Fonte: produção própria.

Repete-se o mesmo procedimento para o eixo Y para fazer com que a cabeça olhe para cima e para baixo.

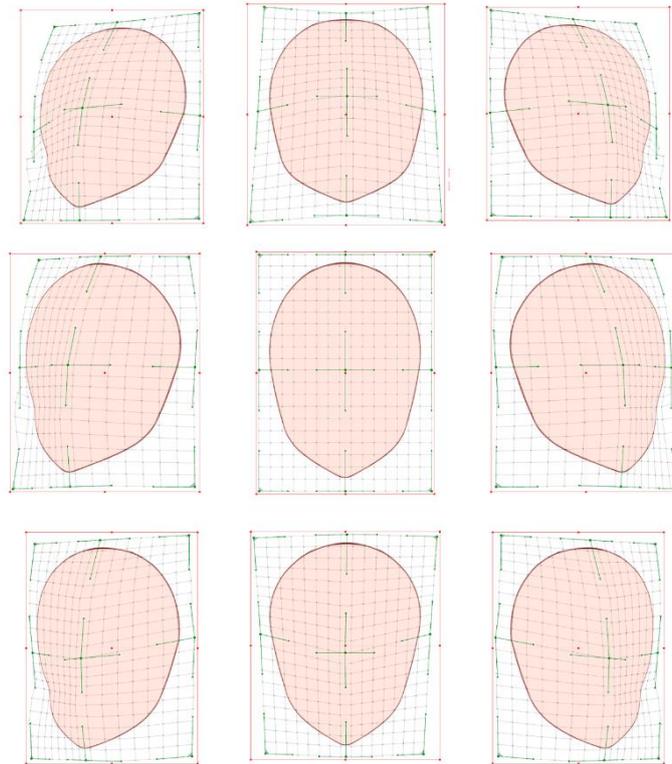
Figura 40 - Demonstração do movimento da cabeça no ângulo Y.



Fonte: produção própria.

Depois dos ângulos x e y feitos, une-se os parâmetros e eles formam uma grade com 9 keys que são os estados das posições, porém os valores dos cantos não misturam os ângulos x e y, para que isso seja feito existe uma opção que se chama *synthesize corners* que irá fazer um cálculo da malha e então irá formar os 4 cantos dos parâmetros, com isso a cabeça já possui a impressão de estar virando em todas as direções. Caso queira fazer algum ajuste em algum dos cantos, é possível modificar a malha nos cantos para se adequar ao desejado, depois de finalizar o ajuste em um canto, o que deve ser feito é desvincular os parâmetros e usar novamente a ferramenta de *reflect motion* para manter a simetria.

Figura 41 - Demonstração do movimento da cabeça com o ângulo X e Y linkados.



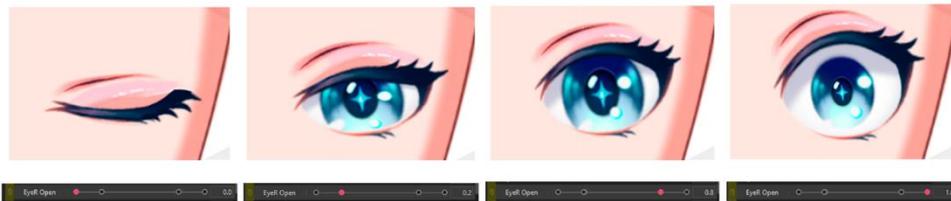
Fonte: produção própria.

Com o movimento da cabeça finalizado, mantém-se ela na posição padrão (olhando para frente) e inicia-se o rig dos olhos começando pelo movimento da íris. Para começar, selecionam-se as camadas das artes que fazem parte da íris, que no caso do modelo são as camadas íris, pupila e os highlights, permitindo criar um *deformer* contendo esses elementos. Com o *deformer* criado, segue um procedimento parecido com o da cabeça criando um parâmetro X com 3 *keys* para a pupila se mover na horizontal e um parâmetro Y também com 3 *keys* para a pupila se mover na vertical, porém desta vez os valores vão variar de -1 até 1 por se tratar de um valor menor. Juntando os dois parâmetros sintetiza-se os cantos, com isso o movimento da pupila já está pronto. Para criar um efeito de tridimensionalidade maior, cria-se um *deformer* para a pupila dentro do *deformer* da íris, novamente 3 *keys* no parâmetro X e no parâmetro Y, fazendo com que o nos extremos a pupila se mova menos que a íris dando a impressão de que a pupila está mais para trás.

Com o movimento da íris pronto, o próximo passo é o processo de riggar o movimento de abrir e fechar o olho. Seleciona-se todas as artes que fazem parte do movimento que no caso seriam todas as camadas de cílios, a pálpebra e a esclera.

Cria-se um parâmetro “eye open” com 4 *keyframes*, para isso deve-se utilizar a ferramenta de criar *keyforms* manualmente, criando *keys* nos valores de 0, 0.2, 0.8 e 1. Definir o valor de 0.8 como padrão para funcionar como o olho aberto. Com os *keys* criados, posiciona-se no *key* 0 e inicia-se a edição das *meshes* para uma posição que pareça com que o olho esteja fechado, ajustando individualmente cada *mesh*. Para auxiliar nessa tarefa é utilizada a ferramenta de *path deform*, no qual cria-se pontos na *mesh* formando uma linha e a ajustando até que fique alinhada com a arte e então ajusta-se os pontos para que a *mesh* fique na forma desejada e aplico a mesma técnica para as outras partes. Para o *key* 0.2, deixa-se o olho em uma posição entreaberta e no *key* 1 eu em posição de um olho arregalado.

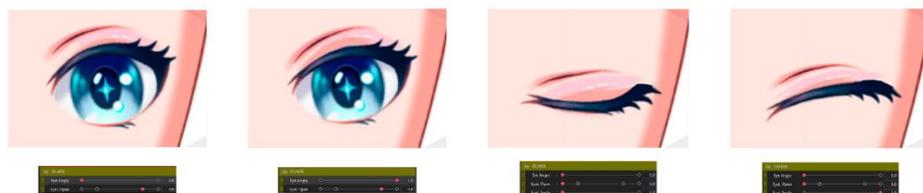
Figura 42 - Demonstração dos *keys* do parâmetro “EyeL open”.



Fonte: produção própria.

Foram feitos também 2 parâmetros para modificar a expressão da personagem, um parâmetro com os olhos como se estivessem com raiva e um parâmetro como se os olhos estivessem sorrindo. Esses parâmetros possuem apenas 2 *keys*, um 0 padrão que é o estado que o olho fica normalmente, e um *key* no 1 no qual é quando o olho fica com aquela expressão. O método de fazer é parecido com o parâmetro de olho aberto apenas ajustando as *meshes* necessárias para dar a impressão da expressão no *key* 1.

Figura 43 - Demonstração dos *keys* do parâmetro “EyeL smile”.

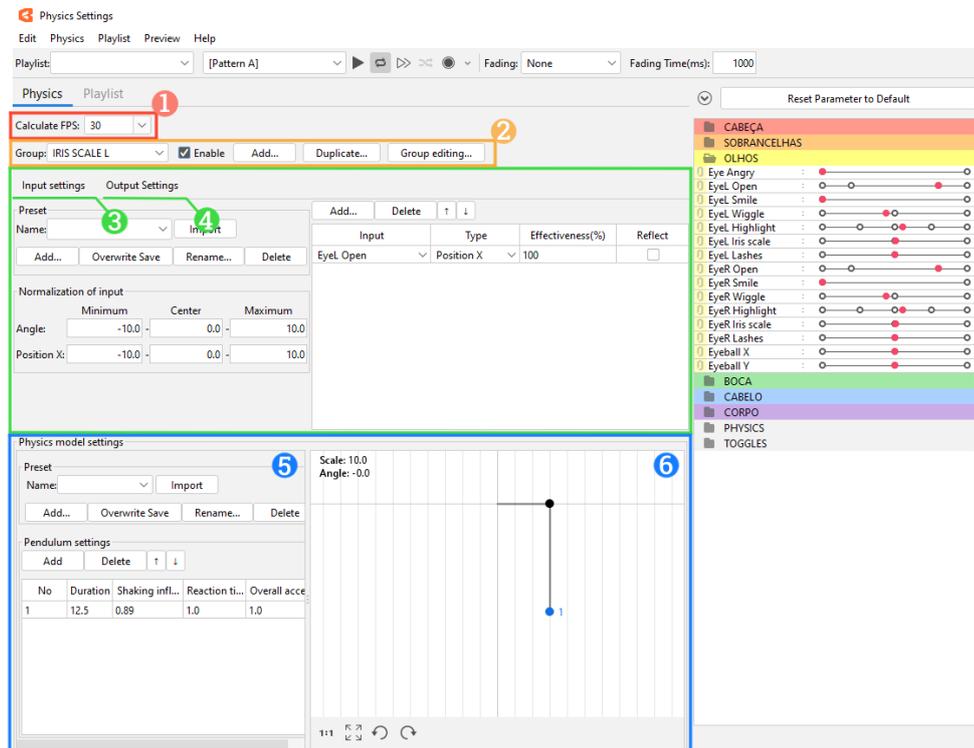


Fonte: produção própria.

Com isso, todos os movimentos básicos do olho já estão prontos, porém o movimento está muito robótico. Para dar mais vida ao movimento dos olhos existem recursos de animação que vão dar mais dinamismo ao movimento, como por exemplo adicionar efeitos de *squash* e *stretch* e movimentos de *overshoot*. Por exemplo para a íris eu fiz com que quando os olhos fecham ela faz o efeito de *squash* e quando os olhos abrem ela faz o efeito de *stretch*. Além disso fiz também com que quando os olhos fossem abertos a íris diminua o tamanho dela. Para fazer esses efeitos primeiramente seleciona-se a *mesh* da íris e cria-se um *deformer* com o nome de “*IRIS SQUISH L*” e também se criam *keyforms* no mesmo parâmetro que controla a abertura do olho, com isso no *key 0* a íris diminuí verticalmente e no *key 1* ela diminui para ficar com a expressão de surpresa. Agora com esse *deformer* selecionado cria-se outro *deformer* chamado “*IRIS SCALE L*” e também se cria um novo parâmetro Chamado “*eyel iris scale*” com um mínimo de -30 e um máximo de 30, o valor é grande para que o movimento seja mais lento na hora de aplicar a física. Com o parâmetro criado, eu seleciono o *deformer* “*IRIS SCALE L*” E cria-se os 3 *keyforms*, no -30 a íris é diminuída e rotacionada o sentido horário movendo apenas alguns pontos do *bezier*, apenas um pouco para dar um mais dinamismo no movimento, e no *key* de 30 é feito o inverso, a íris é aumentada e rotacionada no sentido anti-horário mesmo processo é repetido para a pupila. Os detalhes das luzes nos olhos foram feitos para que cada um gire independentemente um do outro em sentidos diferentes, no parâmetro do fechar de olhos. E por fim eu crio um *deformer* para os cílios chamado “*CÍLIOS DETALHE*” e crio um parâmetro chamado de “*EyeL Lashes*” com valores de -1 até 1 e faço com que o *deformer* dos cílios se balance para a esquerda no -1 e para a direita no 1.

Com esses parâmetros feitos podemos adicionar física ao movimento, para isso abrimos o menu de *modeling* e selecionamos a opção de *physics* que irá abrir a seguinte janela:

Figura 44 – Demonstração da janela de configurações da física.



Fonte: produção própria.

1. **Cálculo de FPS:** determina a quantos frames por segundo a animação da física acontece
2. **Configurações de grupo:** onde as configurações para operações de física adicionadas ao modelo são agrupadas. Aqui são criados grupos para cada parte que vai receber um balanço.
3. **Configurações de *input*:** Esse item é usado para definir parâmetros a serem tratados como inputs para operações físicas. Isso configura a "parte que suspende a linha" do pêndulo. O grau de influência dos parâmetros especificados aqui varia, afetando o valor das configurações de output descritas abaixo.
4. **Configurações de *output*:** Esse item é usado para definir as configurações de saída do balanço calculado.
5. **Configurações da física do modelo:** Esse item é usado para definir a maneira como a peça oscila quando é movida. Isso configura o "peso" do pêndulo. Esse item é usado para cálculos baseados em alterações nos valores dos parâmetros definidos em [*Input Settings*].

6. **Preview do pêndulo:** Os resultados calculados nas configurações de input e nas Configurações da física do modelo são visualizados como o movimento do pêndulo.

Para começar foi criado um grupo novo com o nome de um movimento, por exemplo o tamanho da íris ficará no grupo com o mesmo nome, no caso foi utilizado o nome do *deformer* “ÍRIS SCALE L”. Todos esses parâmetros que foram criados para o olho vão ser controlados pelo piscar do olho, portanto na configuração de input devemos selecionar o parâmetro que controla a abertura do olho que no caso do exemplo é “EyeL Open”. Com o *input* configurado vamos configurar o *output* que no caso seria definir qual movimento reagiria a ação de abrir e fechar o olho que no caso do exemplo foi selecionado o parâmetro “EyeL Iris Scale”. Com isso, quando nosso modelo piscar o movimento de diminuir e aumentar da íris vai acompanhar com um efeito de *follow through*, movimento esse que podemos acompanhar na visualização do pêndulo. Para este exemplo não foi necessário nenhum ajuste extra nas configurações da física. Agora este processo é repetido para todos os outros parâmetros de movimento no olho criados anteriormente.

Com o olho pronto, basta duplicar todos os elementos que compõem ele, para isso foram selecionados todos os *deformers* e todas as *meshes* que fazem parte do olho foram copiadas e coladas. Depois com um clique com o botão direito seleciona-se a opção de *reflect* e marca-se a opção de refletir na horizontal além também marcar a opção de não refletir o parâmetro “Eyeball X” pois caso esse parâmetro seja refletido os olhos vão se movimentar em direções opostas. Depois de refletido podemos desabilitar a visibilidade da pasta contendo os elementos do outro olho na aba de partes pois não será utilizada. Agora com o olho refletido, alguns elementos ficaram em posições incorretas, para arrumá-las basta selecioná-las e ajustá-las individualmente, lembrando sempre que se deve fazer isso para todos os *keyforms* dos parâmetros em que esses elementos estão inseridos. Com os olhos devidamente espelhados, deve-se se atentar que ambos estão ligados aos mesmo parâmetros, portanto se moverão juntos, para que cada olho se mova individualmente deve-se criar outros parâmetros vazios para que se possa transferir o movimento do olho copiado para estes novos parâmetros. Deve-se duplicar cada parâmetro já criado para o olho e renomeá-lo indicando que se trata do novo olho para manter uma organização, por exemplo, duplicar o parâmetro “EyeL Open” e renomeá-lo para “EyeR Open”. Depois

de todos os respectivos parâmetros duplicados, seleciona-se todos os elementos do olho copiado e depois de selecionados clica-se com o botão direito em um dos parâmetros que se deseja transferir e seleciona-se a opção de “change...” e então se escolhe o parâmetro duplicado equivalente a opção selecionada. E por fim deve-se reaplicar a física no novo olho, para isso basta apenas duplicar cada grupo, renomeá-lo e mudar o input para o novo parâmetro duplicado assim como cada output para os parâmetros duplicados.

Figura 45 – Exemplo do resultado final do rig dos olhos.

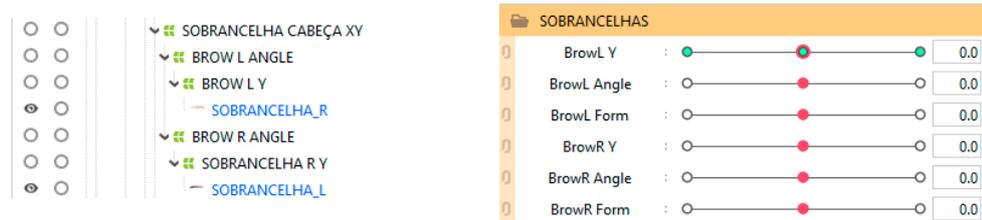


Fonte: produção própria.

Depois dos 2 olhos rigados passa-se para o *rigging* da sobrancelha, para isso o programa oferece parâmetros padrões para que possam ser usados. Inicia-se então escolhendo uma *mesh* da sobrancelha para começar o rig, a seguir no exemplo foi utilizado a sobrancelha da direita, então cria-se um *deformer* chamado “*Browl Y*” e com este *deformer* selecionado cria-se 3 *keyforms* no parâmetro “*Browl Y*”. No valor -1 move-se a sobrancelha para baixo e no valor 1 para cima. Com este *deformer* selecionado cria-se outro *deformer* chamado “*Browl Angle*” e também se cria 3 *keyforms* no parâmetro de mesmo nome. Neste caso, no valor -1 a sobrancelha é rotacionada para dentro, como se a modelo estivesse com expressão de raiva, já no valor 1 a sobrancelha é rotacionada para fora como se estivesse com a expressão de surpresa. E por fim utiliza-se a própria *mesh* para criar *keyforms* no parâmetro que controla a forma, “*Browl Form*”. Utilizando a ferramenta de path *deformer* ao longo da sobrancelha, modela-se a o path de forma que, no valor de -1 a sobrancelha se curve

para baixo como se a modelo estivesse triste, e no valor de 1 a sobrancelha se curva para cima. Para refletir o movimento basta repetir o mesmo procedimento dos olhos.

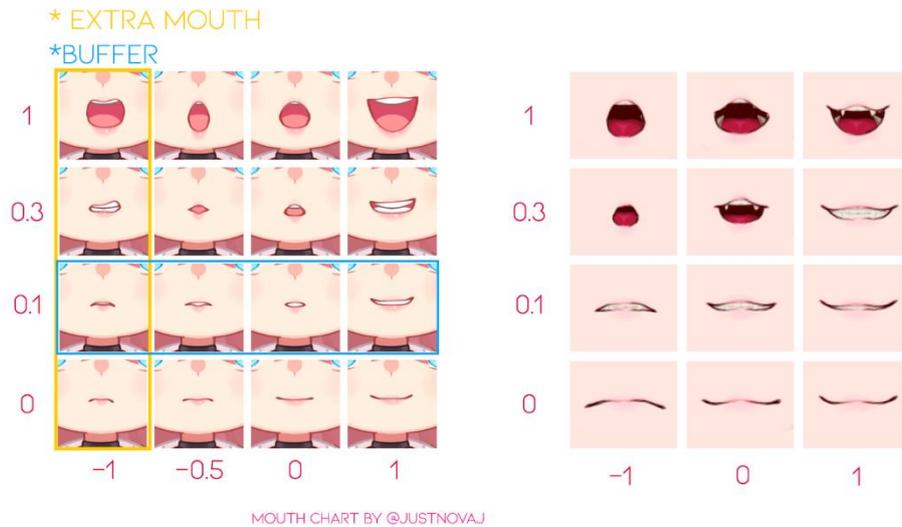
Figura 46 - Exemplo do resultado final do rig das sobrancelhas.



Fonte: produção própria.

Para a boca existem principalmente 2 abordagens para o rig, a primeira é fazer o rig pensando na integração de um *plugin* extra para o programa de captura e o outro que é o que será utilizado que usa apenas 2 parâmetros para o *rig*. Primeiro cria-se um *deformer* contendo todos os elementos da boca, depois cria-se 2 parâmetros, um para a forma da boca que foi nomeado de “*Mouth Form*” com 3 *keyforms*, -1, 0 e 1, além de um para a abertura da boca que foi nomeado de “*Mouth Open*” desta vez com 4 *keyforms*, 0, 0.1, 0.3 e 1. Com estes 2 parâmetros criados, ao junta-los com o link pode se notar que se forma uma grade com 12 *keyforms*, cada um representa um estado da boca. Para auxiliar no processo de fazer cada *key* foi feita uma pesquisa para encontrar alguma tabela que se resulta em algo adequado para o modelo. Com a pesquisa se chegou na seguinte tabela que foi utilizada e adaptada para a se encaixar com o resultado que se esperava. Cada quadrante da tabela representa um *keyform* sendo os números do eixo X representando a forma da boca e o eixo Y representando a abertura da boca.

Figura 47 - Montagem com a referência usada e a versão final boca.



Fonte: Queen Novaj no Twitter. Disponível em:
<https://twitter.com/justNovaj/status/1528061339641561088>

Usando a referência como base inicia-se o processo de criação de *keyforms*, em cada quadrante seleciona-se os elementos que vão ser modificados e então são ajustados para parecer com a referência. Para auxiliar nesse processo, utiliza-se da ferramenta *path form edit* e com ela cria-se pontos ao longo da boca seguindo o formato da linha. Com o *path deform* feito pode-se começar a ir editando cada *keyform* dos parâmetros unidos.

Com os olhos e a boca prontos precisa se resolver o fato de que ambos não acompanham a virada do rosto em nenhum eixo. Para que isso seja resolvido agrupa-se os *deformers* da boca, dos olhos, do nariz (no caso do nariz todas as suas partes estão dentro deste *deformer*) e das sobrancelhas, esse *deformer* foi nomeado Face XY. Com este *deformer* selecionado, cria-se 3 *keyforms* nos parâmetros referentes à cabeça, pois este mesmo deve se mover junto a ela. Basta fazer o mesmo procedimento feito na cabeça e ir ajustando quando necessário.

Agora com a face sincronizada com os olhos pode-se começar a fazer o rig das orelhas. Primeiro cria-se um *deformer* com ambas as *meshes* das orelhas e repete-se os mesmos passos anteriores, usando o parâmetro de x e y da cabeça para criar os *keys*. Porém, para que as orelhas fiquem certas, o correto seria com que uma delas (a que está mais próximo do espectador) fique mais à frente e a outra desapareça atrás da cabeça. Para isso podemos utilizar a ferramenta de *draw order* colocá-lo mais acima na hierarquia, porém nota-se que com isso ambas as orelhas ficam acima da camada do rosto, resultando em algo indesejado.

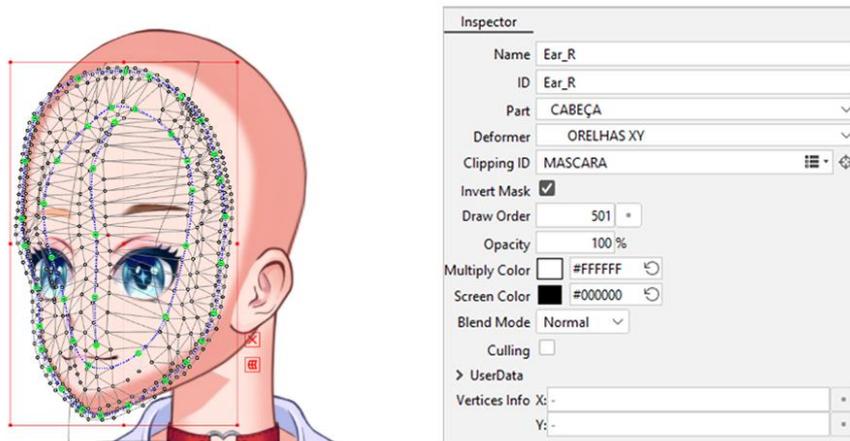
Figura 48 – Demonstração da orelha “rigada” de forma incorreta.



Fonte: produção própria.

Para que isso fosse resolvido, foi feita uma máscara com o formato do rosto. Para isso foi feita uma cópia da camada do rosto e esta camada foi inserida dentro do *deformer* do rosto para que esta siga o movimento do mesmo. Após isso, essa máscara foi colocada com uma opacidade 0 e foi criado um *path deformer* seguindo todo o contorno da malha para fazer correções mais precisas caso necessário. Agora basta selecionar as orelhas e associar o clipping id das orelhas e associar com a máscara criada.

Figura 49 - Demonstração da máscara aplicada a orelha para corrigir o problema anterior.



Fonte: produção própria.

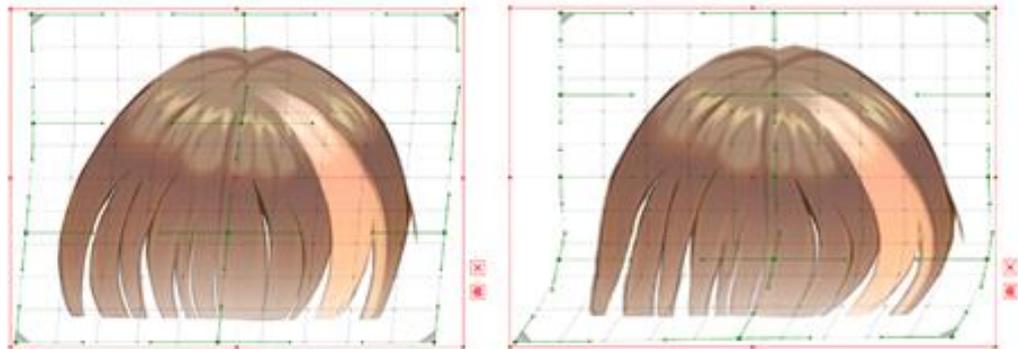
3.1.12 Cabelo

Para os cabelos foi utilizado uma técnica diferente do resto do rosto. Primeiro cria-se um *deformer* para cada *mesh* do cabelo, este *deformer* irá controlar o movimento do cabelo apenas no eixo X, então deve-se criar os *keyforms* apenas no

ângulo X da cabeça. Com todas as *meshes* do cabelo devidamente rigadas no eixo X, deve-se então selecionar individualmente cada *deformer* criado para o eixo X criar um novo *deformer* para o ângulo Y e por fim criar *keyforms* no ângulo Y da cabeça. Para corrigir algumas *deformidades* que podem ser geradas nos cantos, cria-se um novo *deformer* feito com a *mesh* do cabelo selecionada para fazer correções. Cria-se então *keyforms* nos parâmetros de ângulos X e Y da cabeça e nos cantos dos parâmetros linkados fazem-se as alterações necessárias. Dependendo do jeito no qual foram feitas, o cabelo pode ficar na frente do rosto de forma indesejável, para isso basta usar o clipping mask na máscara que foi feita anteriormente.

Com os cabelos prontos, pode-se começar a trabalhar na física dos cabelos. Para isso os cabelos foram separados em 3 categorias, “*front*”, “*side*” e “*back*”. A categoria “*front*” terá 2 parâmetros que controlam o movimento, ambos variando de -1 até 1. Os elementos do cabelo que fazem parte da categoria são aqueles que são mais curtos e que ficam na frente do rosto. Para fazer o movimento, cria-se um *deformer* com a opção de *bezier* em 2x3 e no primeiro parâmetro criamos um movimento movendo apenas os pontos do *bezier* da parte de baixo e aqueles um ponto acima. Já no segundo parâmetro aumenta-se o número do *bezier* para 2x4 e movimenta-se apenas os pontos do *bezier* mais abaixo como nas imagens a seguir.

Figura 50 - Demonstração da deformação do *deformer* da franja.

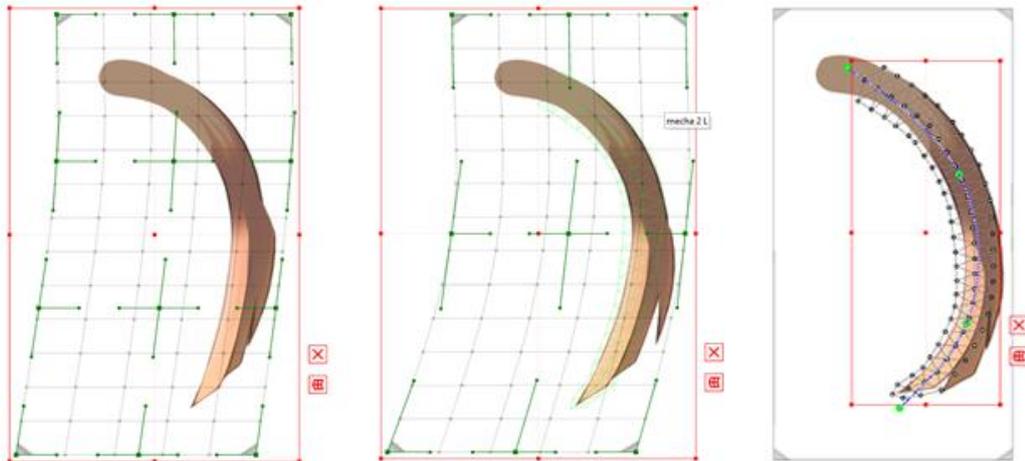


Fonte: produção própria.

Para a categoria de “*side*” são criados 3 parâmetros para controlar o movimento, variando de -1 até 1. Os elementos do cabelo que fazem parte desta categoria são cabelos médios que geralmente ficam ao lado do rosto. Para o movimento basta criar um *deformer* com o *bezier* de 2x4 e se faz o movimento no parâmetro 1, e para o movimento no parâmetro 2 repete-se o mesmo processo, porém

agora com o *bezier* de 2x2. Com a *mesh* selecionada cria-se então um *deform* path seguindo o centro da mecha de cabelo, depois cria-se os *keyforms* no terceiro parâmetro, e nos extremos move-se apenas a ponta do cabelo.

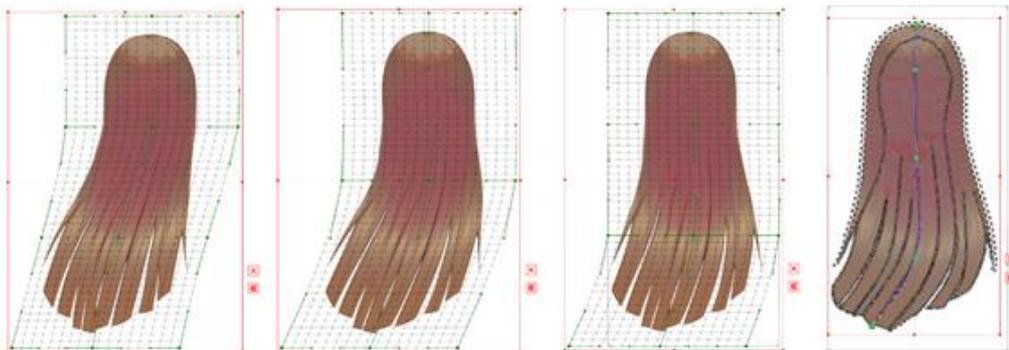
Figura 51 - Demonstração da deformação do *deformer* de cabelos médios.



Fonte: produção própria.

Por fim, a categoria de “back” possui 4 parâmetros para controlar o movimento, com valores de -1 até 1. Nesta categoria se encaixam cabelos longos e tranças. Primeiro repete-se o que foi feito anteriormente, a única diferença é que foi criado um novo *deformer* na malha e com esse *deformer* selecionado cria-se *keyforms* no 3 parâmetro, e com esse *deformer* selecionado volta-se com o *bezier* de 2x4, porém agora apenas movendo os últimos pontos da parte de baixo. E por último, no parâmetro 4, repete-se o movimento com o path *deform*.

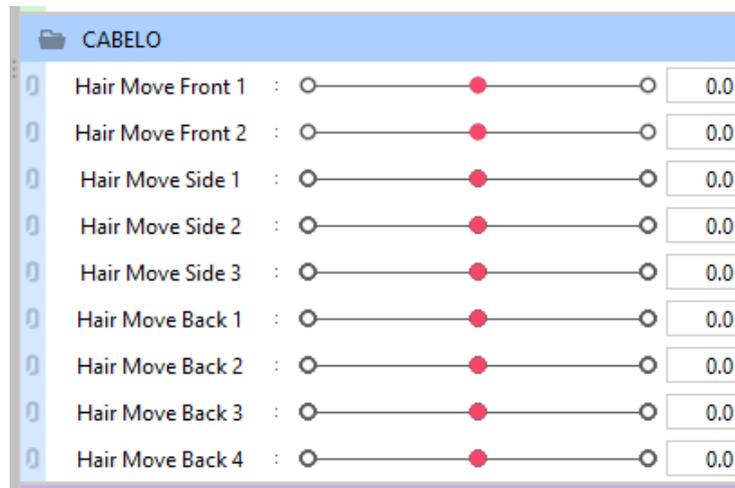
Figura 52 - Demonstração da deformação do *deformer* de cabelos longos.



Fonte: produção própria.

Para a física começa-se criando um grupo para cada categoria. Ao criar o grupo seleciona-se o input preset de *head input* que por padrão já escolhe os parâmetros relacionados à cabeça. Na opção de “*physics model preset*” seleciona-se a opção mais adequada para cada categoria de cabelo sendo: “*front*” a opção “*hair (double pendulum)*”, “*side*” a opção “*hair (triple pendulum)*” e “*back*” também a opção “*hair (triple pendulum)*” adicionando um pêndulo extra para o 4 parâmetro criado.

Figura 53 - Exemplo do resultado final dos parâmetros da física do cabelo.



Fonte: produção própria.

3.1.13 Corpo

Com a cabeça pronta passa-se a rigar o corpo, para isso agrupa-se o corpo em *deformers* específicos:

- “**Head body XY**” para a cabeça: para este *deformer* seleciona-se todos os outros *deformers* já criados relativos à cabeça.
- “**Neck body XY**” para o pescoço e o tórax: primeiro cria-se 2 *deformers*, um contendo todos os elementos do pescoço com o nome de “Neck Z” e outro com elementos do tórax da personagem, exceto o peito, chamado de “chest body XY”.
- “**Bust body XY**” para o peito: cria-se um *deformer* contendo os elementos do peito, incluindo peças de roupa que vão sobre ele, como no caso da modelo, o laço e os suspensórios.
- “**Arms Body XY**” para os braços: primeiro, para cada braço foi criado um *rotation deformer* chamado “shoulder” com todos os elementos do braço, depois cria-se um novo *rotation deformer* com o antebraço e a mão e por fim

cria-se outro *rotation deformer* para a mão. Além dos braços, foram colocados elementos do ombro e da gola da camisa

- “**Skirt Body XY**” para a saia: para a saia foram criados 2 *deformers*, um para o eixo x contendo a *mesh* da saia e com esse *deformer* selecionado foi criado um para o eixo y. Cada um destes 2 *deformers* serão utilizados para configurar a física da saia.
- “**Legs Body XY**” para as pernas: para as pernas repete-se o que foi feito com os braços, porém com os elementos da perna.

Figura 54 - Demonstração da separação dos *deformers* para movimentar o corpo.



Fonte: produção própria.

Com os *deformers* devidamente separados pode-se começar a fazer o rig do corpo no eixo X. Cria-se 3 *keyforms* no parâmetro padrão do programa “*body X*” e então começa-se a mexer nos *deformers* criados. É importante começar pelos elementos que estão mais à frente que no caso da modelo seria o peito, em seguida é só repetir o processo para o resto dos *deformers* criados e fazer a mesma coisa com o eixo Y.

3.1.14 Respiração

Com os movimentos X e Y do corpo prontos, fazemos o movimento de respiração do modelo. Primeiro seleciona-se os *deformers* correspondentes a parte

de cima do corpo e criamos um novo *deformer* chamado “*breath*”. Com este parâmetro selecionado, cria-se 2 *keyforms* no parâmetro padrão de respiração “*breath*” no qual se alterna constantemente entre dois estados, de 0 (expirar) a 1 (inspirar) onde ocorre o movimento de expansão.

3.1.15 Ângulo Z

Por fim basta configurar os parâmetros do ângulo z. Para a cabeça seleciona-se todos os *deformers* que estão dentro do *deformer* “*Head Body XY*” e então cria-se um *rotation deformer* chamado “*Head Z*”. Cria-se 3 *keyforms* no ângulo z da cabeça e então rotaciona-se o *deformer*. Para a parte de cima do corpo seleciona-se o *deformer* “*breath*” e cria-se outro *rotation deformer* chamado “*Body Z*” e repete o mesmo processo anterior no parâmetro “*Body Z*”. Além disso dentro deste *deformer* foi criado um outro *rotation deformer* para controlar a cauda, e o movimento deste *deformer* foi associado ao parâmetro de *breathing* por se tratar de um movimento constante.

EXPORTAÇÃO

Com o modelo finalizado pode-se passar para a fase de exportação do modelo. Dentro do programa utiliza-se a opção “*export for runtime*” e depois “*export as moc3 file*”. Ao selecionar essa opção irá abrir uma caixa de diálogo com algumas configurações de exportação, normalmente não são necessários ajustes nesta parte do processo. Ao clicar em ok abrirá uma janela pop-up perguntando onde o usuário deseja salvar o arquivo.

Ao exportar o arquivo do modelo, o Live2d irá criar 4 arquivos:

- *.moc3 file*: Dados do modelos Live2D para uso em aplicações.
- *.model3.json*: contém informações para vincular arquivos MOC3, arquivos de textura, etc. Este arquivo está no formato JSON (*JavaScript Object Notation*), que é uma formatação leve de troca de dados.
- *.physics3.json*: Esses dados contêm os valores definidos para as físicas criadas para o modelo.
- *.png*: Este arquivo é a imagem gerada do “*texture atlas*”.

VTUBE STUDIO

VTube Studio é um aplicativo disponível para iPhone/iPad, Android, macOS e Windows. O aplicativo usa o smartphone ou webcam para rastrear o rosto e animar o modelo do Live2D de acordo. O modelo é mostrado diretamente no telefone ou PC, transmitindo os dados de rastreamento facial pela rede local do telefone para o PC/Mac.

3.1.16 Interface

Figura 55 - Menu do programa VTube Studio



Fonte: produção própria.

1. Mostra a barra de seleção de modelo, o ícone próximo a ele abre a configuração da captura de tela. Com isso, pode-se fazer capturas de tela diretamente do aplicativo, salvá-las e compartilhá-las nas redes sociais. Também suporta capturas de tela PNG com fundo transparente.
2. Altera o plano de fundo (mostra a lista de seleção de plano de fundo). O ícone próximo a ele (uma estrela) abre a configuração do sistema de itens/objetos para importar itens para a cena. O ícone próximo a esse (três estrelas) abre a janela de configuração da cena do item para configurar coleções de itens que podem ser ativados/desativados por meio de teclas de atalho.

3. Bloqueia a posição do modelo (garantirá que o modelo não possa mover acidentalmente ao tocar na tela, o rastreamento de rosto ainda moverá/animará o modelo). O ícone próximo a ele bloqueia/desbloqueia todos os itens da cena.
4. Abre a parte de configurações do programa e do modelo
5. Mostrar logs (caso ocorra de encontrar algum problema/erro, é recomendado olhar o log).
6. VNet é o sistema de colaboração online integrado do VTube Studio, disponível como DLC pago para Windows e macOS.

3.1.17 Configuração

Ao abrir o programa, clica-se no primeiro botão a qual abre a barra de seleção de modelo. Ao abrir a barra, aparecerão alguns modelos padrões, para carregar o modelo deve-se colocar os itens da exportação do live2d dentro do diretório indicado ao clicar em *"import your own model"*. Com o modelo dentro da pasta indicada pelo programa, pode-se selecioná-lo ao abrir a barra de seleção do modelo. Com o modelo selecionado pode-se configurar ele. Dentro do programa existem 4 abas de configurações: configurações gerais, configurações de câmera, configurações do modelo e configurações de *hotkeys*.

3.1.17.1 *Configurações gerais:*

Nas configurações gerais há a "Configuração de entrada do mouse" que permite que o usuário use a posição do mouse/dedo como parâmetro de entrada para animar seu modelo, além do rastreamento facial. Além disso o VTube Studio suporta sincronização labial baseada em voz. Isso pode substituir ou melhorar o rastreamento da boca da câmera usando sua voz gravada no microfone.

3.1.17.2 *Configurações de câmera:*

Esta guia de configuração permitirá alterar as configurações relacionadas à renderização e à câmera.

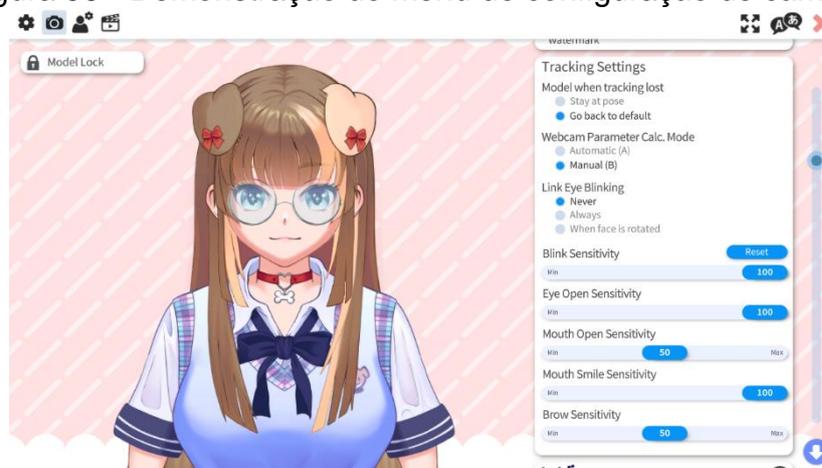
A configuração da câmera AR se refere ao preview da câmera no programa.

A configuração de rastreamento possui como, alterar o comportamento do modelo quando o rastreamento da face é perdido, pode-se fazer com que o modelo congele no lugar quando seu rosto for perdido ou fazer com que ele retorne à sua pose padrão. Alterar o comportamento de vinculação de piscar de olhos, se "Always"

(Sempre) for ligado, os valores de “EyeOpen” do olho esquerdo e direito serão sempre definidos como a média de ambos. Se “When face is rotated” (Quando o rosto é girado) for selecionado, o valor de “EyeOpen” dos olhos será vinculado quando a cabeça for girada muito para o lado. Isso usará o valor do olho que ainda estiver visível quando sua cabeça for girada para um lado.

As configurações de rastreamento do webcam podem ser usadas para personalizar o comportamento de como os valores de rastreamento facial são calculados.

Figura 56 - Demonstração do menu de configuração de câmera.



Fonte: produção própria.

3.1.17.3 Configurações do modelo

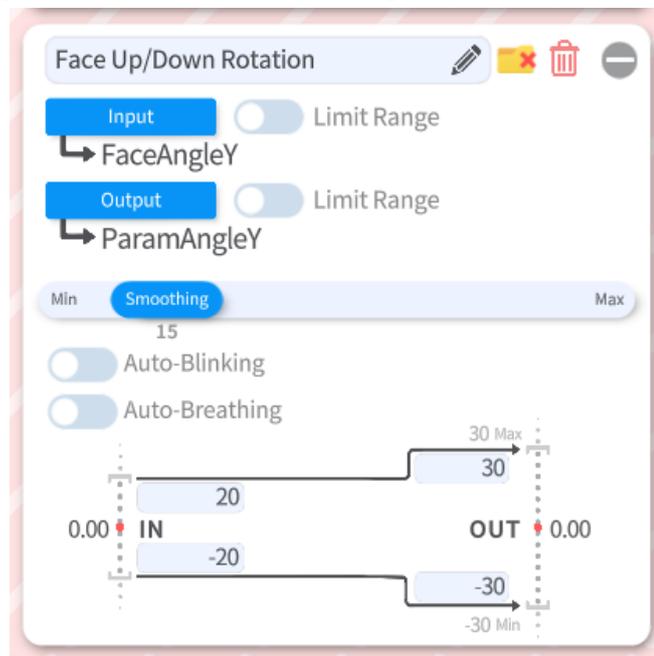
Na parte superior, pode-se inserir um nome para o modelo. Esse nome será mostrado na barra de seleção de modelos. Pode-se escolher um ícone (no ícone no canto superior direito) e a animação ociosa padrão (.motion3.json) para o modelo. Pode-se escolher qualquer arquivo dentro da pasta do modelo. Também pode-se executar a configuração automática do seu modelo do VTube Studio. Isso irá configurar o modelo com base nos nomes e valores padrão dos parâmetros do Live2D.

3.1.17.4 Configuração de parâmetros

Essa é a parte mais importante nas configurações do modelo. Aqui, pode se definir quais parâmetros de rastreamento facial controlam quais parâmetros do Live2D. A ideia geral é que se pode mapear livremente qualquer parâmetro “INPUT” (rastreamento facial, mouse, etc.) para qualquer parâmetro OUTPUT (parâmetro do Live2D). A forma como os valores são mapeados pode ser configurada livremente. No

exemplo a seguir o parâmetro relacionado a mover a cabeça no ângulo a seguir está configurado da seguinte forma, o “IN” se refere a informação recebida pela captura de movimento, que no caso varia de 20 (mover a cabeça para cima) e -20 (mover a cabeça para baixo). Já o “OUT” se refere aos valores máximos do parâmetro, especificados no Live2D, que controla esse movimento. Ao movimentar a cabeça, o ponto vermelho mostrará onde o valor atual está posicionado dentro do intervalo de entrada e saída. O programa irá converter os valores da entrada para valores equivalentes na saída. Estes valores podem ser alterados a qualquer momento e configurados do jeito que melhor atende às necessidades do modelo.

Figura 57 - Exemplo de configuração de parâmetro.



Fonte: produção própria.

Ao ativar "*Limit Range*", pode-se garantir que os valores de entrada ou saída nunca excedam o intervalo definido para a entrada ou saída. Isso também fará com que o valor pare suavemente ao se aproximar dos limites. Em geral, recomenda-se alterar apenas o intervalo de saída e deixar o intervalo de entrada como está.

Mais "*smoothing*" (suavização) tornará os movimentos menos trêmulos, mas introduzirá defasagem.

O *Auto-breath* (respiração automática) fará o parâmetro de saída subir e descer em um movimento semelhante ao da respiração. Nenhum parâmetro de entrada é necessário ao usar essa opção. Qualquer entrada será ignorada.

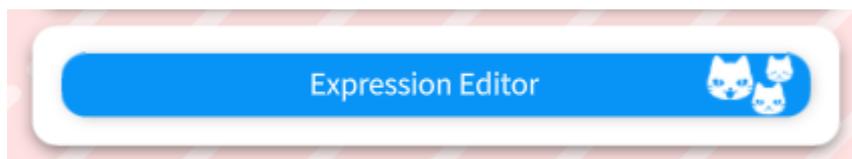
O *Auto-blink* reduzirá aleatoriamente o parâmetro a zero. Isso pode ser combinado com um parâmetro de entrada, mas nenhum é necessário.

Como parâmetro de entrada, pode-se escolher em uma lista de parâmetros disponíveis. Como parâmetro de saída, pode-se escolher qualquer parâmetro do Live2D. Cada parâmetro de saída só pode ser escolhido uma vez, pois, caso contrário, teriam vários parâmetros de entrada gravando no mesmo parâmetro de saída.

3.1.17.5 Configurações de Hotkeys/Expressões

As expressões permitem que se definam parâmetros do Live2D para determinados valores usando teclas de atalho. Usando expressões, pode-se, por exemplo, acionar diferentes expressões faciais ou mudanças de trajés. As expressões podem ser configuradas diretamente dentro do VTube Studio na guia "Hotkeys" (Teclas de atalho).

Figura 58 - Botão para edição de expressões.

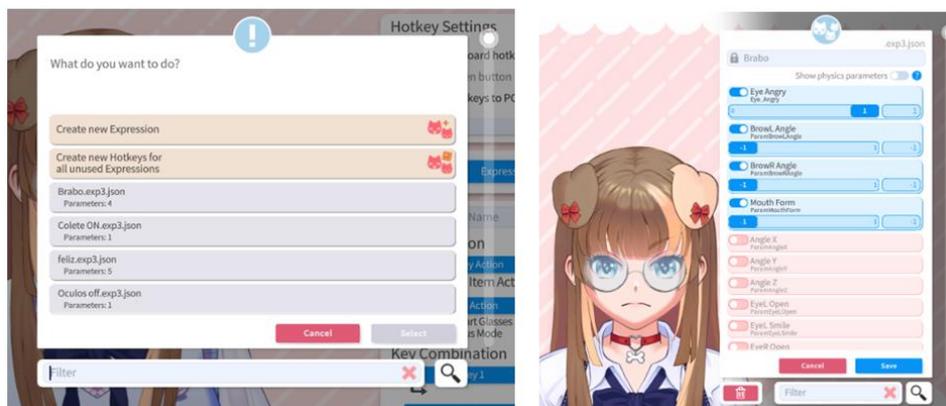


Fonte: produção própria.

A partir daí, pode-se criar uma nova expressão, editar expressões existentes ou criar automaticamente teclas de atalho para todas as expressões que ainda não têm teclas de atalho.

Ao editar expressões ou criar novas expressões, a seguinte janela é exibida. Selecione-se e os parâmetros do Live2D que se deseja que façam parte da expressão.

Figura 59 Exemplo de uma expressão de raiva configurada no menu de expressões.



Fonte: produção própria.

Para usar as expressões no VTube Studio, cria-se uma nova tecla de atalho dentro do VTube Studio, seleciona-se o tipo de tecla de atalho "*Set/Unset Expression*" e então se escolhe um arquivo de expressão na lista.

INTEGRAÇÃO COM SOFTWARE DE STREAMING

Para este trabalho o software escolhido foi o OBS Studio (*Open Broadcaster Software Studio*), que é um programa de streaming e gravação gratuito e de código aberto mantido pelo OBS Project.

No VTube Studio, seleciona-se o fundo do *Color Picker* e marca-se a opção "*Transparent in capture*", o que tornará o plano de fundo da janela transparente quando a janela for gravada no OBS. Para realizar uma stream é necessário configurar as fontes de áudio e vídeo, seguindo os passos:

1. Na tela inicial do OBS Studio, cria-se uma fonte clicando em "+" na seção "Fontes".
2. Seleciona-se a opção de captura de jogo, pode-se renomear o nome da fonte caso desejar.
3. Nas propriedades da captura de jogo altera-se o modo para: "capturar janela específica" e depois seleciona-se a janela do Vtube studio "[VTube Studio.exe]: Vtube Studio". Por fim, marca-se a opção de permitir transparência.

Com isso feito, todas as configurações já estão prontas e pode-se começar uma *stream* com o modelo funcionando.

Figura 60 - Demonstração do modelo em frente a um jogo dentro do software de transmissão.



Fonte: produção própria.

4 CONCLUSÃO

Em conclusão, este trabalho explorou o processo de criação de um Vtuber. Desde a fase inicial de concepção da ideia do design da personagem até a implementação de rigging, cada etapa foi abordada com o objetivo de proporcionar uma compreensão abrangente do que implica criar um personagem virtual envolvente e interativo. Destacou-se a importância do briefing como base para a tradução visual da identidade do Vtuber. Foi demonstrado também como fazer o rigging é um processo complexo e trabalhoso, mas que pode conferir ao modelo não apenas movimentos fluidos, mas também a capacidade de interagir de maneira autêntica com sua audiência.

A cliente adorou como ficou o resultado, o design se encaixou na visão que ela tinha da personagem e as animações ficaram do jeito que ela imaginava. Por fim o resultado foi satisfatório para a cliente e para mim.

Para mim uma jornada escrever sobre este processo devido a paixão que tenho por esse mercado de Vtubers, sempre desejei saber como funcionam os processos para a criação. Esse desejo me embarcou em um mundo muito mais complexo do que era imaginável, quando mais pesquisava sobre os processos mais técnicas novas apareciam e me faziam querer aprender mais. Com este artigo espero que instigue o leitor a querer conhecer mais sobre Vtubers e até mesmo auxiliie na criação do próprio modelo.

Com o conhecimento que possuo agora vejo que muitas coisas do processo poderiam ser melhoradas, cometi vários erros em tomadas de decisões feitas pelo caminho, mas mesmo não sendo o caminho mais simples nem o mais fácil, creio que cheguei em um resultado satisfatório para os padrões do mercado.

REFERENCIAS

DINIZ, Gabriela Rodrigues et al. VTubers: estudo sobre a massificação do consumo de uma cultura de nicho durante a pandemia de COVID-19. In: Anais do III Seminário Iberoamericano de Economia da Cultura. 3. UFMG, Belo Horizonte, MG. 2022. Disponível em: <<https://pesquisas.face.ufmg.br/ecult/wp-content/uploads/sites/32/2022/06/Vtubers-Estudo-Sobre-Massificacao-Do-Consumo-De-Uma-Cultura-De-Nicho-Durante-A-Pandemia-De-Covid-19Diniz-et-al.pdf>>. Acesso em: 06 abr. 2023.

FERREIRA, Julio Cesar Valente; GONÇALVES, Paula; REGIS, Rafael Dirques David. VTubers, cultura participativa e engajamento transmídia: Hololive e Holo no Graffiti. In: Anais do III Congresso TeleVisões. 3. UFF, Niterói, RJ, 2022. Disponível em: <https://www.researchgate.net/publication/366548525_VTubers_cultura_participativa_e_engajamento_transmidia_Hololive_e_Holo_no_Graffiti>. Acesso em: 06 abr. 2023.

GONÇALVES, Paula; SILVA, Beatrice de Melo. THE DIGITAL WON'T LET ME GO: ANÁLISE DA PERCEPÇÃO E INTERAÇÃO DO PÚBLICO DE VTUBERS. In: IV Jornada Internacional GEMInIS (JIG 2021). UFSCar, 2022. Disponível em: <<https://www.doity.com.br/anais/jig2021/trabalho/227455>>. Acesso em: 06 abr. 2023.

LIVE2D CUBISM. **About Deformers**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/deformer/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **About ArtMeshes**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/concept-of-artmesh/>>. Acesso em: 30 out. 2023.

Live2D CUBISM. **About Parameters**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/parameter/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **About Parts**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/parts/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Add/Delete Keys to/from Parameters**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/edit-parameters/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Automatic Generation of Four Corner Forms**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/synthesize-corners/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Clipping Mask**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/clipping-mask/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Deform Paths**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/deformpath/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Glue**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/glue/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **How to Edit**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/how-to-edit/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Keyforms (Make X, Y Movements)**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/keyform-xydirection/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Motion Inversion**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/inversion-of-movement/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Rotation deformer**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/making-and-rotation-of-rotationdeformer/>>. Acesso em: 30 out. 2023.

LIVE2D CUBISM. **Warp deformer**. 2023. Disponível em: <<https://docs.live2d.com/en/cubism-editor-manual/making-and-placement-of-warp-deformer/>>. Acesso em: 30 out. 2023.

REGIS, Rafael Dirques David. Análise do impacto econômico, da expansão de popularidade e das novas tecnologias do fenômeno Vtuber. Arquivos do CMD, [S.l.], v. 9, n. 2, p. 29–55, 2023. ISSN 2318-5422. Disponível em: <<https://periodicos.unb.br/index.php/CMD/article/view/47638>>. Acesso em: 05 abr. 2023.

THE Rise of the VTubers - Like & Describe Podcast #1. S. l.: Youtube, 2022. (20 min.), Video, son., color. Legendado. Série Like & Describe Podcast. Disponível em: <https://www.youtube.com/watch?v=8_igTYoM_VI&t=629s>. Acesso em: 11 mar. 2023.

VTUBE STUDIO. **Expressions (a.k.a. Stickers or Emotes)**. Disponível em: <[https://github.com/DenchiSoft/VTubeStudio/wiki/Expressions-\(a.k.a.-Stickers-or-Emotes\)](https://github.com/DenchiSoft/VTubeStudio/wiki/Expressions-(a.k.a.-Stickers-or-Emotes))>. Acesso em: 06 nov. 2023.

VTUBE STUDIO. **Getting Started**. Disponível em: <<https://github.com/DenchiSoft/VTubeStudio/wiki/Getting-Started>>. Acesso em: 06 nov. 2023.

VTUBE STUDIO. **Loading your own Models**. Disponível em: <<https://github.com/DenchiSoft/VTubeStudio/wiki>Loading-your-own-Models>>. Acesso em: 06 nov. 2023.

VTUBE STUDIO. **Recording Streaming with OBS**. Disponível em: <<https://github.com/DenchiSoft/VTubeStudio/wiki/Recording-Streaming-with-OBS>>. Acesso em: 06 nov. 2023.

VTUBE STUDIO. **VTS Model Settings**. Disponível em: <<https://github.com/DenchiSoft/VTubeStudio/wiki/VTS-Model-Settings>>. Acesso em: 06 nov. 2023.

VTUBE STUDIO. **VTube Studio Settings**. Disponível em: <<https://github.com/DenchiSoft/VTubeStudio/wiki/VTube-Studio-Settings>>. Acesso em: 06 nov. 2023.

WHAT Many VTubers are Missing. S. I.: Yam Albat, 2023. (21 min.), Video, son., color. Disponível em: <<https://www.youtube.com/watch?v=s7OJTip0Rsw&t=898s>>. Acesso em: 23 jun. 2023.

YOUTUBE. **Source information**: innovation in adverse times. Innovation in Adverse Times. 2020. Disponível em: <<https://www.youtube.com/trends/articles/report-sources-tr20/>>. Acesso em: 11 mar. 2023.