

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

Luisa Bernardoni Costa

**Desenvolvimento de *Rig* de personagens para o *teaser* “*Fix-It!*”**

Florianópolis

2021

Luisa Bernardoni Costa

**Desenvolvimento de *Rig* de personagens para o *teaser* “*Fix-It!*”**

Trabalho Conclusão do Curso de Graduação em Animação do Centro de Expressão da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Animação.

Orientador: Prof. Flávio Andaló.

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Costa, Luisa Bernardoni  
Desenvolvimento de rig de personagens para o teaser  
"Fix-it!" / Luisa Bernardoni Costa ; orientador, Flávio  
Andaló, 2021.  
124 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro de  
Comunicação e Expressão, Graduação em Animação, Florianópolis,  
2021.

Inclui referências.

1. Animação. 2. Rig. 3. Animação. 4. Personagens. 5.  
Props. I. Andaló, Flávio . II. Universidade Federal de  
Santa Catarina. Graduação em Animação. III. Título.

Luisa Bernardoni Costa

**Desenvolvimento de *Rig* de personagens para o *teaser* “*Fix-It!*”**

Este Trabalho de Conclusão de Curso (TCC) foi julgado adequado para obtenção do Título de Bacharel em Animação e aprovado em sua forma final pelo Curso de Animação da Universidade Federal de Santa Catarina.

Florianópolis, 07 de maio de 2021.

Prof. Flávio Andaló, Dr. Coordenador do Curso de Animação UFSC

**Banca Examinadora:**

Prof<sup>a</sup> Marta Machado, Dr<sup>a</sup>. (Universidade Federal de Santa Catarina )

Prof Gabriel Prim, Dr. (Universidade Federal de Santa Catarina)

---

Prof. Dr. Flávio Andaló

(Orientador)

Universidade Federal de Santa Catarina



Este trabalho é dedicado a todos os artistas que almejam um dia poder dar vida aos seus personagens.

## AGRADECIMENTOS

Primeiramente gostaria de agradecer a minha amada mãe, por sempre estar do meu lado me motivando a continuar, acreditando em mim mesmo quando nem eu acreditava. Por todas as xícaras cheias de café e carinho que ela me levava tarde da noite no quarto, enquanto eu produzia este trabalho.

Ao meu amado pai, que nunca mediu esforços para trazer todo o conforto possível a mim e ao meu irmão caçula. Que sempre me apoiou nas minhas escolhas e sempre elogiou e incentivou a produção de minhas artes desde pequena.

Aos meus queridos irmãos, que tenho visto pouco devido à pandemia, mas que sempre estão disponíveis para qualquer coisa que eu possa vir precisar.

A minha querida cunhada, que sempre me ajuda com suas palavras de conforto em meio a tanta coisa que vem acontecendo me dando apoio desde sempre.

Aos meus amigos incríveis que tive a sorte de conhecer na faculdade e aos de longa data, que sempre estiveram do meu lado e que sempre me fazem rir e me proporcionam momentos de felicidade.

Aos meus amigos e membros de equipe do projeto inicialmente em 2016, quando ainda éramos calouros e com muita vontade de aprender e criar. Foi através de momentos muito divertidos e de descontração que a ideia do projeto “*Fix-It!*” nasceu. Nunca me esquecerei de vocês, Brito Neto, Gabriela Zanon, Jacob Faria e Victor Caraméz. Sem vocês este pequeno universo ficcional que criamos nunca teria existido.

Ao querido Bruno Shigeo, por fazer uma trilha sonora que se encaixasse perfeitamente neste projeto tão querido pra mim.

Agradeço em especial à Bianca L. Galdino, minha amiga e parceira deste projeto desde 2016, sem ela não teria sido possível dar vida a esses personagens tão maravilhosos, que tivemos a oportunidade de criar juntas e que sinto tanto orgulho.

A todos os professores do curso, em especial ao Flavio Andaló, por ter aceitado ser o meu orientador e ter sido tão paciente ao me orientar e sanar minhas dúvidas. E ao professor Wiliam Machado, que acreditou no projeto desde o início.

*“Ao infinito e além!”*

LIGHTYEAR, Buzz

## RESUMO

Este trabalho tem como objetivo principal o processo da criação de rigs dos personagens principais para a produção do *teaser* da série animada 2D, “*Fix-It!*”. É abordado o processo de criação completo de *rig* de personagem 2D *cutout*, partindo desde o desenvolvimento e exploração de um novo design para os personagens já existentes da série, passando pela etapa de preparação para o *rigging* até a conclusão do objetivo principal. O trabalho também aborda a criação dos *props* para o *teaser* e esclarece alguns dos conceitos, ferramentas e técnicas do *software ToonBoom Harmony* que foi utilizado para a realização do projeto.

**Palavras-chave:** *Rig*. Animação. *Cutout*. *Props*. *ToonBoom Harmony*.

## **ABSTRACT**

This work has as main objective the process of creating rigs of the main characters for the production of the teaser for the 2D animated series, "Fix-It!". It is approached the complete process of creating a 2D cutout character rig, starting since the development and exploration of a new design for the existing characters of the series, going through the preparation stage for rigging until the conclusion of the main objective. The work also approaches the creation of the props for the teaser and clarifies some of the concepts, tools and techniques of the ToonBoom Harmony software that was used to accomplish the project.

**Keywords:** Rig. Animation. Cutout. Props. ToonBoom Harmony.

## LISTA DE FIGURAS

Figura 1 – <i>Print</i> de “ <i>Fix-it!</i> ” que mostra os frames da cena .....	18
Figura 2 – Visão geral da <i>timeline</i> no <i>ToonBoom</i> e indicações de suas partes.....	23
Figura 3 – Demonstração do <i>drawing node</i> na <i>node view</i> .....	24
Figura 4 – Exemplificação visual das 4 diferentes camadas de arte dentro de um <i>drawing</i>	25
Figura 5 – Painel de substituição de desenhos da biblioteca .....	26
Figura 6 – Aba de visualização de desenhos da biblioteca.....	26
Figura 7 – Exemplo da visualização de um conjunto de <i>nodes</i> através da <i>nodeview</i> .....	27
Figura 8 – Exemplo de uma <i>composite</i> na <i>nodeview</i> .....	28
Figura 9 – Demonstração do <i>Cutter node</i> na <i>node view</i> .....	29
Figura 10 – Demonstração de como é conectado o <i>Auto-Patch node</i> na <i>node view</i> .....	30
Figura 11 – Exemplo de conexão do efeito <i>Overlay-Layer</i> na <i>node view</i> .....	31
Figura 12 – Demonstração de uma <i>peg</i> conectada a um <i>drawing</i> na <i>node view</i> .....	31
Figura 13 – Exemplificação do uso do deformador <i>Bone</i> e indicações de duas partes .....	32
Figura 14 – Exemplificação do uso do deformador <i>Game Bone</i> e suas partes.....	33
Figura 15 – Exemplificação do uso do deformador <i>Curve</i> e indicações de suas partes .....	33
Figura 16 – Exemplificação do uso do deformador <i>Envelope</i> e indicações de suas partes.	34
Figura 17 – Exemplificação do uso do deformador <i>Free Form</i> e suas partes .....	35
Figura 18 – Exemplo do <i>Kinematic-Output node</i> funcionando na <i>node view</i> .....	36
Figura 19 – Primeira versão de design aprovado da protagonista, em 2016 .....	40
Figura 20 – Referências para novos elementos do rosto de Lizie .....	40
Figura 21 – Explorações para o novo <i>design</i> de Lizie .....	41
Figura 22 – Definição da proporção e expressões faciais da exploração nº 5 da Lizie .....	41
Figura 23 – Adição de cores e testes de adereços da protagonista .....	42
Figura 24 – <i>Modelsheet</i> com a versão final da protagonista.....	42
Figura 25 – Primeira versão de <i>design</i> aprovado de Sphix, em 2016.....	44
Figura 26 – Algumas referências para o novo design de Shix.....	44
Figura 27 – Primeira exploração para o novo <i>design</i> de Sphix .....	45
Figura 28 – Mais explorações de <i>design</i> e opções de cores para o Sphix .....	45
Figura 29 – Versão final do <i>design</i> de Sphix.....	46
Figura 30 – Primeira versão de <i>design</i> do Sr. Porco, em 2016.....	47
Figura 31 – Referências para as vestimentas do Sr. Porco .....	48

Figura 32 – Nova versão de <i>design</i> do antagonista .....	48
Figura 33 – <i>Modelsheet</i> com a versão final do Sr. Porco .....	49
Figura 34 – Paleta de cores da personagem Lize .....	50
Figura 35 – Visão da <i>node view</i> com a estrutura montada .....	51
Figura 36 – <i>Composite</i> em modo <i>pass trough</i> , localizada na <i>node view</i> .....	52
Figura 37 – As peças soltas de cada personagem durante o processo de preparação .....	53
Figura 38 – Biblioteca de olhos da personagem Lize .....	54
Figura 39 – Biblioteca de olhos do Sr. Porco e Sphix .....	54
Figura 40 – Exemplo de “olho sorrindo”, meio fechado. ....	54
Figura 41 – “Olho sorrindo” da Lize, através do desenho “bochecha” .....	55
Figura 42 – Biblioteca de bocas dos personagens Lize e Sr. Porco .....	55
Figura 43 – Biblioteca de bocas do personagem Sphix .....	56
Figura 44 – Biblioteca de línguas do Sphix .....	56
Figura 45 – Biblioteca de mãos da Lize .....	57
Figura 46 – Biblioteca de mãos do Sr. Porco .....	58
Figura 47 – Biblioteca de mãos do Sphix .....	58
Figura 48 – Biblioteca de pés do Sphix .....	59
Figura 49 – Biblioteca de pés do Sr. Porco.....	59
Figura 50 – Biblioteca do desenho “pontape” da personagem Lize .....	59
Figura 51 Primeira etapa de <i>rigging</i> do olho .....	60
Figura 52 – Exemplificação da penúltima etapa de <i>rigging</i> do olho .....	61
Figura 53 – Exemplificação da última etapa de <i>rigging</i> do olho .....	61
Figura 54 – Os <i>drawings node</i> da boca e nariz, na <i>node view</i> .....	62
Figura 55 – Elementos da boca do Sphix e a visão na <i>node view</i> .....	63
Figura 56 – Boca de Sphix já mascarada e a visão da <i>node view</i> .....	63
Figura 57 – Desenho da língua separada e funcionando no encaixe da boca.....	64
Figura 58 – Visão das estrelas do rosto de Lize na <i>node view</i> .....	64
Figura 59 – Visão do desenho da cabeça de Lize na <i>node view</i> .....	65
Figura 60 – Visão da <i>peg</i> da cabeça na aba <i>Camera view</i> e <i>node view</i> .....	65
Figura 61 – Visão da <i>node view</i> que mostra a máscara das estrelas do rosto .....	66
Figura 62 – Peg das estrelas e a máscara funcionando .....	66
Figura 63 – Desenhos da orelha na camada <i>LineArt</i> e <i>ColourArt</i> .....	67
Figura 64 – Desenho da orelha na camada <i>UnderlayArt</i> e <i>OverlayArt</i> .....	67
Figura 65 – Propriedades dos desenhos da orelha .....	68

Figura 66 – Visão da node view que mostra a conexão para o efeito de mescla da orelha .	69
Figura 67 – Efeito da orelha sendo mostrado e funcionando na pose $\frac{3}{4}$ .....	69
Figura 68 – Conexões dos desenhos das franjas na <i>node view</i> .....	70
Figura 69 – Conexão do efeito Overlay-Layer do olho na node view .....	70
Figura 70 – Cabeça de Lizie antes e depois do efeito OverLay-Layer se adicionado .....	71
Figura 71 – Posição dos Pivot points de cada franja e da peg “franjas-master” .....	71
Figura 72 – Desenho do <i>drawing</i> do cabelo e a visão da <i>nodeview</i> .....	71
Figura 73 – Visão da node view que mostra as conexões da lente dos óculos .....	72
Figura 74 – Desenho completo do brilho-lente e com a máscara e a peg .....	72
Figura 75 – Desenho da lente sendo mascarada pela base e vista da <i>node view</i> .....	73
Figura 76 – Efeito Overlay-Layer funcionando no desenho e a visão da node view .....	73
Figura 77 – Visão do <i>rigging</i> dos óculos na <i>node view</i> .....	74
Figura 78 – Versão final do lado esquerdo dos óculos da Lizie .....	74
Figura 79 – Desenho do pescoço e seu <i>drawing</i> node na <i>node view</i> .....	75
Figura 80 – Vista do desenhos inteiros dos elementos da jardineira e eles mascarados .....	76
Figura 81 – Desenho do ombro solto e desenho do ombro mascarado no tronco .....	76
Figura 82 – Visão do <i>rigging</i> do torso na node view .....	77
Figura 83 – Demonstração de onde o Pivot point é posicionado nos elementos do braço ..	77
Figura 84 – Demonstração dos desenhos nas camadas LineArt e Colour Art .....	78
Figura 85 – Visão dos elementos do braço na <i>node view</i> .....	79
Figura 86 – Exemplificação do efeito de “dobra no cotovelo” .....	80
Figura 87 – Visão do <i>rigging</i> do braço na node view .....	80
Figura 88 – Desenho da manga da camiseta e a visão da <i>node view</i> .....	81
Figura 89 – Desenhos dos elementos que compõe a mão de Lizie e vista da <i>node view</i> ...	81
Figura 90 – Propriedades dos desenhos das partes da luva .....	82
Figura 91 – Desenho final da mão com o elemento da luva e a vista da node view .....	83
Figura 92 – Linhas adicionadas na camada OverlayArt e a luva funcionando .....	83
Figura 93 – Vista do <i>rigging</i> da mão na node view .....	84
Figura 94 – Demonstração do encaixe das duas partes da perna e os efeitos funcionando .	84
Figura 95 – Rig da perna na <i>node view</i> .....	85
Figura 96 – Desenhos da bota nas camadas <i>LineArt</i> , <i>ColourArt</i> e a união da duas .....	85
Figura 97 – Desenho da “bota-tras” na camada OverlayArt e o resultado dos desenhos ...	86
Figura 98 – Visualização do <i>rig</i> da bota na <i>node view</i> .....	86

Figura 99 – Demonstração da mascara do detalhe da bota funcionando.....	87
Figura 100 – Visualização do calcanhar sem o <i>Auto-Patch</i> .....	87
Figura 101 – Visualização do calcanhar com o <i>Auto-Patch</i> funcionando.....	88
Figura 102 – Demonstração do desenho “pontape” nas <i>ArtLayers</i> .....	88
Figura 103 – Visão do <i>rig</i> da bota na <i>node view</i> .....	89
Figura 104 – Demonstração da bota sem o <i>Overlay-Layer</i> e com o <i>Overlay-Layer</i> .....	89
Figura 105 – <i>Rig</i> final do pé de Lizie .....	90
Figura 106 – Deformadores envelope nas franjas de Lizie.....	91
Figura 107 – Deformadores na gola e no bolso da jardineira.....	91
Figura 108 – Deformador envelope aplicado nas partes da jardineira e tronco .....	92
Figura 109 – Deformador aplicado na tira dos óculos e cabelo.....	92
Figura 110 – Uso do deformador nas sobrancelhas.....	93
Figura 111 – Comparação do braço com deformador e sem deformador.....	93
Figura 112 – Comparação da perna com deformadores e sem deformadores .....	94
Figura 113 – Deformador aplicado no detalhe da bota.....	94
Figura 114 – Propriedades da <i>layer</i> “ <i>socket</i> ”.....	95
Figura 115 – Opção a ser selecionada após a seleção de toda a hierarquia .....	96
Figura 116 – Hierarquia do lado esquerdo dos óculos na <i>node view</i> .....	97
Figura 117 – Hierarquia final dos óculos na <i>node view</i> .....	98
Figura 118 – Hierarquia do braço esquerdo da Lizie na <i>node view</i> .....	98
Figura 119 – Hierarquia dos braços de Lizie na <i>node view</i> .....	99
Figura 120 – Hierarquia da perna de Lizie na <i>node view</i> .....	100
Figura 121 – Hierarquia das pernas de Lizie na <i>node view</i> .....	101
Figura 122 – Cadeia hierárquica de Lizie .....	102
Figura 123 – <i>Peg</i> do rosto de Lizie.....	103
Figura 124 – <i>Peg</i> do rosto e óculos de Lizie.....	103
Figura 125 – Controlador da “ <i>CABECA-MASTER</i> ” da Lizie.....	104
Figura 126 – <i>Peg</i> “ <i>CABECA-PESCOCO</i> ” da personagem Lizie .....	104
Figura 127 – <i>Peg</i> que controla a cabeça e os braços na hierarquia da Lizie.....	105
Figura 128 – <i>Peg</i> que agrupa os elementos da parte de cima da personagem .....	105
Figura 129 – Controlador final da personagem Lizie. ....	106
Figura 130 – <i>Print</i> da <i>timeline</i> com os <i>keyframes</i> das poses de giro do <i>rig</i> .....	106
Figura 131 – Demonstração do novo cabelo, da aplicação do deformador e nova cabeça	107
Figura 132 – Demonstração dos novos desenhos do tronco e da biblioteca.....	107

Figura 133 – Demonstração do novo desenho da tira e da biblioteca .....	107
Figura 134 – Biblioteca do <i>prop</i> caixa tecnológica .....	108
Figura 135 – Biblioteca do <i>prop</i> chave inglesa .....	109
Figura 136 – <i>Prop</i> disquete à esquerda e <i>prop</i> SSD à direita .....	109
Figura 137 – Etapa de animação, com a personagem interagindo com os <i>props</i> .....	110
Figura 138 – Personagem interagindo com os <i>props</i> na cena.....	110
Figura 139 – Partes da moto .....	111
Figura 140 – <i>Rig</i> da visão traseira da moto .....	112
Figura 141 – Divisão da moto em <i>pegs</i> .....	112
Figura 142 – <i>Rig</i> do chaveiro da moto.....	113
Figura 143 – Guidão da moto com o chaveiro.....	113
Figura 144 – <i>Rig</i> da parte traseira da moto .....	114
Figura 145 – Divisão da parte de trás da moto em <i>pegs</i> .....	114
Figura 146 – <i>Prop</i> da moto com os personagens .....	115
Figura 147 – Processo de ajuste de detalhe do braço na etapa de animação .....	116
Figura 148 – Novo design da jaqueta sem precisa de ajuste para as articulações .....	117

## **LISTA DE ABREVIATURAS E SIGLAS**

2D Duas dimensões

3D Três dimensões

FK Forward Kinematics

IK Inverse Kinematics

SSD Solid States Drive

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
1.1	OBJETIVOS .....	15
1.1.1	<b>Objetivo Geral .....</b>	<b>15</b>
1.1.2	<b>Objetivos Específicos .....</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1	ANIMAÇÃO .....	17
2.1.1	<b>ANIMAÇÃO TRADICIONAL .....</b>	<b>18</b>
2.1.2	<b>ANIMAÇÃO DIGITAL 2D.....</b>	<b>19</b>
2.2	RIG.....	20
2.2.1	<b>ARTICULAÇÕES.....</b>	<b>21</b>
2.3	PROPS.....	22
<b>2.4</b>	<b>BREVE DOCUMENTAÇÃO DO TOONBOOM HARMONY PREMIUM .....</b>	<b>23</b>
2.4.1	<b>VISÃO GERAL DA <i>TIMELINE</i> .....</b>	<b>23</b>
2.4.2	<b>DRAWING.....</b>	<b>24</b>
2.4.3	<b>ART LAYERS .....</b>	<b>24</b>
2.4.4	<b><i>LIBRARY</i> OU BIBLIOTECA .....</b>	<b>25</b>
2.4.5	<b>NODEVIEW .....</b>	<b>26</b>
2.4.6	<b>COMPOSITE NODE.....</b>	<b>27</b>
2.4.7	<b>CUTTER NODE.....</b>	<b>28</b>
2.4.8	<b>AUTO-PATCH NODE .....</b>	<b>29</b>
2.4.9	<b>OVERLAY-LAYER NODE .....</b>	<b>30</b>
2.4.10	<b>PEG.....</b>	<b>31</b>
2.4.11	<b>DEFORMADORES .....</b>	<b>32</b>
2.4.12	<b>KINEMATIC-OUTPUT NODE .....</b>	<b>35</b>
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>37</b>

<b>3.1.1</b>	<b>PERSONAGENS.....</b>	<b>37</b>
3.1.1.1	<i>LIZIE</i> .....	38
3.1.1.2	<i>SPHIX</i> .....	43
3.1.1.3	<i>SR. PORCO</i> .....	46
3.2	PREPARAÇÃO PARA O RIG.....	49
<b>3.2.1</b>	<b>ESTRUTURA .....</b>	<b>50</b>
<b>3.2.2</b>	<b>BIBLIOTECA .....</b>	<b>53</b>
3.2.2.1	<i>OLHOS</i> .....	53
3.2.2.2	<i>BOCAS</i> .....	55
3.2.2.3	<i>MÃOS</i> .....	56
3.2.2.4	<i>PÉS</i> .....	59
<b>3.2.3</b>	<b>RIGGING.....</b>	<b>60</b>
3.2.3.1	<i>ROSTO</i> .....	60
3.2.3.2	<i>CABEÇA</i> .....	65
3.2.3.3	<i>TORSO</i> .....	75
3.2.3.4	<i>ARTICULAÇÕES E EXTREMIDADES</i> .....	77
3.2.3.5	<i>APLICAÇÃO DE DEFORMADORES</i> .....	90
3.2.3.6	<i>DUPLICANDO AS PARTES</i> .....	95
3.2.3.7	<i>HIERARQUIA</i> .....	101
3.2.3.8	<i>O TURN AROUND</i> .....	106
3.3	PROPS.....	108
<b>3.3.1</b>	<b>A MOTO .....</b>	<b>110</b>
<b>4</b>	<b>RESULTADOS.....</b>	<b>116</b>
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>118</b>
	<b>REFERÊNCIAS .....</b>	<b>119</b>

## 1 INTRODUÇÃO

“*Fix-It!*” é um projeto de animação 2D, criado inicialmente em 2016 pela autora deste trabalho, em conjunto de seis outros estudantes da Universidade Federal de Santa Catarina, com o intuito de criar um universo ficcional de uma série animada para TV voltada para o público infantil. A etapa de desenvolvimento criativo teve a participação de todos os integrantes do grupo, onde foram discutidas e definidas as propostas da série, premissa, narrativa, título e estrutura da história.

Com a vontade de continuar desenvolvendo o projeto a fim de gerar conteúdo e material suficiente para uma possível venda e comercialização, foi necessário conseguir o aval dos integrantes anteriores para que apenas duas graduandas prosseguissem com o projeto.

Os conteúdos optados pelas duas graduandas foram: construção de uma bíblia para a série, preparação para *pitch* e produção de um *teaser*.

O projeto será realizado em parceria com Bianca L. Galdino, com divisões claras de trabalho de acordo com suas áreas de maior afinidade. Assim, a autora está envolvida na parte visual da obra. Já a parte de roteiros e textos de personagens, pesquisas e estruturação da bíblia, além da organização da produção são de autoria de Bianca L. Galdino.

Portanto, no presente trabalho será abordada a parte de desenvolvimento visual das personagens principais da série e o processo de criação de *rig* de personagem e *props* para a produção do *teaser*.

Para a etapa de produção de *rigs* de personagens, *props* e animação foi optado pela autora deste trabalho, a utilização do software *ToonBoom Harmony Premium* (na versão de estudante com a mensalidade de \$10,00 aproximadamente R\$57,00), pois é um dos *softwares* mais completos do mercado de animação nos dias atuais. O artista tem a liberdade de escolher qual técnica de animação 2D utilizar, podendo inclusive mesclá-las, e até adicionar câmera e efeitos de pós-produção para um resultado mais satisfatório, tudo dentro do programa. Além de possuir uma interface nodal para a programação dos *rigs*, sistema que facilita e muito a vida de qualquer profissional *rigger* e facilitará para a finalidade deste projeto.

### 1.1 OBJETIVOS

#### 1.1.1 Objetivo Geral

Desenvolvimento de *rig* dos personagens principais da série autoral “*Fix-It!*” para produção de um *teaser*.

### 1.1.2 Objetivos Específicos

- Refazer o *design* dos personagens principais da série já existentes, adaptando o estilo para a técnica de animação 2D escolhida;
- Desenvolvimento de *rig* das personagens principais para a produção do *teaser*;
- Desenvolver os *props* para produção do *teaser*.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para começar a desenvolver o *rig* de um personagem 2D para uma série de animação, é fundamental entender os conceitos e as técnicas de animação 2D que existem no mercado, as maneiras de criar um *rig*, e principalmente o *software* que for utilizado.

O trabalho de um *rig* é altamente técnico, mas requer um toque fino de um artista. Os riggers precisam de uma compreensão profunda do funcionamento interno de qualquer software e hardware que estejam usando, [...] O conhecimento da anatomia, humano ou animal, também é útil. Como o trabalho principal dos riggers é ajudar os animadores, eles precisam de uma compreensão bastante boa da animação de personagem e objeto para saber como criar controles eficientes. (BEANE, 2012, p.178, tradução livre da autora)

Com estes controles eficientes citados por Beane, é possível que o animador simule movimentos críveis, gerando um resultado satisfatório da animação e fazendo com que o público compre a ideia de que o personagem realmente está vivo e se movimentando na tela.

### 2.1 ANIMAÇÃO

BEANE (2012, p.194), nos descreve em seu livro, de maneira sucinta que “a animação ocorre quando um grupo de imagens paradas que são ligeiramente diferentes umas das outras são mostradas de tal forma — ou seja, em ordem sequencial e em uma velocidade suficiente — que nossos olhos acreditam que algo está se movendo.”

Em 1824, Peter Mark Roget descobriu [...] o princípio vital, "a persistência da visão". Este princípio baseia-se no fato de que nossos olhos temporariamente mantêm a imagem de qualquer coisa que eles acabaram de ver. Se não fosse assim, nunca teríamos a ilusão de uma conexão ininterrupta em uma série de imagens, e nem filmes nem animação seriam possíveis. Muitas pessoas não percebem que os filmes não se movem de fato, e que eles são imagens paradas que parecem se mover quando são projetadas em uma série. (WILLIAMS, 2009, p.13, tradução livre da autora)

Estas “imagens paradas” são chamadas de quadros, ou frames em inglês, que são produzidas pelos animadores. A taxa de quadros padrão em filmes é de 24 quadros por segundo, ou seja, para se produzir uma animação de 90 minutos (129.600 quadros) é necessária uma equipe de animadores que trabalhem nas poses dos personagens ou objetos e analisem cada quadro para que o

produto final esteja funcionando. Esta é a razão da produção de uma animação demandar tanto tempo e orçamentos com custos altos. (BEANE, 2012, p.194)

Figura 1 – Print de “Fix-it!” que mostra os frames da cena



Fonte: A autora

Para que uma animação cumpra seu papel, de divertir o público fazendo com que a obra tenha apelo suficiente para fazê-los acreditarem que os personagens estejam vivos, os artistas precisam ter a compreensão e noções de desenho, além de conhecer e dominar os 12 princípios da animação descritos no livro “*Illusion of Life*” de Frank Thomas e Ollie Johnston, 1981. E “*The Animator’s Survival Kit*”, livro escrito por Richard Williams, 2009. O conceito de *keyframe*, ou “quadro chave” traduzido do inglês, é explicado com clareza nos livros, assim como *timing* e *spacing*, que são essenciais para se produzir uma animação de qualidade.

Ao longo dos anos o termo animação 2D se tornou muito amplo devido ao surgimento de novos *softwares* próprios para a sua produção, com ferramentas que facilitam o trabalho e economizam o tempo dos profissionais que passam a utilizar vários processos e técnicas novas através das tecnologias digitais. (WINDER; DOWLATABADI, 2011, p.247)

### 2.1.1 ANIMAÇÃO TRADICIONAL

Na animação tradicional, todos os quadros são desenhados à mão, tendo as suas etapas divididas em *layout*, seguido de animação, *cleanup* (ou “limpeza dos quadros”) e efeitos. Possuindo uma equipe de pessoas especializadas para cada etapa. (WINDER; DOWLATABADI, 2011, p.248)

Dependendo de como a produção é configurada, é possível que o artista desenhe diretamente em um tablet digital (processo sem papel) ou use papel e lápis (processo baseado em papel) para criar imagens que sejam digitalizadas e integradas ao pipeline digital. Em ambos os pipelines sem papel e baseados em papel, os desenhos da linha vão de rough ao cleanup e são então submetidos ao processo de tinta digital e pintura para aplicação de cores.

### 2.1.2 ANIMAÇÃO DIGITAL 2D

Na animação digital 2D os personagens são quebrados em partes, a partir de seu *modelsheet*, são separados por *layers*, cada uma se refere a uma parte do corpo (como cabeça, pescoço, tronco etc.). Durante a produção da animação digital 2D algumas dessas peças precisam ser trocadas para dar a impressão de mudança de ângulo dos personagens e para alterar as expressões faciais, (SANDERS, 2018), todas são adicionadas à biblioteca do personagem para que possam ser reutilizadas novamente.

Um sistema de controladores é criado através de uma hierarquia das peças, baseando-se em como o corpo humano, animal ou objeto funciona. Conforme essas peças são posicionadas, os *keyframes* são criados na *timeline* pelo *software*, sendo interpolados ou não de um para o outro, podendo ser modificado a qualquer momento pelo animador, que possui a vantagem de reutilizar o *keyframe* quantas vezes quiser, além de definir o *ease* em cada *keyframe* existente, ou seja, ajustar o *timing* da animação em poucos segundos. Algo que na animação tradicional ou quadro a quadro digital, não seria possível.

Winder e Dowlatabadi (2011, p.248), distinguem os tipos de animação tradicional e digital, dizendo que no tradicional 2D, “o artista cria animação desenhada à mão em departamentos selecionados, como *layout*, animação, *cleanup* e efeitos. No digital 2D, a animação é baseada principalmente em *assets*, usando as imagens separadas dos personagens criadas durante a fase de pré-produção.”

Dependendo do visual desejado para um projeto e seus parâmetros orçamentais, a animação tradicional 2D e digital 2D têm seus prós e contras. A animação tradicional 2D, que se beneficia da opção de animação completa, é normalmente usada para projetos de alto orçamento, incluindo longas-metragens. O digital 2D, por outro lado, é usado principalmente para projetos de TV com cronogramas mais curtos que são projetados para um estilo mais limitado. (WINDER; DOWLATABADI, 2011, p.248, tradução livre da autora)

A técnica *cutout*, está sendo muito utilizada nas séries de TV, por possuir o método de reutilização de *assets*, a biblioteca de bocas, mãos e olhos são elaborados na etapa de pré-produção, sendo possível adicionar mais desenhos e alterar os que já existem ao longo da produção dos episódios. Também é possível e extremamente útil guardar poses e ciclos de animações inteiras para serem utilizados em alguma outra cena ou episódio, sendo capaz de manter bibliotecas com diferentes ciclos de caminhadas, corrida, pulos, entre outros. Através desta técnica se podem reaproveitar os *assets* de roupas de um personagem, por exemplo, e adaptá-lo para outro personagem a partir do desenho já criado. Com esse tipo de reutilização de desenhos, esse estilo de animação faz com que a demanda da produção seja muito mais rápida e com orçamento mais baixo do que a produção de animação tradicional.

Do ponto de vista do produtor, é importante notar que essas duas técnicas são frequentemente combinadas porque são mutuamente benéficas. Os melhores pipelines de produção misturam e combinam com esses métodos com base nas necessidades de um shot: por exemplo, um projeto de recurso tradicional pode usar técnicas 2D digitais para preencher perfeitamente e eficientemente uma tomada que precisa de uma multidão. Enquanto isso, um programa de televisão 2D digital pode utilizar um artista que é tradicionalmente treinado para melhorar a animação limitada através da adição de poses desenhadas. (WINDER; DOWLATABADI, 2011, p.248, tradução livre da autora)

## 2.2 RIG

Como BEANE (2012, p.177) aponta em seu livro, “toda geometria 3D que vai ser animada precisa de algum tipo de sistema que forneça aos animadores o controle e flexibilidade necessários para mover esse objeto de alguma forma. Este sistema de controle é chamado de *rig*”.

Os artistas de rigging criam o sistema de controle para os animadores usarem no processo de animação, assim como os fabricantes de fantoches criam os sistemas de controle para marionetes. Riggers criam este sistema de controle altamente avançado de tal forma que os animadores verão apenas o que é necessário para articular os personagens e objetos em uma interface simples. O trabalho principal do rigger é facilitar o trabalho do animador. (BEANE, 2012, p.177, tradução livre da autora)

Na animação digital 2D que não se utiliza da técnica tradicional de quadro a quadro, o *rig* possui o mesmo peso de importância. Como explica WINDER e DOWLATABADI (2011, p.192), “para que o artista gere animação, ele deve controlar e manipular os *assets*. Isso é feito através do *rig* dos modelos pelos quais os símbolos são combinados para formar o personagem completo e seus *pivot points* são selecionados e ajustados para o movimento desejado”.

Em seu livro, BEANE (2012, p.180), esclarece que o objeto tem como ponto de partida para seu movimento o local onde o *pivot point* é posicionado, ou seja, o objeto será escalonado ou girado a partir do *pivot*. Por isso é importante saber exatamente onde posicioná-lo em cada peça, pois irá definir como o objeto começará o movimento.

Para a construção de um *rig*, é essencial organizar as peças de cada parte do corpo na sua ordem certa para que as partes possam ser animadas em grupos. A hierarquia básica para esta formação é chamada de *parenting*.

Todos os rigs são baseados em uma hierarquia de sistemas e controles que trabalham em uma ordem sequencial para criar a articulação do seu objeto. Essa hierarquia em sua forma mais básica é uma relação pai/filho — um objeto é o pai (parent), e outro objeto é o filho(child). Criar essa relação é chamado de paternidade (parenting). Um objeto filho pode se mover, girar e escalar independentemente do objeto pai, mas quando o objeto pai se move, o filho o seguirá. Você pode ter vários filhos sob um pai, chamados irmãos (simblings), e pode até ter filhos de filhos. [...], você pode mudar essa relação entre pai e filho a qualquer momento. Você pode colocar objetos como pai de objetos de uns aos outros, ou pode agrupar objetos para se mover como uma unidade. (BEANE, 2012, p.179, tradução livre da autora)

Através de um *rig* básico apenas com relação pai/filho já é possível animar. Porém para animações mais elaboradas de personagens principais de alguma série, por exemplo, que terá que ser animados em diversas situações criadas por roteiristas, é necessário um sistema mais avançado de *rig*. Estes sistemas mais avançados são construídos através de esqueletos ou articulações.

### 2.2.1 ARTICULAÇÕES

Como explica BEANE (2012, p.183), existem vários tipos de movimentos de articulações que funcionam para partes específicas do corpo, como por exemplo, uma articulação do tipo dobradiça, para que seja possível simular o movimento de um cotovelo. Ou um grupo de articulações, uma abaixo da outra, para que se possa rotacionar ao mesmo tempo, e desta forma, servindo como uma espinha dorsal.

BEANE (2012, p.184), afirma que “o movimento das articulações pode ser dividido em duas categorias: *Foward Kinematics* (FK) e *Inverse Kinematics* (IK). Esses dois tipos de movimento articular fornecem ao animador diferentes maneiras de abordar tarefas específicas de animação”.

*Foward Kinematics*, que traduzindo do inglês significa “Cinemática Direta” é a articulação que se comporta através da relação de pai/filho, ou seja, é possível manipular o braço inteiro através

do movimento de rotação da articulação ombro que é pai do cotovelo, que por sua vez é pai do pulso. Ao começar a rotação através do cotovelo, apenas o pulso irá seguir o movimento, pois o pulso é o único filho do cotovelo, portanto o único que é influenciado. Esta articulação é útil, pois como é usada apenas rotação, ela cria uma linha de movimento em arco nas juntas. Os arcos fazem com que a animação fique muito mais próxima do real, visto que nossos movimentos não acontecem em linhas retas, além de fazer parte de um dos doze princípios da animação.

Já a *Inverse Kinematics*, ou “Cinemática Inversa” traduzida para o português, é a articulação que possui sua hierarquia na ordem inversa. Ela tem a capacidade de mover a cadeia inteira apenas movimentando a articulação do pulso. Desta forma, movimenta-se o pulso fazendo com que as juntas tracem um trajeto reto, tendo que ser deixados em arco manualmente, caso seja necessário. É mais utilizado para as pernas, assim os pés sempre ficarão em contato com o chão.

"FK e IK podem ser criados para permitir que o animador troque entre esses dois sistemas para dar flexibilidade para criar um desempenho específico, se necessário." (BEANE, 2012, p.185)

### 2.3 PROPS

*Props*, que traduzindo do inglês quer dizer “adereços” é o termo usado para se referir aos objetos que os personagens irão interagir em algum momento dentro da história. Os *props* podem ser móveis, veículos, objetos pequenos como, celular, caneca ou objetos grandes como violão, barraca etc. Estes precisam ser coerentes com o estilo visual de toda a animação, sendo criado por um *prop designer*. De acordo com Winder e Dowlatabadi (2011, p.186), o guia de estilo abrange as seguintes áreas para *props*:

- Frente, parte traseira e interior (se aplicável);
- Construção (se o prop tiver elementos complexos, adicionar anotações)
- Comparação de tamanho com personagem e/ou layout de fundo
- Diretrizes sobre como o prop funciona (se for um dispositivo complicado)

“Paulo Muppet, sócio e diretor de animação da Produtora de Animação Birdo, na cidade de São Paulo, afirma que “os *props* são a interface pela qual os personagens se conectam aos cenários”. ELLER (2018, p.45).

## 2.4 BREVE DOCUMENTAÇÃO DO TOONBOOM HARMONY PREMIUM

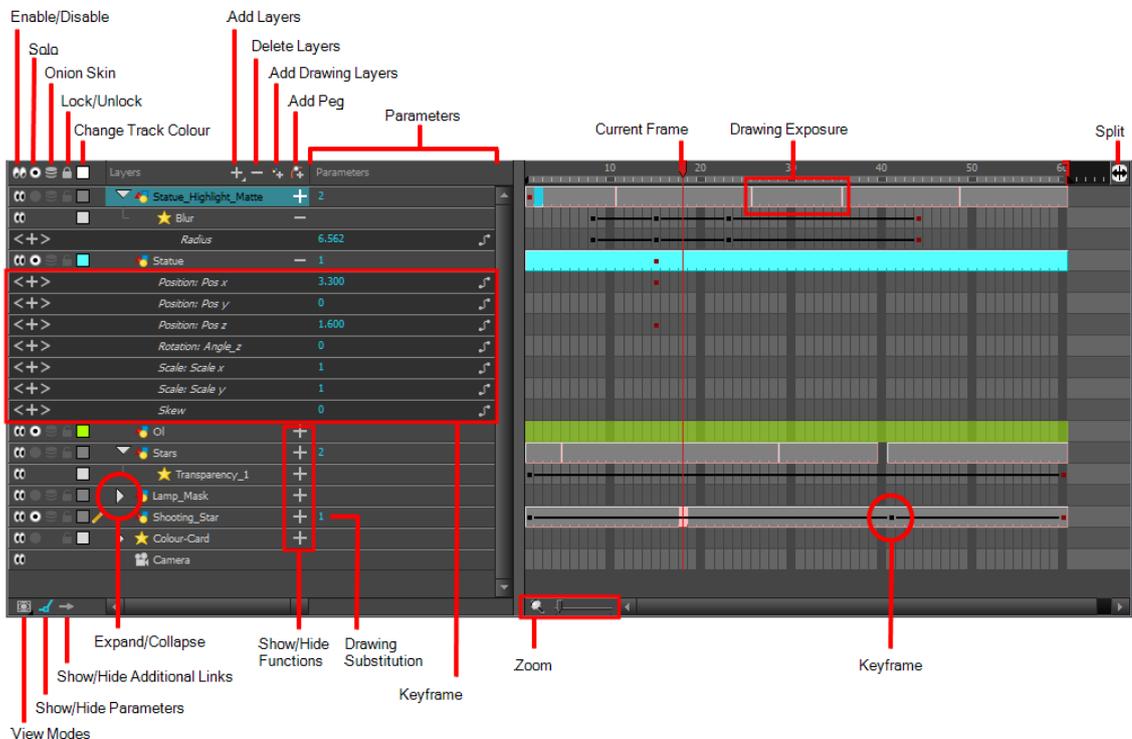
Para que o *rig* dos personagens funcione de maneira satisfatória e com facilidades para a animação, é preciso ter um conhecimento das ferramentas que o software disponibiliza. Neste tópico serão descritas algumas ferramentas que foram essenciais para a realização dos *rigs* deste projeto.

### 2.4.1 VISÃO GERAL DA *TIMELINE*

Saber como a *timeline* (ou traduzindo para o português que significa “linha do tempo”) funciona e ter conhecimento de suas partes é de extrema importância para qualquer finalidade de utilização do *software*, seja para animação, para criação de *rig* etc.

A *timeline* serve para ajustar os tempos dos desenhos alterando as exposições, adicionar os *keyframes* e ordenar as camadas. Ela permite a visualização das camadas, efeitos, sons, valores dos *keyframes*, comprimento de cena e tempo.

Figura 2 – Visão geral da *timeline* no *ToonBoom* e indicações de suas partes

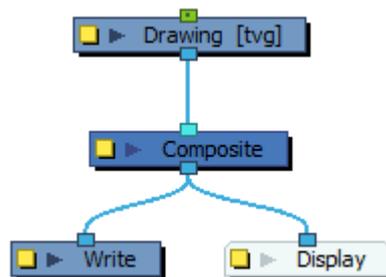


Fonte: < <https://docs.toonboom.com/help/harmony-14/premium/reference/view/timeline-view.html?Highlight=timeline> > (acesso em 26 de abril de 2021)

## 2.4.2 DRAWING

O *drawing* é o elemento na qual as artes são armazenadas. Na aba “*camera view*” é possível visualizar e editar o desenho, adicionando cores, ajustando linhas etc. Na aba “*timeline*” é mostrada a exposição do *drawing* na cena e a sua posição através dos eixos XYZ, sendo possível altera-las. Na *node view* é visto como o elemento “*drawing*” *node*, e é possível ser visualizado na cena apenas se estiver conectado a um *composite* que tem sua saída conectada ao “*Display*” *node*.

Figura 3 – Demonstração do *drawing node* na *node view*



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/reference/node/generator/element-node.html?Highlight=draweing%20node%20> > (acesso em 25 de abril de 2021)

## 2.4.3 ART LAYERS

Em cada *drawing* estão contidas quatro camadas de desenho, sendo elas:

- *OverlayArt*;
- *LineArt*;
- *ColourArt*;
- *UnderlayArt*;

Que permitem que a arte seja dividida nessas camadas. Um exemplo de utilização dessas camadas de arte é separar o contorno do preenchimento do desenho, processo similar ao da animação tradicional.

Estas camadas adicionais são úteis para a criação de máscaras e articulações durante o processo de *rig*.

Nas propriedades da *layer*, na aba “*Drawing*” é possível escolher quais destas *Artlayers* serão visíveis ou não.

Figura 4 – Exemplificação visual das 4 diferentes camadas de arte dentro de um *drawing*



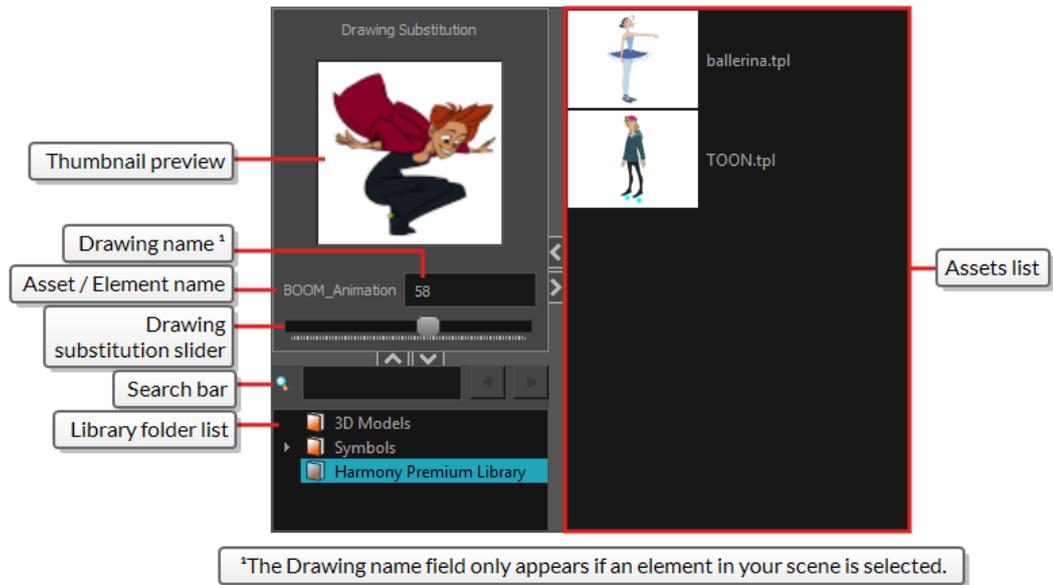
Fonte: < [https://docs.toonboom.com/help/harmony-16/advanced/paperless-animation/about-art-layer.html?Highlight=auto%20patch#drawingandanimating\\_4228305222\\_1169081](https://docs.toonboom.com/help/harmony-16/advanced/paperless-animation/about-art-layer.html?Highlight=auto%20patch#drawingandanimating_4228305222_1169081) > (acesso em 25 de abril de 2021)

#### 2.4.4 LIBRARY OU BIBLIOTECA

A biblioteca é onde se guarda as peças das personagens, no software *ToonBoom Harmony* é possível guardar não apenas peças, mas poses, *rigs*, cenários e até ciclos de animações para serem reutilizadas durante a produção.

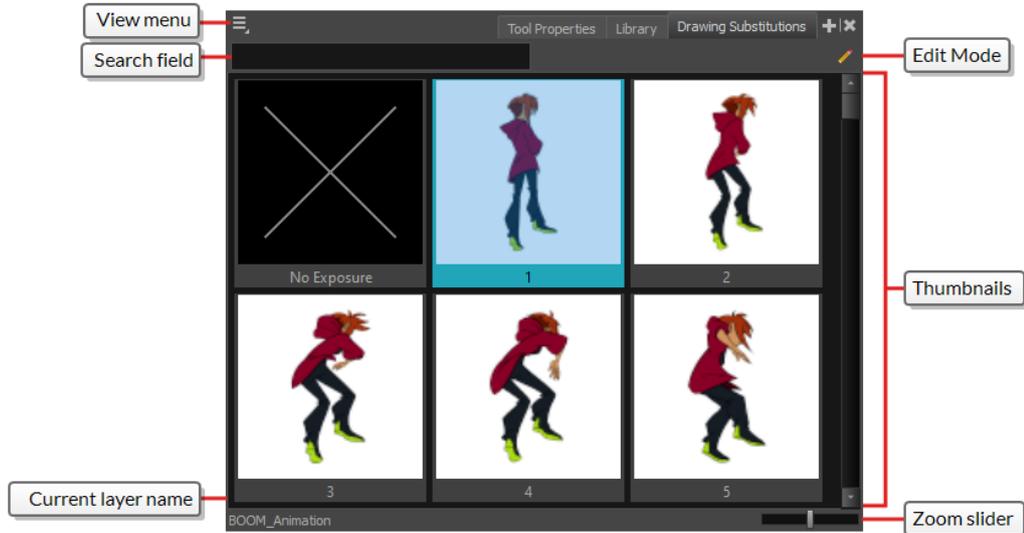
Através do painel de substituição de desenhos da biblioteca, é facilitada a troca de desenhos alterando rapidamente a exposição do quadro atual para o novo. Tal troca de desenhos é bastante utilizada na técnica de animação *cutout*, para animar bocas, mãos, pés etc.

Figura 5 – Painel de substituição de desenhos da biblioteca



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/reference/view/library-view.html?Highlight=library%20> > (acessado em 26 de abril de 2021)

Figura 6 – Aba de visualização de desenhos da biblioteca



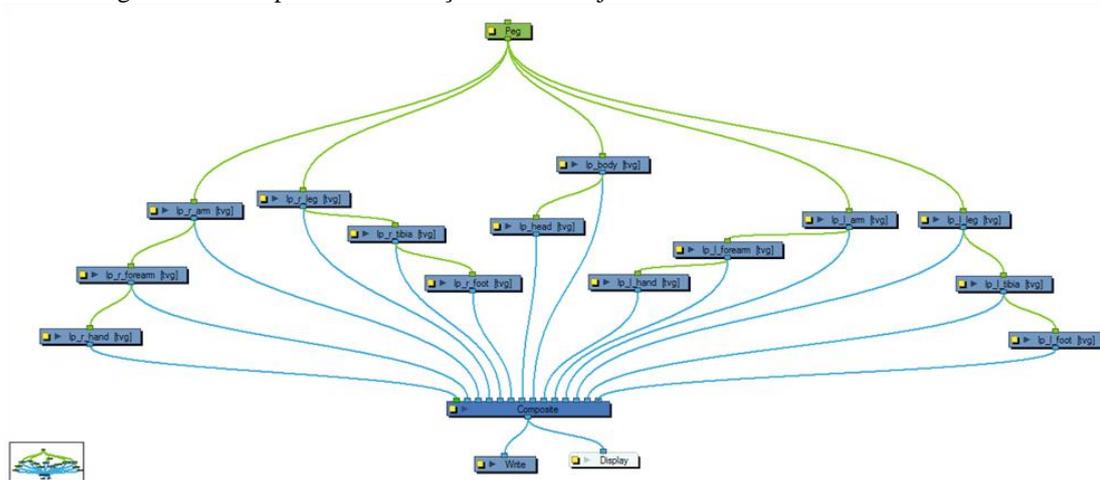
Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/reference/view/drawing-picker-view.html> > (acessado em 26 de abril de 2021)

## 2.4.5 NODEVIEW

A *nodeview* facilita a visualização geral de qualquer elemento e efeito que compõe a cena. Cada node é correspondente a uma *layer* (camada) na *timeline*.

Com este tipo de visualização que a *nodeview* fornece e o sistema entre nodes, que podem se conectar uns aos outros, se torna possível e fácil a criação de máscaras, filtragens e mesclas de desenhos para a construção de um *rig*.

Figura 7 – Exemplo da visualização de um conjunto de *nodes* através da *nodeview*

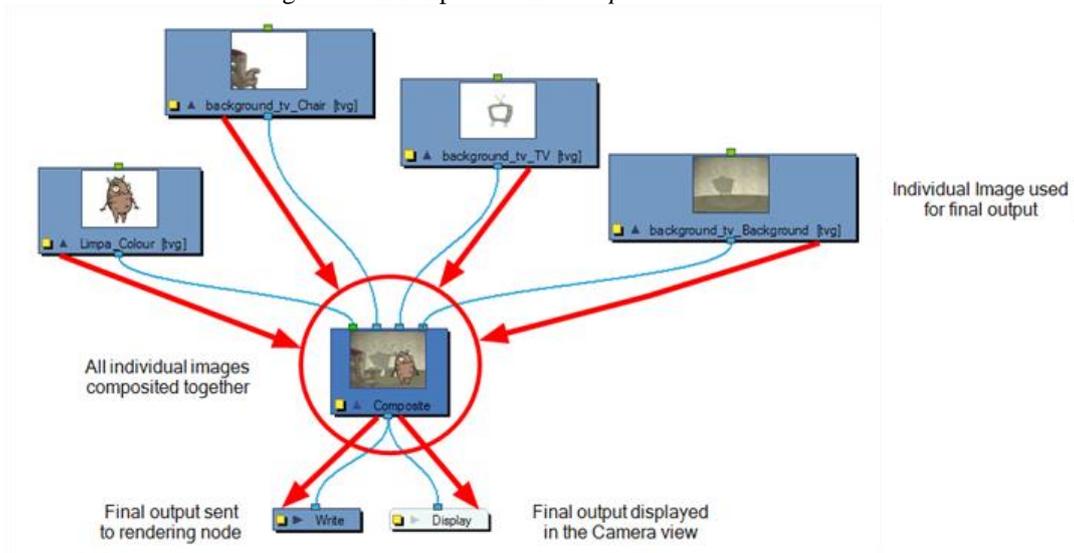


Fonte: < <https://docs.toonboom.com/help/harmony-14/premium/reference/node/node.html> > (acesso em 25 de abril de 2021)

#### 2.4.6 COMPOSITE NODE

A *composite* permite usar vários elementos para produzir uma única imagem de resultado. Ou seja, através das conexões de vários *drawings*, elementos e nodes de efeitos conectados à ele é possível a renderização de uma imagem, seja ela em *bitmap*, vetor, ou formatos de vídeo. Essas configurações de *render* são decididas nas configurações do node “*Write*”.

Através da *composite* também é possível determinar a ordem das camadas se elas estiverem na mesma posição de eixo Z (para frente e para trás).

Figura 8 – Exemplo de uma *composite* na *nodeview*

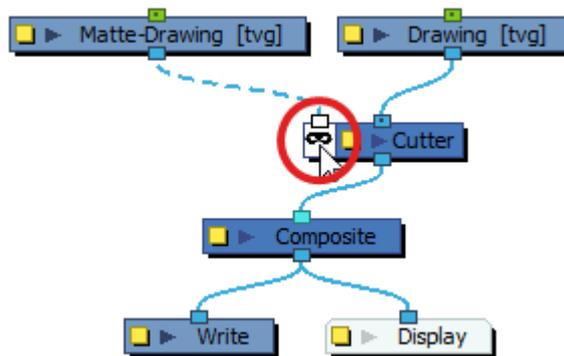
Fonte:< <https://docs.toonboom.com/help/harmony-14/premium/reference/node/combine/composite-node.html> > (acesso em 25 de abril de 2021)

## 2.4.7 CUTTER NODE

A função do *Cutter*, ou traduzindo para o português que significa “cortador”, tem sua função como o nome já diz, ele corta uma parte da imagem. Para isso acontecer, o *Cutter* precisa ter sua entrada conectada a um desenho que se deseja cortar, é preciso também conectar ao lado esquerdo do *node*, onde possui um ícone de máscara, um desenho que servirá como *matte*. O *Cutter* assumirá a formato do *matte* para fazer o corte.

Tal efeito é útil para várias coisas, um exemplo é utilizar o *Cutter* para a construção do olho, determinando que a pupila seja visível apenas na parte branca do olho.

Como todos os efeitos que utilizam um *matte*, o *Cutter* possui a opção “*Inverted*” que tem a função de inverter o efeito.

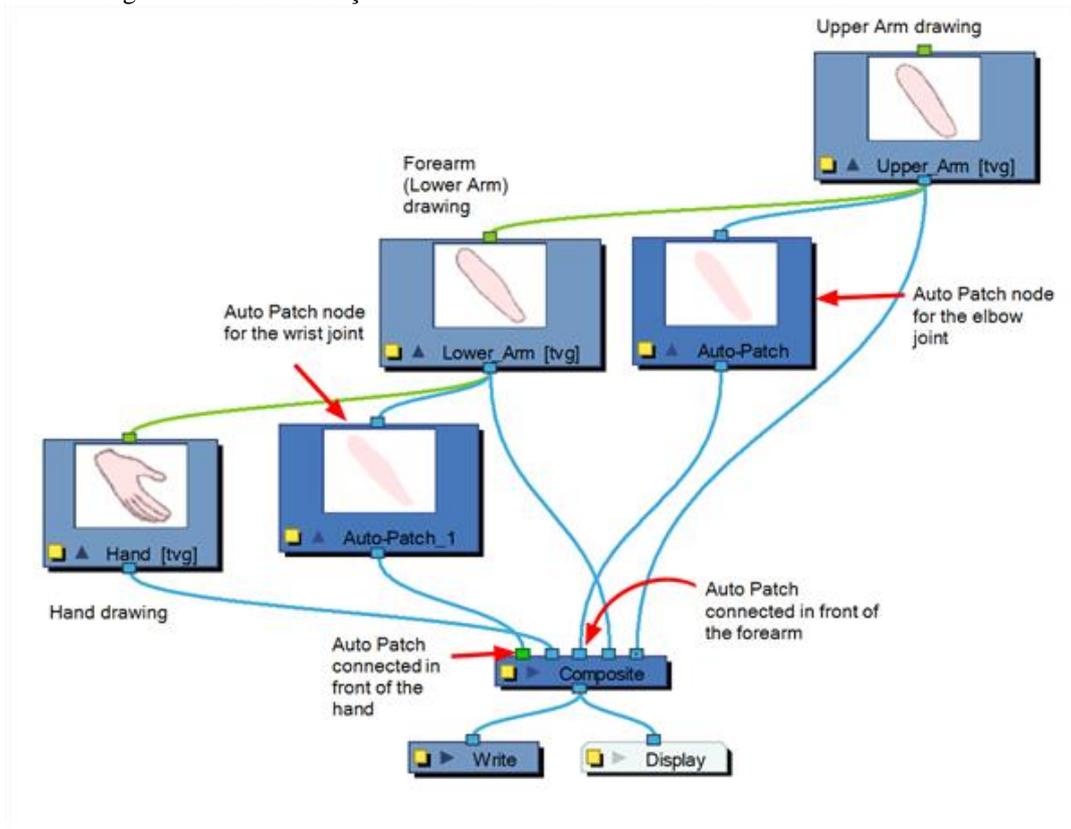
Figura 9 – Demonstração do *Cutter node* na *node view*

Fonte:< <https://docs.toonboom.com/help/harmony-20/premium/reference/node/combine/cutter-node.html?Highlight=cutter> > (acesso em 25 de abril de 2021)

## 2.4.8 AUTO-PATCH NODE

Este efeito é utilizado principalmente para criar os remendos das juntas de braços e pernas. Ao invés de remover a linha nas junções do braço, para que elas não apareçam no meio é possível deixa-las inteiras, sendo separadas as linhas na camada *LineArt* do *drawing* e o preenchimento na camada *ColourArt* e plugando o *Auto-Patch*, as linhas das juntas serão automaticamente escondidas.

Figura 10 – Demonstração de como é conectado o *Auto-Patch node* na *node view*

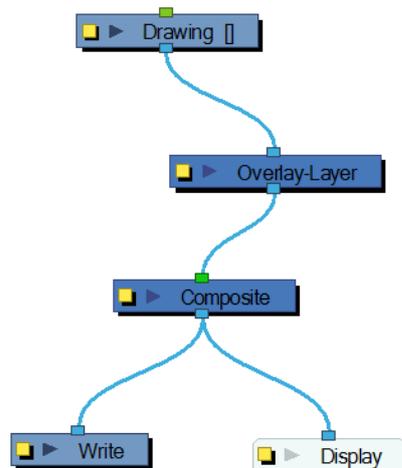


Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/reference/node/filter/auto-patch-node.html?Highlight=auto%20patch%20node> > (acesso em 25 de abril de 2021)

## 2.4.9 OVERLAY-LAYER NODE

Este efeito tem a função de filtrar qualquer desenho que se encontra na camada de *OverlayArt* do *drawing* na qual está plugada, sendo possível trazer para frente dos outros *drawings* ao conectar a frente de outras camadas na *composite*. Um exemplo da utilização deste efeito é adicionar uma parte da linha do antebraço na camada *OverlayArt* do *drawing* e plugar o *Overlay-Layer* à frente do *Auto-Patch* do braço, para que cause um efeito de dobra no interior do cotovelo. Deixando mais próximo de como um braço real funciona.

Figura 11 – Exemplo de conexão do efeito *Overlay-Layer* na *node view*



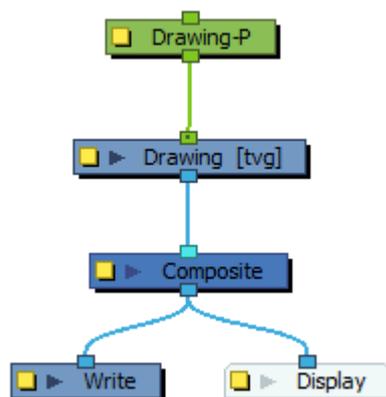
Fonte:< <https://docs.toonboom.com/help/harmony-20/premium/reference/node/filter/overlay-layer-node.html?Highlight=overlay-layer%20node> > (acesso em 26 de abril de 2021)

#### 2.4.10 PEG

As *pegs* são camadas de trajetórias que não possuem desenhos. São utilizadas como controladores, que controlam o desenho na qual estão conectadas. Animar uma *peg* faz com que as informações de posição e animação do elemento se mantenham guardadas separadas do *drawing*. Caso a animação precise ser apagada, basta excluir a *peg* que continha essas informações de *keyframes*.

As *pegs* são bastante utilizadas para controlar mais de um desenho por vez, isso facilita na hora da criação de uma hierarquia de *rig*, por exemplo.

Figura 12 – Demonstração de uma *peg* conectada a um *drawing* na *node view*



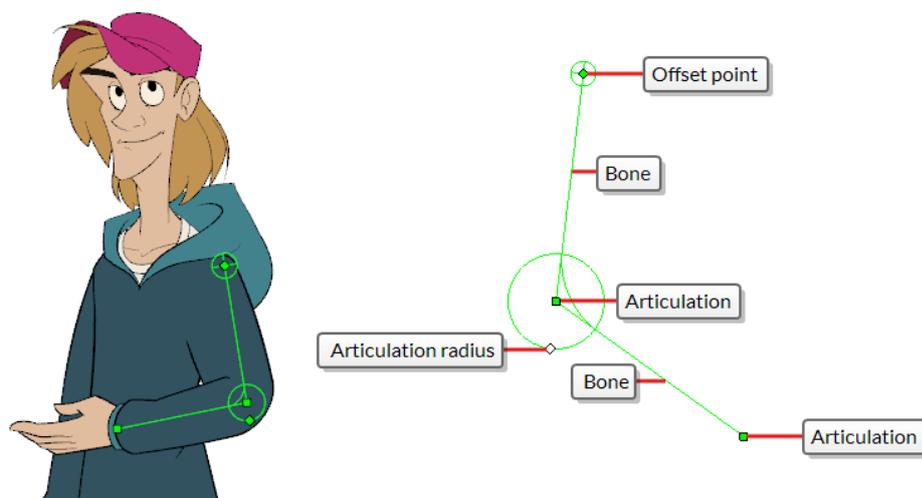
Fonte:< <https://docs.toonboom.com/help/harmony-20/premium/reference/node/generator/element-node.html?Highlight=draweing%20node%20> > (acesso em 25 de abril de 2021)

### 2.4.11 DEFORMADORES

Através dos deformadores é possível animar qualquer tipo de imagem, como *bitmaps*, vetores e até gradientes e texturas. Elas atuam como um esqueleto, com ossos e juntas (articulações), que agem diretamente com o desenho que está conectado à deformação. Existem de vários tipos, para situações específicas. São eles:

- *Bone*: O deformador por *bones*, ou traduzindo para o português, a “deformação óssea”, permite criar uma estrutura de articulações que são flexíveis em uma imagem. Esta deformação é comumente utilizada para a animação de braços ou pernas sem a necessidade de manter estes membros separados em diferentes *drawings*, com apenas uma *layer* com o desenho do braço completo é possível dividir entre a articulação do ombro, cotovelo e pulso. Estes pontos podem ser animados através de uma distorção calculada pelo *software* conforme é movimentado fazendo com que tenha o mesmo resultado se as peças tivessem sido separadas em *drawings* diferentes. Este deformador também pode ser utilizado para animar articulações do desenho do torso ou dedos da mão, por exemplo.

Figura 13 – Exemplificação do uso do deformador *Bone* e indicações de duas partes

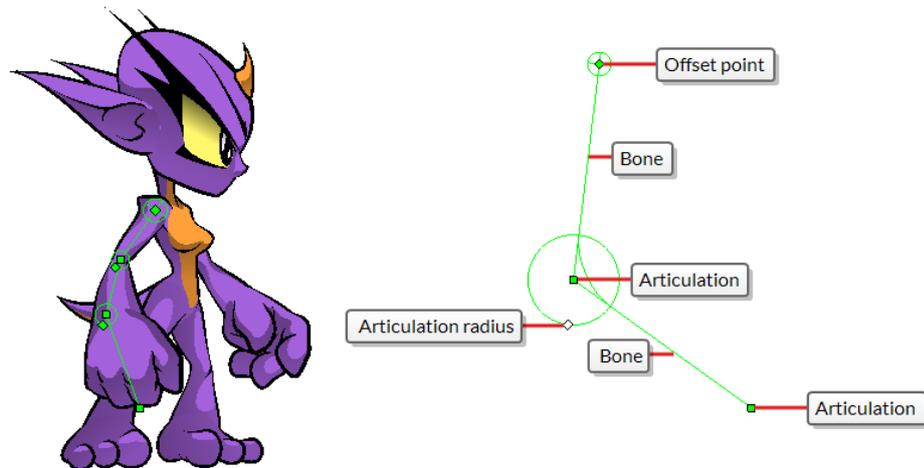


Fonte: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-bone-deformation.html>> (acesso em 25 de abril de 2021)

- *Game Bone*: A deformação de *Game Bone* ou, em português, “deformação óssea de jogo” tem a mesma função do deformador de *bones*. Deformar, esticar, encurtar ou

girar as articulações já criadas para cada junta, seja do braço, como perna ou qualquer outra estrutura que necessite de articulação. As suas diferenças ficam na parte de otimização, visto que, a deformação óssea de jogo é otimizada para *engines* de jogos como a *Unity*. Ou seja, é normalmente utilizado para o desenvolvimento de animações para jogos.

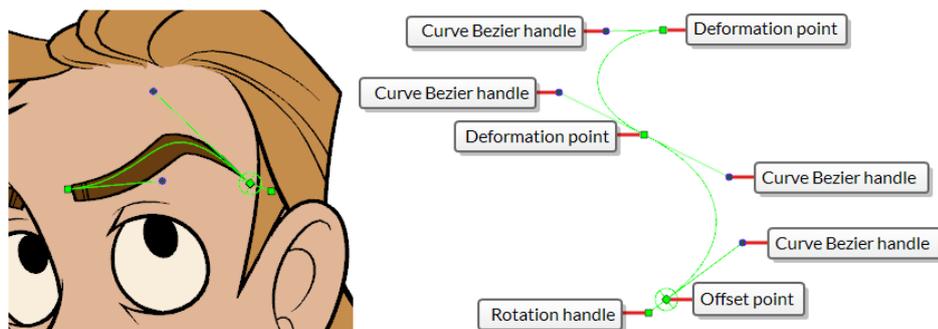
Figura 14 – Exemplificação do uso do deformador *Game Bone* e suas partes



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/deformation/about-game-bone-deformation.html> >  
(acesso em 25 de abril de 2021)

- *Curve*: O deformador *Curve*, ou traduzindo do inglês, “deformador de curvas”, é exatamente o que nome já revela, a deformação é feita através da edição de curvas de *bezier*. A principal diferença entre os deformadores de *bones*, é que estes são conectados de um ponto ao outro através de linhas curvas, sem muitas limitações de movimento. É comumente utilizado para deformar sobrancelhas, cabelos, torsos e até braços e pernas dependendo do estilo visual optado pelo artista.

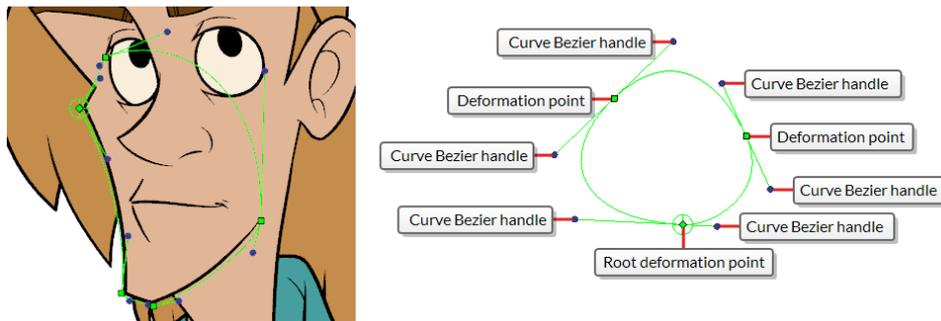
Figura 15 – Exemplificação do uso do deformador *Curve* e indicações de suas partes



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/deformation/about-curve-deformation.html> > (acesso em 25 de abril de 2021)

- **Envelope:** A deformação de envelope, é bastante semelhante às deformações feitas através do deformador de curvas, porém, o último ponto do envelope é conectado ao primeiro ponto, criando uma forma fechada, sendo possível modificar completamente o formato do desenho através da manipulação dos pontos de deformação e das curvas de *bezier*. A utilização deste deformador possui várias finalidades, pode ser usado para animar as partes que são mais leves como: cabelos, capas, ombros, mandíbula, roupas etc. Ou partes mais sólidas dos desenhos, para que pareçam que estão em ângulos diferentes, sem a necessidade de haver trocas de desenhos.

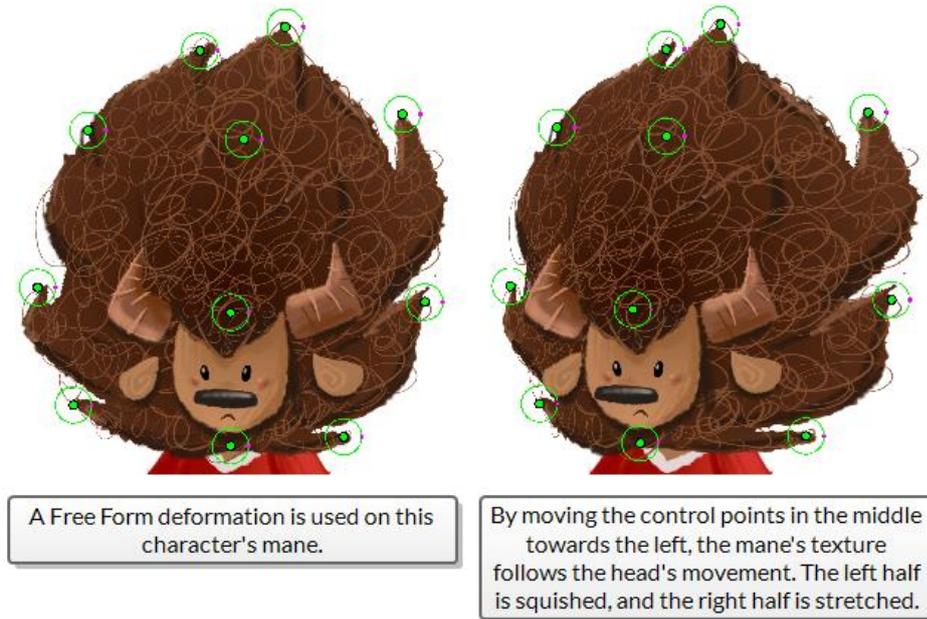
Figura 16 – Exemplificação do uso do deformador Envelope e indicações de suas partes



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/deformation/about-envelope-deformation.html> > (acesso em 25 de abril de 2021)

- **Free Form:** O deformador *Free Form*, traduzido como “forma livre”, consiste em criar pontos em uma área do desenho que seja texturizada ou que possua um padrão complexo, para ser animado conforme a peça texturizada seja movimentada. Ou seja, os pontos de deformação criam animações aleatórias, esticando e amassando a textura do desenho. Essas deformações podem ser controladas através dos pontos adicionados, definindo exatamente onde as deformações ocorrerão.

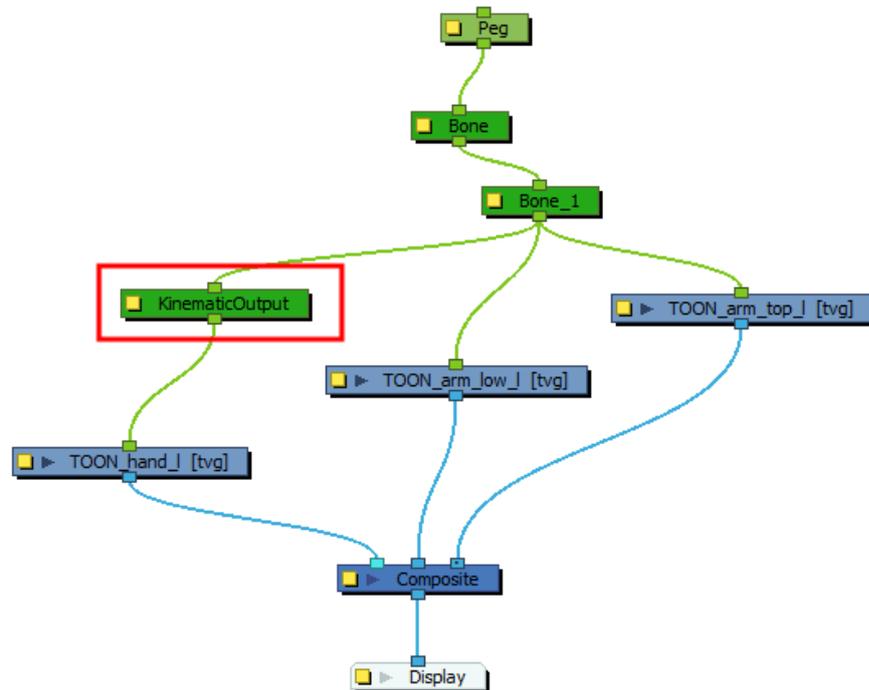
Figura 17 – Exemplificação do uso do deformador *Free Form* e suas partes



Fonte: < <https://docs.toonboom.com/help/harmony-20/premium/deformation/about-free-form-deformation.html> >  
(acesso em 25 de abril de 2021)

#### 2.4.12 KINEMATIC-OUTPUT NODE

Com este *node* é possível fazer com que um desenho siga o movimento de uma peça que se utiliza de deformadores para se mover, sem sofrer a deformação. Por exemplo, conectando a saída do *Kinematic-Output* ao *drawing* da mão, que se deseja ser influenciado no movimento. E a saída de deformação de um braço sendo plugado à entrada do *Kinematic-Output*.

Figura 18 – Exemplo do *Kinematic-Output node* funcionando na *node view*

Fonte:< <https://docs.toonboom.com/help/harmony-20/premium/reference/node/deformation/kinematic-output-node.html?Highlight=kinematic%20output> > (acesso em 26 de abril de 2021)

### 3 DESENVOLVIMENTO

De acordo com os objetivos deste trabalho, o foco está no desenvolvimento visual das personagens, na criação do *rig* e *props* utilizados para a animação das cenas do *teaser*. A autora se aprofundará somente no desenvolvimento das personagens principais da série “*Fix-It!*”, pois terão mais importância visto que estarão na maioria das cenas do *teaser*, portanto, irão precisar de uma elaboração maior.

Após as etapas iniciais de exploração e pesquisa de referências, foi possível desenvolver toda a parte de estrutura de história e narrativa como, a premissa da série e a invenção e fechamentos do universo ficcional onde as personagens estão inseridas, ambas ficaram definidos como:

“Procurando respostas para tempestades de lixos, uma dupla de mecânicos viaja por planetinhas e descobre esquemas irresponsáveis de uma indústria poluente, além de ajudar a todos em seus caminhos com reciclagem e perspicácia.”

Universo:

“Em uma aventura cômica e misteriosa, Lizie e Sphix passam por diferentes planetas até chegar ao maior deles: o Planetão. Uma espécie de capital do pequeno sistema solar e principal fonte de toda sucata despejada nos outros planetas. O Planeta Sucata, Abóbora e Chumbo são os pequenos planetinhas que orbitam o maior.

Depois que as indústrias Bacon se instalaram no Planetão, tempestades de lixo começam a se formar e vão para os outros planetas, dando o nome a cada um.

Passando por alguns planetas diferentes e conhecendo milhares de espécies de aliens, a dupla ajuda a todos que pode com suas ferramentas tecnológicas e habilidade biológica para tentar minimizar os danos causados pelo lixo das empresas. Além disso, milhares de moradores do planeta capital somem quando a fábrica se instala, deixando bairros quase desertos.

Sphix é o único ser “mágico”, com a habilidade de engolir sucata e cuspir algo completamente renovado, auxiliando Lizie e o tornando um alien muito precioso.”

#### 3.1.1 PERSONAGENS

Na etapa de exploração e definição em foram compostos painéis semânticos para cada personagem, baseando-se nas características de cada um, definidas na criação da premissa e desenvolvimento do universo, onde surgiram as ideias de como as personagens seriam fisicamente.

Os conceitos das personagens foram feitos inicialmente em 2016 e sofreram algumas mudanças da autora, pois foi levado em consideração a técnica de animação 2D *cutout*, escolhida para a realização do *teaser* e para a série em geral. Foi preciso simplificar o design das personagens e isso será descrito nas seções de cada personagem.

### 3.1.1.1 LIZIE

Um pouco da construção da protagonista é descrito na bíblia da série, como:

“Lizie é a alma do negócio. Extremamente dedicada e habilidosa, ela é uma garota capaz de arrumar qualquer estrago utilizando a mecânica e a reciclagem, mesmo às vezes se atrapalhando no processo. Destemida, ela topa qualquer desafio e com a ajuda de seu melhor amigo e sua chave de fenda boca 2000 (criada por ela mesma), Lizie pode superar qualquer obstáculo e aprender muito com as situações em que se envolve.

Sua vida muda completamente quando seu planeta é entulhado de sucata provinda do Planetão e seus pais somem buscando ajuda para o lugar onde moram. Assim, a menina decide ir atrás deles, ajudando todos aqueles que encontra em seu caminho com seus consertos incríveis.”

O processo de novo design dos personagens começou primeiro com a protagonista, por ser a principal personagem da história e que teria mais tempo em cena, foi optado por fazer as decisões de estilo nela. Para que assim, os demais personagens se adaptassem com seu novo estilo, deixando-os todos concisos.

Para começar com as novas explorações para uma nova definição de estilo da protagonista, foi analisado o que já se tinha em seu design e que as autoras gostariam que se mantivessem. Portanto, as características que seriam mantidas e sofreriam apenas algumas adaptações seriam:

- As cores: desde o começo do projeto, em 2016 as cores tinham sido escolhidas e aprovadas em unanimidade pelos integrantes do grupo. Pela personalidade da protagonista, de ser sempre ativa nas cenas e prestativa em todas nas situações, foram escolhidas cores vibrantes e marcantes. Até para se destacar de cenários que são o oposto dela, de tonalidades escuras e sem muita saturação. Fazendo que a personagem carregue e espalhe vida nas cenas.
- As estrelas no rosto: as estrelas foram escolhidas, para deixar uma característica marcante na personagem, como se fossem marcas de nascença. Distinguindo de

qualquer outro personagem que possa vir a surgir e que tenham as mesmas características de sua espécie (exemplo: seus pais).

- As luvas: elas são de fundamental importância para seu design, já que seu trabalho de mecânica envolve mexer com ferramentas que possam ser perigosas de se manipular, sendo essencial para exercer suas habilidades.
- Os óculos: também se tornou muito característico em seu design, podendo ser utilizado nas cenas dos episódios para dar um tom de forma dramática, para quando a personagem colocar os óculos, o público compreender que a personagem se tornou mais determinada naquele instante a resolver as situações. Acessório também muito característica para quando a personagem se aventura com sua moto pelo espaço.
- A jardineira: desde 2016 foi decidida a vestimenta da personagem como sendo uma jardineira. Pois se a personagem estivesse com o macacão de mecânico ficaria muito taxado de profissão o que ela faz na série, e este não é o intuito.
- As botas: as botas de galocha amarelas, também ficaram muito característica da personagem. Muito útil para poder se deslocar em seu pequeno planetinha no qual vive, no meio de vários destroços, peças de veículos e sucata. Além de se aventurar com elas em muitos outros terrenos através da galáxia. Essas botas são muito comuns também no vestuário do público-alvo da série, causando certa representatividade.

Figura 19 – Primeira versão de design aprovado da protagonista, em 2016



Fonte: A autora

Os principais elementos analisados pela autora, para o novo design da Lizie, foram os elementos que compõe o rosto. Parte principal para a personagem transmitir seus sentimentos.

Os olhos e nariz, no design de 2016 estavam muito pequenos em comparação ao tamanho de sua cabeça e corpo. Foi feita uma rápida pesquisa de referências para ser feita a nova versão dos elementos do rosto.

Figura 20 – Referências para novos elementos do rosto de Lizie

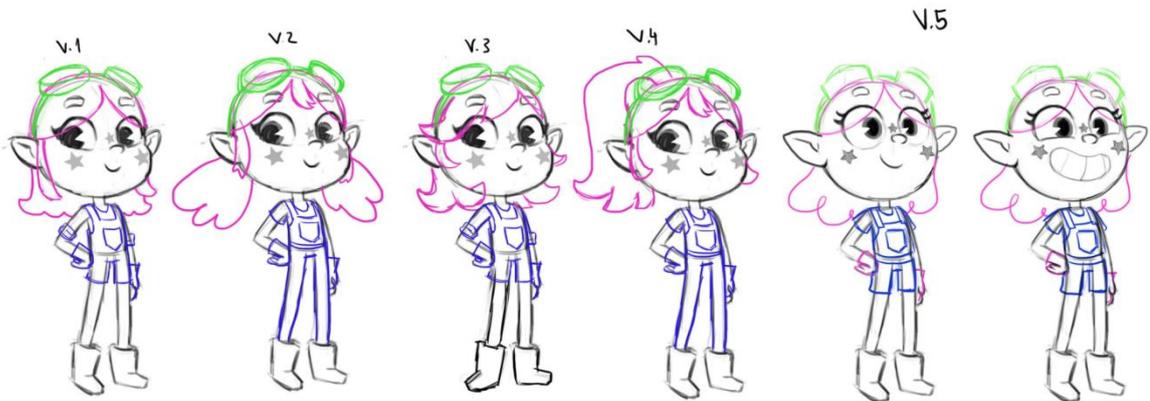


Fonte: A autora

Através das referências coletadas, foram feitas algumas explorações para a mudança dos elementos do rosto, além de novos testes de cabelo e tentativa de deixar sua silhueta mais esguia,

sem tanto volume da vestimenta. Para que não seja necessário animar o *follow through* das roupas, foi pensando em deixar os detalhes da jardineira mascarados no tronco, para serem animadas apenas através de deformadores.

Figura 21 – Explorações para o novo *design* de Lizie

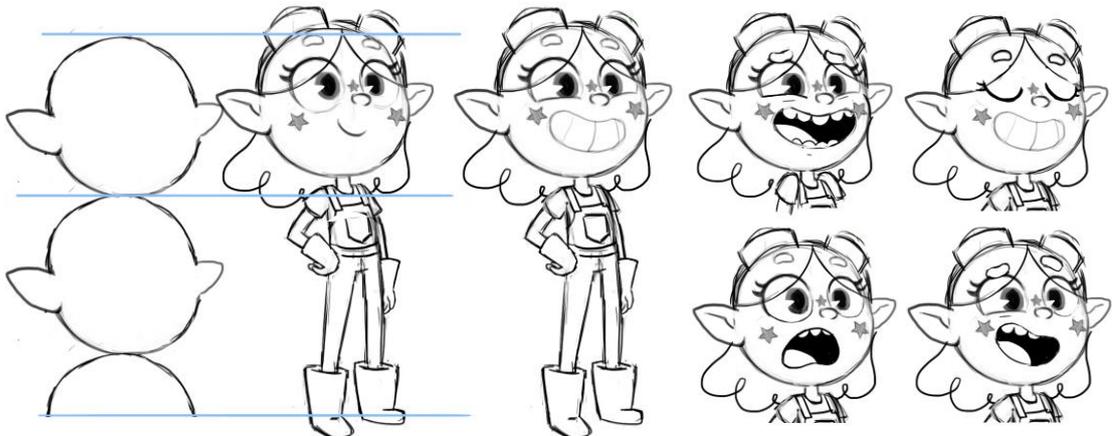


Fonte: A autora

A camiseta listrada também foi rejeitada do design, sendo trocada para uma branca, deixando sua leitura mais limpa. A opção escolhida para o cabelo foi uma versão um pouco mais simplificada da já existente do design de 2016. Por ser mais simples para animar, com menos mechas e detalhes.

Foi feito então, mais uma exploração final da sua proporção de cabeças e de como ficariam as expressões da personagem.

Figura 22 – Definição da proporção e expressões faciais da exploração n° 5 da Lizie



Fonte: A autora

Foram feitos mais testes de design de acessórios da exploração nº 5, até chegar à versão final do novo design da personagem.

Figura 23 – Adição de cores e testes de adereços da protagonista



Fonte: A autora

Figura 24 – Modelsheet com a versão final da protagonista



Fonte: A autora

### 3.1.1.2 SPHIX

Um pouco da construção do parceiro da protagonista é descrito na bíblia da série, como:

“Sphix é a coisa mais estranhamente fofa da galáxia. Ninguém sabe de onde veio e é o único da sua espécie. É o melhor amigo de Lizie e adora ajudar com os consertos utilizando sua habilidade de transformar sucata em objetos úteis! Não é muito amigável com quem não gosta e é bem precipitado. Curioso e atrapalhado, sempre arranja confusões quando tenta resolver os problemas sozinho, fazendo com que Lizie preste atenção para que não tenha que trabalhar dobrado.”

Com o design e estilo da protagonista já decidido, começou a exploração dos demais personagens. Para o parceiro de Lizie, inicialmente ele seria um alien que, além do poder de modificar as peças de sucata que é mastigado por ele, ele também serviria como uma espécie de bolsa. Mesma função da bolsa da Hermione, personagem da saga “Harry Potter”. Sphix teria todo um universo em sua barriga, onde seria possível a protagonista guardar suas invenções. E seus tentáculos quando unidos, teriam a função de alças. Mas isto foi reformulado quando a autora deste trabalho e sua parceira, Bianca Galdino revisaram a história e universo. Portanto, o design precisaria ser mudado do zero.

As principais mudanças que seriam feitas no design de Sphix:

- Elementos faciais;
- Troca dos tentáculos para braços;
- Proporção de cabeças;
- Cores;

Figura 25 – Primeira versão de *design* aprovado de Sphix, em 2016



Fonte: A autora

Para as novas explorações de Sphix, foram coletadas algumas referências para partir como base, já que até o formato em si do personagem ainda não estava decidido.

Figura 26 – Algumas referências para o novo design de Shix



Fonte: A autora

Foi feita então, uma primeira exploração do design de Sphix. Sua principal mudança foi a troca de tentáculos por braços com mãos e dedos. No topo de sua cabeça, foi adicionada uma espécie de “antena” natural, que pode ajudar o personagem a reconhecer o local com mais facilidade. A estrela que se encontrava na barriga, foi mudada de lugar, e adicionando um *shape* de cor diferente para diferenciar o rosto do resto do corpo.

Figura 27 – Primeira exploração para o novo *design* de Sphix

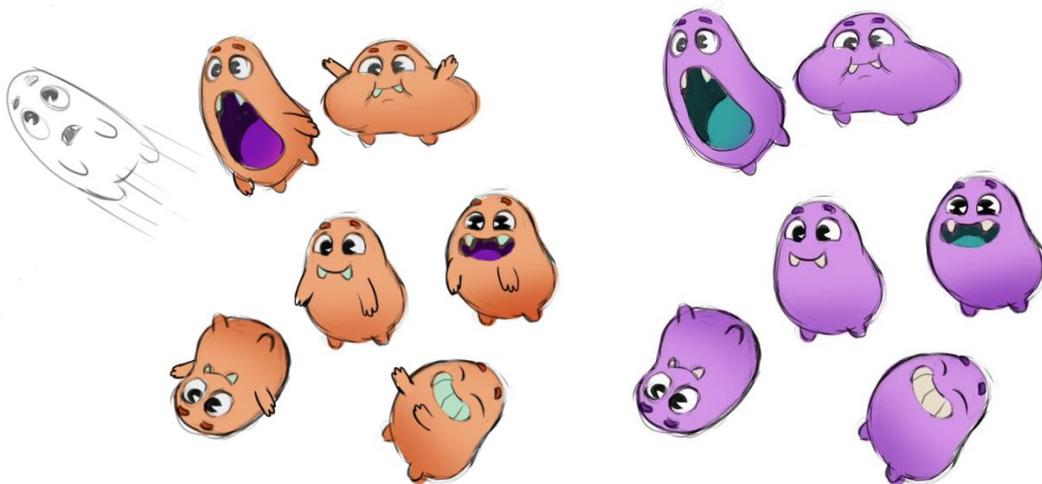


Fonte: A autora

Algumas coisas deste design foram repensadas, por se tratar de uma animação *cutout*, o *shape* do rosto, teria que ser feito separado e mascarado ao corpo e adicionado um deformador ao desenho. Para simular as mudanças de ângulo do personagem. Algo que se tornaria muito trabalhoso para a animação. E a ideia de ter mais um personagem complexo para animar além da protagonista Lizie, não agradaram as autoras do projeto.

Foi feita uma nova exploração, abordando uma ideia mais simples do personagem, deixando sua boca com mais evidência, visto que sua principal habilidade ocorre na sua mastigação.

Figura 28 – Mais explorações de *design* e opções de cores para o Sphix



Fonte: A autora

Mais uma vez, se buscou analisar na exploração, como seria a movimentação do personagem na série e as interações com a protagonista. Esta exploração foi pensada em deixar o personagem bastante maleável, quase sem peso nenhum, se movimentando através de pulos e até flutuando algumas vezes. Os membros foram reduzidos e um teste foi feito com apenas pernas, sem braços.

Uma coisa que agradou as autoras nesta exploração, foi a combinação das duas cores: o laranja e o roxo. Os dentes pontiagudos e separados do personagem o deixaram com características divertidas e fofas.

Decidiu-se fazer mais uma exploração com estas características citadas. Foi adicionada a antena de formato de folha feito na primeira exploração, pois pode ser feita a conexão da reciclagem. Os olhos também foram padronizados como os da protagonista, adicionando o brilho de formato triangular. Por fim, surge a versão final de Sphix.

Figura 29 – Versão final do *design* de Sphix



Fonte: A autora

### 3.1.1.3 SR. PORCO

Um pouco da construção do antagonista da série é descrito na bíblia, como:

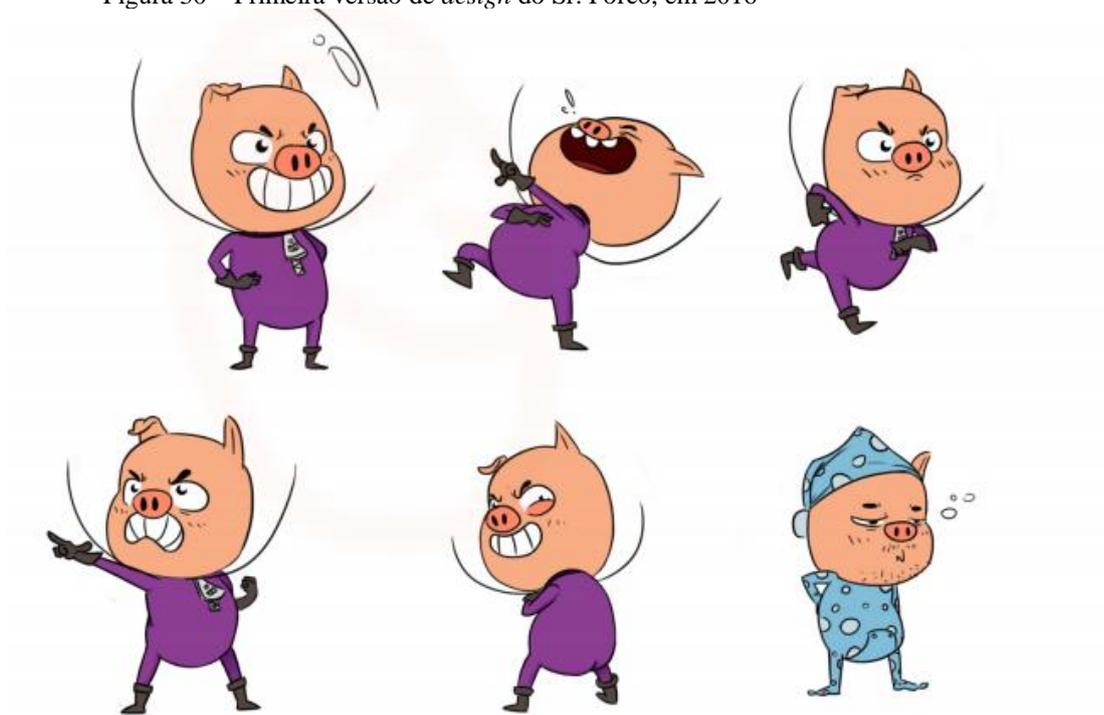
“Sr. Porco, o Supremo, é um grande magnata do Planetão. Construiu sua a empresa do zero com suas próprias mãos e suor... pelo menos é o que ele conta para todo mundo. Sempre quer cortar gastos e ganhar muito dinheiro. Sendo assim, sua empresa acaba se tornando uma grande poluidora. Quando descobre Sphix, decide pegar o alien para sí, já que tem uma habilidade incrível capaz de transformar lixo em peças de grande qualidade, o que levaria o suíno a gastar muito pouco ou nada para produzir seus produtos. Possui uma personalidade um tanto irritadiça, porém se sente melhor e mais disposto ouvindo músicas de ninar, fazendo exercícios e dançando balé.”

Antes de seu design ser passado para o papel em 2016, os integrantes do projeto já tinham em mente como o antagonista seria visualmente. Ele seria pequeno, menor até que a protagonista. Com características que deixam com aparência de fofo, e não de vilão.

Para a versão nova e atualizada de 2020, está ideia continuou a mesma. Portanto para o novo design do Sr. Porco, não foi preciso ser feita uma busca muito ampla de referências. As características principais que seriam alteradas em seu design eram:

- Proporção de cabeça;
- Padronização do estilo de olho já feito na protagonista e no Sphix;
- Nova vestimenta;

Figura 30 – Primeira versão de *design* do Sr. Porco, em 2016



Fonte: Victor Caraméz

Figura 31 – Referências para as vestimentas do Sr. Porco

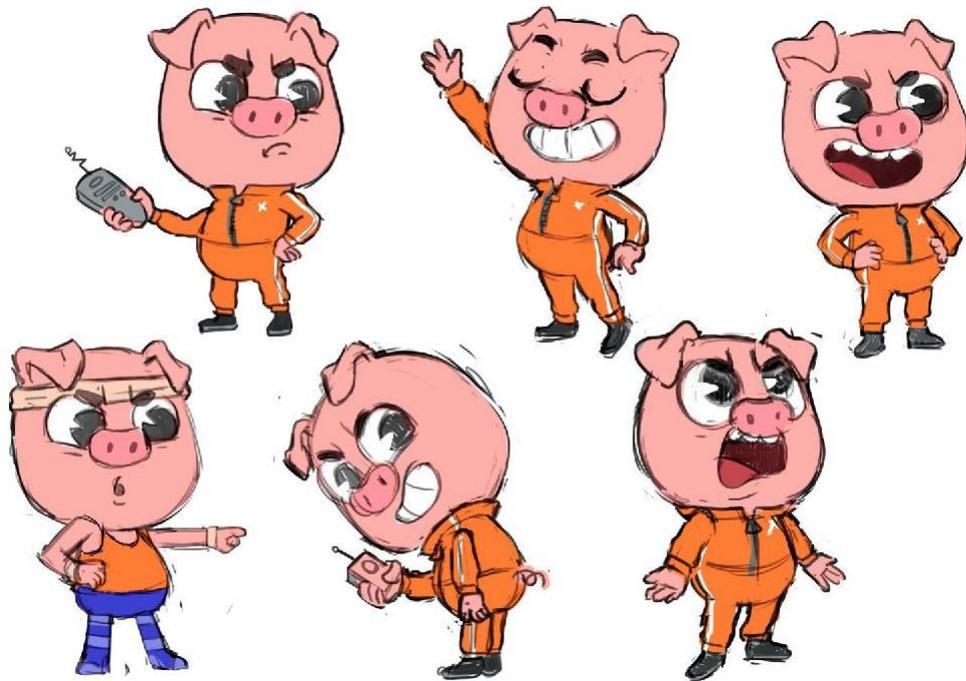


Fonte: A autora

Para o novo design do Sr. Porco, foi buscado referências de roupas que se encaixassem melhor em sua personalidade narcisista e egocêntrica.

A proporção de cabeças do personagem foi pouco alterada, e com a atualização do estilo dos elementos do rosto já padronizados pela protagonista, gerou então, a nova versão do antagonista da série “*Fix-It!*”.

Figura 32 – Nova versão de *design* do antagonista



Fonte: A autora

Figura 33 – *Modelsheet* com a versão final do Sr. Porco

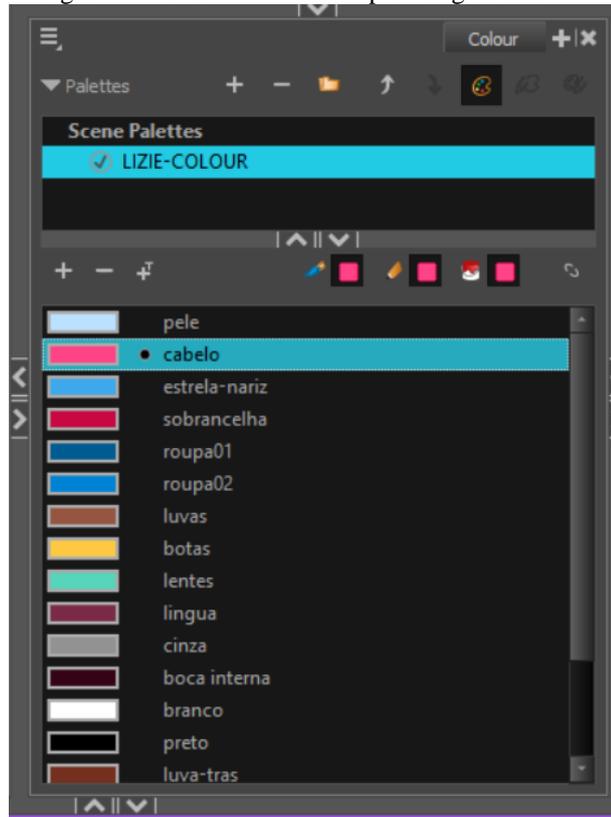
Fonte: A autora

### 3.2 PREPARAÇÃO PARA O RIG

Com o trabalho de design e o *modelsheet* dos personagens concluído, começa a etapa de preparação para o *rig*, ou seja, fazer a separação das várias partes do corpo dos personagens em diferentes *layers*, para que possam ser independentes uma das outras e assim fazer com que se movimentem mais tarde, quando estiverem riggadas.

Nesta etapa, a imagem do *modelsheet* é importada no *ToonBoom Harmony Premium*, que servirá de referência para o trabalho de vetorização de parte do desenho. Começando pela paleta de cor, utilizando da ferramenta *Dropper*, “conta-gotas” em português, e clicando nas cores da imagem do *modelsheet*, é copiado o código da cor para o dentro do *software*. É de suma importância criar uma paleta organizada para cada personagem, para não ter problemas futuros importando os *rigs* nas demais cenas do projeto. Nesta parte é preciso clicar no “+” na aba “*Colour*”, e ir adicionando cada cor do personagem e nomeando de acordo com que cada uma representa. Após a coleta das cores, é renomeada a paleta para o nome do personagem que está sendo feito.

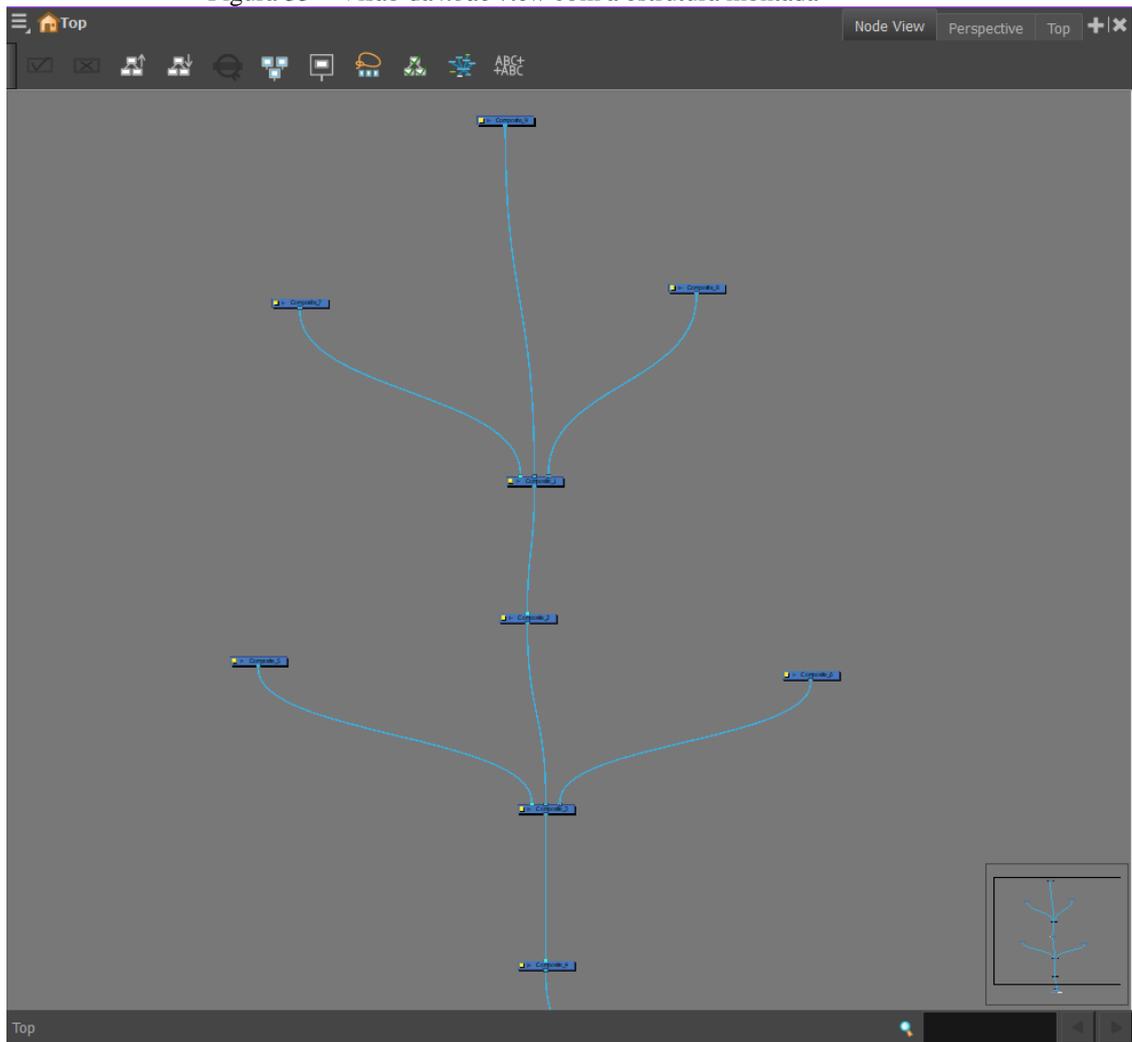
Figura 34 – Paleta de cores da personagem Lize



Fonte: A autora

### 3.2.1 ESTRUTURA

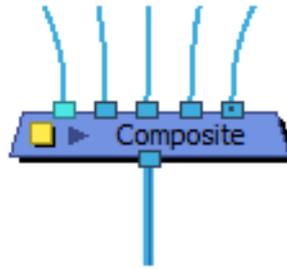
Para melhor organização, foi criada uma “espinha dorsal” da personagem na *node view*, através da inserção de várias *composites*, conectando-as umas às outras, com espaçamentos suficientes para que outras possam ser adicionadas ao longo do processo de criação do *rig*. Esta estrutura inicial já serve para ter uma visão do personagem ao todo. Deixando clara a divisão das partes dos membros do corpo.

Figura 35 – Visão da *node view* com a estrutura montada

Fonte: A autora

Todas as *composites* foram alteradas para o modo *pass through*, através das configurações da *composite*. O modo *pass through*, traduzido para o português como “passe através”, tem a função que o nome já revela, passar informações como: o desenho, efeitos e transformações desta *composite* para próxima a qual está conectada. Este modo é muito útil, pois além de passar os efeitos que serão criados, é dada liberdade de trazer um desenho para frente ou para trás de outro desenho que se encontra em outra *composite* ao longo do *rig*, um exemplo disto é o desenho da manga da camiseta, que será posicionada ora para trás, ora para frente do tronco, dependendo da pose em que a personagem se encontra.

Figura 36 – *Composite* em modo *pass trough*, localizada na *node view*



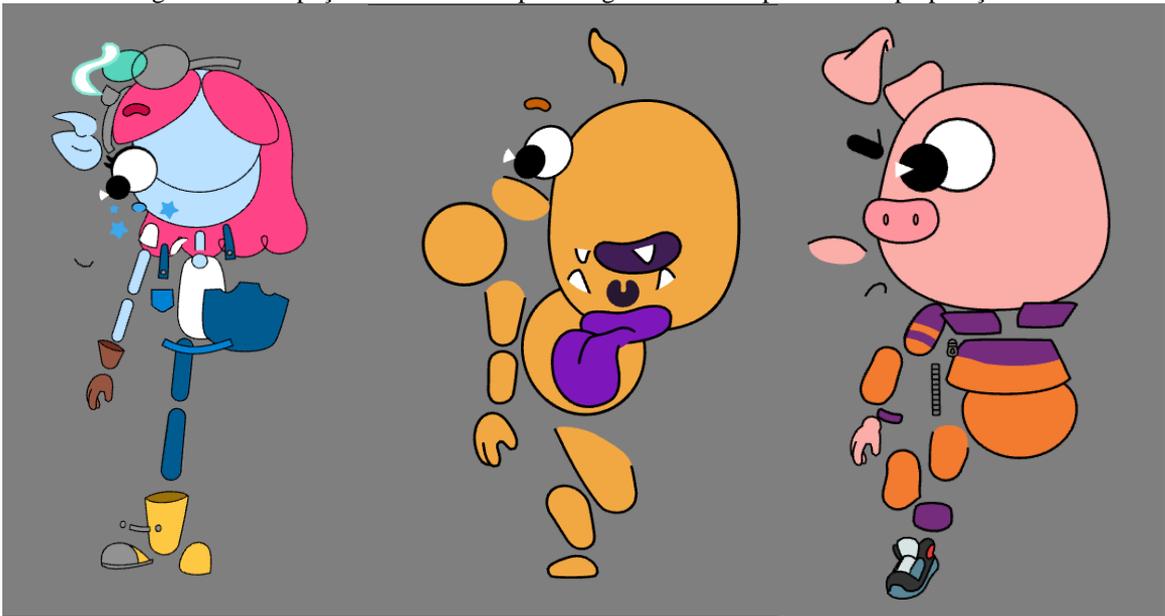
Fonte: A autora

O trabalho de vetorização é feito através das ferramentas de desenho disponíveis no *software*. Os desenhos podem começar a ser feitos partindo de uma forma básica como, o retângulo e a elipse e editando os pontos com a ferramenta de contorno. Ou pode ser feito através da ferramenta *Polyline*. A utilização do conjunto das ferramentas de desenho é a forma correta de se produzir as imagens, pois todos se complementam no processo. É criado um *drawing* novo a cada parte do corpo que é feito. Nomeando de acordo com a sua parte desenhada, para maior organização e facilidade na hora de encontrar as peças futuramente durante o processo de *rig* e animação.

Para economia de tempo e trabalho, foram criadas as peças de apenas um lado do corpo, o esquerdo, para que depois de *riggadas* no futuro, apenas sejam duplicadas para assim servirem como o lado direito sem a necessidade de *riggar* cada membro duas vezes. O olho, além de ter sido *riggado* apenas uma vez e duplicado depois, foi utilizado o mesmo desenho para os três personagens, adicionando apenas detalhes de cílios para Lizie, seguindo a referência do *modelsheet*.

Na figura podemos ver como foi feita a separação dos membros e de cada peça que compõe os personagens.

Figura 37 – As peças soltas de cada personagem durante o processo de preparação



Fonte: A autora

### 3.2.2 BIBLIOTECA

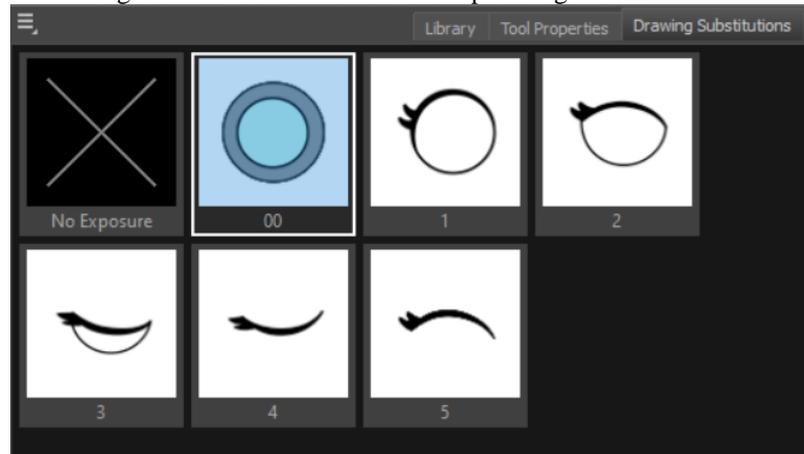
É de extrema importância e necessidade que cada personagem tenha sua biblioteca com diferentes desenhos de olhos, sobrancelhas e bocas para animar diferentes expressões e diálogos. Assim como é importante também criar uma biblioteca com diferentes mãos e pés para que as personagens possam interagir, gesticular, e transmitir suas emoções.

#### 3.2.2.1 OLHOS

Para a biblioteca do olho, foram gerados desenhos do olho fechado e seus intermediários para fechar um ciclo de piscada.

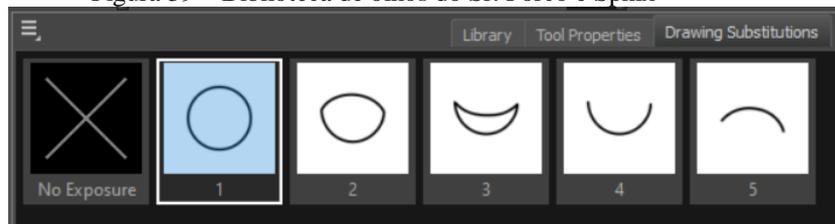
Como a personagem Lizie, aparece usando os óculos em algumas cenas do *teaser*, foi optado pela autora adicionar as lentes dos óculos já no desenho do olho. Tornando mais fácil o processo de animar.

Figura 38 – Biblioteca de olhos da personagem Lizie



Fonte: A autora

Figura 39 – Biblioteca de olhos do Sr. Porco e Sphix



Fonte: A autora

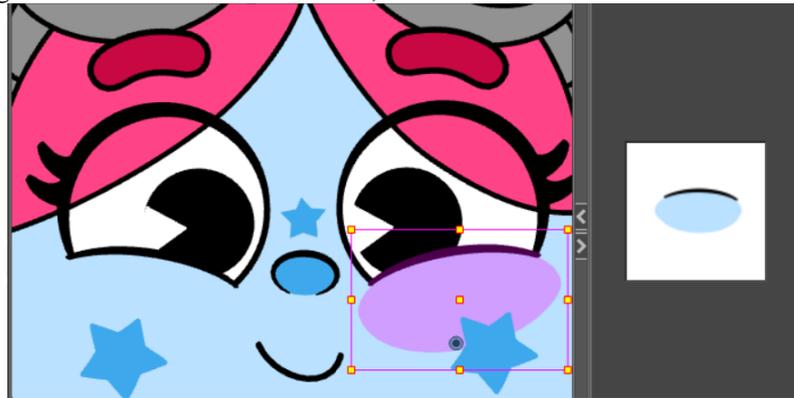
Por escolha da autora, ao invés de gerar um desenho de olho meio fechado para representar o “olho sorrindo”, foi criada uma peça adicional chamada “bochecha” para que ela fique por cima e cubra metade do olho quando a personagem estiver com um sorriso.

Figura 40 – Exemplo de “olho sorrindo”, meio fechado.



Fonte: < [https://br.freepik.com/fotos-gratis/menininha-em-orelhas-bunny-ficar-com-coelho\\_4076796.htm#page=4&query=kid+smiling&position=20](https://br.freepik.com/fotos-gratis/menininha-em-orelhas-bunny-ficar-com-coelho_4076796.htm#page=4&query=kid+smiling&position=20) > (acessado em 25 de abril de 2021)

Figura 41 – “Olho sorrindo” da Lizie, através do desenho “bochecha”



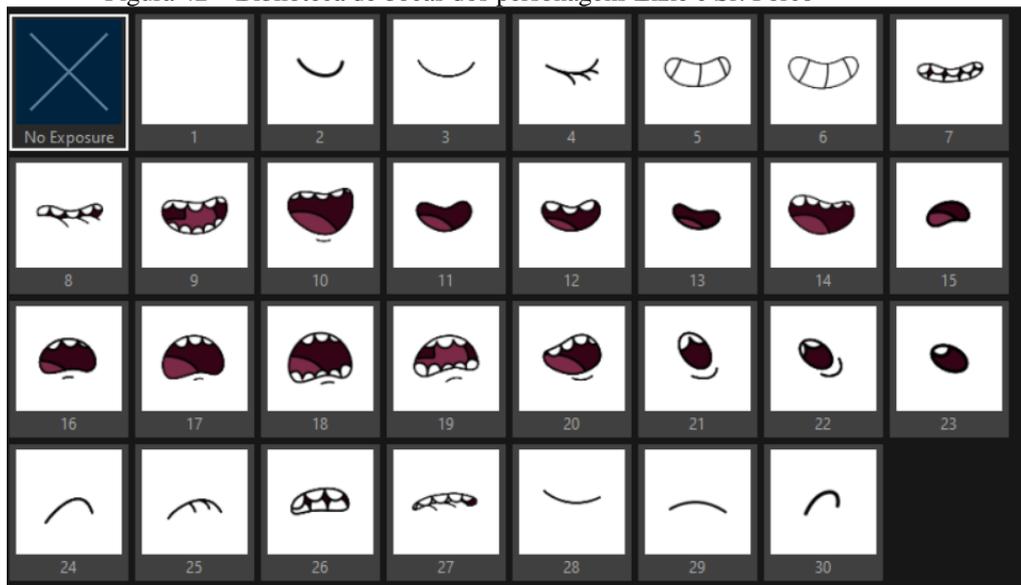
Fonte: A autora

Esta escolha deu mais liberdade para as animadoras do projeto decidirem o quanto fechar e abrir os olhos e qual o *timing* que isso acontece na cena, sem a necessidade de ficar gerando novos desenhos de olhos para a biblioteca.

### 3.2.2.2 BOCAS

Para as bocas, foram feitos os desenhos de sorriso de boca aberta, sorriso de boca fechada, bocas tristes e as bocas que representam os fonemas tristes e felizes, para que seja possível animar diálogos, o chamado *lipsync*.

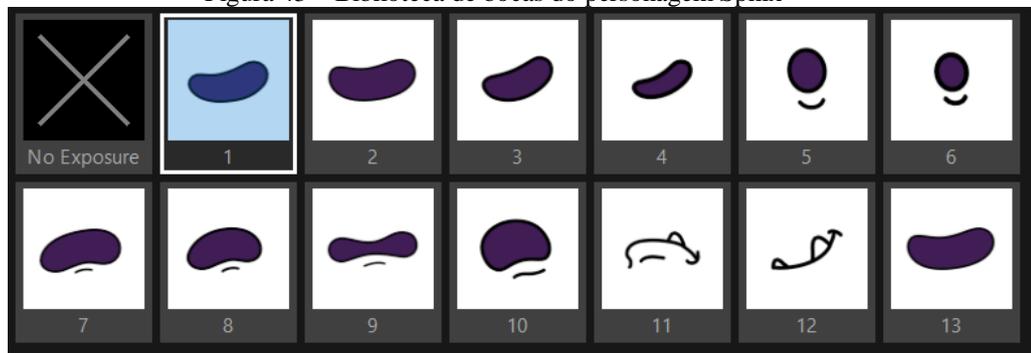
Figura 42 – Biblioteca de bocas dos personagens Lizie e Sr. Porco



Fonte: A autora

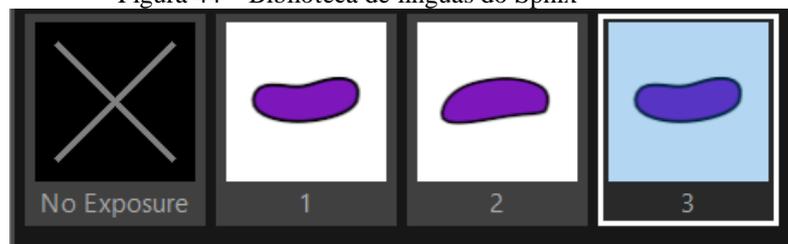
Surgiu a necessidade de se fazer a boca do personagem Sphix um pouco mais complexa comparado aos outros dois personagens que compartilham da mesma biblioteca. Embora o personagem não possuir a habilidade da fala, ele possui um poder que será possível somente através de suas mastigações. Por conta deste fato, e das cenas que são mostradas no *animatic*, percebe-se que o personagem precisaria ser animado mastigando e de boca aberta. Dividiu-se a boca em cinco *drawings* diferentes, são eles: “boca”, representada pela parte interna da boca, onde os outros elementos serão mascarados. “garganta”, “dentes”, “língua-interna”, este representando a parte da língua que será mascarada dentro da boca e o desenho “língua”, que ficará para fora da máscara.

Figura 43 – Biblioteca de bocas do personagem Sphix



Fonte: A autora

Figura 44 – Biblioteca de línguas do Sphix



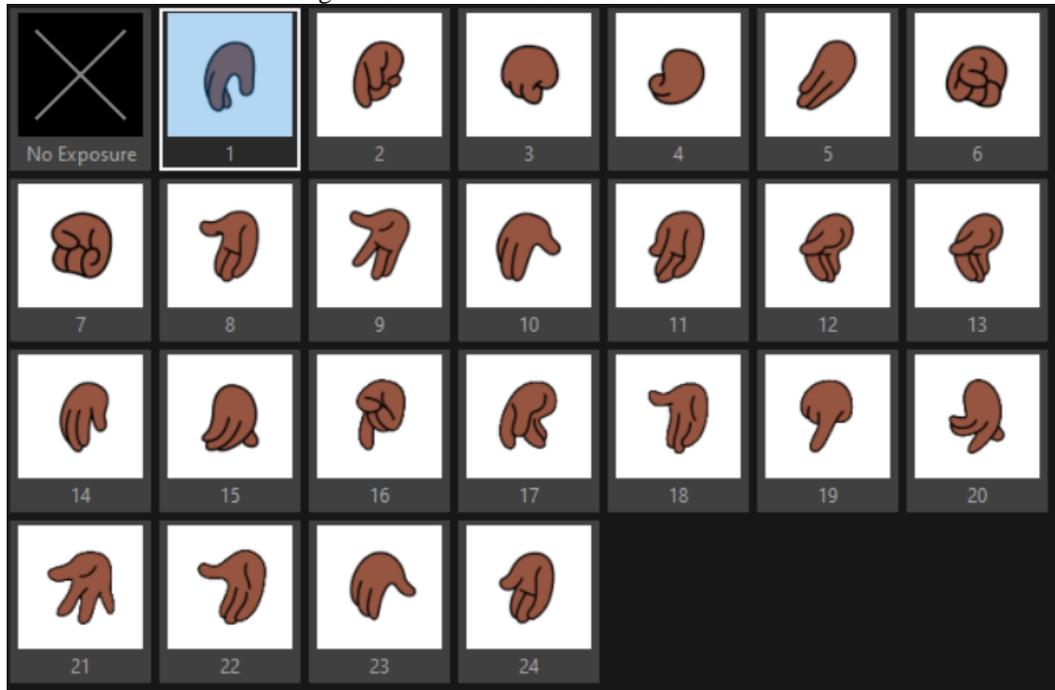
Fonte: A autora

### 3.2.2.3 MÃOS

Logicamente os dedos se movem independentes da palma da mão, porém para algumas produções de animação 2D *cutout* e para a o resultado deste projeto, isto se tornaria algo muito complexo e desnecessário visto que os movimentos das mãos ocorrem em poucos *frames*, é algo que acontece tão rápido que seria super refinado e que tomaria muito tempo devido a criação de *keyframes* para cada dedo da mão e que tais animações não seriam perceptíveis para o expectador. Por isso a técnica de troca de desenho se aplica para essa parte do corpo.

Foram feitas poses de mãos genéricas além de a autora utilizar como referência as cenas dos desenhos do *storyboard/animatic*.

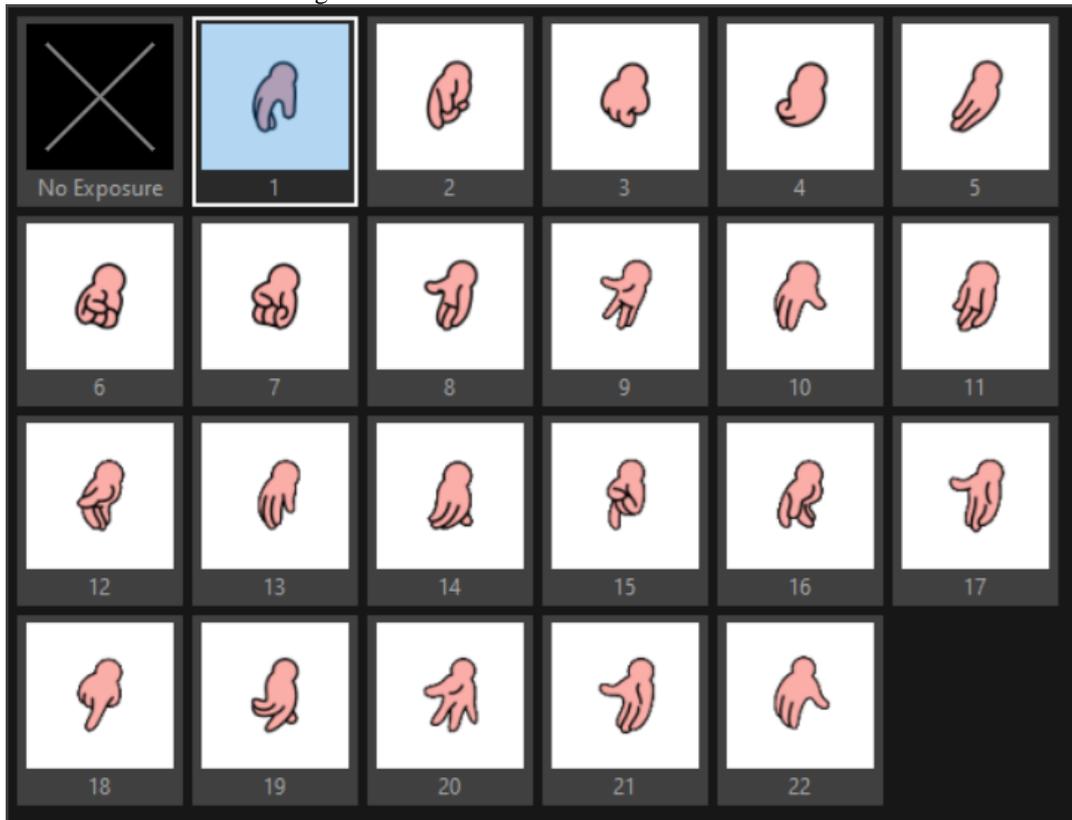
Figura 45 – Biblioteca de mãos da Lizie



Fonte: A autora

Através de pequenas modificações foi possível reutilizar as mãos da protagonista Lizie, para gerar as mãos do antagonista o Sr. Porco, economizando tempo de trabalho.

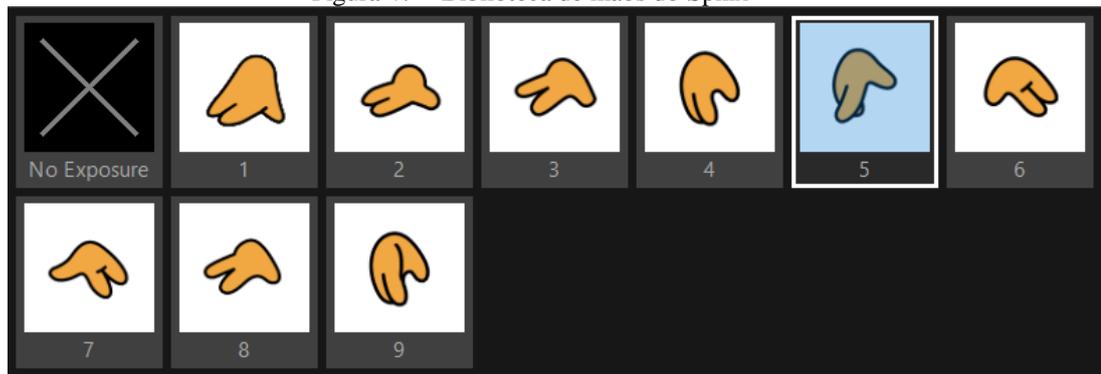
Figura 46 – Biblioteca de mãos do Sr. Porco



Fonte: A autora

Como não há muita troca de poses do personagem Sphix nas cenas do *teaser*, para se economizar tempo, as mãos foram feitas apenas as necessárias para as cenas e para seu *modelsheet*.

Figura 47 – Biblioteca de mãos do Sphix



Fonte: A autora

Na etapa de animação surgiu-se a necessidade de mais desenhos de mãos, que foram produzidas na hora e em seguida adicionadas na biblioteca.

### 3.2.2.4 PÉS

Os pés dos três personagens possuem o *design* diferente um do outro, devido as suas personalidades, por conta disto não foi possível reutilizar as peças.

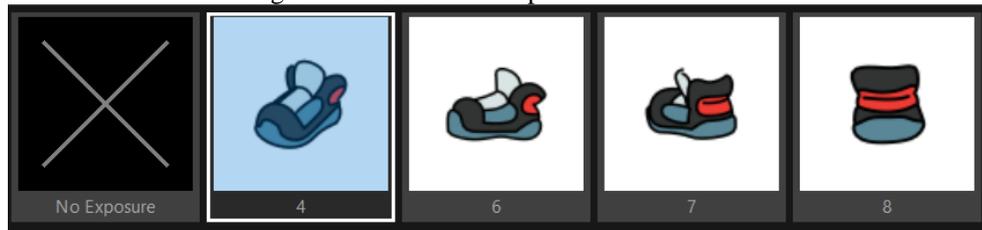
Foram feitos então, os desenhos dos pés dos personagens, Sr. Porco e Sphix baseados nas poses do *modelsheet*, e nas cenas do *animatic*.

Figura 48 – Biblioteca de pés do Sphix



Fonte: A autora

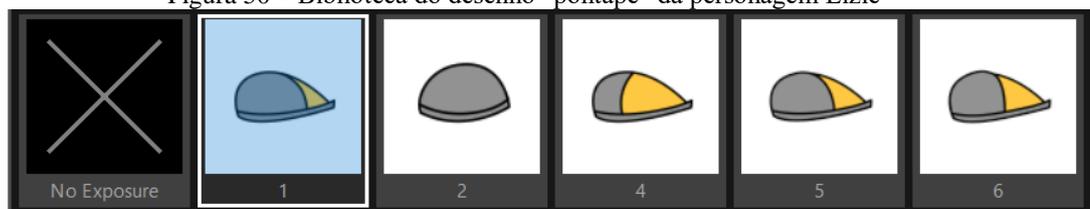
Figura 49 – Biblioteca de pés do Sr. Porco



Fonte: A autora

A bota da protagonista foi dividida em quatro partes, sendo elas, “bota-frente” é a maior parte da bota, esta será duplicada na etapa de *rigging* para que sirva também como a parte de trás da bota, “calcanhar”, “pontape” e “detalhe-bota”. Esta divisão foi feita de forma que, durante a animação, seja possível animar apenas partes específicas para que gere a impressão de troca de ângulo. Os elementos que mais ajudam a causar a mudança no ângulo são os detalhes da bota e a ponta do pé, que foram gerados variações de desenho nesta etapa de preparação da biblioteca.

Figura 50 – Biblioteca do desenho “pontape” da personagem Lizie



Fonte: A autora

### 3.2.3 RIGGING

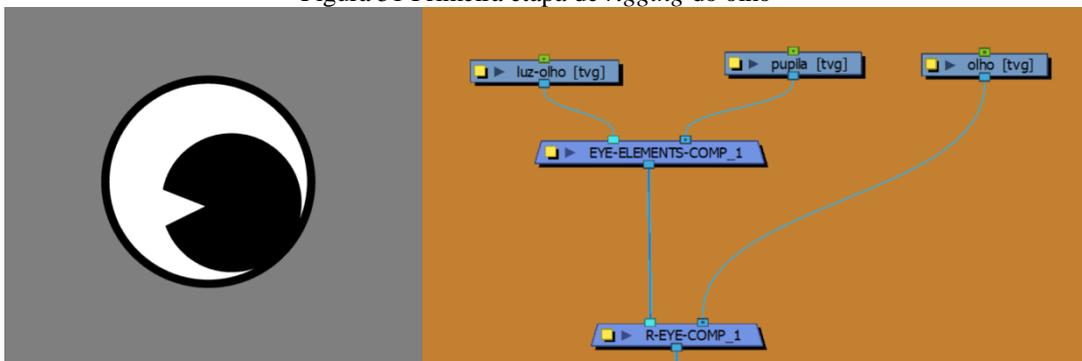
Depois de prontas as peças separadas por layers, chega à etapa de *rigging*. Etapa esta que se ajusta a posição dos *pivot points* (já citados na seção [2.2](#) deste trabalho) de cada peça, onde se trabalha para que as articulações funcionem, adição de máscaras, além dos controladores e a montagem da hierarquia onde tudo se conecta. As ferramentas que serão utilizadas nesta etapa foram citadas na seção [2.4](#) deste trabalho.

#### 3.2.3.1 ROSTO

Uma das partes de *rig* mais complexa do rosto foi o olho, pois foi necessário separá-los em partes para que a pupila pudesse ser movimentada e mascarada para que ficasse apenas visível na parte interna do olho.

É criada uma nova *composite* para plugar os *drawings* que fazem parte do olho. São eles: “pupila”, com seu desenho localizado na camada *LineArt*, “luz-olho”, este é o desenho que ficará em cima da pupila para um efeito de reflexo de luz nos olhos. “Olho” é o desenho que servirá como *matte* para que a pupila seja mascarada, sendo plugado em uma *composite* abaixo da *composite* que foram plugados seus elementos. O desenho foi dividido, sendo a linha na camada de *LineArt* do *drawing* e seu preenchimento localizado na *ColourArt*.

Figura 51 Primeira etapa de *rigging* do olho



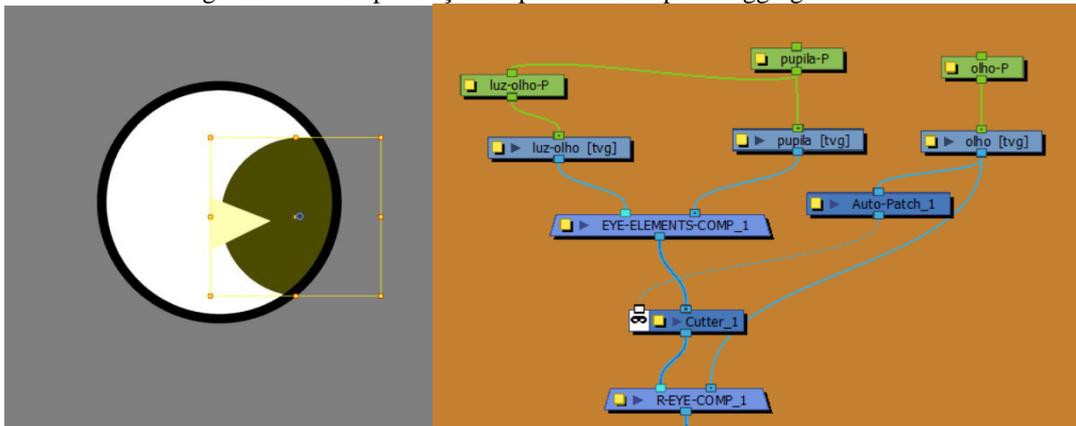
Fonte: A autora

É importado para *node view* os *nodes*: *Cutter* e *Auto-Patch*, sendo o *Cutter* conectando sua entrada a *composite* dos elementos da pupila e sua saída na *composite* que está plugada o desenho

do olho. O “olho” é conectado ao *Auto-Patch* e a saída deste na porta de entrada do *Cutter*, indicando para o *software* que este desenho do olho servirá como *matte* para a máscara.

É adicionada uma *peg* para cada *drawing*, os *pivot points* são posicionados todos no centro de cada desenho.

Figura 52 – Exemplificação da penúltima etapa de rigging do olho

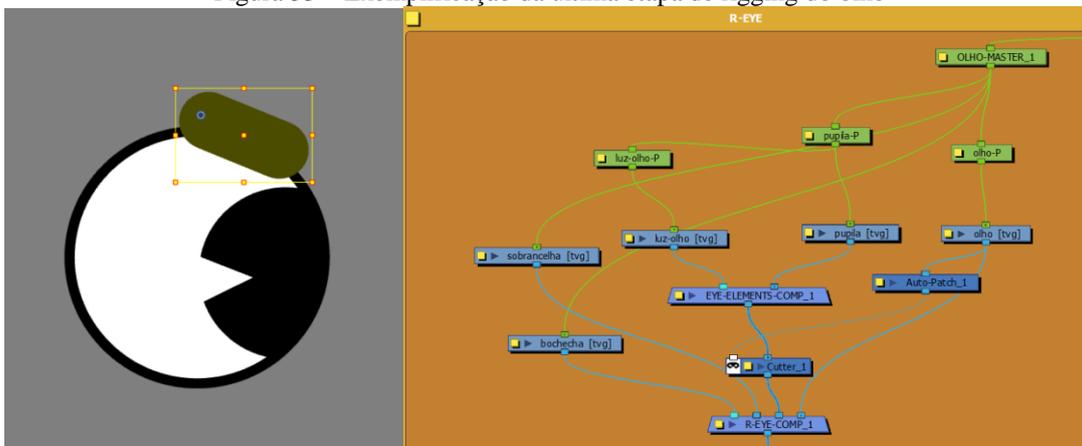


Fonte: A autora

É conectado o desenho “sobrancelha” na *composite* à frente do *Cutter*. O *pivot point* é posicionado no canto esquerdo do desenho. Como já citado na seção 3.2.2.1, foi criado o desenho da bochecha, para posicionar na frente do olho quando for necessário. Conecta-se este *drawing* a frente do desenho dos olhos na entrada da *composite*.

Por fim, adiciona-se uma *peg* “OLHO\_MASTER\_1” que é plugada nas demais *pegs*.

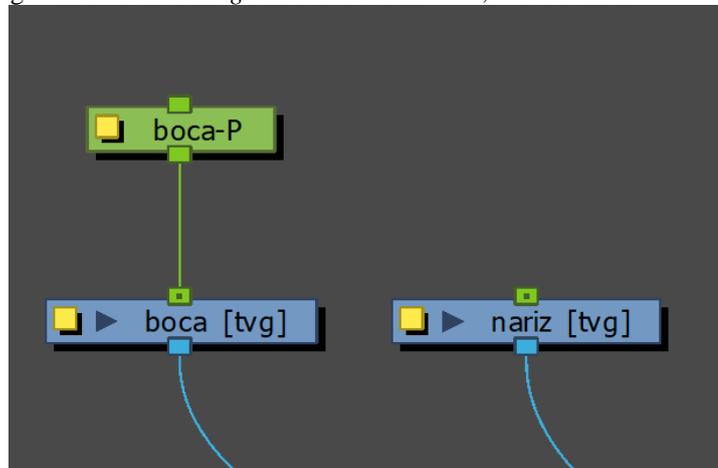
Figura 53 – Exemplificação da última etapa de rigging do olho



Fonte: A autora

São criados os *drawings* chamados de “nariz” e “boca”, ambos são conectados à *composite* que servirá para os elementos da cabeça, criado anteriormente na etapa de estruturação da espinha dorsal do *rig*. O local do *pivot point* do nariz fica na sua base. É adicionada uma *peg* ao *drawing* da boca, pois esta peça sofrerá bastantes modificações na animação, como o *stretch* e *squash*, por exemplo. Por isso surge a necessidade do controlador, para ter certeza de que o desenho estará intacto até o fim da cena. Os desenhos de boca mostrados na seção [3.2.2.2](#) são adicionados à biblioteca a partir deste *drawing*, através da criação de *frames*, podendo se trocados manualmente como já explicado na seção [2.4.4](#).

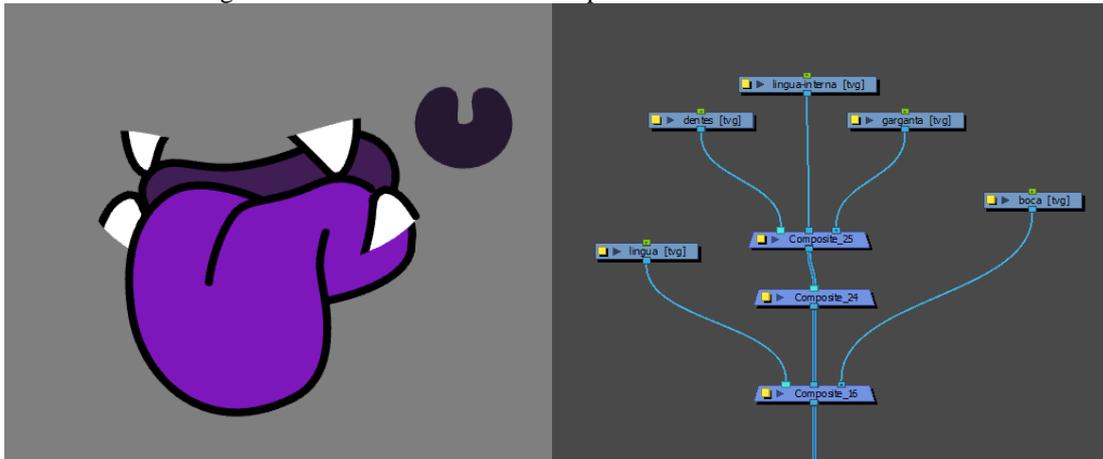
Figura 54 – Os *drawings node* da boca e nariz, na *node view*



Fonte: A autora

Como dito anteriormente na seção [3.2.2.2](#), a boca do personagem Sphix foi dividida em cinco partes. Os elementos de dentro da boca, “garganta”, “língua-interna” e “dentes” foram plugados em uma *composite* separada acima da *composite* que são conectadas as outras duas partes, “boca” e “língua”.

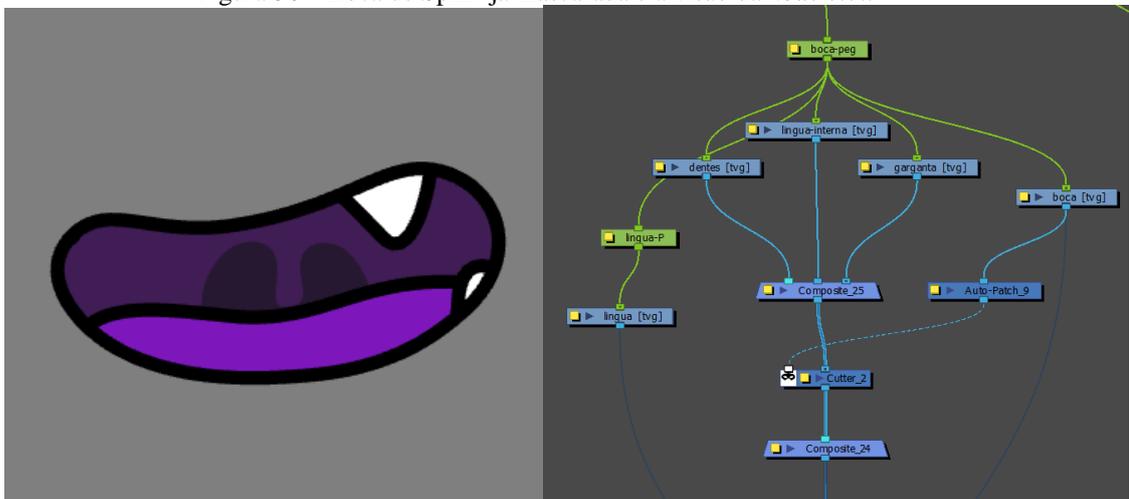
Figura 55 – Elementos da boca do Sphix e a visão na *node view*



Fonte: A autora

Para que os elementos internos da boca sejam mascarados, são inseridos os *nodes Auto-Patch* e *Cutter*, a saída do desenho da boca é plugado na entrada do *Auto-Patch*, e este tem sua saída conectada ao ícone de máscara no lado direito do *node Cutter* que, por sua vez tem sua saída conectada na *composite* abaixo dos elementos. Uma *peg* “boca-p” é conectada aos cinco elementos que representam a boca do personagem.

Figura 56 – Boca de Sphix já mascarada e a visão da *node view*

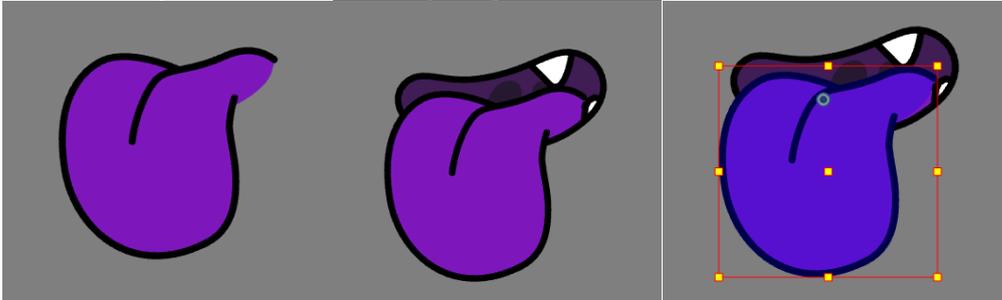


Fonte: A autora

O *drawing* “lingua” é criado para utilizar somente quando o personagem precise ficar com a língua para fora da boca, esta peça é encaixada de forma que o desenho se conecte com o desenho “língua-interna” e fique com a impressão de ser uma só. O *pivot point* da língua é posicionado na

parte superior do desenho para que seja possível transformar a língua de modo que o *squash* e *stretch* da peça seja feito na etapa de animação.

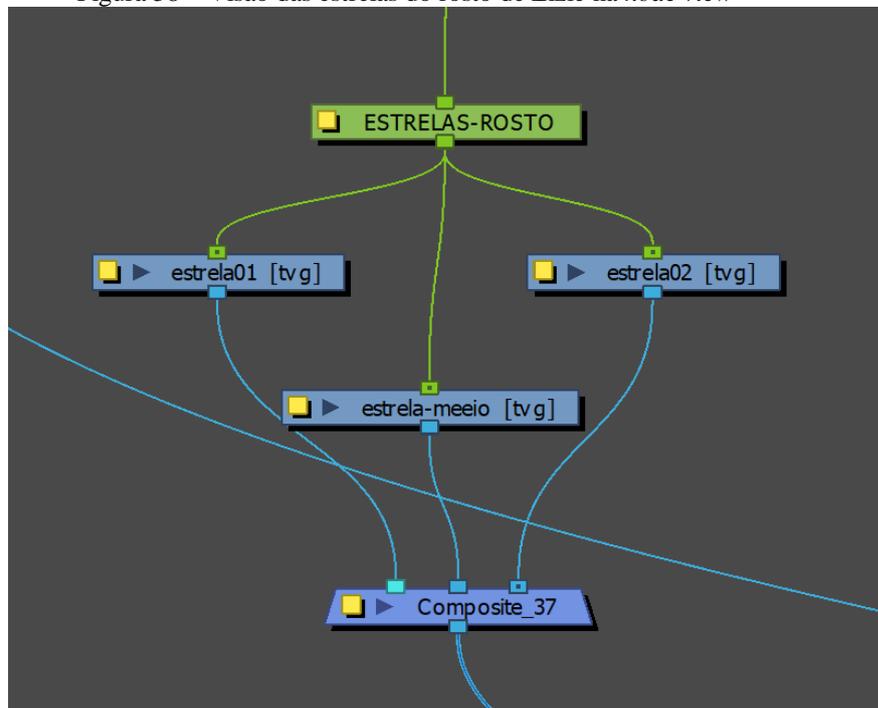
Figura 57 – Desenho da língua separada e funcionando no encaixe da boca



Fonte: A autora

Na personagem Lizie, é criada uma nova *composite* para agrupar os *drawings* das estrelas azuis que fazem parte do *design* de Lizie, e que se localizam no rosto. As três foram feitas na *ColourArt* dos *drawings*, nomeadas respectivamente de “estrela01”, “estrela-meio” e “estrela-02”. Os *pivot points* das estrelas são posicionados no centro de cada uma, por fim é inserida uma *peg* chamada “ESTRELAS-ROSTO”, para que seja movido em grupo caso necessário.

Figura 58 – Visão das estrelas do rosto de Lizie na *node view*



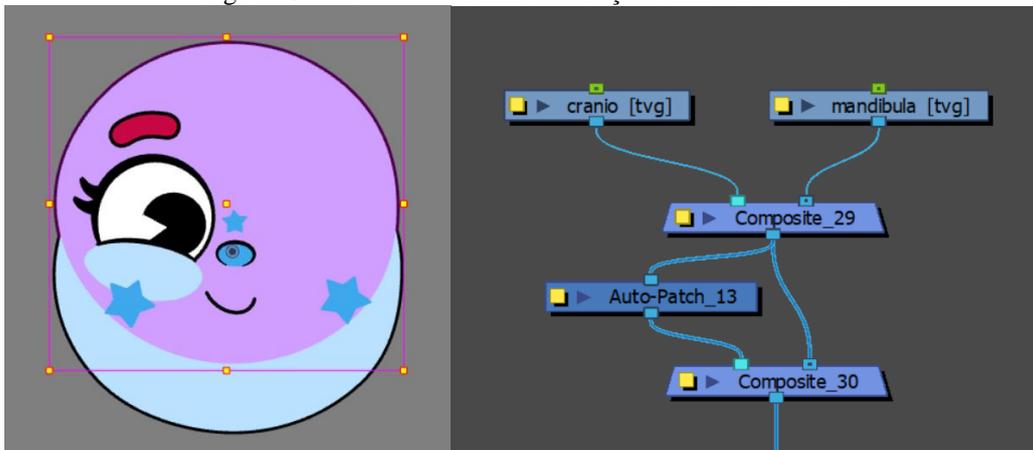
Fonte: A autora

### 3.2.3.2 CABEÇA

O desenho da cabeça foi dividido em duas partes: cabeça e mandíbula, caso as animadoras preferissem animar a mandíbula quando a boca da personagem se move, deixando desta forma mais realista, visto que, é através do movimento de rotação da mandíbula que a boca é aberta.

É adicionado um *Auto-Patch* entre a *composite* que possui os desenhos que formam a cabeça.

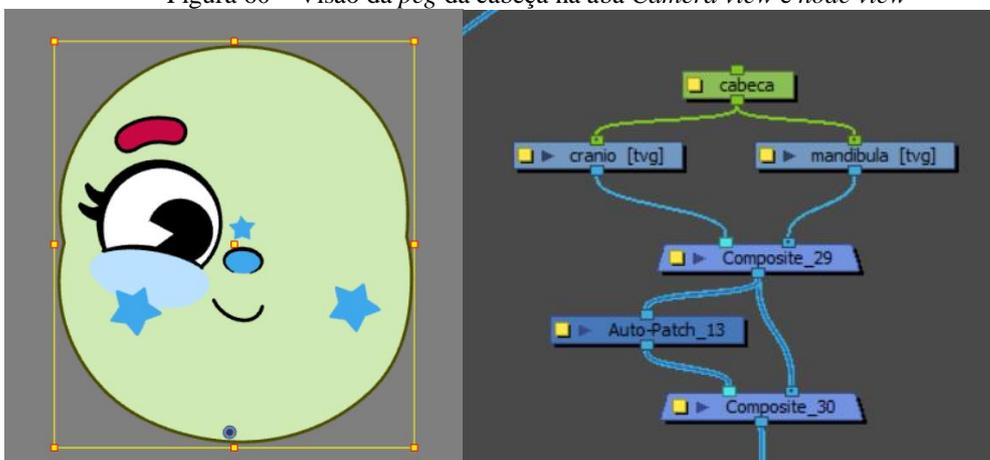
Figura 59 – Visão do desenho da cabeça de Lizie na node view



Fonte: A autora

É criada uma *peg* nomeada “cabeça” para que seja possível as duas partes se moverem em conjunto. O *pivot point* é posicionado na base, um pouco fora do centro. Para deixar o movimento mais real, já que o pescoço é conectado na parte de trás da cabeça, e não na garganta.

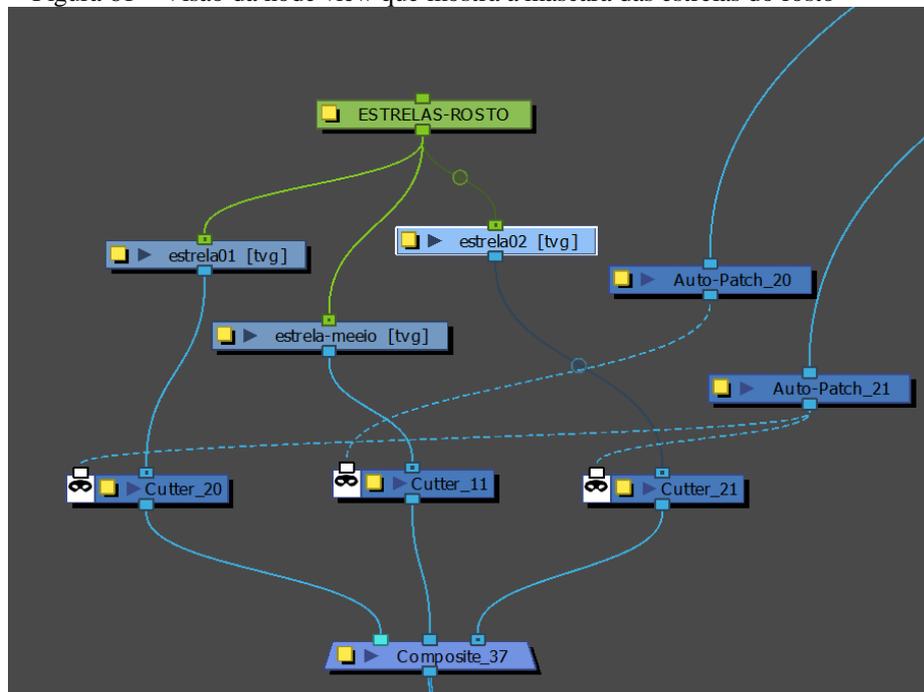
Figura 60 – Visão da *peg* da cabeça na aba *Camera view* e *node view*



Fonte: A autora

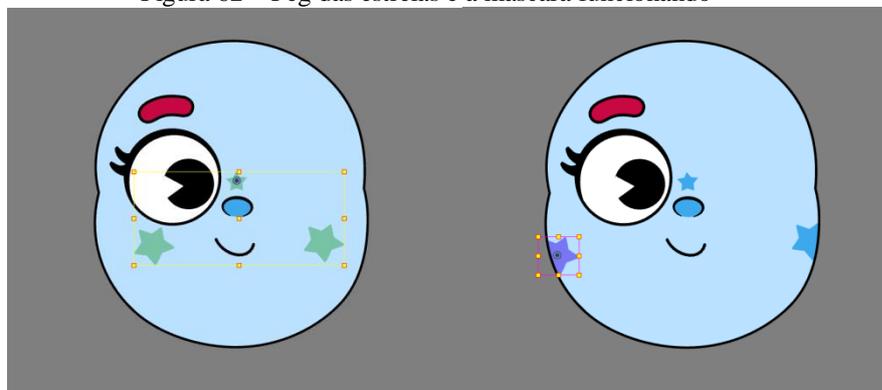
Para as estrelas no rosto da personagem não ultrapassarem o desenho da cabeça, é feito a mesma máscara que fora utilizada para mascarar a pupila dentro do olho. Abaixo de cada uma foi adicionado um *Cutter* com sua saída conectada na *composite*. Dois novos *Auto-Patch* são inseridos no *node view*, um deles tem sua entrada conectada pelo desenho da cabeça, pois a “estrela-meio” se localiza nela. Em contra ponto, as outras estrelas se localizam por cima do desenho da “mandíbula”, é adicionada uma conexão no outro *Auto-Patch*. Em seguida estes *Auto-Patch nodes* são conectados na entrada de *matte* do *node Cutter*, ativando a máscara.

Figura 61 – Visão da node view que mostra a máscara das estrelas do rosto



Fonte: A autora

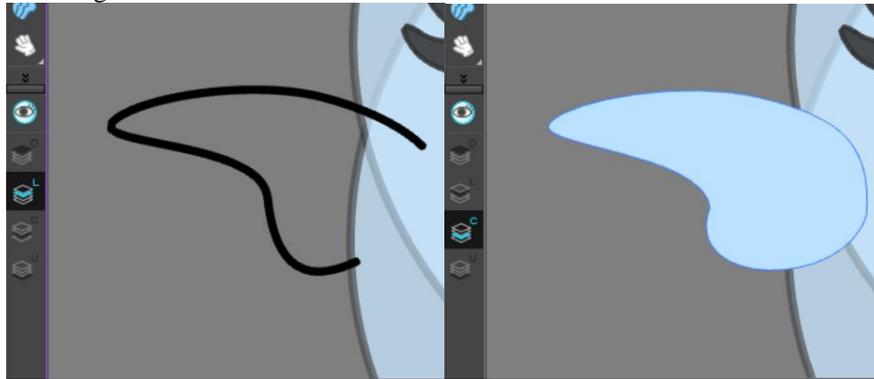
Figura 62 – Peg das estrelas e a máscara funcionando



Fonte: A autora

A orelha tem suas partes divididas através das camadas de *Artlayers* do *drawing* para que seja possível por meio dos *nodes Auto-Patch* e *Cutter* a mescla da orelha com a cabeça. Separa-se as linhas da parte da frente da orelha na camada de *LineArt*, e a de preenchimento na *ColourArt*.

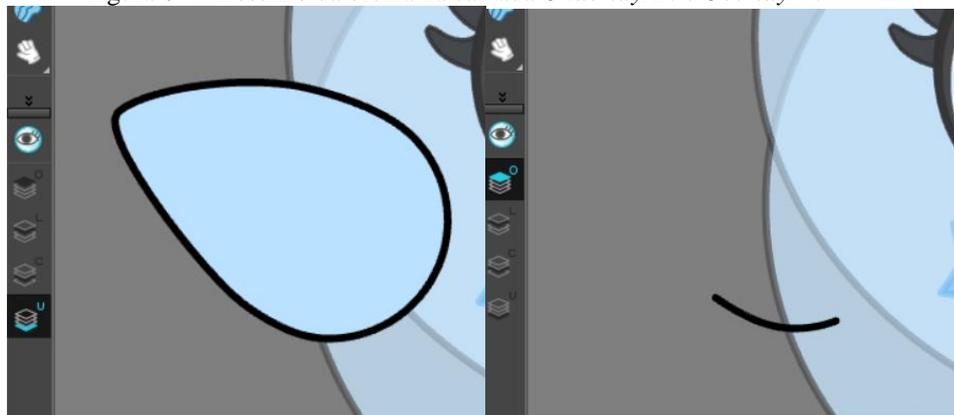
Figura 63 – Desenhos da orelha na camada *LineArt* e *ColourArt*



Fonte: A autora

O desenho da parte de trás da orelha é feito inteiro na *UnderlayArt*. É copiada para a camada de *OverlayArt*, um pedaço da linha do desenho que ultrapassa um pouco da cabeça. Isto é para efeito de tridimensionalidade da orelha, quando a personagem estiver em posição de  $\frac{3}{4}$ ,

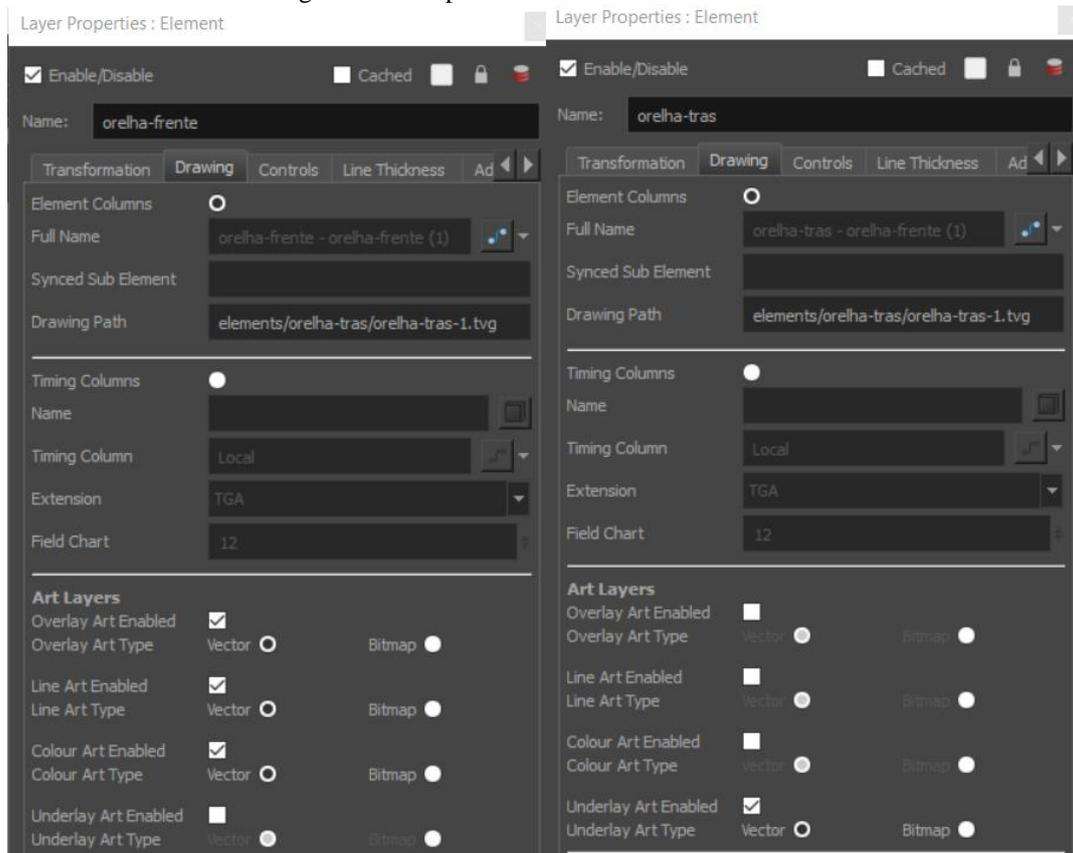
Figura 64 – Desenho da orelha na camada *UnderlayArt* e *OverlayArt*



Fonte: A autora

O desenho da orelha é duplicado, sendo um nomeado como “orelha-frente” e o outro como “orelha-tras”. Nas propriedades da *layer* “orelha-frente”, na aba “*drawing*” é desabilitada a opção “*UnderlayArt*”, deixando esta *Artlayer* invisível. E nas propriedades da *layer* “orelha-tras” a única *Artlayer* habilitada é a *OverlayArt*.

Figura 65 – Propriedades dos desenhos da orelha

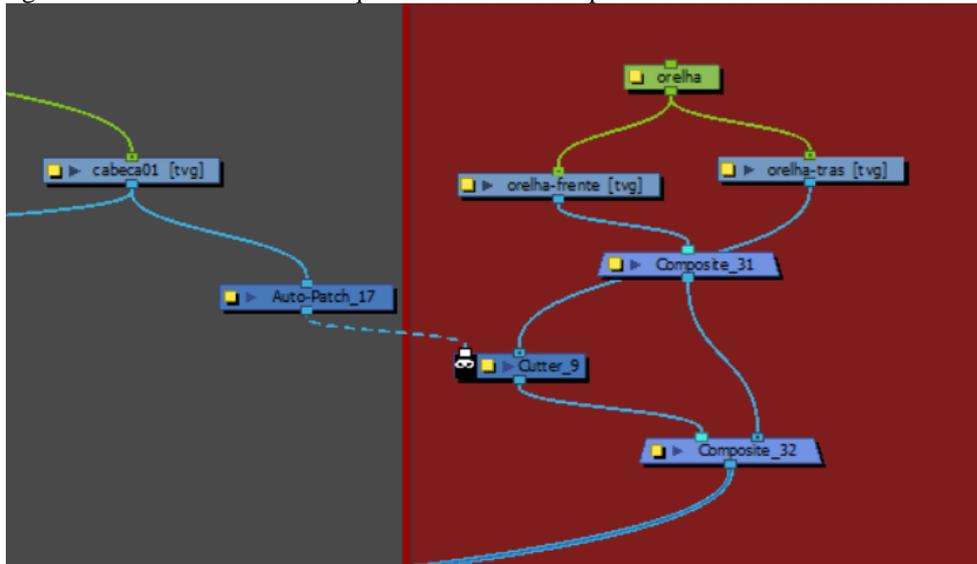


Fonte: A autora

O *Auto-Patch* é importado para *nodeview*, junto do *Cutter*. A “orelha-trás” é plugada na entrada no *Cutter*, que é conectado a *composite* abaixo das que se encontram os dois elementos da orelha. É puxada uma saída da *layer* da cabeça e conectada a entrada do *Auto-Patch*, que por sua vez é plugado na saída de *matte* do *Cutter node*. Indicando que a cabeça servirá de *matte* para que haja o corte da orelha.

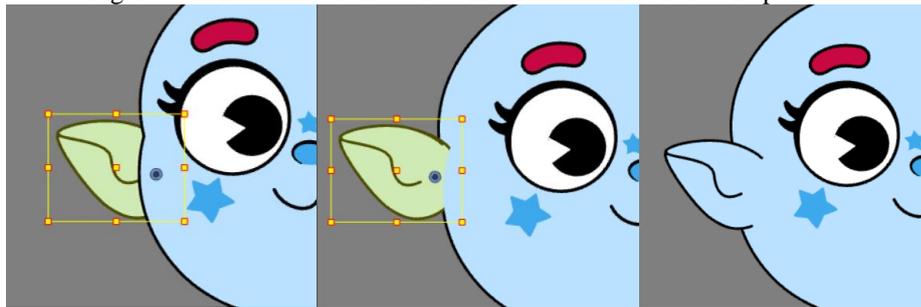
Em seguida é adicionada uma *peg* chamada “orelha” para que seja possível mover as duas peças ao mesmo tempo.

Figura 66 – Visão da node view que mostra a conexão para o efeito de mescla da orelha



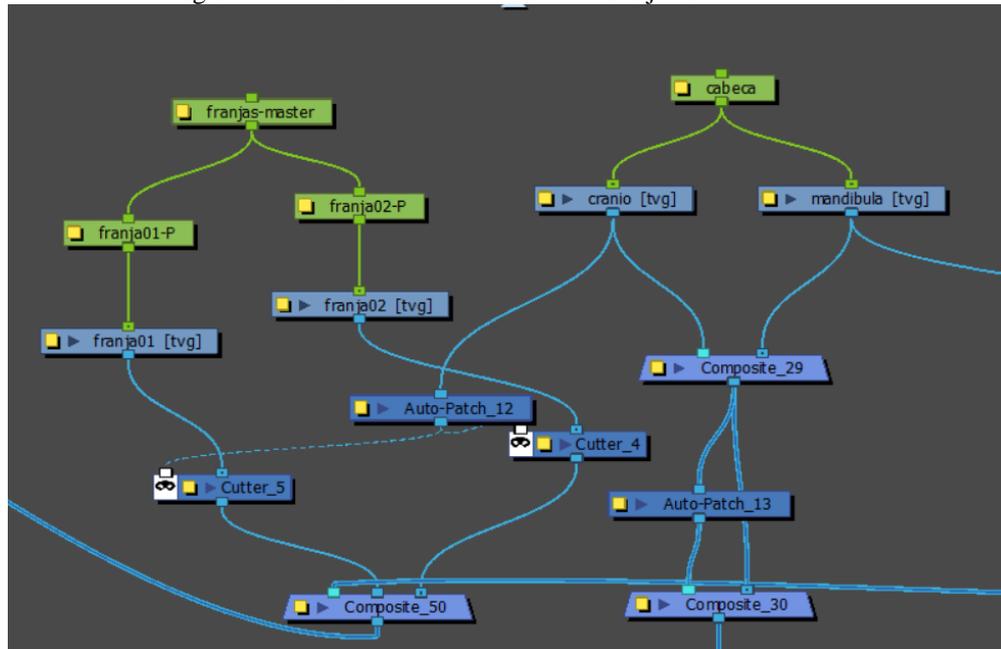
Fonte: A autora

Figura 67 – Efeito da orelha sendo mostrado e funcionando na pose ¾



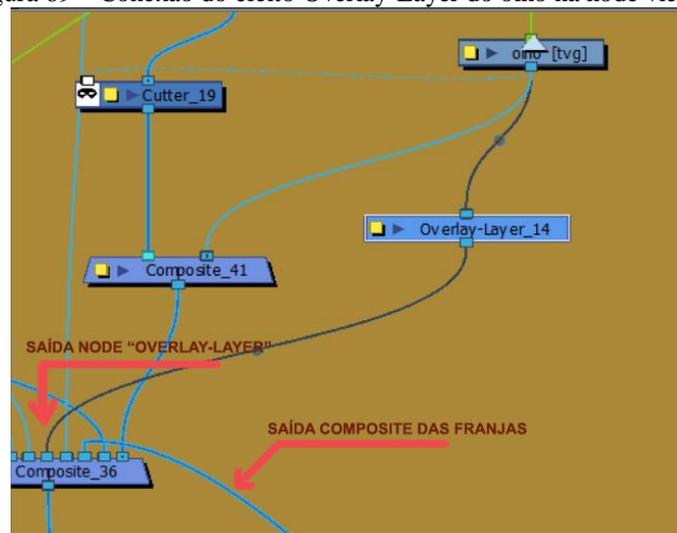
Fonte: A autora

O cabelo de Lizie é dividido em três partes ao todo, para que possa ser animado em mechas individualmente. Para o cabelo na parte da frente da cabeça ficou dividido como: “franja01” a franja da esquerda e “franja02” a franja da direita, que foram mascaradas através dos *nodes Cutter* e *Auto-Patch* usando o desenho do crânio como *matte* para que o desenho não ultrapassasse a cabeça. O *pivot point* é adicionado no topo da franja, para que possa funcionar a deformação de *squash* e *stretch*, caso for necessário. Logo em seguida, adiciona-se o controlador “franjas-master” para caso haja a necessidade de move-las em conjunto.

Figura 68 – Conexões dos desenhos das franjas na *node view*

Fonte: A autora

Para que a linha de contorno do olho da personagem fique por cima da franja, característica de seu *design*, é adicionada à *node view* o *node Overlay-Layer*, que quando plugada à saída do *drawing* “olho” em sua entrada, é filtrado o desenho do contorno do olho que se encontra na *OverlayArt* da *layer*. A saída do *Overlay-Layer node* é plugado na *composite* trazendo a informação da *layer* “olho” para frente da *composite* que se encontram os desenhos das franjas.

Figura 69 – Conexão do efeito *Overlay-Layer* do olho na *node view*

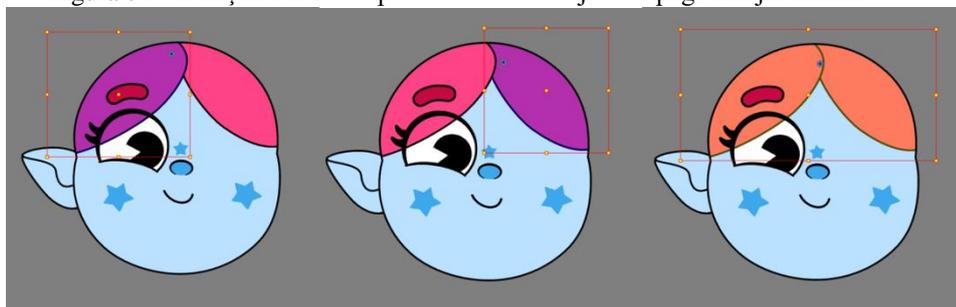
Fonte: A autora

Figura 70 – Cabeça de Lizie antes e depois do efeito OverLay-Layer se adicionado



Fonte: A autora

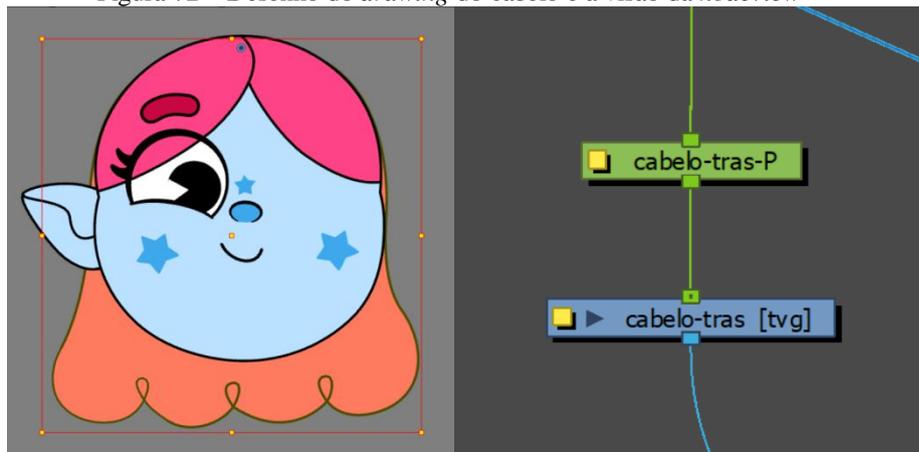
Figura 71 – Posição dos Pivot points de cada franja e da peg “franjas-master”



Fonte: A autora

A outra parte ficou posicionada para trás da cabeça.

Figura 72 – Desenho do *drawing* do cabelo e a visão da *nodeview*

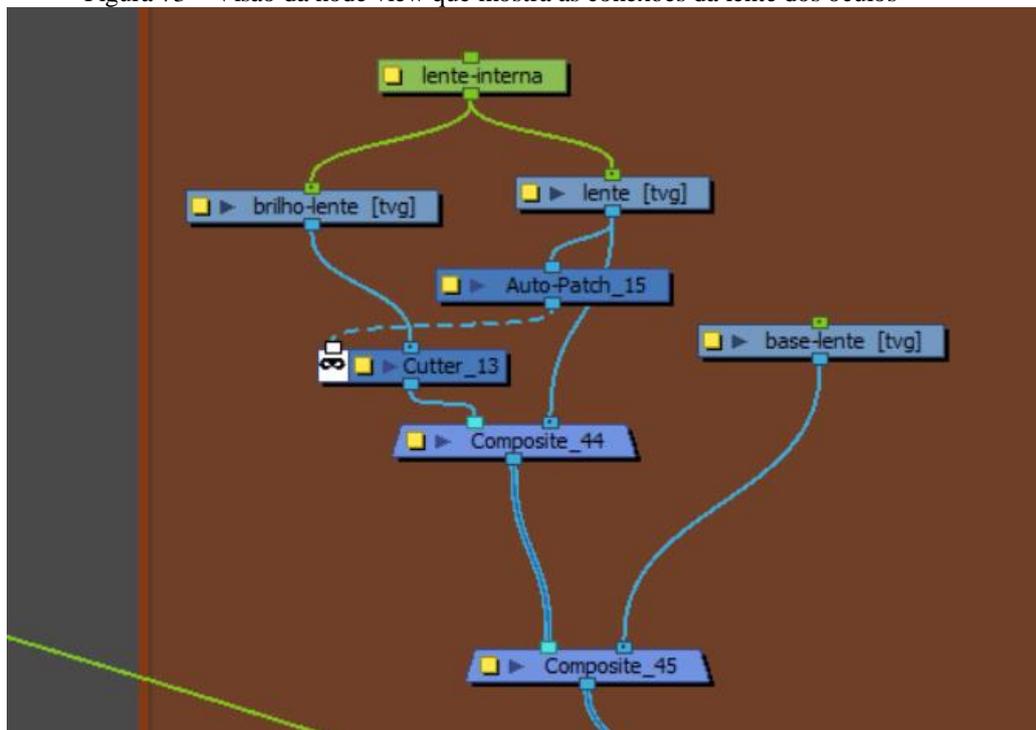


Fonte: A autora

Também foi necessário ser feito o *rigging* do acessório característico da protagonista, os óculos. Dividiu-se assim, as partes: “base-lente”, que fora plugada em uma *composite* abaixo da

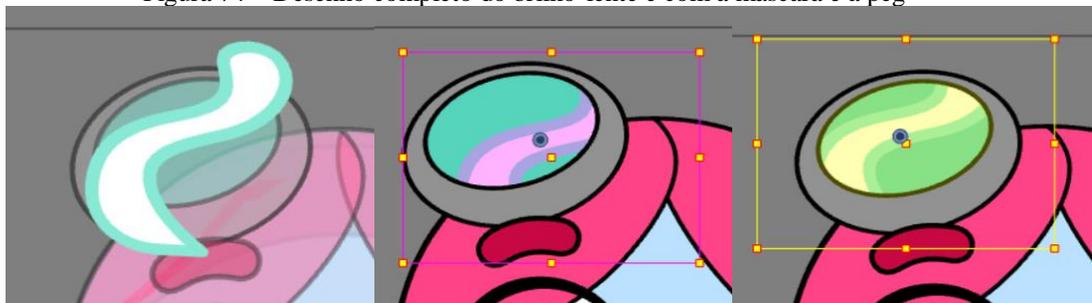
*composite* criada para plugar as partes “lente” e “brilho-lente”. Estes desenhos formam a peça que contém a lente e a armação da lente. O desenho “lente” foi utilizado como *matte* para mascarar o “brilho-lente” que serve para ser o reflexo da luz, portanto, plugou-se a lente na entrada do node *Auto-Patch* que por sua vez, fora plugado na entrada do *Cutter*. A peça do brilho possui um desenho maior para que pudesse ser animado dentro da máscara. A “lente” e “brilho-lente” foram unidos em uma *peg* nomeada “lente-interna”.

Figura 73 – Visão da node view que mostra as conexões da lente dos óculos



Fonte: A autora

Figura 74 – Desenho completo do brilho-lente e com a máscara e a peg

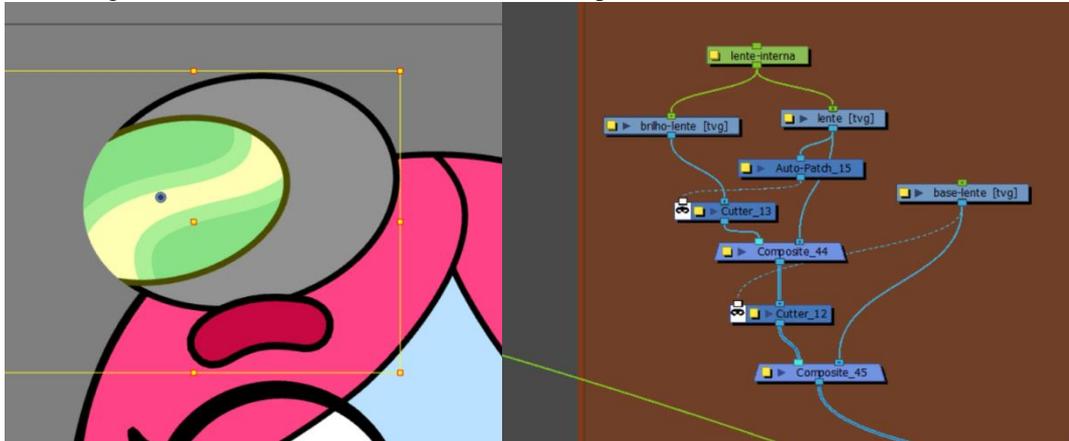


Fonte: A autora

Para fazer com que a *peg* “lente-interna” não ultrapasse a armação da lente, caso precise reposicioná-la na hora da animação, foi importado outro *Cutter* que, se conecta entre as duas

*composites* dos elementos dos óculos. O *drawing* “base-lente” é plugado na entrada de *matte* do *Cutter* para que a máscara funcione.

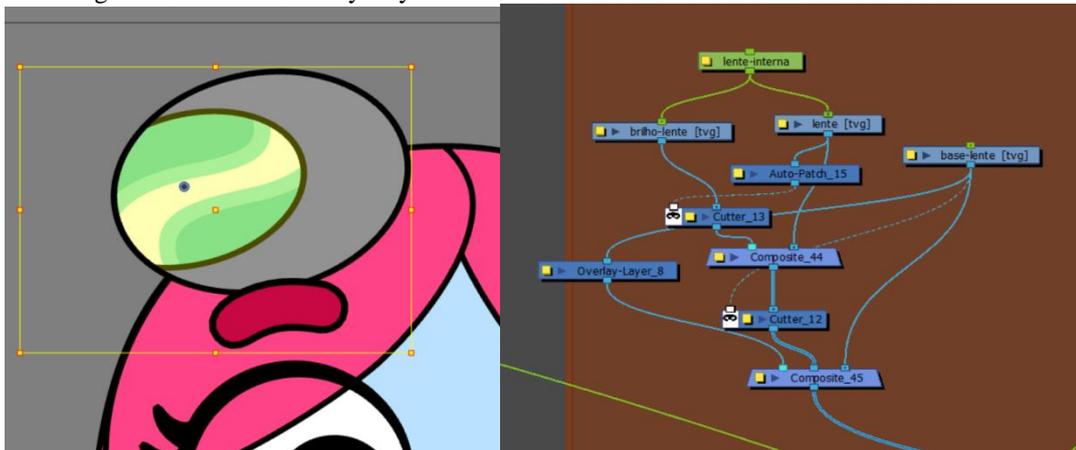
Figura 75 – Desenho da lente sendo mascarada pela base e vista da *node view*



Fonte: A autora

Por questões de acabamento é importado à *node view* o *Overlay-Layer*, como já dito antes, este *node* é responsável por filtrar o desenho que se encontra na *OverlayArt* do *drawing* em que é plugado. Portanto, para que quando a *peg* “lente-interna” seja trazida para o canto da armação e a lente não fique por cima das linhas, é feita a separação das linhas da “base-lente” sendo inseridas na *OverlayArt* e o preenchimento na *ColourArt*.

Figura 76 – Efeito *Overlay-Layer* funcionando no desenho e a visão da *node view*

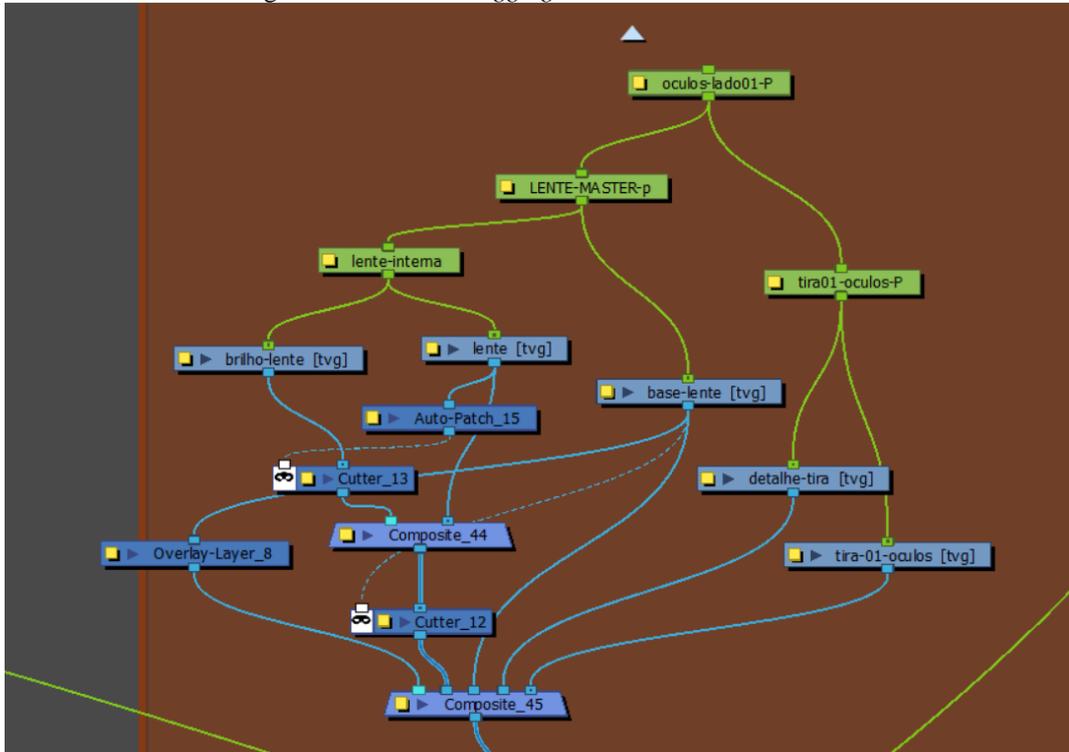


Fonte: A autora

A tira dos óculos ficou dividida como: “detalhe-tira”, é a peça que conecta a base da lente com a “tira-01-oculos” que vai até a orelha. Ambas são conectadas a *peg* intitulada “LENTE-MASTER-P” que também agrupa à *peg* “lente-interna” criada anteriormente. Por fim, é criada a *peg*

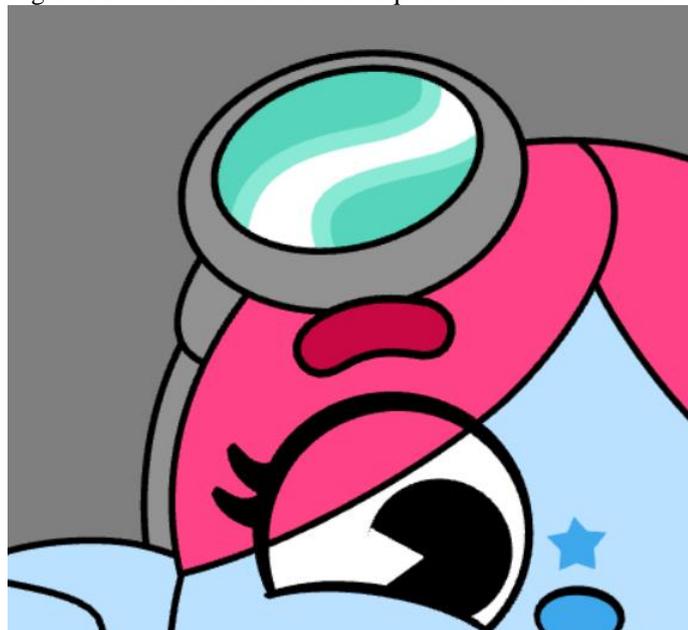
“oculos-lado01-P” que agrupa todas as *pegs* das partes dos óculos feitas anteriormente e é posicionado o *pivot point*, como de praxe.

Figura 77 – Visão do *rigging* dos óculos na *node view*



Fonte: A autora

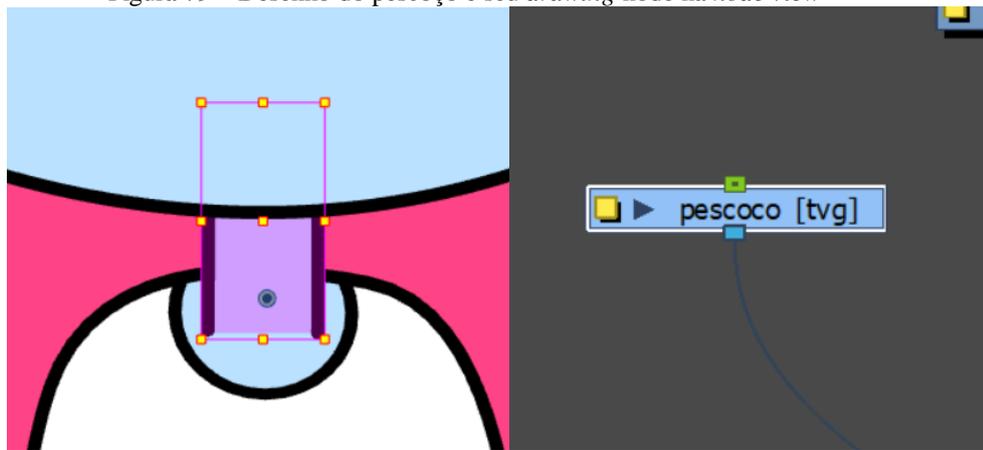
Figura 78 – Versão final do lado esquerdo dos óculos da Lizie



Fonte: A autora

Terminando o *rigging* da cabeça, posiciona-se o *pivot point* do pescoço na sua base, que se conecta com a *layer* “gola” com a mesma posição de *pivot point* do pescoço, assim indo de encontro ao torso.

Figura 79 – Desenho do pescoço e seu *drawing node* na *node view*

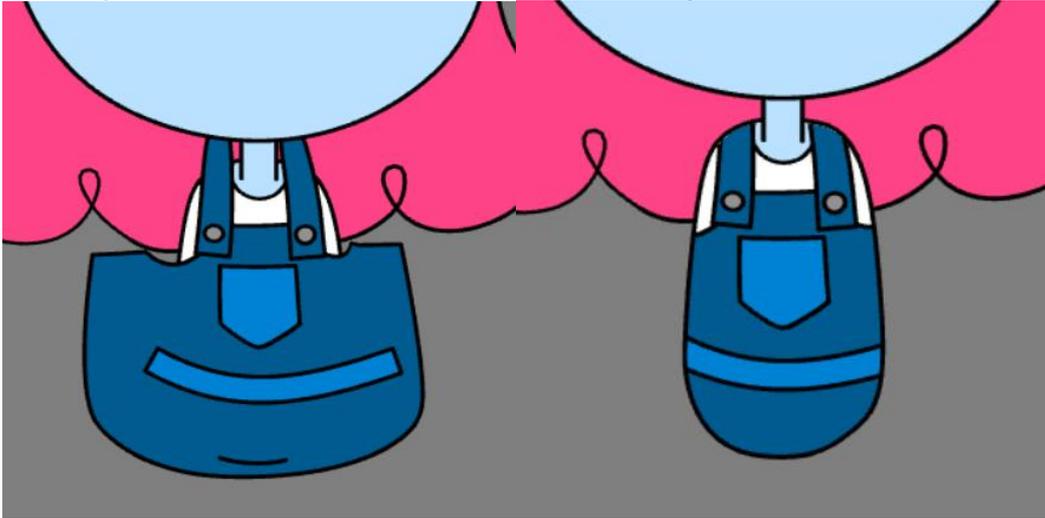


Fonte: A autora

### 3.2.3.3 TORSO

Para o torso, foi feito o desenho com a forma base “tronco” que serviu como *matte* para mascarar a peça “gola”, o desenho da “jardineira” e seus detalhes: “cinto”, “bolso”, “tira” e “tira01”. Estes desenhos que compõe a jardineira foram feitos maiores para que pudessem ser animados conforme a personagem gira em seu eixo, dando a sensação de tridimensionalidade.

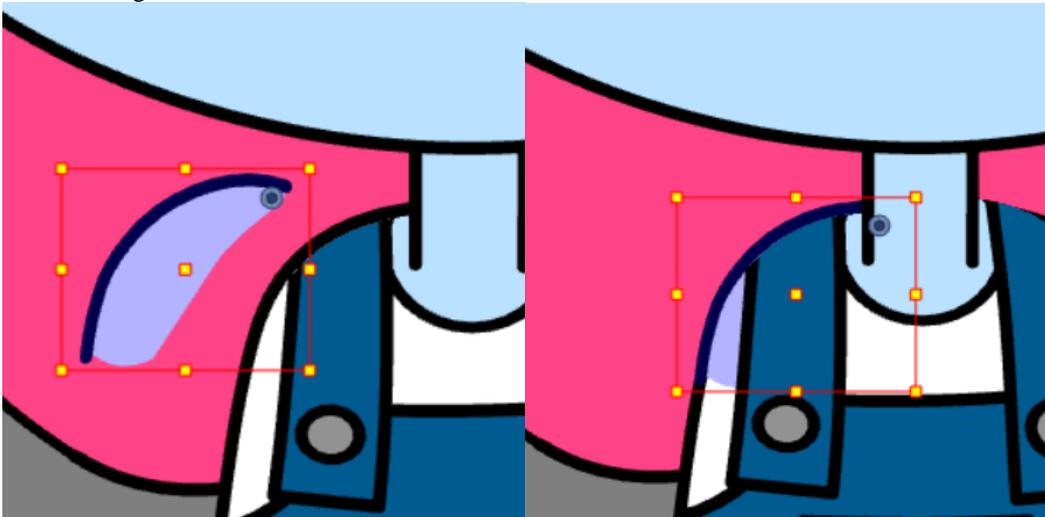
Figura 80 – Vista do desenhos inteiros dos elementos da jardineira e eles mascarados



Fonte: A autora

Além das peças da jardineira, também foi mascarada o desenho do “ombro”, peça esta que ajuda a acentuar a expressão corporal da personagem.

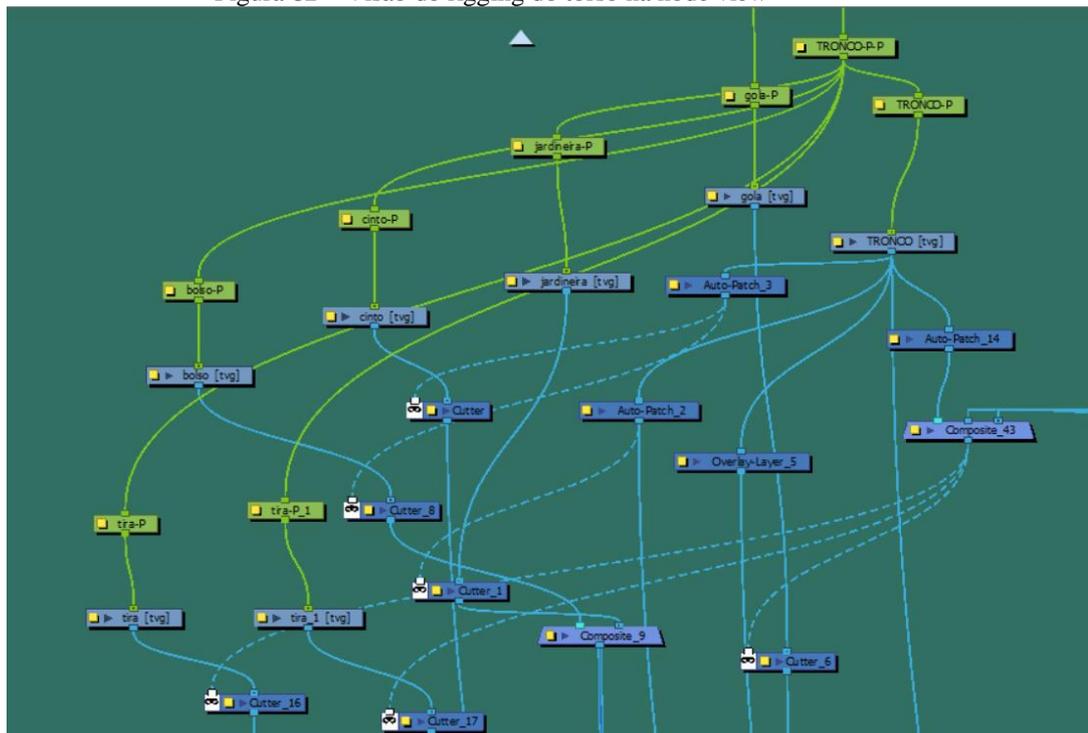
Figura 81 – Desenho do ombro solto e desenho do ombro mascarado no tronco



Fonte: A autora

Para cada desenho desta etapa foi gerado uma *peg* que é conectada a uma *peg master* intitulada “TRONCO-P”.

Figura 82 – Visão do rigging do torso na node view

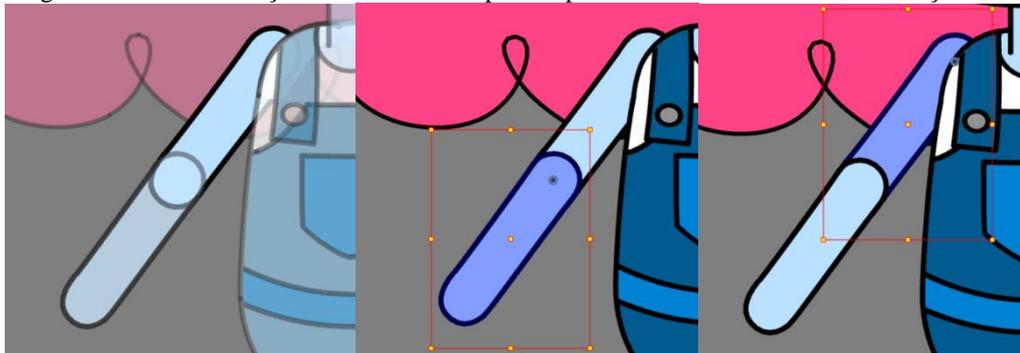


Fonte: A autora

### 3.2.3.4 ARTICULAÇÕES E EXTREMIDADES

Para que as articulações do braço funcionem corretamente, primeiro é criada uma nova *composite* onde serão conectados os desenhos do membro. É fundamental que o desenho do “braco” e “antebraco” se encaixe formando um círculo e bem ao centro é posicionado o *pivot point* do “antebraco” para que assim quando houver a rotação do antebraço, ele não se desloque do desenho do braço. E o *pivot point* do braço é posicionado no topo do desenho, próximo ao ombro.

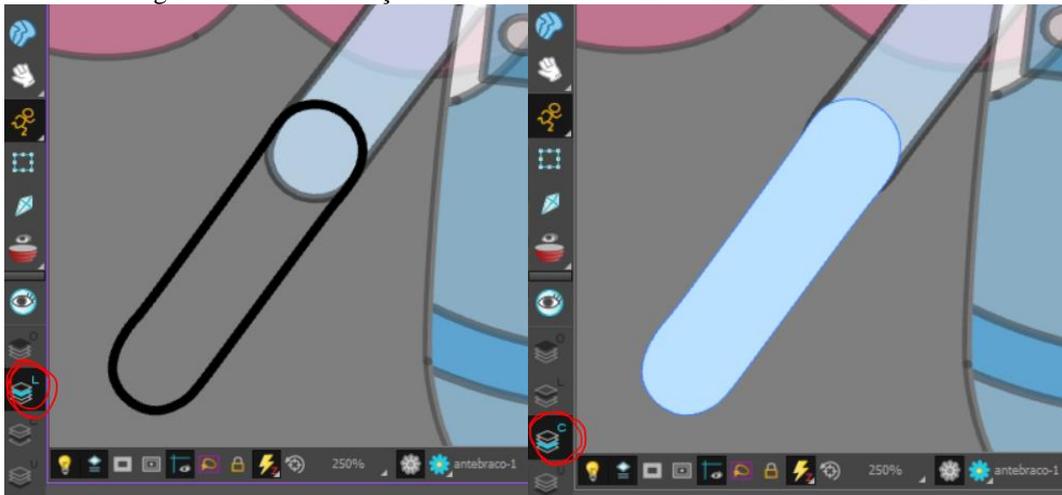
Figura 83 – Demonstração de onde o Pivot point é posicionado nos elementos do braço



Fonte: A autora

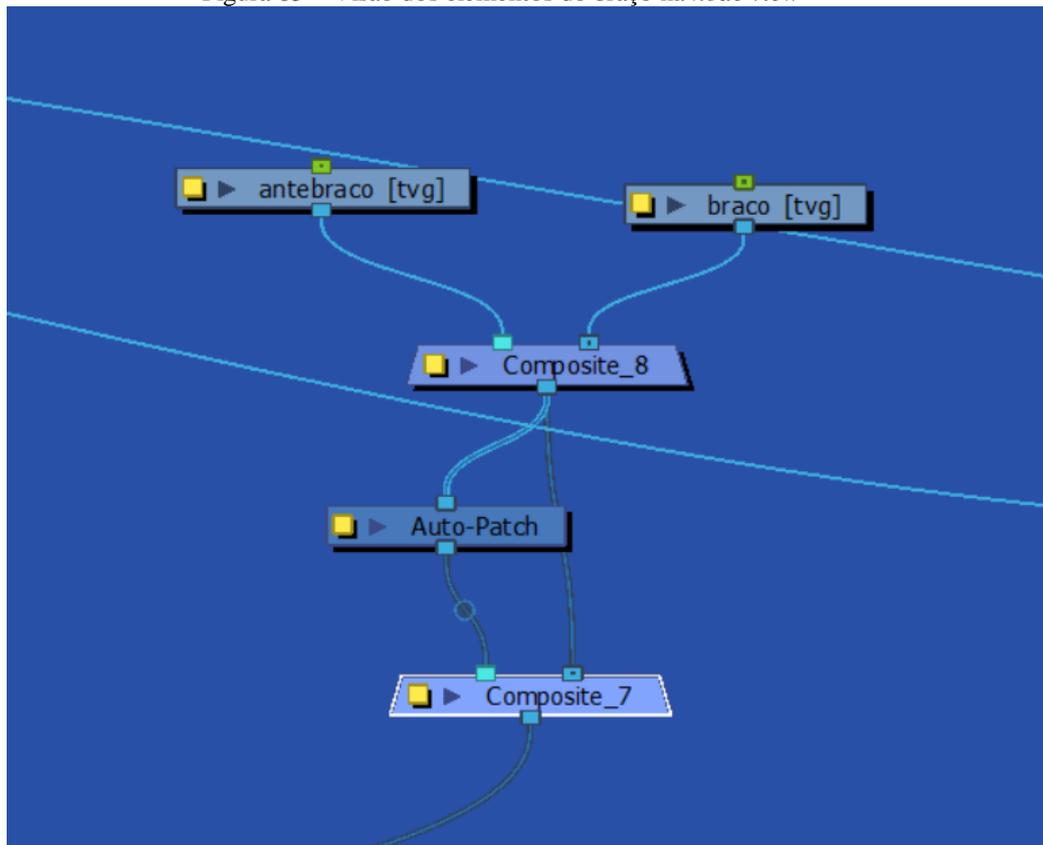
Para que não haja a linha do antebraço por cima, mostrando a divisão das partes, é necessário separá-la do preenchimento, portanto coloca-se as linhas do “braco” e “antebraco” na camada de *LineArt* e o preenchimento na camada *ColourArt*.

Figura 84 – Demonstração dos desenhos nas camadas LineArt e Colour Art



Fonte: A autora

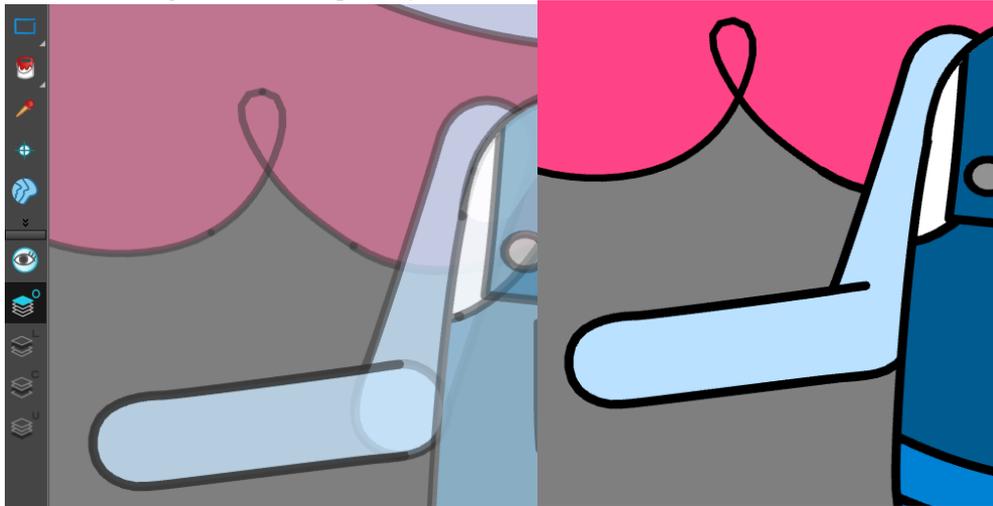
Feito isso, é importado à *node view* o *Auto-Patch*, este filtrará as camadas dos desenhos colocando o preenchimento para cima, deixando as linhas para baixo fazendo com que desapareça. Plugando a saída da *composite* antiga na entrada do *Auto-Patch*, e plugando a saída do *Auto-Patch* na frente da saída da *composite* que fora criada para os desenhos do braço.

Figura 85 – Visão dos elementos do braço na *node view*

Fonte: A autora

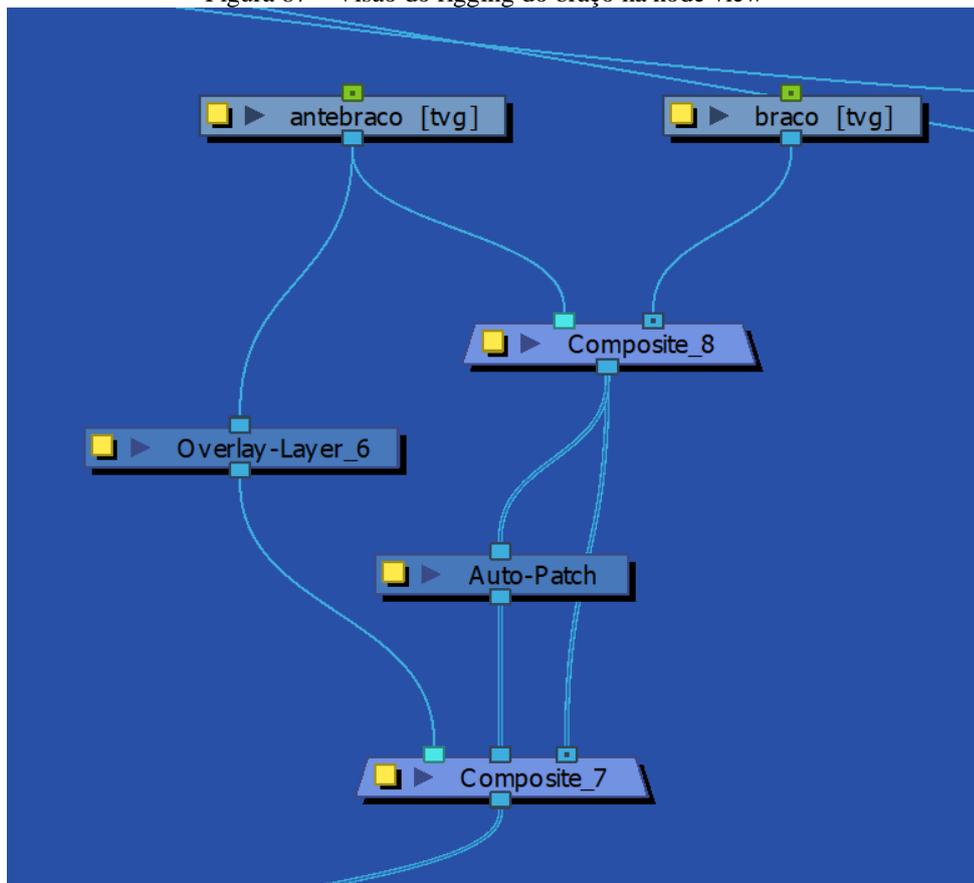
Para dar um efeito de dobra na parte interna do cotovelo, como citado na seção [2.4.9](#), é copiada uma parte da linha que atravessa um pouco o braço, e colado na camada *OverlayArt* do desenho do antebraço. É importado à *node view* o *node Overlay-Layer*, sendo este tendo a sua entrada plugada pelo desenho “antebraço” e sua saída conectada a *composite*, à frente do *Auto-Patch*.

Figura 86 – Exemplificação do efeito de “dobra no cotovelo”



Fonte: A autora

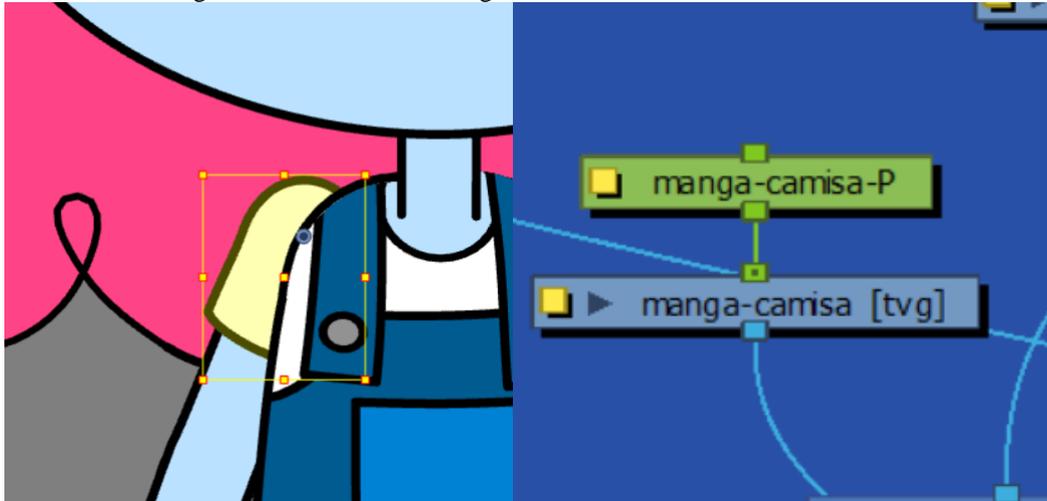
Figura 87 – Visão do rigging do braço na node view



Fonte: A autora

É criado um desenho individual chamado “manga-camisinha” para que fique por cima de boa parte do braço, e para que as animadoras tenham a liberdade de animarem a manga separadamente. Cria-se uma *peg* para a manga, com o *pivot point* na mesma posição do *pivot* do braço.

Figura 88 – Desenho da manga da camiseta e a visão da *node view*

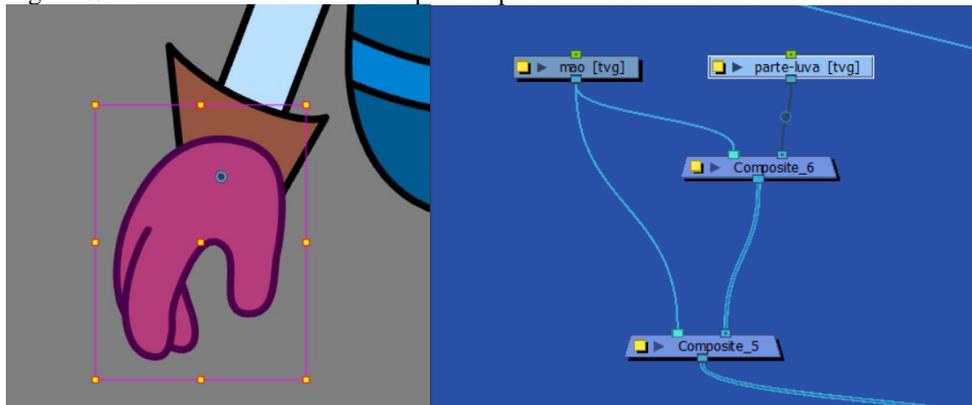


Fonte: A autora

A personagem principal Lizie, utiliza-se de luvas. Portanto para que se possa fazer a conexão da mão ao braço cria-se uma nova *composite*, onde são plugados os desenhos “mao” e “parte-luva”, destes separa-se as linhas do preenchimento, colocando nas camadas de *LineArt* e *ColourArt* de seus respectivos *drawings*.

Figura x -

Figura 89 – Desenhos dos elementos que compõe a mão de Lizie e vista da *node view*

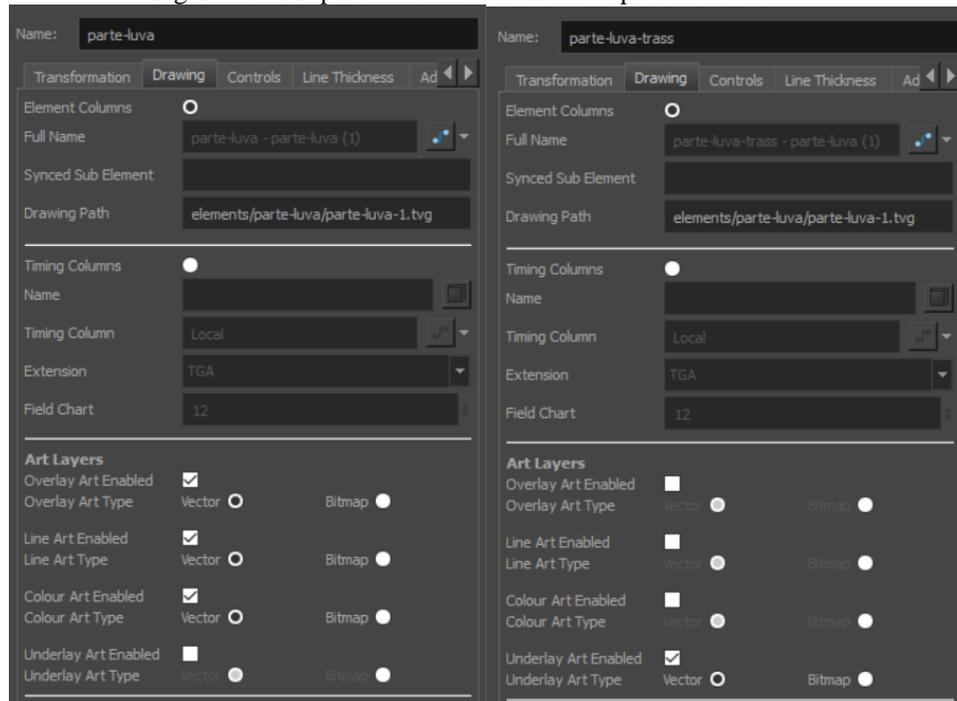


Fonte: A autora

Duplica-se “parte-luva” como “parte-luva-tras” e o desenho é copiado das camadas *LineArt* e *ColourArt*, e colado na camada *UnderlayArt* sendo modificado para se encaixar na perspectiva da

parte de trás da luva. Nas propriedades do *drawing* “parte-luva” é desmarcada a opção de “*UnderlayArt*” para que tudo que esteja nesta camada fique invisível. É marcado apenas a opção “*UnderlayArt*” nas propriedades do *drawing* “parte-luva-tras” para que assim apenas o desenho que está nesta camada seja mostrada.

Figura 90 – Propriedades dos desenhos das partes da luva

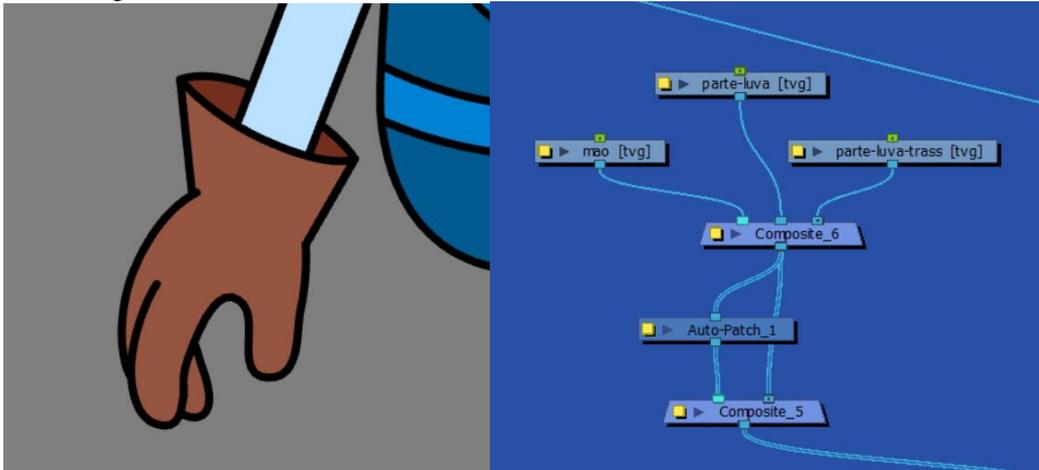


Fonte: A autora

É trazido o node *Auto-Patch* para a *node view*, sendo plugada entre as duas *composites*. A *composite* que contém os *drawings* da mão e da luva é plugado na entrada do *Auto-Patch* e a saída do *AutoPatch* é plugada em frente a esta *composite*.

Os *pivot points* de todos os desenhos desta etapa da mão são posicionados na base da mão.

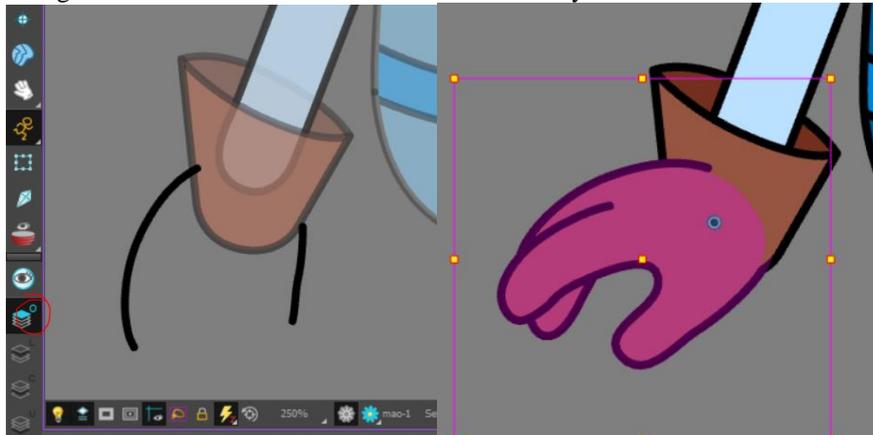
Figura 91 – Desenho final da mão com o elemento da luva e a vista da node view



Fonte: A autora

Para dar a impressão de tridimensionalidade para a luva quando a mão é girada, é adicionada as linhas da mão mais profundas do desenho e que foram mescladas no *Auto-Patch*, na camada *OverlayArt* do *drawing* da mão. Para que isto apareça no desenho foi necessário adicionar o node *Overlay-Layer* na *node view*, plugando na saída do desenho da mão e conectando a saída da *Overlay-Layer* na frente do *Auto-Patch*, assim o desenho adicionado na camada *OverlayArt* é mostrado na frente da mescla entre a mão e a parte da luva.

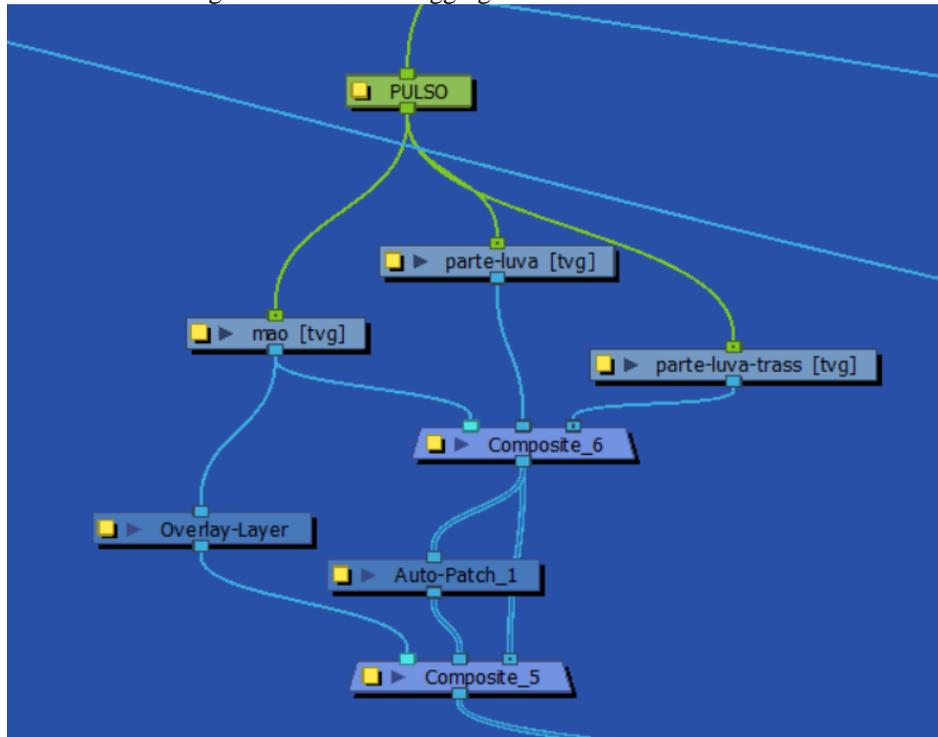
Figura 92 – Linhas adicionadas na camada *OverlayArt* e a luva funcionando



Fonte: A autora

Cria-se uma *peg* “PULSO” onde é conectada aos *drawings* e possui a mesma posição de *pivot point*: na base da mão.

Figura 93 – Vista do rigging da mão na node view

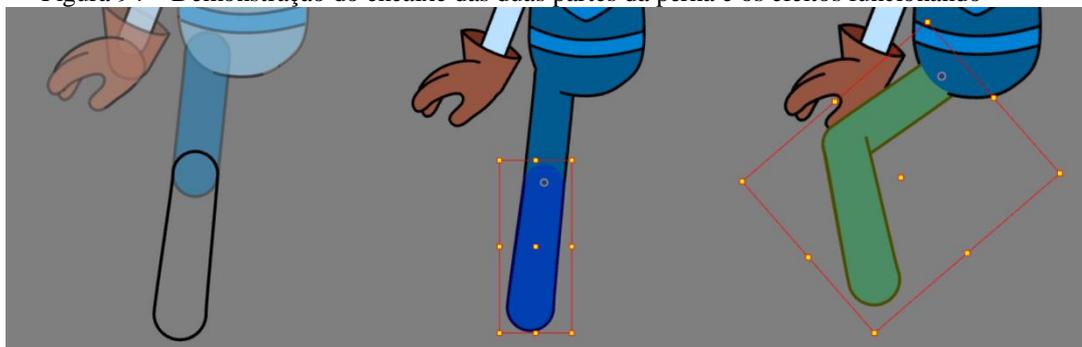


Fonte: A autora

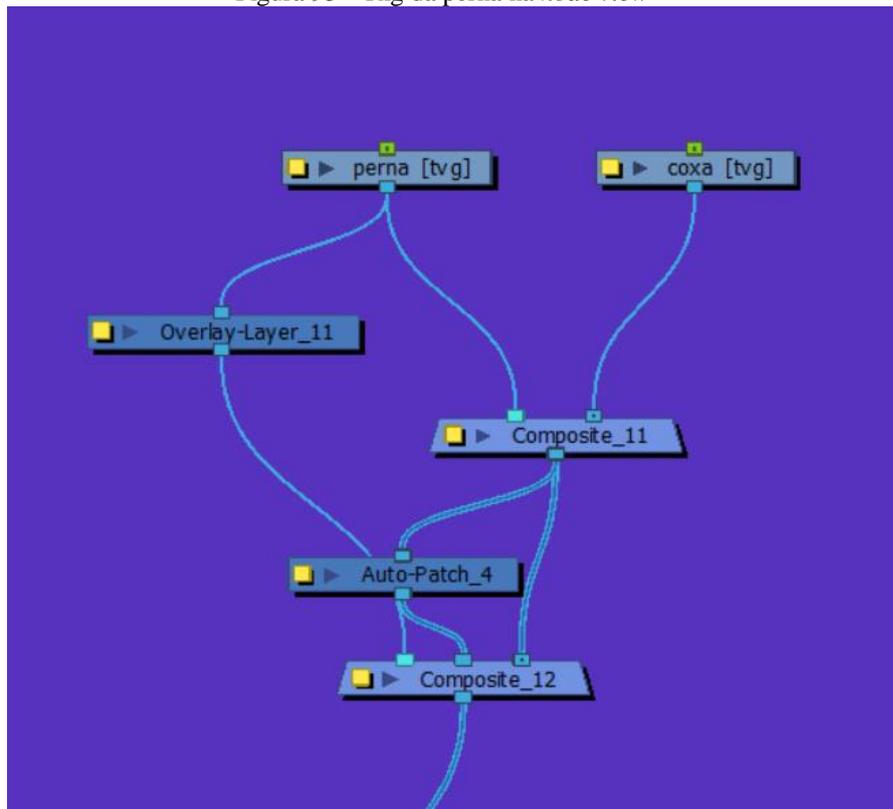
A etapa da perna possui exatamente o mesmo processo de *rigging* do braço. Fora dividido em duas partes, “coxa” e “perna” separado os desenhos em *OverlayArt*, *LineArt*, e *ColourArt* para que o *Auto-Patch* funcione mesclando estas duas peças dando impressão que é apenas uma.

O *pivot point* da coxa localiza-se na base do tronco e da perna no centro do círculo formado pela união das duas peças.

Figura 94 – Demonstração do encaixe das duas partes da perna e os efeitos funcionando

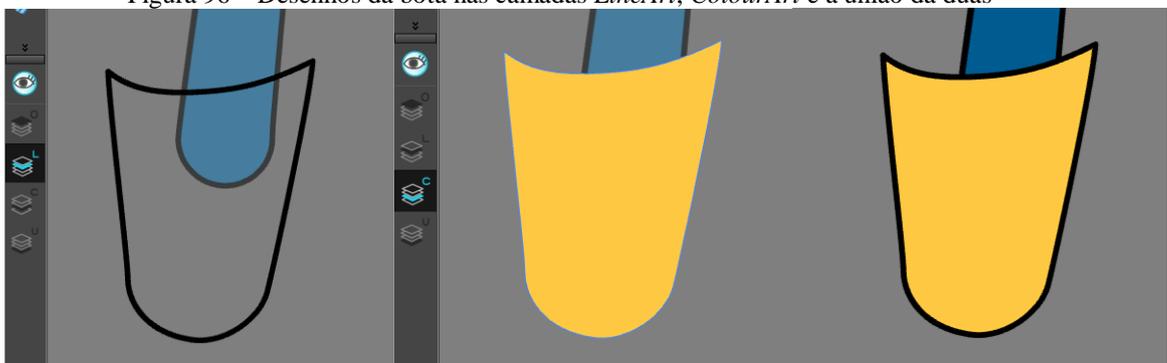


Fonte: A autora

Figura 95 – Rig da perna na *node view*

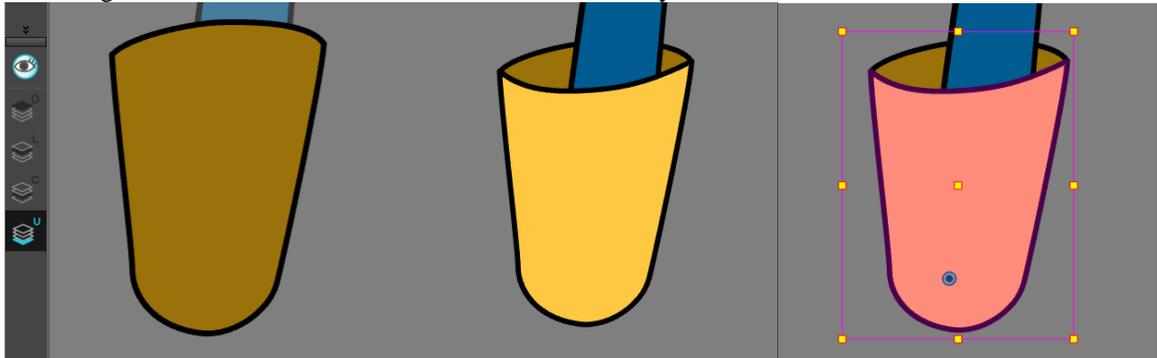
Fonte: A autora

Para o *rigging* de sua extremidade, a bota, como dita na seção [3.2.2.4](#), separou-se as partes: “bota-frente”, feita com o mesmo processo da luva. Duplicou-se o desenho da “bota-frente” sendo renomeada para “bota-tras” copiando o desenho das camadas *LineArt* e *ColourArt* para a *UnderlayArt* e modificando para se adequar a visão da parte de trás. As propriedades dos *drawings* também foram alteradas para que fique apenas visível o desenho que está descrito no nome da camada.

Figura 96 – Desenhos da bota nas camadas *LineArt*, *ColourArt* e a união da duas

Fonte: A autora

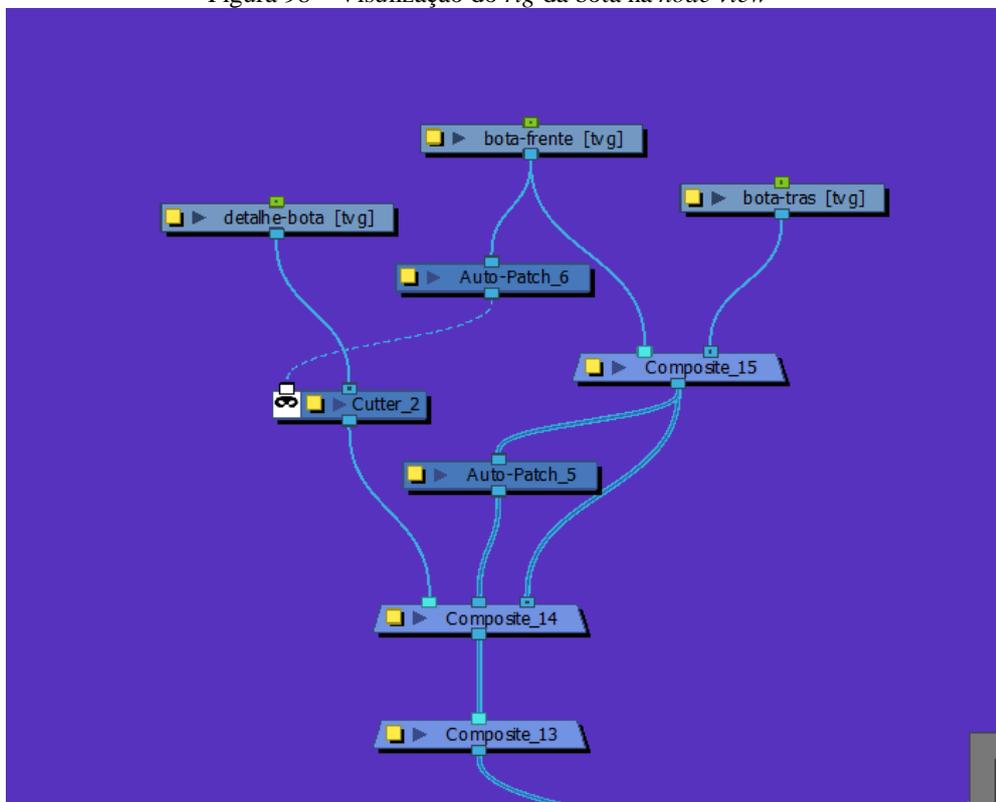
Figura 97 – Desenho da “bota-tras” na camada OverlayArt e o resultado dos desenhos



Fonte: A autora

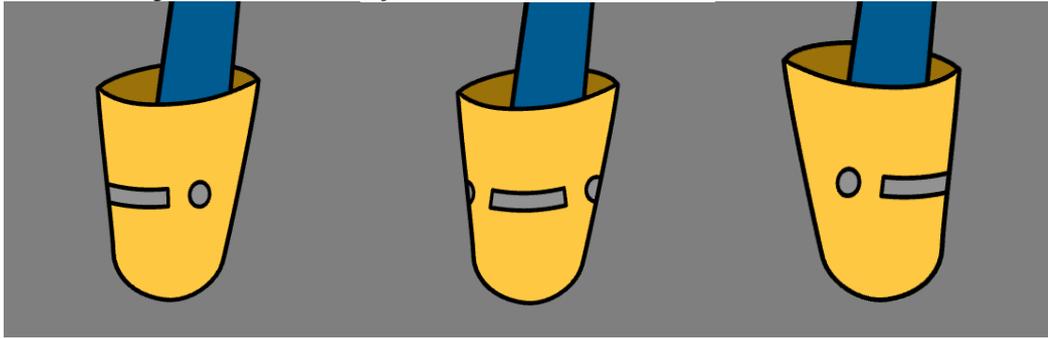
Para que os detalhes da bota que ficam por cima desta parte recém riggada, sejam mascarados por ela, é inserida uma nova *composite* abaixo da *composite* na qual os desenhos da “bota-frente” e “bota-tras” foram plugados, o *drawing* “detalhe-bota” é plugado na entrada do *Cutter node*, e a saída deste, plugada na última *composite* criada. Adiciona-se o *Auto-Patch* a *node view*, levando as informações de *matte* do desenho “bota-frente” para a entrada do *Cutter*.

Figura 98 – Visualização do rig da bota na *node view*



Fonte: A autora

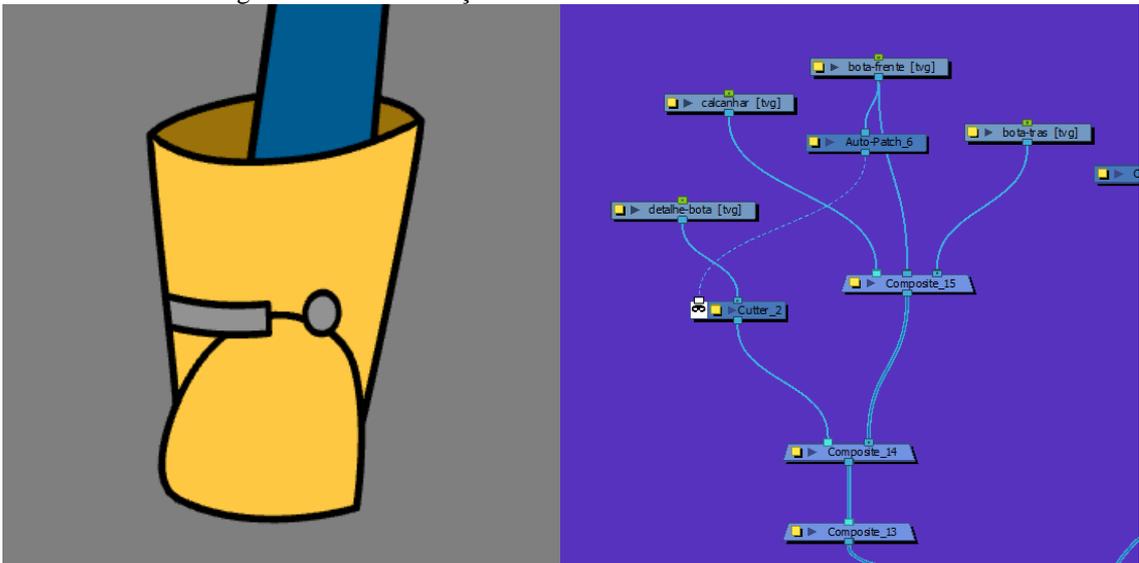
Figura 99 – Demonstração da máscara do detalhe da bota funcionando



Fonte: A autora

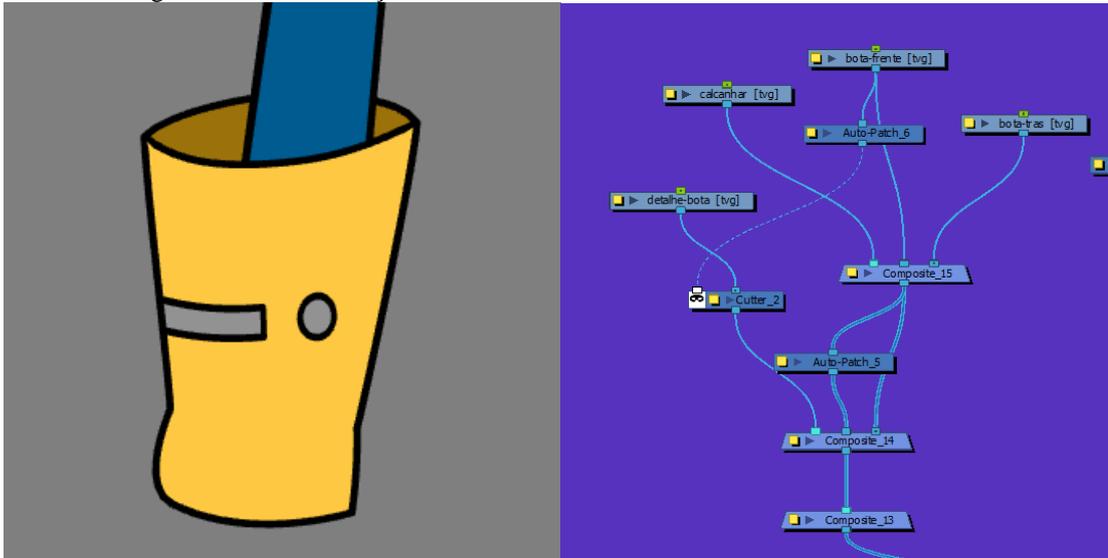
Em seguida é adicionado o desenho complementar para a formação da bota, o calcanhar é com suas linhas na camada *LineArt* e o preenchimento na camada *ColourArt*, o *drawing* é plugado na mesma *composite* em que se encontram as peças “bota-frente” e “bota-tras”. Pluga-se um novo *Auto-Patch*, entre as duas *composites* que possuem os elementos da bota, para que assim, o preenchimento fique à frente das linhas do calcanhar, na parte interna da bota.

Figura 100 – Visualização do calcanhar sem o *Auto-Patch*



Fonte: A autora

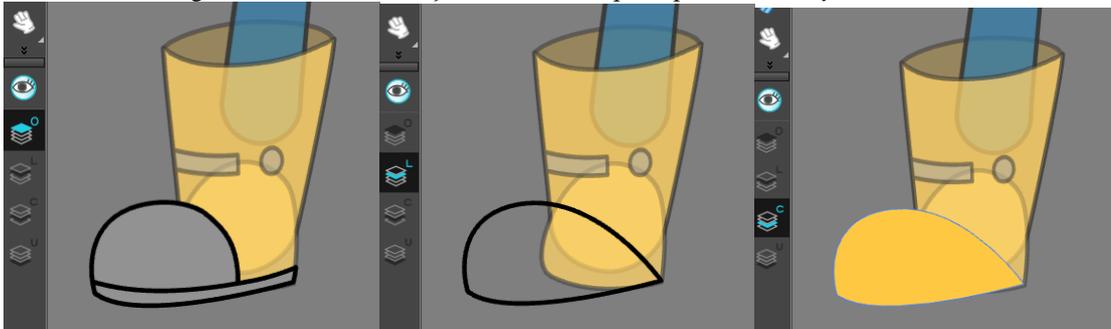
Figura 101 – Visualização do calcanhar com o Auto-Patch funcionando



Fonte: A autora

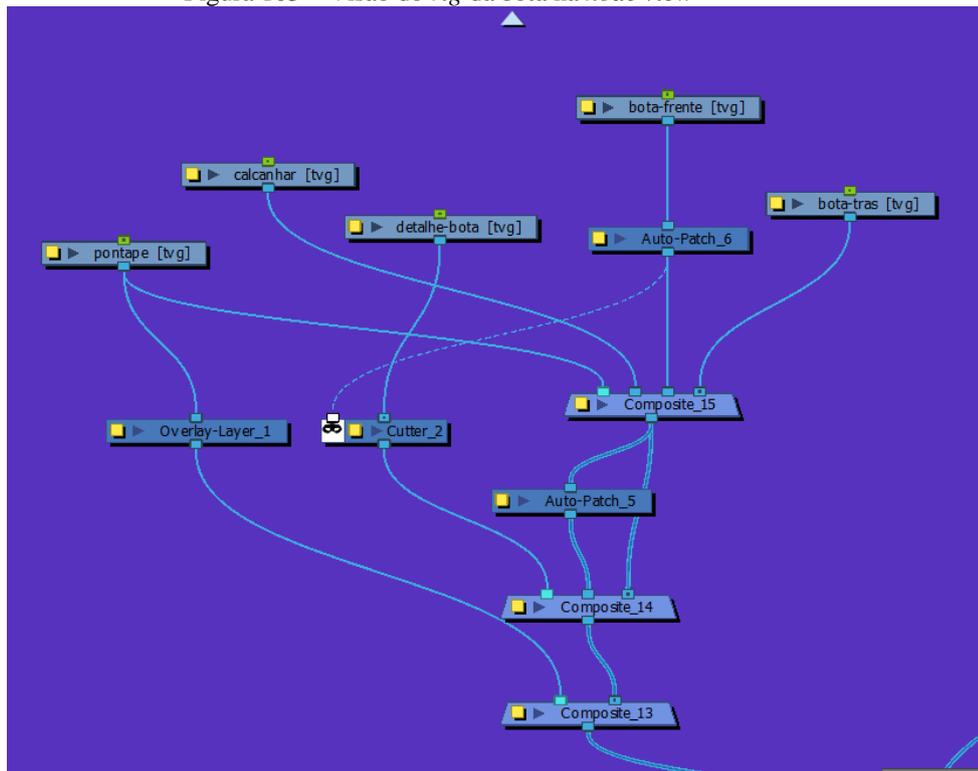
A última peça para o *rigging* da bota trata-se da “pontape” que tem sua arte dividida entre as *ArtLayers* do *drawing*. A parte da ponta e do detalhe cinza da base do pé, está localizado na *OverlayArt*. Já as linhas que vão de encontro da ponta até o canto do calcanhar, se localiza na *LineArt* e seu preenchimento na *ColourArt*.

Figura 102 – Demonstração do desenho “pontape” nas *ArtLayers*

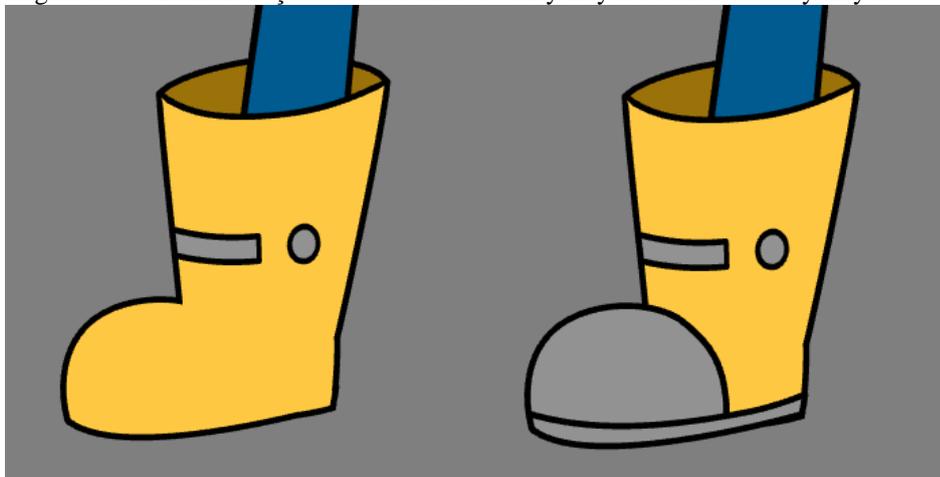


Fonte: A autora

Para o acabamento final desta etapa da bota, é inserido o *Overlay-Layer* no node *view*, conectando a saída do último elemento criado da bota, em sua entrada. E a saída do *Overlay-Layer* na frente do *Auto-Patch* na *composite*. Assim, o desenho que se encontra na *OverlayArt* da “pontape” possa ficar visível.

Figura 103 – Visão do *rig* da bota na *node view*

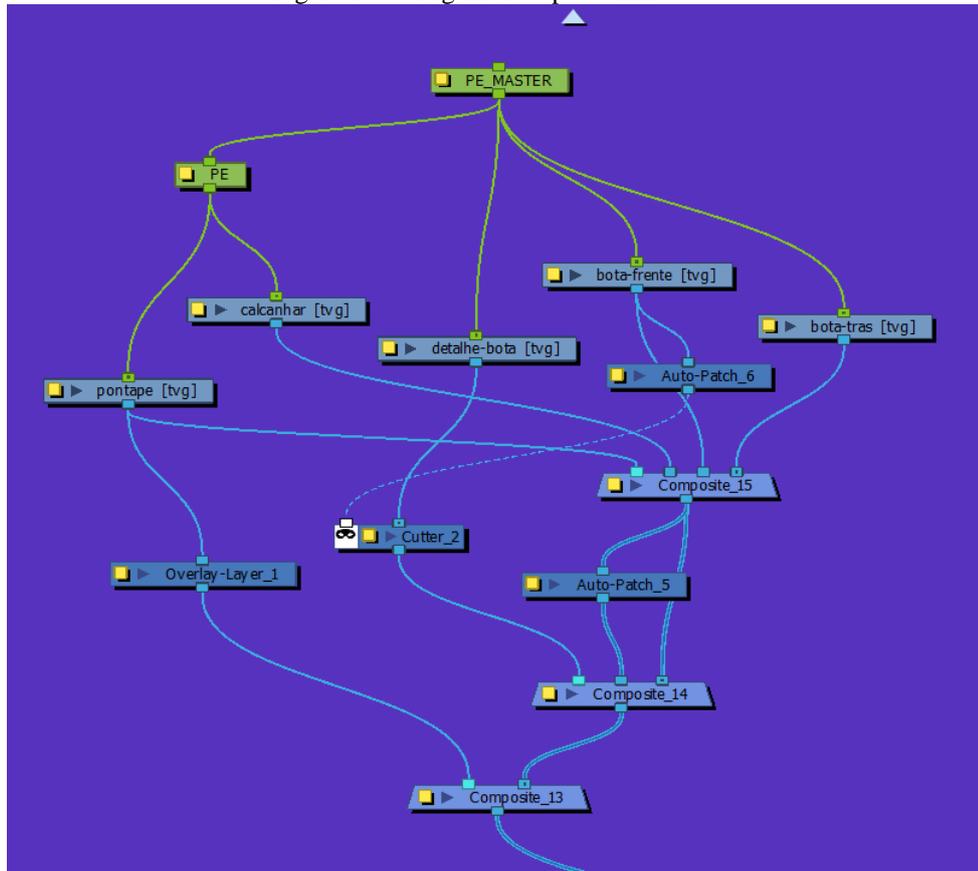
Fonte: A autora

Figura 104 – Demonstração da bota sem o *Overlay-Layer* e com o *Overlay-Layer*

Fonte: A autora

Por último, é criada uma *peg* que une a os desenhos da ponta do pé e do calcanhar, para serem animados em conjuntos e quando necessário animado separadamente. O *pivot point* é localizado no centro do calcanhar. “PE\_MASTER” é a *peg* que é plugada em todos os elementos que formam a bota. Também é posicionado o *pivot point* no centro do calcanhar.

Figura 105 – Rig final do pé de Lizie



Fonte: A autora

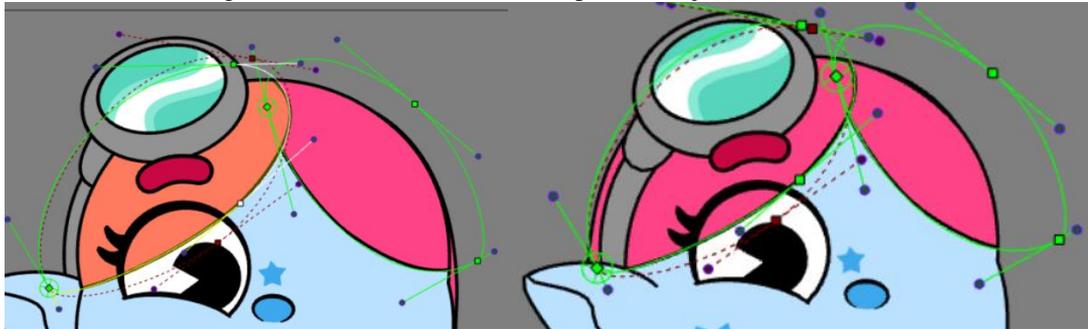
### 3.2.3.5 APLICAÇÃO DE DEFORMADORES

Os deformadores são uma parte importante do *rig*, através deles podemos alterar o desenho para que cause a sensação de diferentes ângulos, sem precisar criar um novo desenho. Além de fazer com que as trocas de poses fiquem mais fluidas, pois é possível animar fazendo com que as peças se deformem gradativamente.

O deformador de Envelope, para as franjas de Lizie foi utilizado para fazer com que a personagem mudasse o ângulo da cabeça sem precisar criar novos desenhos. Além da possibilidade de animar a franja sem precisar esticar o desenho com a ferramenta *Transform*, deixando de modo mais sutil e fluido.

É possível adicionar o deformador Envelope como se estivesse aplicando o deformador de curvas, desta forma, o envelope cumpre o mesmo papel com um “porém” de grande utilidade. Os pontos que são criados são independentes, portanto o primeiro ponto a ser criado do deformador não possui influências sob os outros pontos na cadeia de deformação. Coisa que o deformador *Curve* possui.

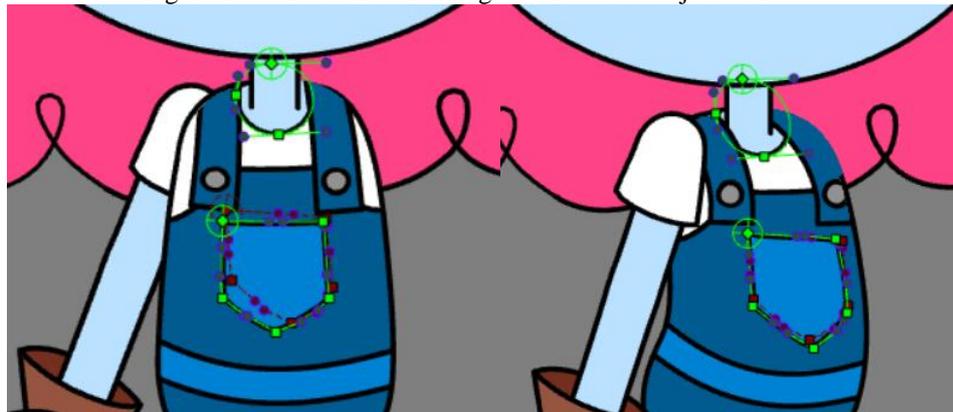
Figura 106 – Deformadores envelope nas franjas de Lizie



Fonte: A autora

Foi utilizado também nos desenhos da gola e do bolso da jardineira, para posar a personagem em ângulos diferentes.

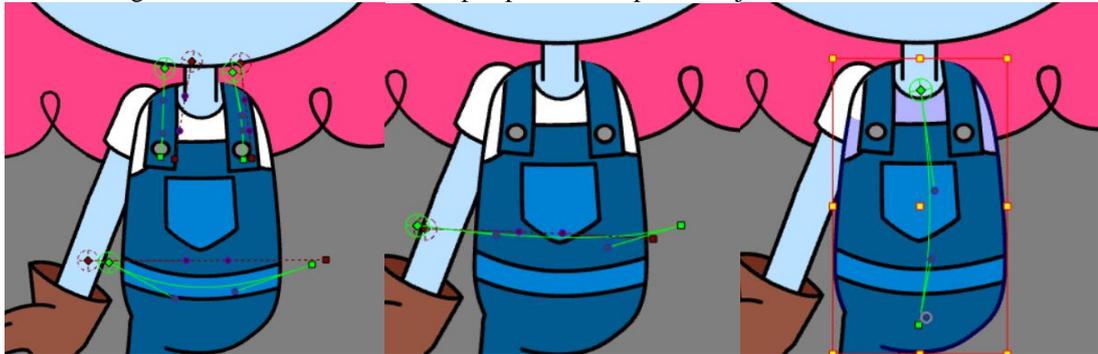
Figura 107 – Deformadores na gola e no bolso da jardineira



Fonte: A autora

A partir desta etapa o deformador Envelope será aplicado da forma que se aplica o deformador *Curve*. Desta forma, foi utilizado nas partes das tiras verticais e da faixa horizontal da jardineira, podendo esticar livremente caso seja necessário. A jardineira também ganhou um deformador, que parte do lado esquerdo, atravessando o tronco até o lado direito, o deformador é adicionado para fora do desenho do tronco, visto que, o restante do desenho da jardineira se encontra mascarado. E por último o tronco, que tem o deformador aplicado de forma vertical, bem no meio do desenho, isto é, para que o tronco consiga ser deformado de uma forma que possa se exagerar na silhueta.

Figura 108 – Deformador envelope aplicado nas partes da jardineira e tronco

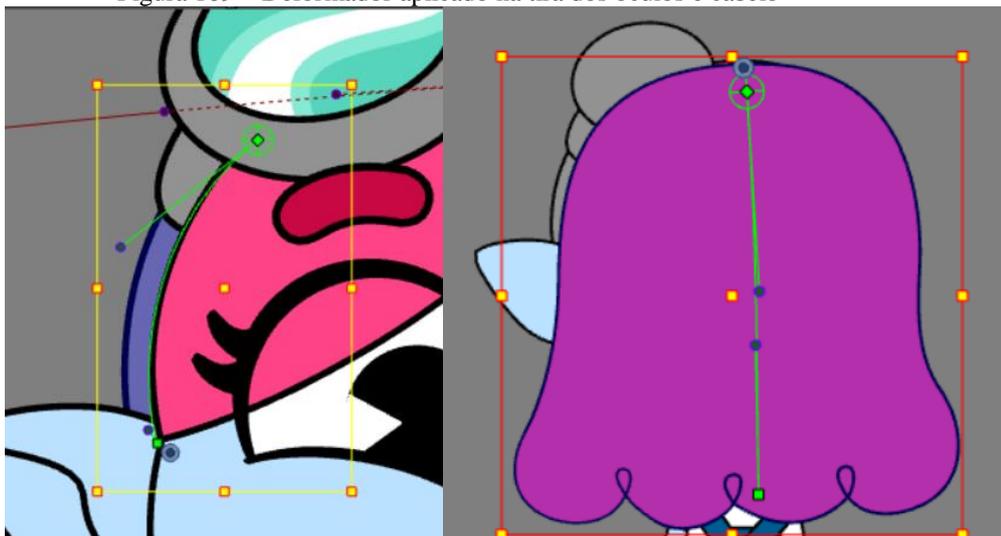


Fonte: A autora

As tiras dos óculos também recebem o deformador, para facilitar na hora de posar o desenho quando a personagem virar a cabeça.

No cabelo da personagem o deformador foi aplicado na vertical, para pegar toda a sua extensão.

Figura 109 – Deformador aplicado na tira dos óculos e cabelo



Fonte: A autora

Para as sobrancelhas que possuem um peso importante na expressão facial, foi optado pelo deformador, para dar flexibilidade, podendo exagerar e deixá-las retas ou curvas. Assim, evitando ficar adicionando desenhos de sobrancelhas à biblioteca sempre que precisar de alguma nova.

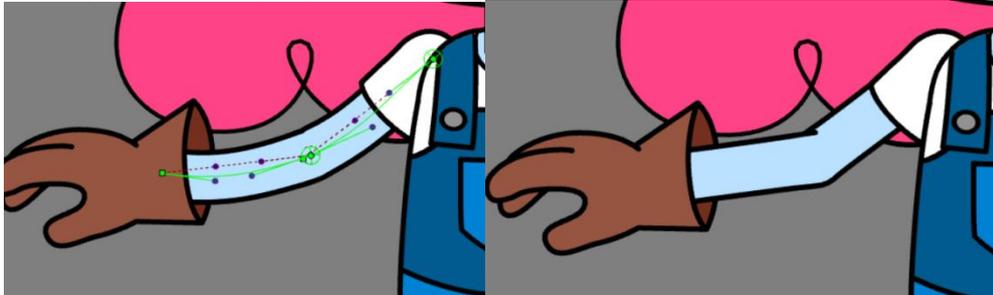
Figura 110 – Uso do deformador nas sobrancelhas



Fonte: A autora

Os deformadores também foram aplicados nos braços, pois deixam com a liberdade de esticar e exagerar na linguagem corporal e silhuetas, quando forem trabalhadas as poses.

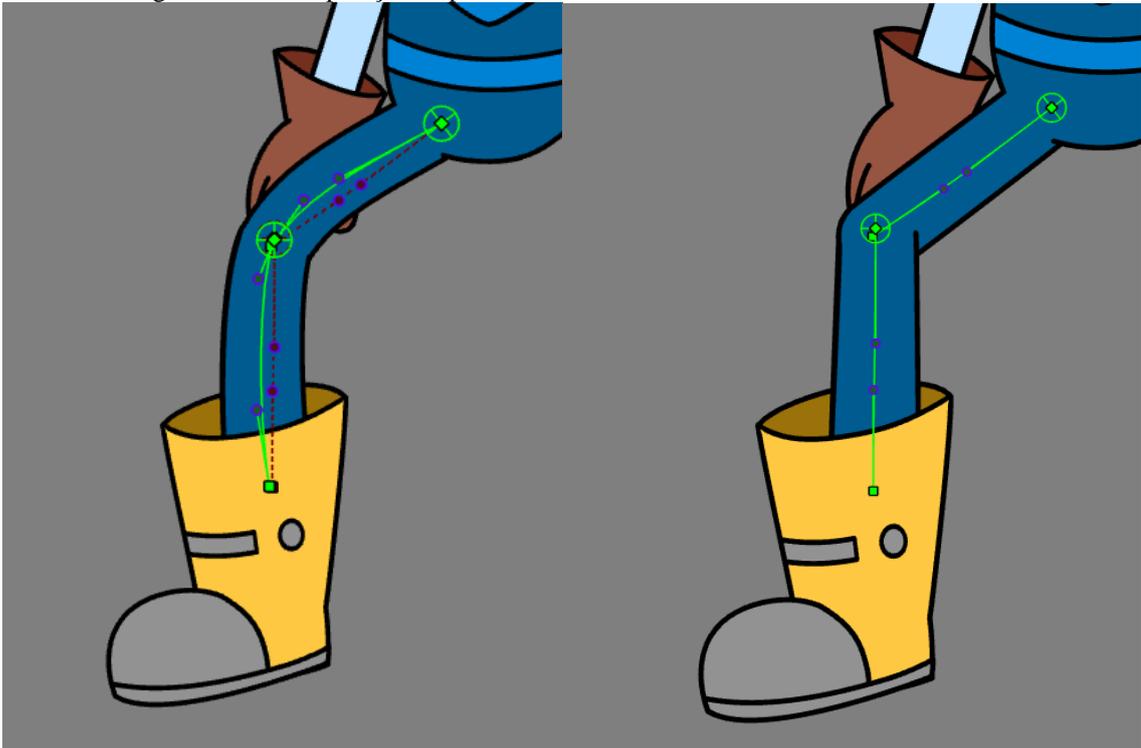
Figura 111 – Comparação do braço com deformador e sem deformador



Fonte: A autora

Também foram adicionados os deformadores na coxa e na perna com o mesmo intuito do braço. Exagerar nas poses quando necessário.

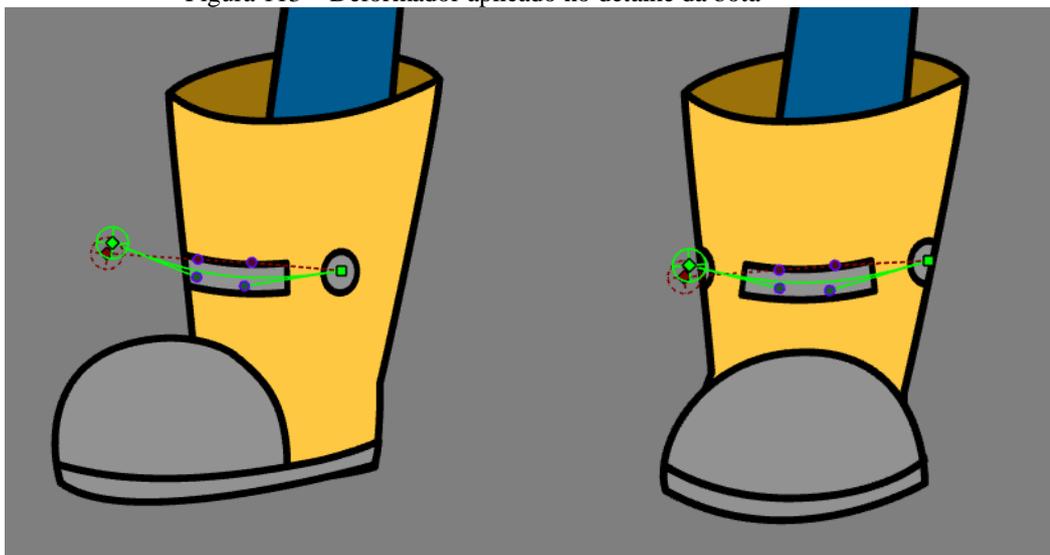
Figura 112 – Comparação da perna com deformadores e sem deformadores



Fonte: A autora

Os detalhes da bota receberam um deformador, pois facilita na hora de posar a bota em diferentes ângulos. Apenas com a ferramenta *Transform* isso já seria possível, porém o deformador permite que o desenho seja curvado, causando ainda mais a sensação de profundidade em algumas poses.

Figura 113 – Deformador aplicado no detalhe da bota



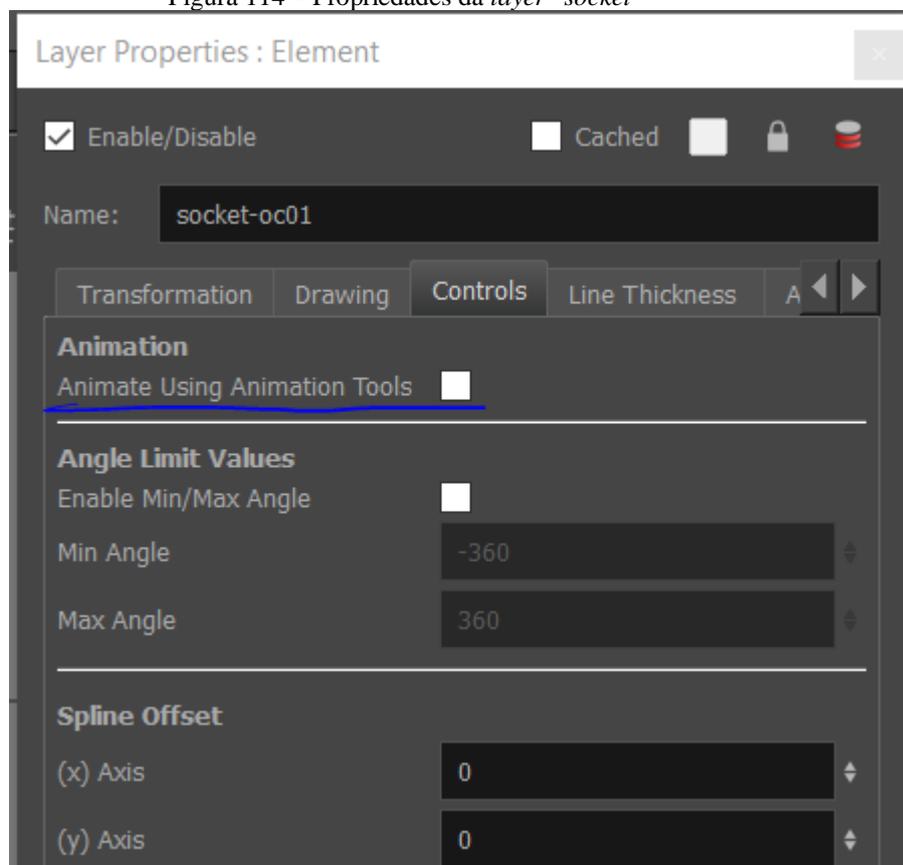
Fonte: A autora

Após adicionar os deformadores nos desenhos, cada *drawing*, passa a ter um grupo com o nome “*Deformation* - (o nome do *drawing* que foi adicionado o deformador)” conectado a si. Em seguida, é necessário criar uma *peg* conectada a este grupo, para que se possa movimentar o desenho e o controlador ao mesmo tempo. Desta forma o deformador sempre estará no lugar correto que fora adicionado.

### 3.2.3.6 DUPLICANDO AS PARTES

Primeiro passo a ser feito, é verificar a hierarquia dentro de cada parte e membros, e criando *pegs* para uni-los, até chegar a um controlador que é “pai” de todos. Em seguida é criado um *drawing*, nomeado de “*socket*-(objeto ou parte do corpo)01” e plugado neste controlador que está no topo da hierarquia. Nas propriedades desta *layer*, na aba “*Controls*” é desativada a opção que permite que o *drawing* seja animado pela ferramenta *Transform*, que é a utilizada para animar dentro do *software*.

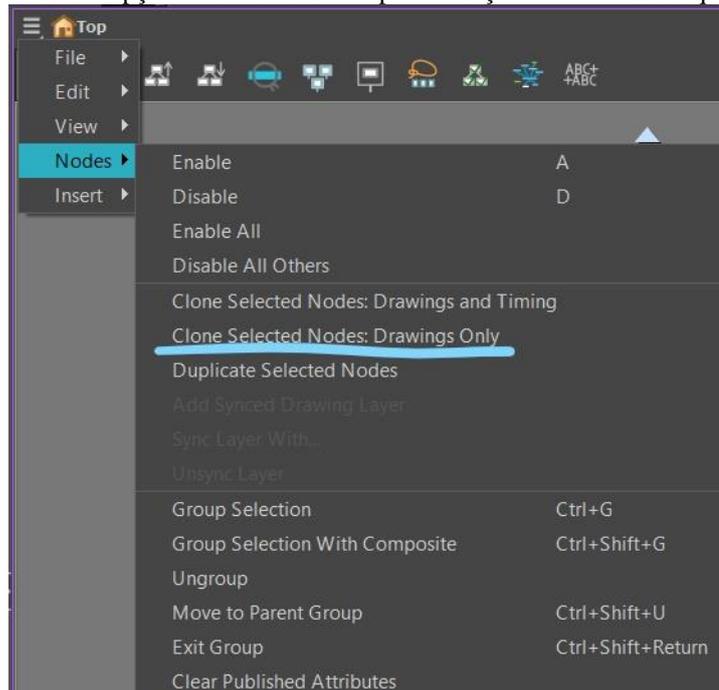
Figura 114 – Propriedades da *layer* “*socket*”



Fonte: A autora

É feita uma seleção de todos os elementos daquela hierarquia, o *drawing* “socket” também é selecionado. Em seguida nas opções de “*Nodes*” dentro das opções da *node view*, é selecionada a opção “*Clone Selected Node: Drawinds Only*”. Esta opção duplicará a hierarquia e quando for necessário adicionar um novo desenho à biblioteca, o desenho será adicionado automaticamente para o outro lado que fora duplicado.

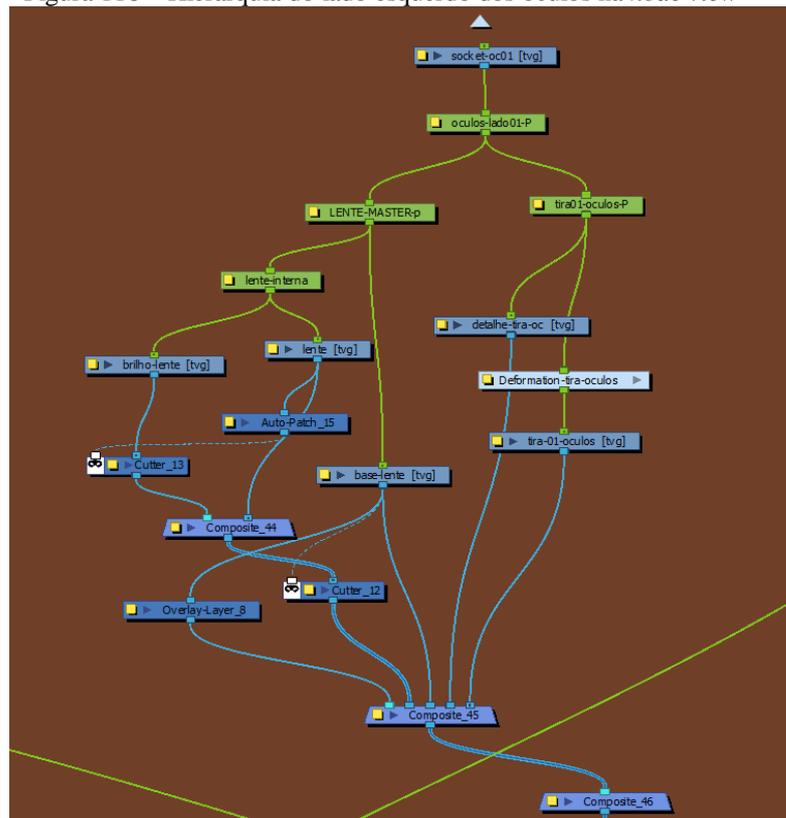
Figura 115 – Opção a ser selecionada após a seleção de toda a hierarquia



Fonte: A autora

Com o *rig* duplicado, desliga-se o modo animação do *ToonBoom Harmony* para poder inverter o lado das peças e se encaixarem no lado direito sem que haja nenhuma animação. O *rig* é invertido através da ferramenta *Flip Vertical* na aba *Tool Properties* do programa. Após o reposicionamento do *rig* através do *drawing* “socket”, é reativado o modo de animação do *software*.

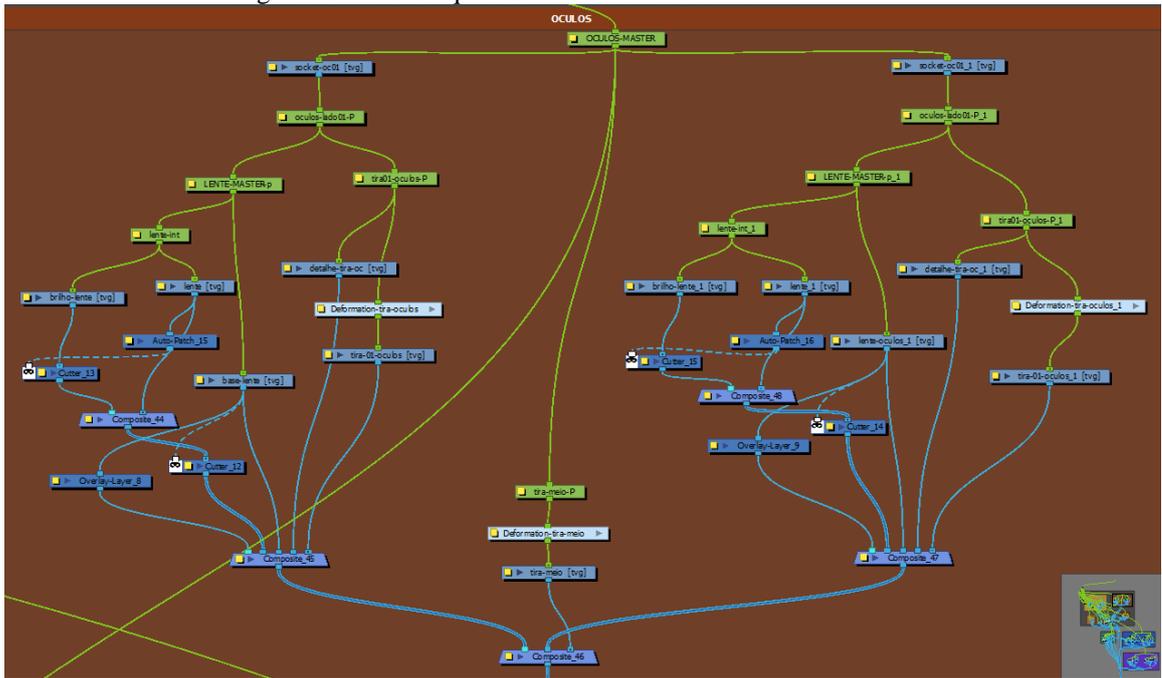
Figura 116 – Hierarquia do lado esquerdo dos óculos na *node view*



Fonte: A autora

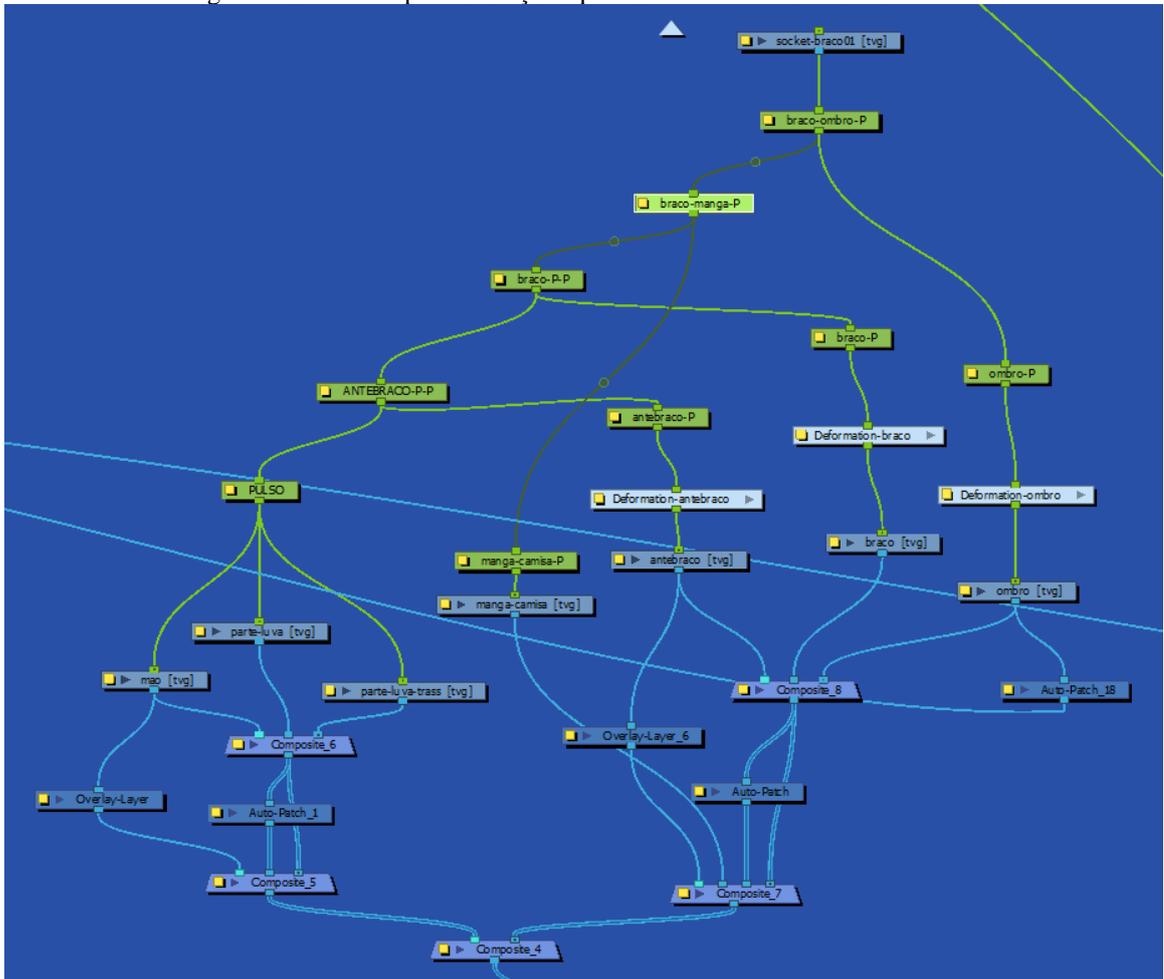
Com a hierarquia revisada do lado esquerdo dos óculos, foi duplicada e invertida para servir de direita, renomeia-se, portanto a *peg* para “oculos-lado02-P” sendo agrupada por uma última *peg*, a “OCULOS-MASTER” que também agrupa a “tira-meio” que conecta as duas bases da lente.

Figura 117 – Hierarquia final dos óculos na node view



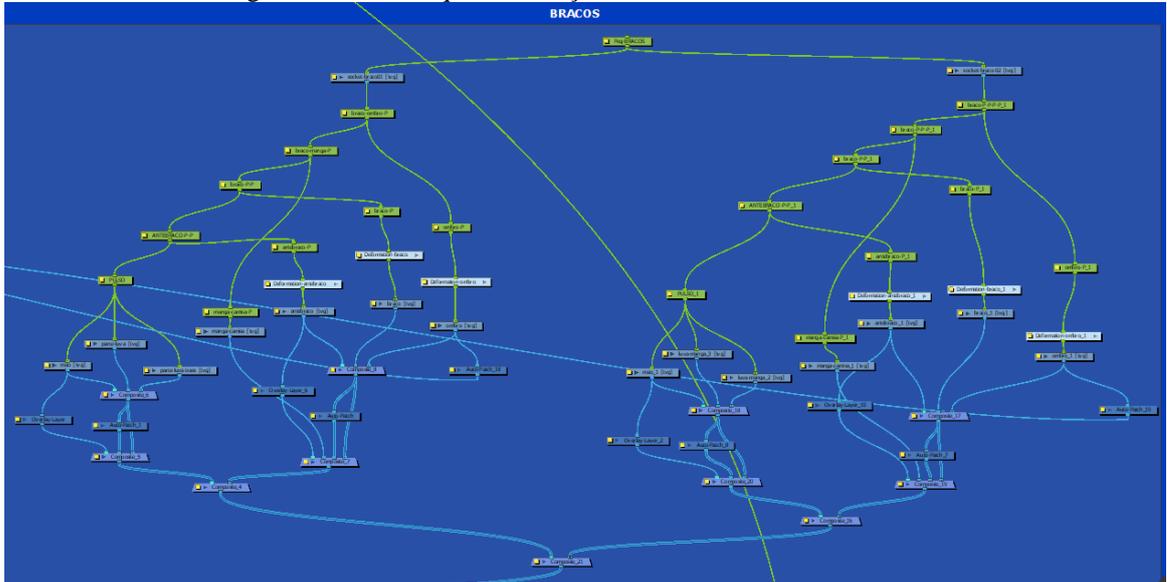
Fonte: A autora

Figura 118 – Hierarquia do braço esquerdo da Lizie na node view



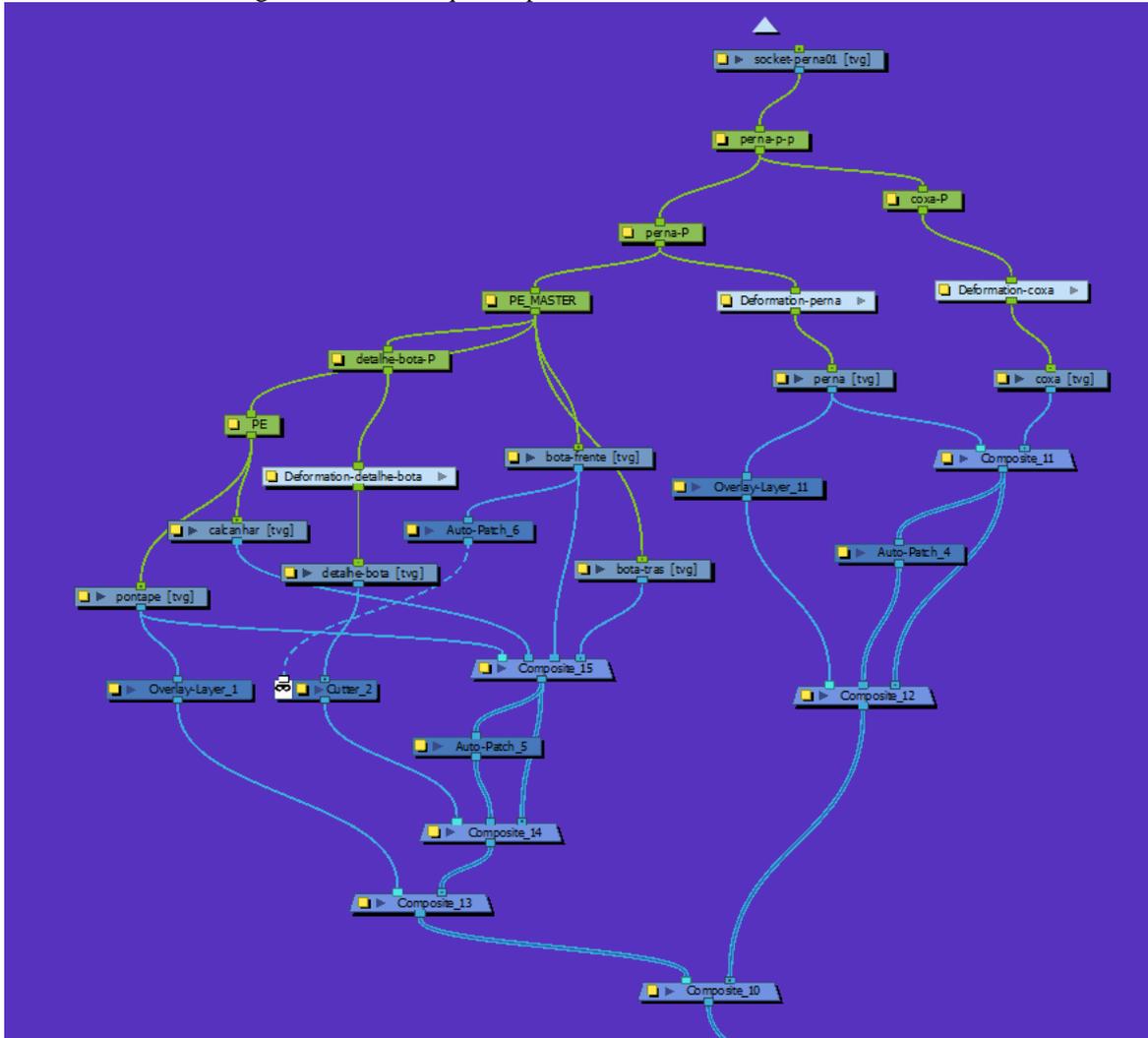
Fonte: A autora

Figura 119 – Hierarquia dos braços de Lizie na node view

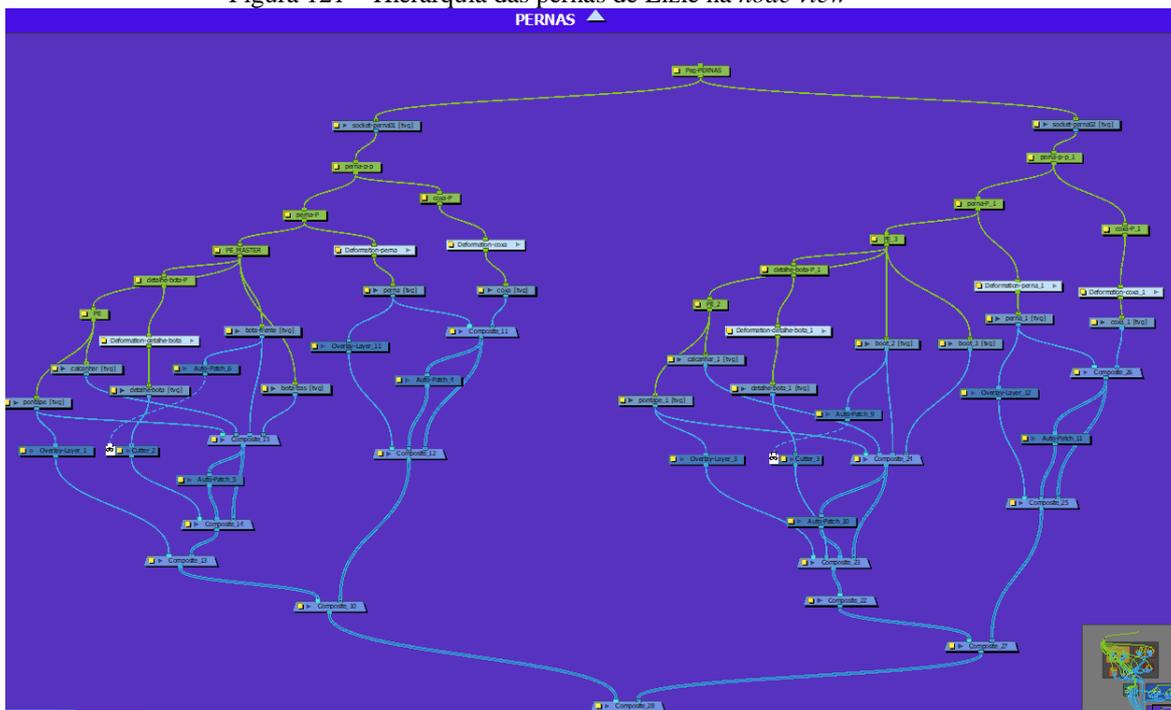


Fonte: A autora

Figura 120 – Hierarquia da perna de Lizie na *node view*



Fonte: A autora

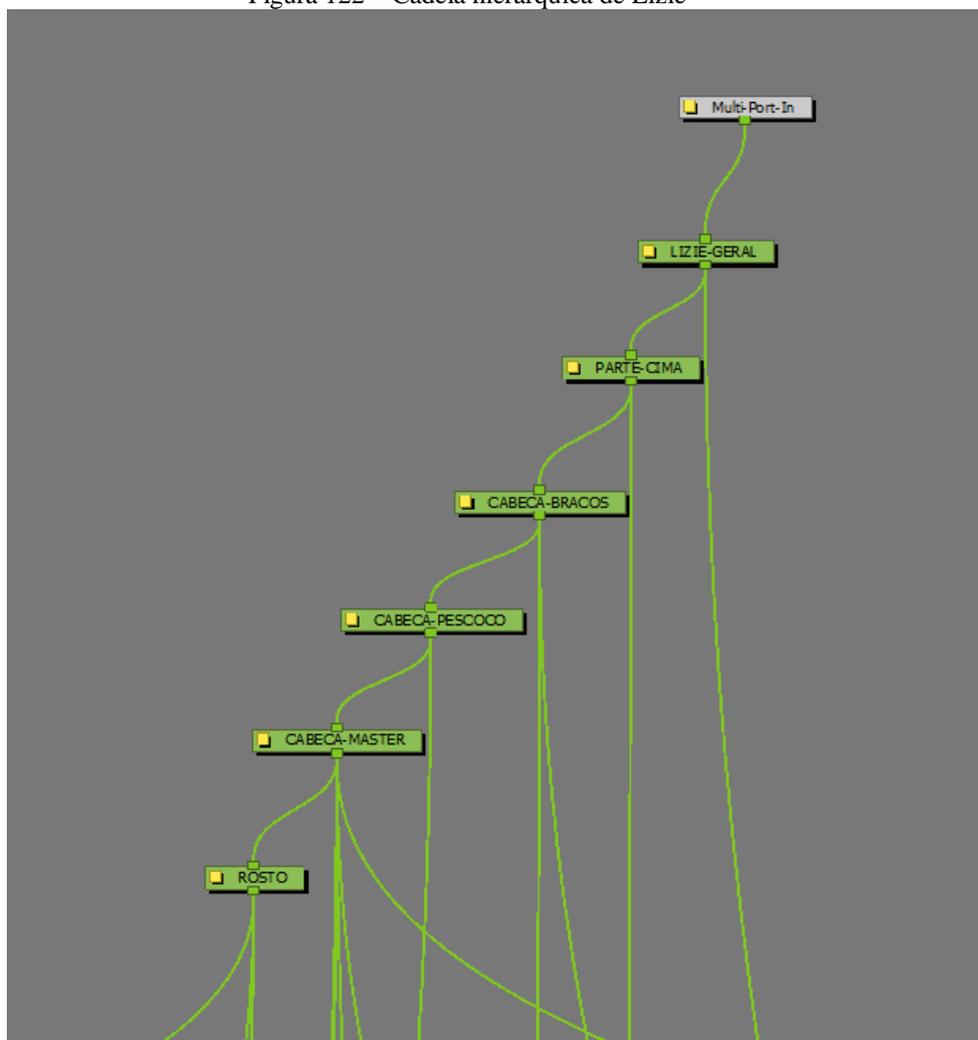
Figura 121 – Hierarquia das pernas de Lizie na *node view*

Fonte: A autora

### 3.2.3.7 HIERARQUIA

Estas hierarquias que são formadas unindo as pequenas hierarquias de cada parte/membros do corpo são muito importantes, pois serão as primeiras que serão animadas, visto que se começa a blocar as poses a partir delas.

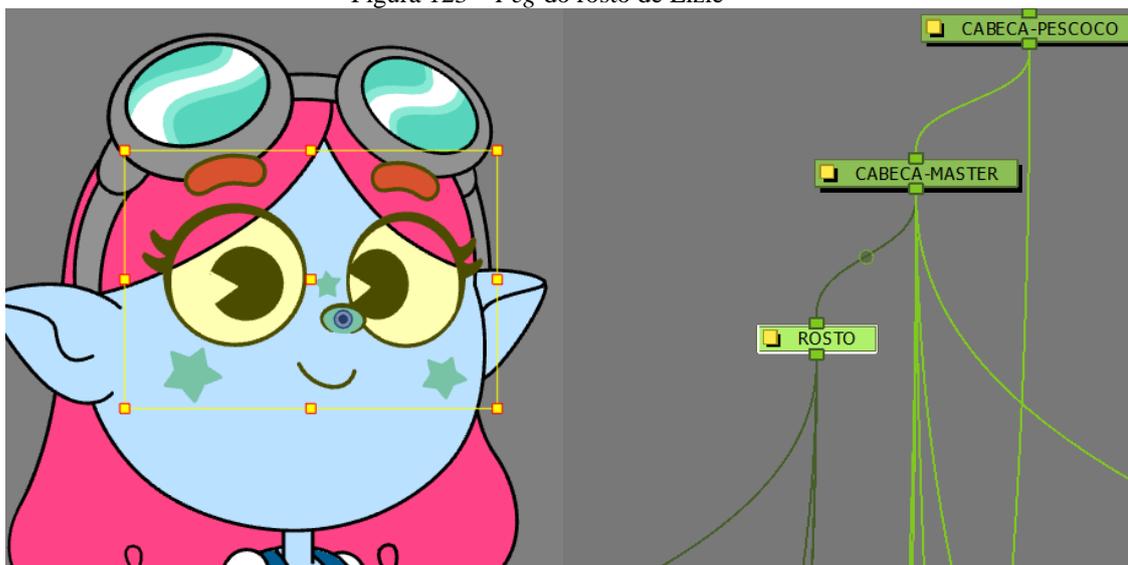
Figura 122 – Cadeia hierárquica de Lizie



Fonte: A autora

Para facilitar na hora de animar o rosto da personagem, foi criada uma *peg* que controle apenas os elementos do rosto.

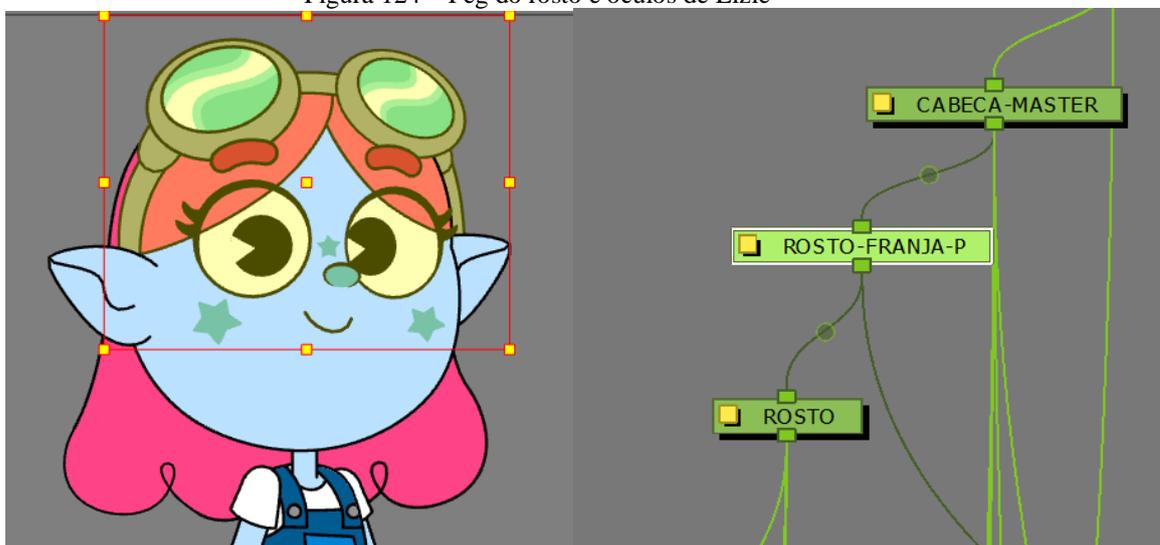
Figura 123 – Peg do rosto de Lizie



Fonte: A autora

Para facilitar na mudança de ângulo no rosto da personagem, foi criada um peg que controle os elementos que formam o rosto, as franjas e seus óculos. Com pequenos movimentos e ajustes a sensação de mudança de ângulo na personagem já é notada.

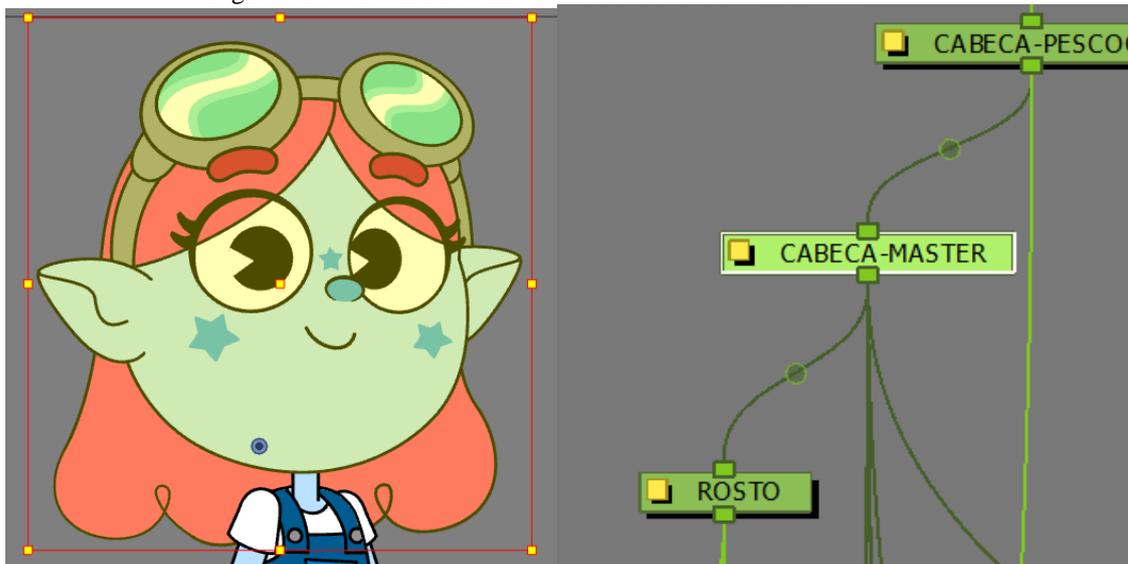
Figura 124 – Peg do rosto e óculos de Lizie



Fonte: A autora

A *peg* “CABECA-MASTER” é a que facilita na hora de movimentar a cabeça da personagem para cima e para baixo. Seu *pivot point* é posicionado no mesmo local que fora adicionada na *peg* com apenas os desenhos da cabeça, como é explicado na seção [3.2.3.2](#) deste trabalho.

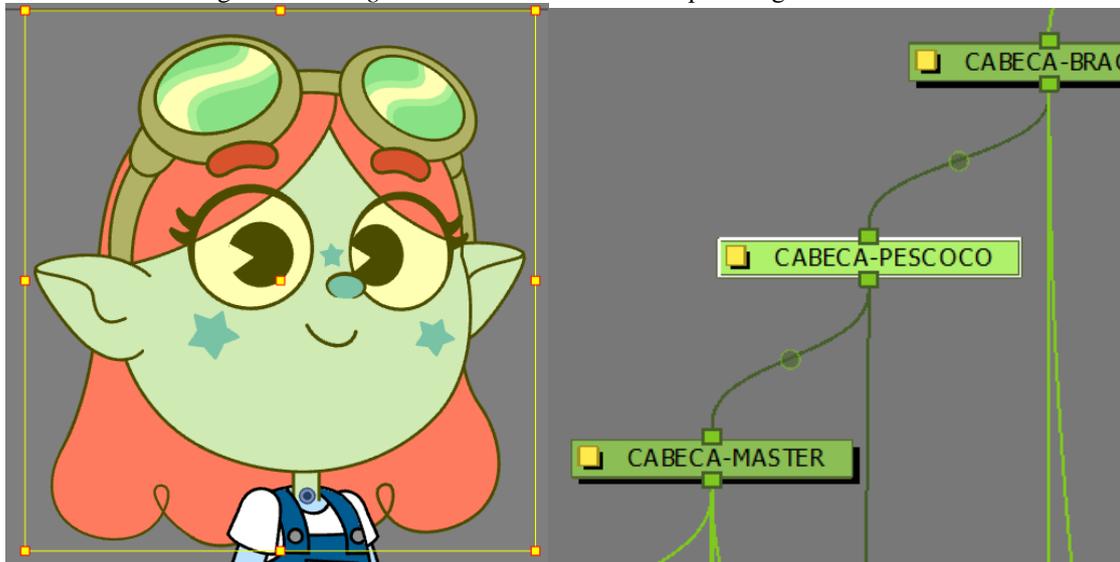
Figura 125 – Controlador da “CABECA-MASTER” da Lizie



Fonte: A autora

O controlador “CABECA-PESCOCO” foi criado para facilitar na hora de encaixar a cabeça da personagem ao tronco na etapa de blocagem de poses da animação.

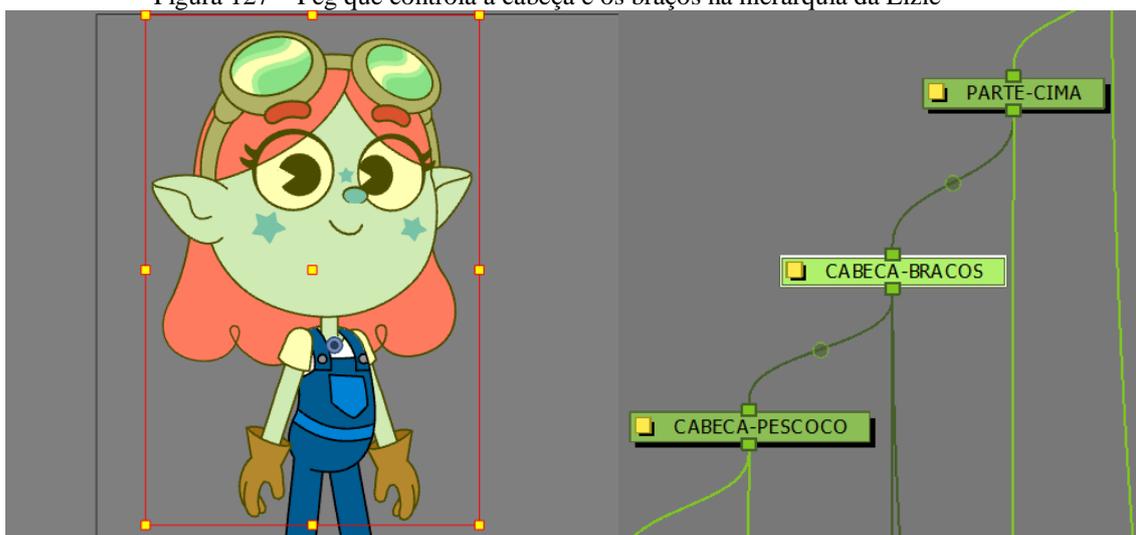
Figura 126 – Peg “CABECA-PESCOCO” da personagem Lizie



Fonte: A autora

A *peg* “CABECA-BRACOS” é bastante utilizada para o posicionamento dessas partes na blocagem de poses para animação, quando é necessário que se mova os dois braços em conjunto.

Figura 127 – Peg que controla a cabeça e os braços na hierarquia da Lizie



Fonte: A autora

“PARTE-CIMA” é a *peg* que controla todos os elementos que compõe o corpo da personagem a partir de seu tronco. Útil para girar a parte de cima da personagem e para deformações exageradas como o *squash* e *stretch*.

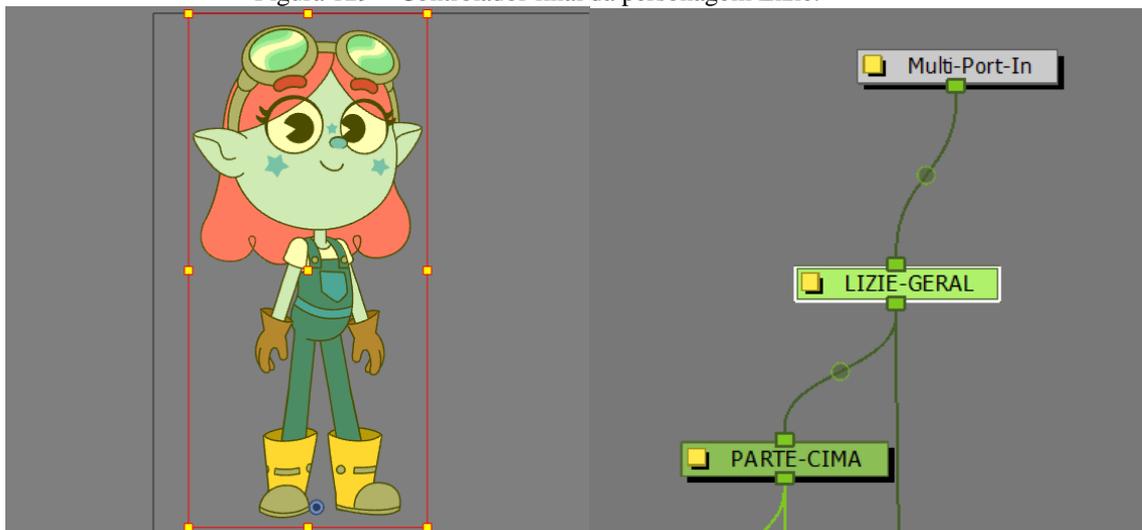
Figura 128 – Peg que agrupa os elementos da parte de cima da personagem



Fonte: A autora

E por fim, a *peg* geral da personagem, onde abrange dos os elementos do corpo.

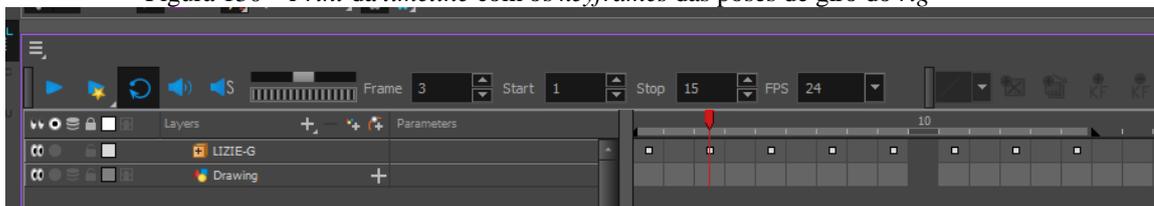
Figura 129 – Controlador final da personagem Lizie.



Fonte: A autora

### 3.2.3.8 O TURN AROUND

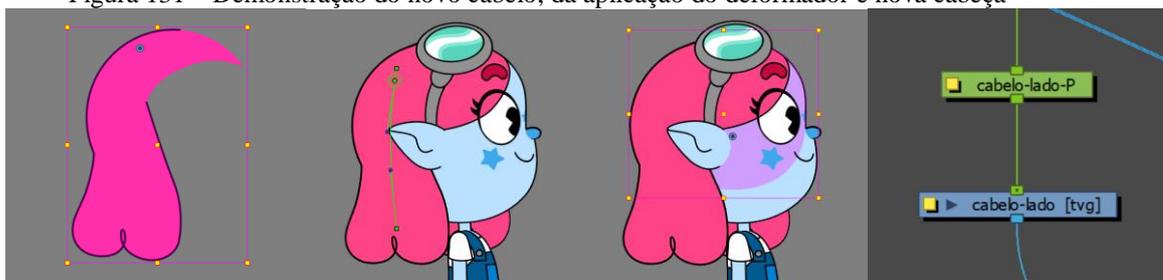
Para que o *rig* esteja completo e pronto para animação, é necessário ter todas as poses de giro do personagem. Para adicioná-las ao *rig*, é preciso que as peças criadas sejam reposicionadas nas poses, criando *keyframes* na *timeline*. Ou seja, as peças e controladores são animados nesta etapa. Os deformadores adicionados anteriormente cumprem seu papel, fazendo com que poupe a maioria do trabalho de criação de novos desenhos para as peças.

Figura 130 – Print da *timeline* com os *keyframes* das poses de giro do *rig*

Fonte: A autora

Um novo formato de cabeça é adicionado à biblioteca do desenho “cabeça”, assim como um novo desenho de nariz e uma nova parte do cabelo, para ser utilizada na pose de perfil da personagem. O *pivot point* do cabelo adicional foi posicionado na parte superior do desenho, e aplicado o um deformador envelope na vertical para caso seja feito um *squash* e *stretch* na etapa de animação.

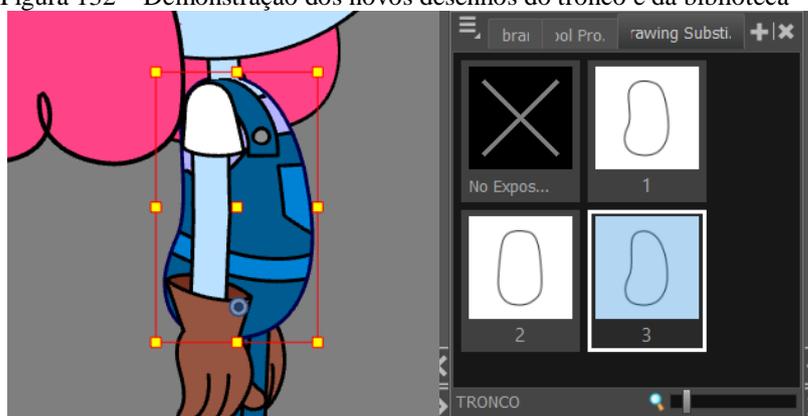
Figura 131 – Demonstração do novo cabelo, da aplicação do deformador e nova cabeça



Fonte: A autora

Dois novos desenhos são adicionados à biblioteca do *drawing* “TRONCO”, um para ser trocado quando a personagem estiver pose  $\frac{3}{4}$  e perfil.

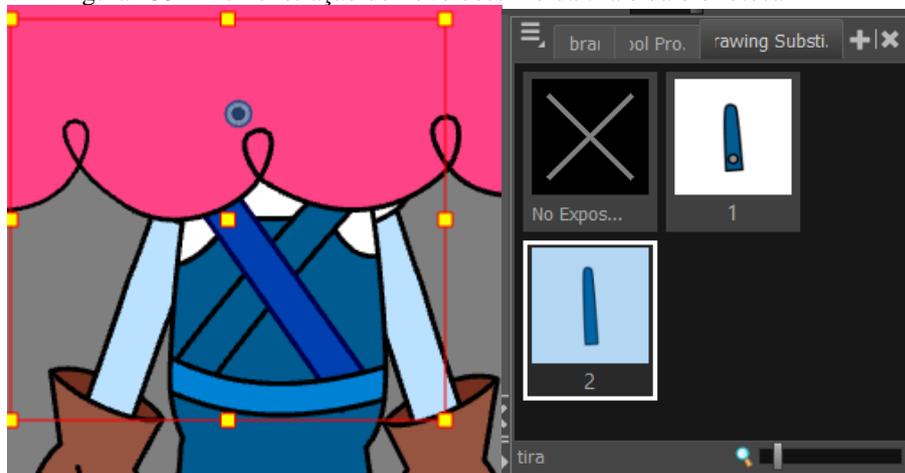
Figura 132 – Demonstração dos novos desenhos do tronco e da biblioteca



Fonte: A autora

Para a pose de costas, foi acrescentado à biblioteca o desenho das tiras da jardineira da personagem.

Figura 133 – Demonstração do novo desenho da tira e da biblioteca



Fonte: A autora

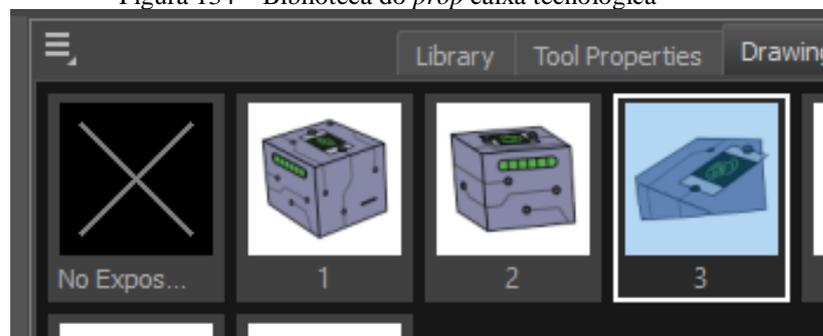
### 3.3 PROPS

Os desenhos dos objetos pequenos nos quais a personagem interage, foram feitos direto nos arquivos de cada cena e adicionados à biblioteca. Assim, depois de criada a primeira pose do objeto, para criar a próxima pose já havia um desenho de base, sem precisar começar do zero.

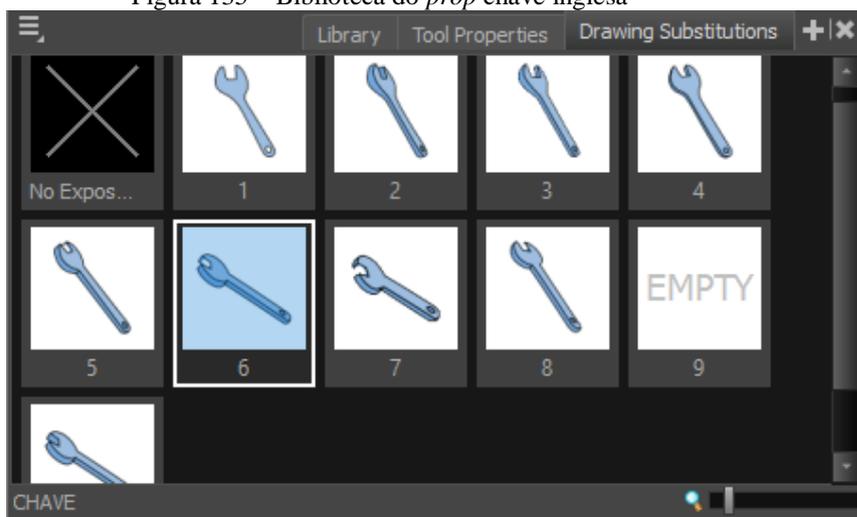
Os *props* que interagiram diretamente com os personagens nas cenas do *teaser*, são:

- Uma caixa tecnológica, consertada por Lizie e que não é mostrada sua função no *teaser*;
- Uma chave de boca utilizada pela protagonista;
- Um disquete velho e estragado;
- Um SSD super tecnológico, obra de Sphix;
- A moto de Lizie.

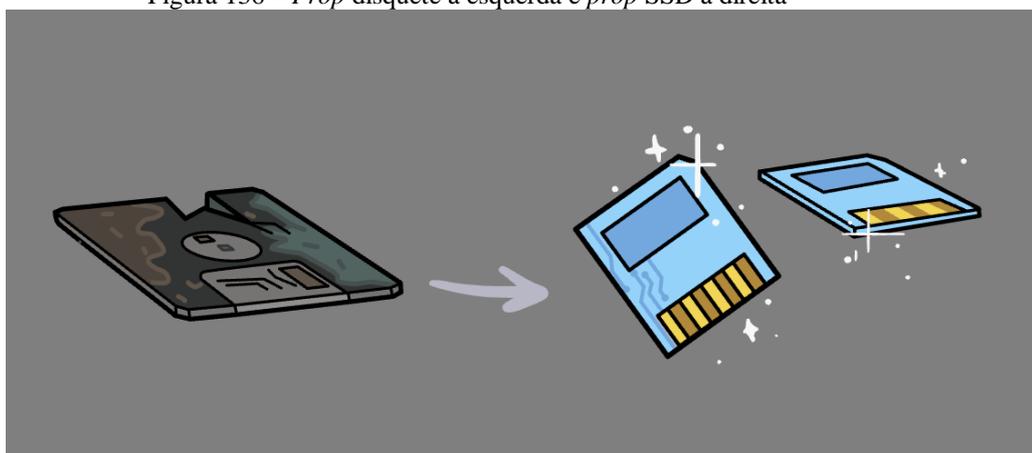
Figura 134 – Biblioteca do *prop* caixa tecnológica



Fonte: A autora

Figura 135 – Biblioteca do *prop* chave inglesa

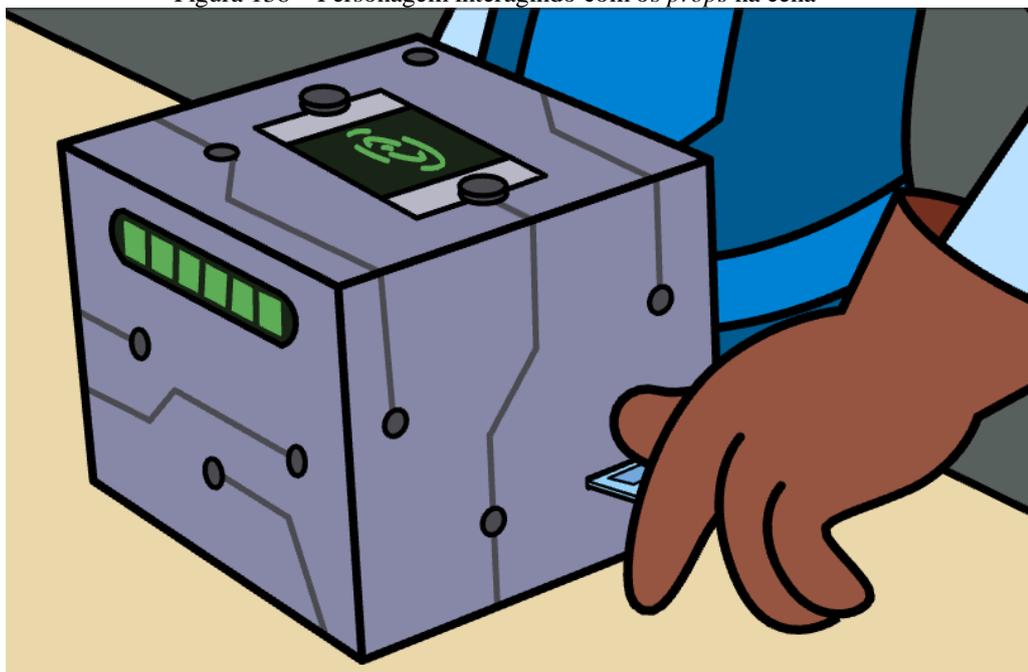
Fonte: A autora

Figura 136 – *Prop* disquete à esquerda e *prop* SSD à direita

Fonte: A autora

Figura 137 – Etapa de animação, com a personagem interagindo com os *props*

Fonte: A autora

Figura 138 – Personagem interagindo com os *props* na cena

Fonte: A autora

### 3.3.1 A MOTO

A moto da personagem foi o *prop* que demandou mais tempo para ser finalizado, porque precisaria ser dividido em partes para encaixar os personagens, já que eles interagiriam diretamente nas cenas.

Foi importado ao *ToonBoom Harmony* o *concept* da moto para vetorização. A pose de frente da moto foi criada com a intenção de servir para todas as cenas nas quais aparece, exceto na cena em que a traseira da moto é mostrada.

A divisão das partes da moto foi pensada de acordo com a ordem dos objetos que se sobrepõe, portanto, dividiram-se as partes: “banco”, “moto-frente”, “moto-pt2”, “L-helice”, “R-helice” e o grupo “chaveiro”, onde foram agrupados os elementos que compõe os pingentes pendurados ao guidão da moto. As hélices foram separadas da moto, já que seria criado um ciclo de animação de giro de hélice para as cenas na qual a moto estaria se movimentando.

Figura 139 – Partes da moto

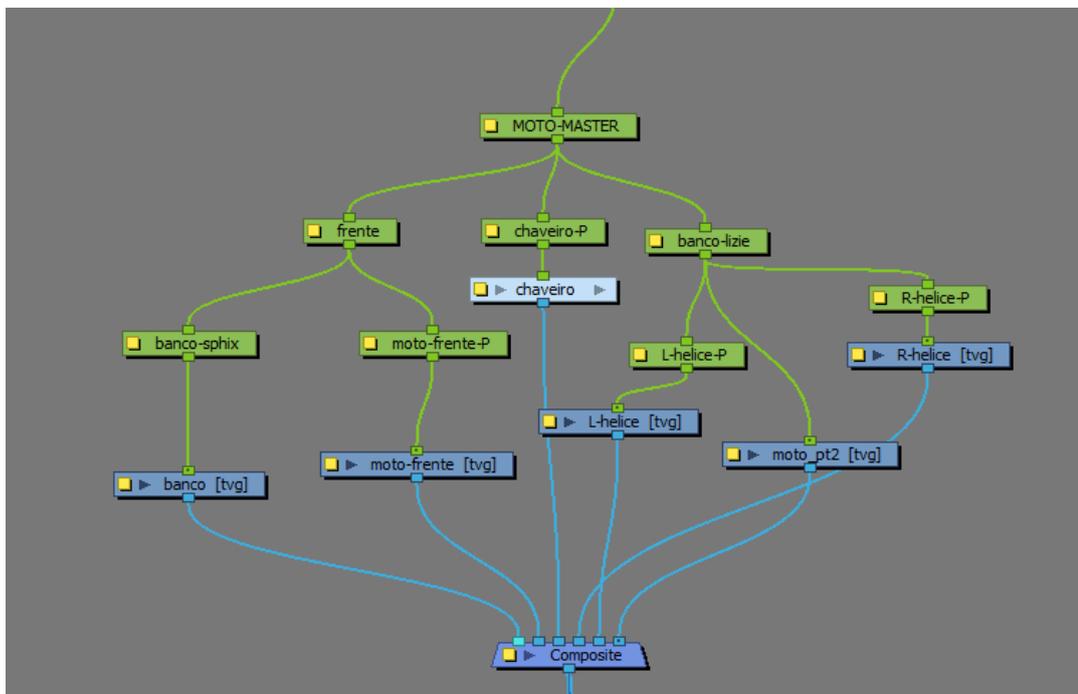


Fonte: A autora

Foram adicionados os controladores e estabelecida a hierarquia. “MOTO-MASTER” é a *peg* pai, plugada nas *pegs* que controlam cada parte. A *peg* “frente” controla o desenho do banco que se encaixa na frente de Sphix e o desenho “moto-frente” que abrange as parte da roda, painel e

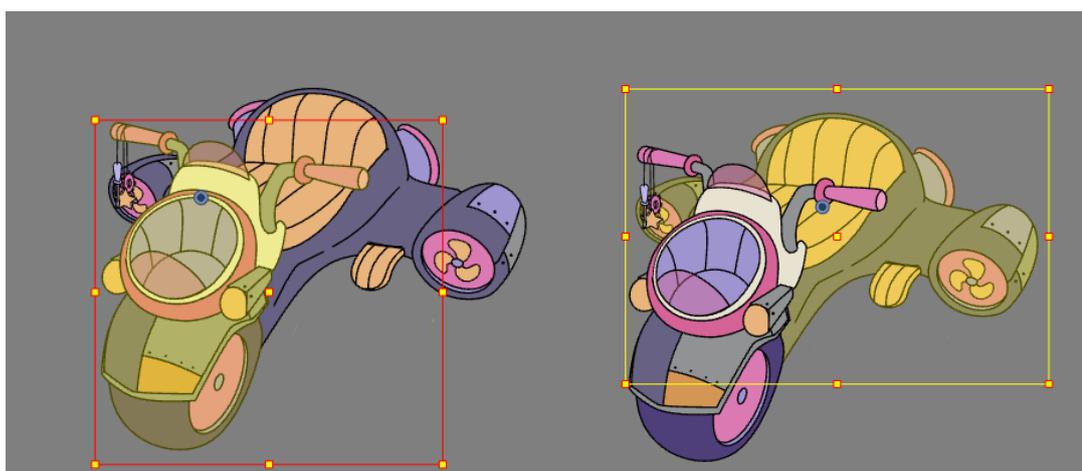
guidão. A peg “banco-lizie” fora criada para agrupar as partes de trás da moto, como toda a sua traseira incluindo as hélices. E o controlador para o grupo do chaveiro.

Figura 140 – *Rig* da visão traseira da moto



Fonte: A autora

Figura 141 – Divisão da moto em *pegs*

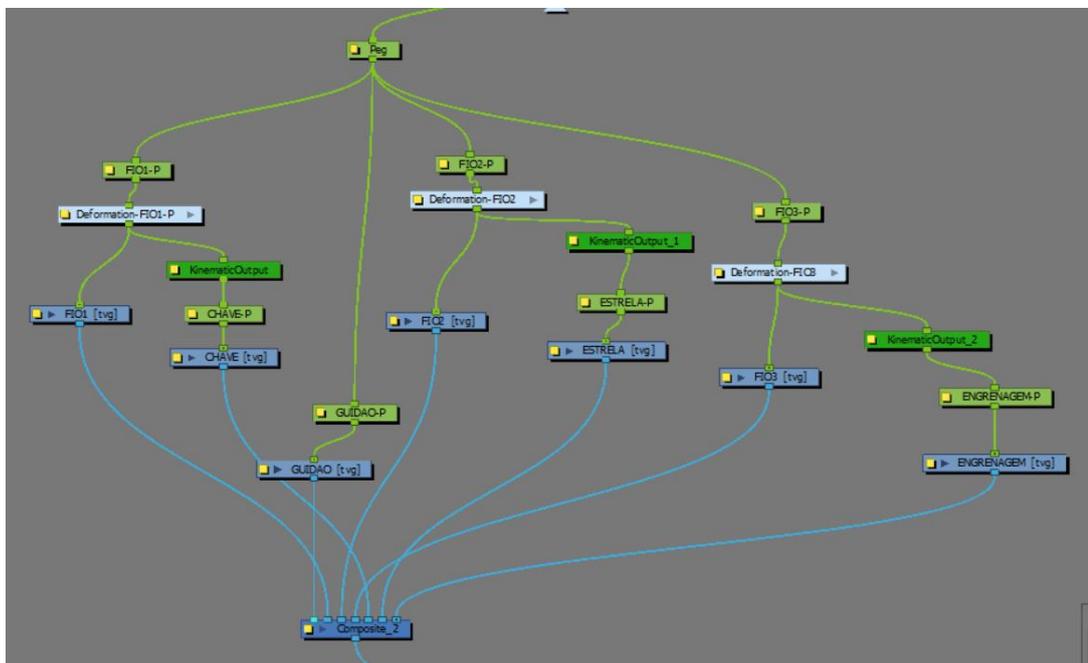


Fonte: A autora

O chaveiro precisou ser dividido em elementos, pois teria uma cena de *close-up* na qual o chaveiro apareceria se movimentando. Cada pingente fora separado em uma *layer* diferente e sendo plugado a uma *peg*, assim como os fios que são conectados ao guidão da moto.

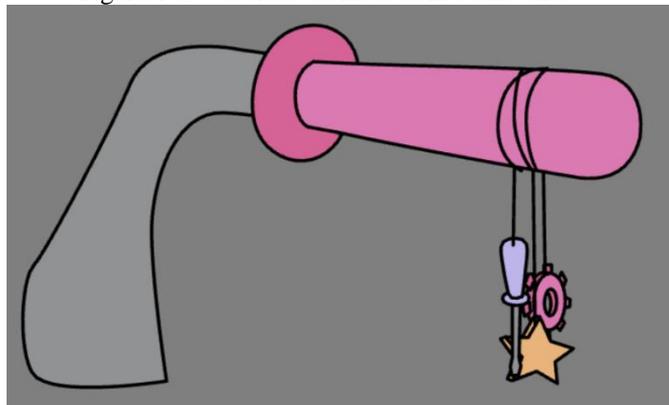
Nos fios de cada pingente, é inserido o deformador de curvas, descritos na seção [2.4.11](#), para reproduzir os movimentos mostrados no *animatic*. Para que os pingentes sigam o caminho feito pelos fios, é plugada à saída de cada grupo de deformação dos fios, o *node Kinematic\_Output*, citado na seção [2.4.12](#) que por sua vez, tem sua saída conectada às *pegs* dos pingentes. Por fim é inserida uma *peg* geral que une cada fio.

Figura 142 – Rig do chaveiro da moto



Fonte: A autora

Figura 143 – Guidão da moto com o chaveiro

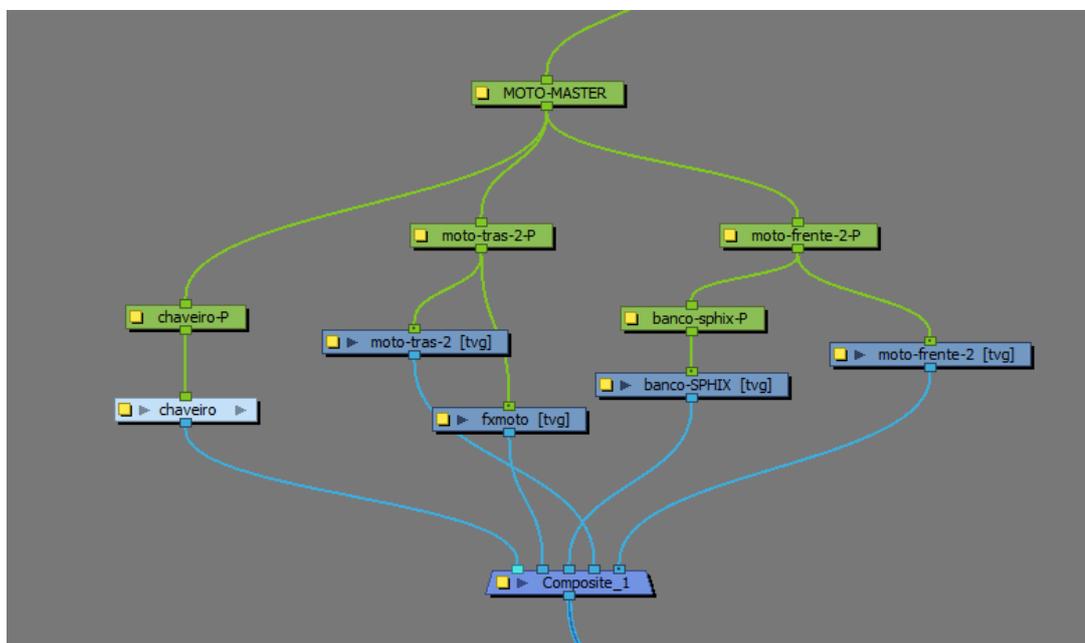


Fonte: A autora

A visão da traseira da moto também foi pensada e dividida em duas partes principais, a parte de trás do banco de Lizie, com suas turbinas traseiras, e a parte da frente com o painel, guidão e banco de Sphix, este último não é possível ser visto devido a perspectiva que foi desenhada.

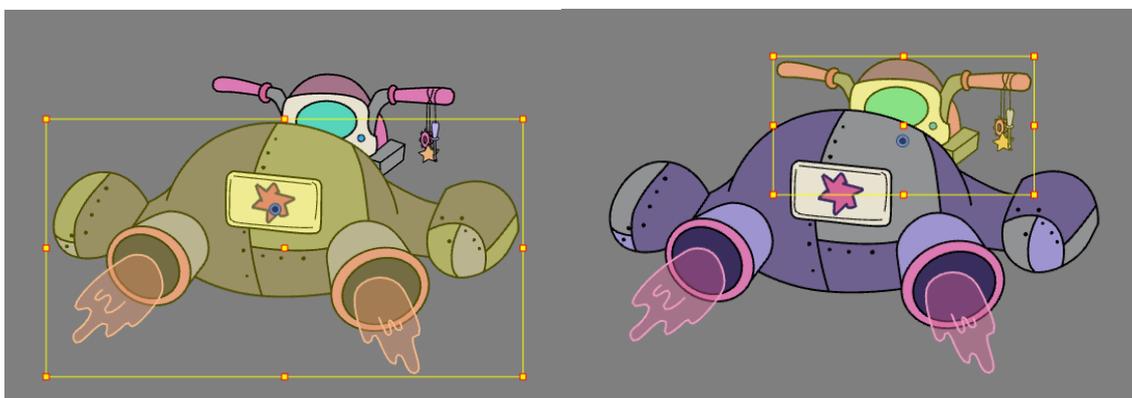
A *peg* geral da moto se conecta ao chaveiro que se mantém o mesmo, apenas invertido o lado, a traseira do banco da protagonista e a parte da frente.

Figura 144 – *Rig* da parte traseira da moto



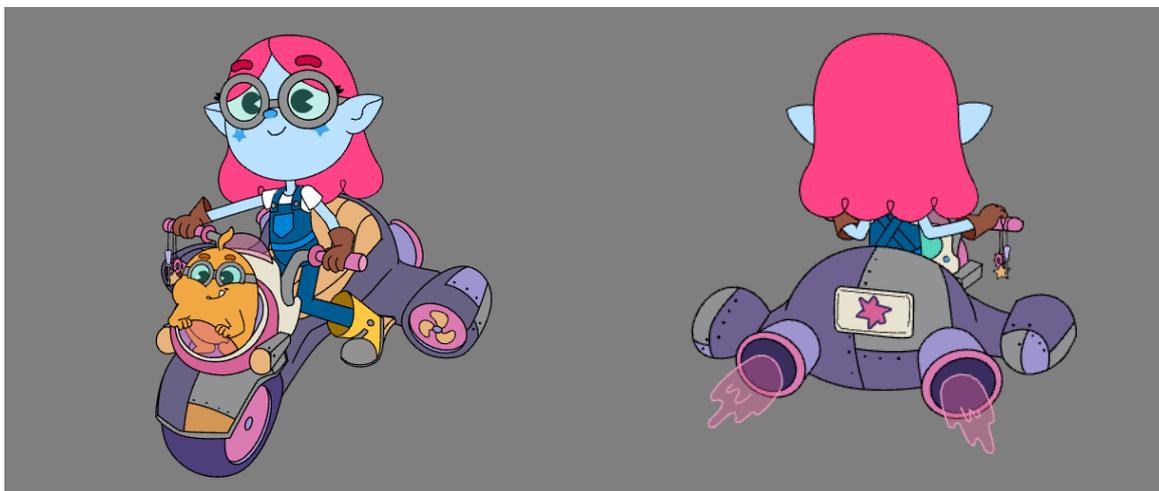
Fonte: A autora

Figura 145 – Divisão da parte de trás da moto em *pegs*



Fonte: A autora

Figura 146 – Prop da moto com os personagens



Fonte: A autora

## 4 RESULTADOS

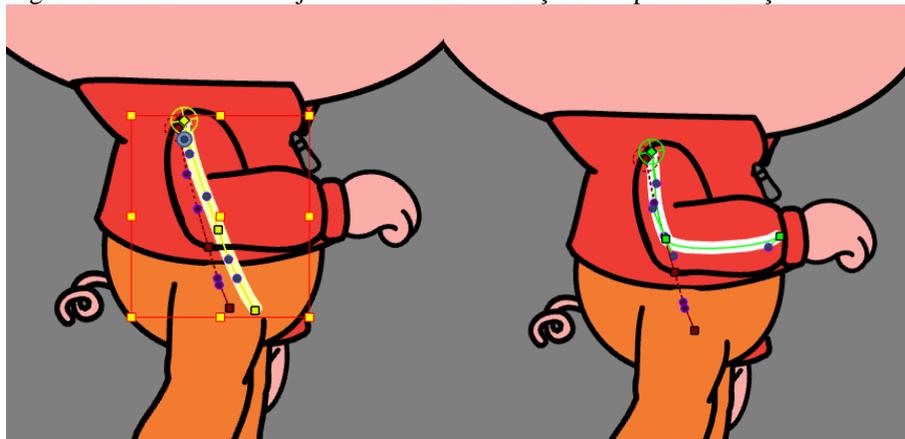
Em uma produção de animação *cutout*, onde o prazo de entrega é relativamente curto, é essencial personagens com *designs* simples, sem nada muito complexo e rebuscado, e um *rig* bem estruturado, que funcione para todas as poses e ações que o roteiro pede.

Durante o processo de desenvolvimento deste projeto, a parte de redefinição de estilo dos personagens já existentes foi a etapa que demandou menos tempo. Pois, ao retirar o projeto de 2016 da gaveta, a autora e sua parceira já tinham concluído todas as disciplinas do curso de Animação e expandido seus conhecimentos e habilidades, que na época não eram tão evoluídos. Portanto, já sabiam o que poderia ser modificado, tornando esta etapa concluída sem muita demora.

A parte de criação do *rig* dos personagens foi uma etapa extensa, que definitivamente demandou tempo da autora. Devido a pouca experiência em criação de *rig*, foi necessário tempo para pesquisar e estudar o software escolhido.

Após o processo de criação do *rig*, e início da etapa de animação percebeu-se uma certa dificuldade de animar as articulações dos braços do personagem Sr. Porco, devido ao design de sua jaqueta. Sendo necessário mudar para que fosse mais fácil de animar, sem perder muito tempo no acabamento de detalhes. Tal imprevisto não teria ocorrido, se a autora tivesse separado um tempo no cronograma de produção para testar os rigs antes da etapa de animação.

Figura 147 – Processo de ajuste de detalhe do braço na etapa de animação



Fonte: Luisa Costa

Figura 148 – Novo design da jaqueta sem precisa de ajuste para as articulações



Fonte: Luisa Costa

Contudo, os outros *rigs* funcionaram perfeitamente para o propósito das cenas do *teaser*, sem necessidade de alterações bruscas de *rig*, sendo necessário apenas, adicionar alguns desenhos a mais na biblioteca de mãos, pés, e bocas dos personagens. Coisa que é comum em qualquer produção de animação que trabalhe com troca de imagens.

A elaboração dos *props* foi feita seguindo os desenhos do *storyboard* como referencia para as poses. Definir uma pose da moto de frente que servisse para a maioria das cenas em que aparece no *teaser* foi algo que economizou muito tempo de produção.

Devido ao curto tempo de produção, seguindo o novo cronograma criado no início de 2021, o projeto conseguiu ser concluído com resultados satisfatórios para a autora e sua parceira, Bianca Galdino.

## 5 CONCLUSÃO

O presente projeto relatou as escolhas de design dos personagens já existentes, fazendo uma readaptação para um estilo mais conciso e apelativo para o público-alvo. Além de discorrer como foi o processo de preparação dos protagonistas da série “*Fix-It!*” para a etapa de *rigging*. Etapa esta que fora fundamental para o objetivo da dupla, Luisa Costa e Bianca L. Galdino que consistia em produzir um *teaser* para a série, mostrando para o público um pouco do universo e a apresentação dos personagens principais.

Devido à pandemia, em março de 2020 o andamento do projeto sofreu uma pausa, mas ainda eram feitas reuniões semanais pelo *Discord* para discussões relacionadas às partes ainda em aberto da história. As integrantes do projeto criaram um cronograma em início de 2021 para que o projeto pudesse ser finalizado. E mesmo com alguns imprevistos e o prazo curto, ocasionando na falta de teste do *rig*, por exemplo, o projeto foi finalizado com sucesso.

Portanto, a autora conclui que o desenvolvimento deste trabalho foi satisfatório, pois nunca havia feito todo o processo de *rig* de um personagem completo, do início ao fim. O que gerou conhecimento e portfólio para a autora.

## REFERÊNCIAS

BEANE, Andy. **3D Animation Essentials**. Indianapolis: John Wiley & Sons, Inc., 2012.

DOWLATABADI, Zahra; WINDER, Catherine. **Producing Animation**. 2ª Edição. Oxford: Focal Press, 2012.

JOHNSTON, Ollie; THOMAS, Frank. **The Illusion of Life: Disney Animation**. 1ª Edição. California: Disney Editions, 1995.

HARMONY, Toonboom. **About Art Layers**. Disponível em: <[https://docs.toonboom.com/help/harmony-16/advanced/paperless-animation/about-art-layer.html?Highlight=auto%20patch#drawingandanimating\\_4228305222\\_1169081](https://docs.toonboom.com/help/harmony-16/advanced/paperless-animation/about-art-layer.html?Highlight=auto%20patch#drawingandanimating_4228305222_1169081)>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **About Bone Deformations**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-bone-deformation.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **About Curve Deformations**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-curve-deformation.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **About Envelope Deformations**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-envelope-deformation.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **About Free Form Deformations**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-free-form-deformation.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **About Game Bone Deformations**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/deformation/about-game-bone-deformation.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Auto-Patch Node**. Disponível em: <<https://docs.toonboom.com/help/harmony-20/premium/reference/node/filter/auto-patch-node.html?Highlight=auto%20patch%20node>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Composite Node**. Disponível em: <<https://docs.toonboom.com/help/harmony-14/premium/reference/node/combine/composite-node.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Cutter Node**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/node/combine/cutter-node.html?Highlight=cutter>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Drawing Substitutions View**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/view/drawing-picker-view.html>>. Acesso em: 26 abr. 2021.

HARMONY, Toonboom. **Element / Drawing Node**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/node/generator/element-node.html?Highlight=draweing%20node%20>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Kinematic Output Node**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/node/deformation/kinematic-output-node.html?Highlight=kinematic%20output>>. Acesso em: 26 abr. 2021.

HARMONY, Toonboom. **Library View**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/view/library-view.html?Highlight=library%20>>. Acesso em: 26 abr. 2021.

HARMONY, Toonboom. **Nodes**. Disponível em: <<https://docs.toonboom.com/help/harmony-14/premium/reference/node/node.html>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Overlay Layer Node**. Disponível em:

<<https://docs.toonboom.com/help/harmony-20/premium/reference/node/filter/overlay-layer-node.html?Highlight=overlay-layer%20node>>. Acesso em: 25 abr. 2021.

HARMONY, Toonboom. **Timeline View**. Disponível em:

<<https://docs.toonboom.com/help/harmony-14/premium/reference/view/timeline-view.html?Highlight=timeline>>. Acesso em: 26 abr. 2021.

SANDERS, Adrien-Luc. **Learn About Cutout Animation**. 2019. Disponível em:

<<https://www.lifewire.com/what-is-cutout-animation-140519>>. Acesso em: 26 abr. 2021.

WILLIAMS, Richard. **The Animators Survival Kit: a manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators**. Nova York: Farrar, Straus And Giroux, 2009.