



UNIVERSIDADE FEDERAL DO CEARÁ
INSTITUTO UNIVERSIDADE VIRTUAL – UFC VIRTUAL
CURSO DE BACHARELADO EM SISTEMAS E MÍDIAS DIGITAIS

FERNANDO DE ARAÚJO ALVES

PIPELINE DE CONSTRUÇÃO TÉCNICO ARTÍSTICA DE UM PERSONAGEM
TRIDIMENSIONAL PARA GAME EM 3ª PESSOA

FORTALEZA

2021

FERNANDO DE ARAÚJO ALVES

PIPELINE DE CONSTRUÇÃO TÉCNICO ARTÍSTICA DE UM PERSONAGEM
TRIDIMENSIONAL PARA GAME EM 3ª PESSOA

Relatório Técnico apresentado ao Curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito à obtenção do título de Bacharelado em Sistemas e Mídias Digitais. Área de concentração: Sistemas de Informação Multimídia.

Orientador: Dr. Adriano Anunciação Oliveira

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- A479p Alves, Fernando de Araújo.
Pipeline de construção técnico artística de um personagem tridimensional para game em 3ª pessoa /
Fernando de Araújo Alves. – 2021.
98 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual,
Curso de Sistemas e Mídias Digitais, Fortaleza, 2021.
Orientação: Prof. Dr. Adriano Anunciação Oliveira.
1. Desenvolvimento de personagem. 2. Jogos digitais. 3. Desenvolvimento de visualidade. 4. Fluxo de
produção. 5. Renderização Baseada em Física. I. Título.

CDD 302.23

FERNANDO DE ARAÚJO ALVES

PIPELINE DE CONSTRUÇÃO TÉCNICO ARTÍSTICA DE UM PERSONAGEM
TRIDIMENSIONAL PARA GAME EM 3ª PESSOA

Relatório Técnico apresentado ao Curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito à obtenção do título de Bacharelado em Sistemas e Mídias Digitais. Área de concentração: Sistemas de Informação Multimídia.

Aprovado em: ___/___/_____.

BANCA EXAMINADORA

Prof. Dr. Adriano Anunciação Oliveira
Universidade Federal do Ceará (UFC)

Prof. Dr. José Gilvan Rodrigues Maia
Universidade Federal do Ceará (UFC)

Prof. Dr. Natal Anacleto Chicca Junior
Universidade Federal do Ceará (UFC)

Aos meus pais, Genival e Lucinda, pelos sacrifícios que fizeram para que me tornasse quem sou.

AGRADECIMENTOS

Ao meu orientador, Prof. Adriano Anunciação, por todo o auxílio, soluções eficientíssimas e incentivo durante o desenvolvimento deste trabalho.

À minha família, pela paciência e apoio que sempre me deram, especialmente o meu irmão, Felipe Alves, por me fazer rir até nos momentos mais estressantes.

Aos professores que tive no SMD, que me deram as ferramentas para chegar até aqui.

À Camilo Cunha e Lucas Falcão, pela disponibilidade e pelas recomendações que me ajudaram em tomadas de decisão importantes para o projeto.

Aos amigos Alexandre Magno, Carmen Li Juy e Leonardo Sambonet, pelas conversas bem humoradas, desabafos e apoio emocional que me deram desde o começo do trabalho.

À todos os amigos que fiz no PET Computação UFC, que me ensinaram muito e com quem sempre posso contar, em especial Matheus Magalhães, pelas conversas sinceras que instigaram minha criatividade, e Débora Moura, que sempre tive como exemplo de determinação e profissionalismo.

À todos os amigos que fiz durante a graduação, com os quais dividi muitas alegrias e espero dividir muitas mais.

RESUMO

O desenvolvimento de um jogo digital envolve uma série de etapas e processos multidisciplinares, promovendo a integração do esforço e conhecimento de diversos profissionais da indústria do entretenimento, passando por ilustradores, animadores, modeladores 3D, programadores, dentre outros. Cada projeto precisa integrar esses profissionais e suas atividades de modo que a produção de um produto seja otimizada, evitando desperdício de recursos e refações das atividades envolvidas. A estrutura de planejamento, organização e ajustes das atividades e processos desenvolvidos por cada um desses profissionais para cada etapa de um projeto é chamado de *pipeline* de produção. Essa estrutura é escalável e ajustável a uma infinidade de projetos na indústria de jogos, podendo ser utilizada para organização de projetos inteiros, etapas de produção ou mesmo na criação de conteúdos específicos de um projeto, como cenários, objetos ou mesmo personagens de um jogo. Sob essa perspectiva, o presente trabalho descreve os processos de ideação e produção, dentro de uma pipeline de produção, de uma personagem para jogos em terceira pessoa de gênero ação e aventura, que teve como resultado um modelo tridimensional adaptado para utilização na Unreal Engine 4 com texturas desenvolvidas em uma lógica de representação para Renderização Baseada em Física (PBR).

Palavras-chave: Desenvolvimento de personagem, Jogos digitais, Desenvolvimento de visualidade, Fluxo de produção, Renderização Baseada em Física.

ABSTRACT

The development of videogames wrap a series of multidisciplinary steps and processes, promoting the integration of the effort and knowledge of many professionals in the entertainment industry, including illustrators, animators, 3D modelers, programmers, and much more. Each project needs to integrate these professionals and their activities to optimize a product development, avoiding wasting resources and reworking of the activities involved. The structure for planning, organizing and adjusting the activities and processes developed by each of these professionals for each stage of a project is called production pipeline. This structure is scalable and adjustable to a multitude of projects in the game industrie, and can be used to organize entire projects, production stages or even in the creation of specific contents in a project, such as scenarios, objects or even characters of a game. From this perspective, this report describes the processes of conception and production, within a production pipeline, of a three-dimensional character for a third-person action-adventure game, which resulted in a three-dimensional model adapted for use in Unreal Engine 4 with textures developed in a representation logic of Physically Based Rendering (PBR).

Key Words: Character design, Videogame, Look dev, Production pipeline, Physically Based Rendering.

LISTA DE FIGURAS

Figura 1 –	Pipeline de produção da personagem Joana	18
Figura 2 –	Amplificação através da simplificação	20
Figura 3 –	Representação visual realista de personagem em jogo digital	22
Figura 4 –	Representação de personagem da década de 1990	23
Figura 5 –	Texturas de personagem tridimensional para game	24
Figura 6 –	Exemplo de uma pipeline de produção para um projeto de VFX	25
Figura 7 –	Ilustração conceitual de personagem	26
Figura 8 –	Etapas de produção de objeto tridimensional	28
Figura 9 –	Mapeamento UV do modelo tridimensional de personagem	30
Figura 10 –	Mapas de material no padrão PBR	31
Figura 11 –	Foto de um vaqueiro	34
Figura 12 –	Painel semântico com referências de roupas de vaqueiro	35
Figura 13 –	Um dos painéis semânticos utilizados no projeto	37
Figura 14 –	<i>Model sheet</i> da personagem Joana	38
Figura 15 –	Blocagem do modelo tridimensional da personagem	39
Figura 16 –	Modelagem não-destrutiva da malha do cabelo da personagem	40
Figura 17 –	Uma das primeiras versões da modelagem do corpo	41
Figura 18 –	Proposta final de modelagem do rosto da personagem	42
Figura 19 –	Modelagem não-destrutiva da malha do cinto da personagem	43
Figura 20 –	Modelo <i>high poly</i> da roupa da personagem	44
Figura 21 –	Modelagem não destrutiva do chapéu da personagem	44
Figura 22 –	Modelagem <i>high poly</i> final da personagem	45
Figura 23 –	Modelos <i>high poly</i> e <i>low poly</i> das roupas da personagem	47
Figura 24 –	<i>Rig</i> gerado pelo <i>addon</i> Rigify	48

Figura 25 – Animação feita através do Rigify	48
Figura 26 – Problemas na atribuição de pesos das luvas	50
Figura 27 – Antes e depois do ajuste de pesos das sobrancelhas e boca	50
Figura 28 – Pintura de vértices do topo do chapéu	50
Figura 29 – Pintura de vértices do cordão do chapéu	51
Figura 30 – Antes e depois do ajuste de pesos da região do pescoço	51
Figura 31 – Defeitos na topologia do rosto	52
Figura 32 – Ajustes de topologia no queixo e pescoço	53
Figura 33 – Ajuste de topologia no rosto	53
Figura 34 – Áreas da topologia da roupa que não sofreram alterações	53
Figura 35 – Marcação do mapeamento UV das roupas da personagem	55
Figura 36 – <i>Layout</i> de UDIM do modelo da personagem	55
Figura 37 – Pintura de vértices da malha <i>high poly</i> da personagem	57
Figura 38 – Mapa de ID gerado a partir da pintura de vértices da malha <i>high poly</i>	58
Figura 39 – Mapa de Oclusão Ambiente gerado a partir do contato entre geometrias na malha <i>high poly</i> da personagem	58
Figura 40 – Mapa de Normal gerado a partir da superfície da malha <i>high poly</i>	59
Figura 41 – Artefatos gerados pelo <i>bake</i> na boca da personagem	60
Figura 42 – Artefatos gerados pelo <i>bake</i> no pescoço da personagem	60
Figura 43 – Separação da malha para <i>bake</i> do modelo	61
Figura 44 – Mapas de <i>bake</i> gerados para o projeto	61
Figura 45 – Camadas e canais PBR atrelados ao material das luvas	63
Figura 46 – Painel semântico com referências para o material dos calçados	64
Figura 47 – Projeção esférica de uma textura sobre a malha das roupas	65
Figura 48 – Artefatos na divisão entre materiais	65
Figura 49 – Artefatos isolados em uma máscara	66

Figura 50 – Correção de artefatos das roupas	66
Figura 51 – Resultado final dos materiais das roupas	67
Figura 52 – Material base para o rosto da personagem	68
Figura 53 – Umas das máscaras para detalhamento da pele	68
Figura 54 – Ajustes nos canais de normal do material	69
Figura 55 – Ajustes nos canais de rugosidade do material	69
Figura 56 – Adição de queimaduras de sol na pele	70
Figura 57 – Adição de sardas na pele	70
Figura 58 – Comparativo entre pele sem e com adição de imperfeições	71
Figura 59 – Material base da “armadura” de couro	72
Figura 60 – Textura de couro aplicada à superfície da “armadura”	73
Figura 61 – Desgaste leve aplicado à superfície da “armadura”	73
Figura 62 – Desgaste acentuado aplicado à superfície da “armadura”	74
Figura 63 – Máscara usada para o desgaste leve da “armadura”	74
Figura 64 – Máscaras usadas para o desgaste pesado da “armadura”	75
Figura 65 – Resultado final do material de couro da “armadura”	75
Figura 66 – Material PBR do chapéu da personagem	76
Figura 67 – Cílios da personagem, exibidos com canal de transparência	77
Figura 68 – <i>Shader</i> PBR dos cílios na Unreal Engine	78
Figura 69 – Exemplos de ciclos de animação para validação da malha da personagem	80
Figura 70 – Malha e rig do Mannequin, personagem padrão da Unreal Engine	81
Figura 71 – Rig gerado pelo Auto-Rig Pro aplicado à personagem	82
Figura 72 – Personagem exportada para projeto padrão da Unreal Engine	83
Figura 73 – Personagem utilizando animações do projeto da Unreal Engine	83
Figura 74 – Comparação entre os modelos do Mannequin e Joana	84
Figura 75 – Comparação entre câmera padrão e os ajustes feitos	84

Figura 76 – Renderização em tempo real do material PBR da personagem	86
Figura 77 – Exibição do canal PBR de cor	87
Figura 78 – Exibição do canal PBR de normal	87
Figura 79 – Exibição do canal PBR de rugosidade	88
Figura 80 – Exibição dos <i>bones</i> da personagem	88
Figura 81 – Comparação entre visualizações da personagem na Unreal	89
Figura 82 – Joana no ambiente virtual do Memorial Interativo	90
Figura 83 – Joana avançando sobre um obstáculo	91
Figura 84 – Joana explorando o percurso do Memorial Interativo	91
Figura 85 – Joana em frente a um expositor da etapa conceitual do projeto	92
Figura 86 – Joana explorando a sala da etapa de modelagem tridimensional	93
Figura 87 – Expositores com assets da etapa de look dev	93
Figura 88 – Exemplo de animação utilizada no Memorial Interativo	94

LISTA DE ABREVIATURAS E SIGLAS

3D	<i>Tridimensional</i>
PC	<i>Personal computer</i>
PBR	<i>Physically Based Rendering</i>
<i>Look dev</i>	<i>Look development</i>
UE4	<i>Unreal Engine 4</i>
VFX	<i>Visual effects</i>
<i>High poly</i>	<i>High polygon modeling</i>
<i>Low poly</i>	<i>Low polygon modeling</i>
FBX	<i>Autodesk Filmbox</i>
RGB	<i>Red Green Blue</i>
HDRI	<i>High Dynamic Range Image</i>
UV	<i>UV Map</i>
UDIM	<i>U-Dimension</i>
PNG	<i>Portable Network Graphics</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Pipeline de produção	17
1.2	Objetivos	18
2	REFERENCIAL TEÓRICO	20
2.1	Desenvolvimento conceitual de personagem	20
2.2	Desenvolvimento técnico e visual de personagem	21
2.3	Pipeline de produção para jogos	24
2.3.1	<i>Conceituação visual</i>	25
2.3.2	<i>Modelagem tridimensional</i>	26
2.3.3	<i>Rigging</i>	28
2.3.4	<i>Look dev</i>	29
2.3.5	<i>Animação</i>	31
3	REALIZAÇÃO	33
3.1	Conceituação Visual	34
3.2	Modelagem tridimensional	38
3.2.1	<i>Escultura do corpo</i>	39
3.2.2	<i>Modelagem das roupas</i>	42
3.2.3	<i>Retopologia do modelo</i>	45
3.3	Rigging	47
3.3.1	<i>Ajustes de pesos dos vértices</i>	48
3.3.2	<i>Ajustes de topologia</i>	51
3.4	Look dev	54
3.4.1	<i>Mapeamento UV</i>	54
3.4.2	<i>Baking de texturas</i>	56

3.4.3	<i>Shaders PBR e pintura 3D</i>	62
3.5	Animação	78
3.5.1	<i>Workflow no Blender com Rigify</i>	79
3.5.2	<i>Workflow na Unreal Engine</i>	80
4	PRODUTO FINAL	85
4.1	Personagem Joana	85
4.2	Memorial Interativo	90
5	CONSIDERAÇÕES FINAIS	95
	REFERÊNCIAS	99

1. INTRODUÇÃO

Em obras de ficção, os personagens normalmente são os agentes fundamentais que movem uma narrativa. Tanto suas características quanto as formas de interação desses personagens com seu entorno são, na maioria das vezes, os principais responsáveis pelo engajamento emocional da audiência em uma estória (SMITH, 1995). Em termos narrativos, os personagens criados pelo autor precisam expressar as características arquetípicas demandadas pela história para facilitar a interpretação da audiência com a narrativa e guiar a expectativa do público (VOGLER; MACHADO, 2006).

Em função da necessidade de expressão visual desses arquétipos e diante do avanço expressivo das potencialidades da computação gráfica, nos últimos anos o desenvolvimento visual de personagens tem ganhado mais notoriedade na indústria de entretenimento, seja para aplicação em filmes ou jogos digitais. O avanço da capacidade gráfica acessível ao grande público em computadores pessoais (PCs) e consoles domésticos vêm possibilitando convergências nos modos de produção de indústrias distintas no mercado de entretenimento, que por sua vez possibilitam na indústria de jogos digitais o desenvolvimento de personagens cada vez mais expressivos visualmente. Nessa indústria, tanto grandes estúdios como produtoras independentes vêm aperfeiçoando técnicas e processos de desenvolvimento da representação visual e interativa (*e.g.* animação, simulação de física de objetos etc.) dos personagens de seus jogos, em abordagens cada vez mais alinhadas com os processos e exigências de produção já amplamente empregados no campo de efeitos especiais para cinema e vídeo (DUNLOP, 2014).

É possível afirmar que, apesar das suas especificidades persistirem, as indústrias do cinema e de jogos – especialmente os de gêneros narrativos que mais utilizam efeitos digitais em sua expressão visual –, atualmente compartilham fluxos de produção semelhantes. Além disso, ambas indústrias empregam os mesmos perfis de profissionais técnicos e artistas para criar representações de universos ficcionais detalhados e dos personagens complexos que os habitam. A exemplo desses perfis temos artistas conceituais, ilustradores, animadores digitais, modeladores 3D, programadores, dentre outros perfis técnicos habilitados a trabalhar em ambas as indústrias.

Um dos eixos da convergência entre os modos de produção de ambas indústrias de jogos e cinema é a emergência recente de uma série de princípios de representação visual computacional conhecidos como *Physically Based Rendering* (PBR) (renderização baseada

em física). Esses princípios alinham as tecnologias de geração computacional de imagens (*i.e. render*) offline (para cinema e vídeo) e online (para games) em torno da emulação virtual hiper-realista de como as superfícies reais interagem fisicamente com a luz no mundo natural (MCDERMOT, 2018).

1.1. Pipeline de produção

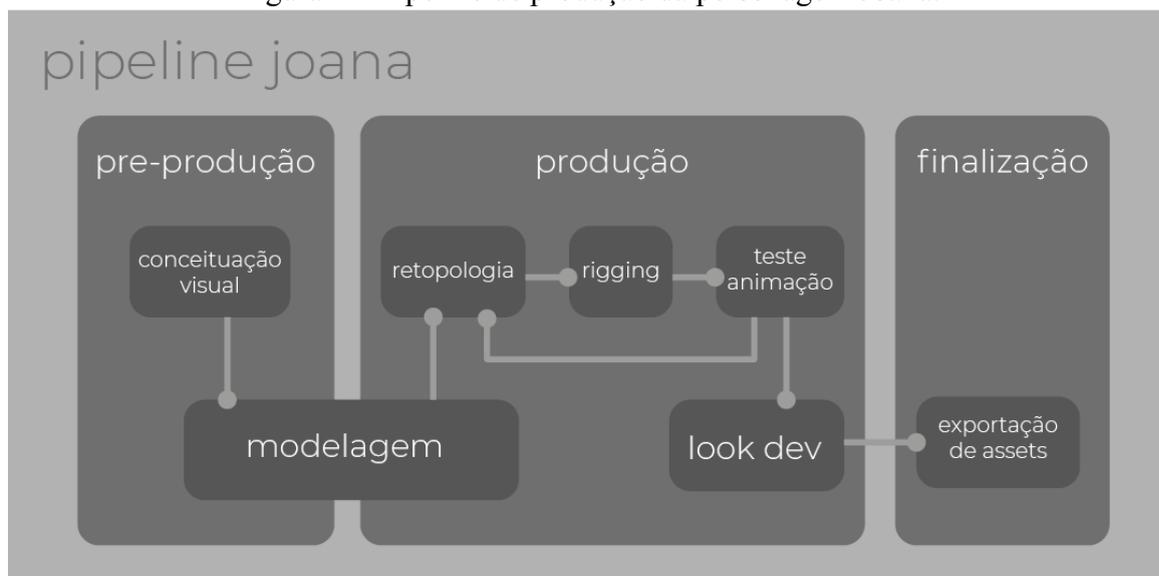
A abordagem PBR juntamente com outros novos paradigmas vêm promovendo a unificação de antigos processos de desenvolvimento técnico artístico, constituindo um novo fluxo de produção (*i.e. workflow*) da representação visual de personagens que passa a integrar processos de produção idênticos em games e produtos audiovisuais tradicionais.

Esse fluxo de produção é composto de inúmeras etapas, que podem ser agrupadas de modo abrangente em (1) conceitualização visual, (2) modelagem, (3) aplicação de estruturas animáveis ao modelo (*rigging*), (4) desenvolvimento de visualidade (*i.e. look dev*, que engloba finalização de texturas, materiais e iluminação) e (5) animação de movimentos e interações do personagem (DUNLOP, 2014).

Para este projeto o fluxo de produção utilizado foi definido a partir de adaptação de pipelines discutidas por Renee Dunlop, de modo que ela orientasse a produção de uma personagem 3D chamada Joana. A pipeline proposta (Figura 1) busca organizar apenas o conjunto de atividades necessárias para a produção de Joana dentro de uma divisão de etapas que contempla a pré-produção, produção e finalização de assets para jogos digitais, tendo como alvo a Unreal Engine 4 (UE4), que é uma das plataforma de desenvolvimento de jogos digitais com suporte para PCs e consoles da atual geração de videogames.

A Figura 1 mostra o fluxo de produção da pipeline adaptada para o desenvolvimento da personagem Joana, dividindo as etapas em pré-produção, produção, e finalização.

Figura 1 – Pipeline de produção da personagem Joana.



Fonte: Elaborado pelo autor.

A pipeline proposta para este trabalho foi planejada tendo em mente um fluxo de produção individual e linear, visto que as atividades desenvolvidas foram executadas por apenas uma pessoa. Além disso, durante a produção do trabalho foi adotado um processo iterativo entre as atividades de retopologia, rigging e teste de animação, o que permitiu maior liberdade para ajustes no modelo tridimensional da personagem desenvolvida.

1.2. Objetivos

A partir desse ponto de vista convergente entre processos e fluxos de trabalho dentro da indústria do entretenimento, o desenvolvimento de um personagem para jogos digitais acaba se beneficiando da expressão narrativa do cinema, explorando, como produto de mídia, a representação visual das características de um personagem. Nesse contexto, o presente projeto se propôs a explorar as potencialidades e os limites das etapas do desenvolvimento visual de um personagens aplicados à indústria de jogos digitais.

Levando em conta também o contexto de uma produção com limite de recursos materiais e profissionais, tendo como principais ferramentas de produção um computador pessoal com dois sistemas operacionais – um de base Linux e outro Windows – e softwares gratuitos ou com licença educacional, o projeto teve bastante limitação de recursos.

Tendo em vista essas limitações, foram definidos como objetivos específicos: (1) definir um conjunto de características visuais relevantes, que sejam capazes de expressar elementos narrativos em torno do personagem; (2) reproduzir, de maneira realista, o conjunto

de características visuais do personagem utilizando técnicas de representação PBR; e por fim (3) desenvolver um modelo tridimensional para o personagem que fosse tecnicamente otimizado, que atinja os requisitos de um ambiente de desenvolvimento para um jogo digital na modalidade de 3ª pessoa, respeitando as potencialidades e limitações dos sistemas atuais.

Como forma de apresentação e teste de interação com a personagem, foi produzido um Memorial Interativo do trabalho desenvolvido, consistindo em uma aplicação para computador em que o usuário assume o controle da personagem, a fim de experimentar as possibilidades de movimentação e de interação com os obstáculos de um cenário de jogo tridimensional em terceira pessoa.

2. REFERENCIAL TEÓRICO

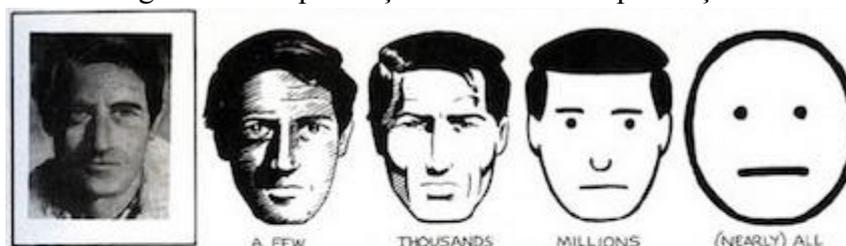
Neste capítulo serão apresentados os conceitos necessários para a compreensão das etapas de produção abordadas no trabalho. Serão abordados os conceitos de desenvolvimento técnico visual e animação de personagem e a *pipeline* da produção de ativos (*i.e. assets*) para jogos.

2.1. Desenvolvimento conceitual de personagem

Na arte sequencial, precursora do cinema e dos jogos digitais, quando uma imagem tem seus detalhes retirados até que ela atinja o seu significado essencial, o artista consegue amplificar essa essência de um modo que a arte realista não consegue. A esse processo, é dado o nome de amplificação através da simplificação (MCCLLOUD, 1993). O mesmo conceito pode ser aplicado a qualquer tipo de imagem abstrata, incluindo personagens de games.

A Figura 1 mostra o processo de simplificação do rosto de um homem, cujas características principais podem ser amplificadas para representar cada vez mais indivíduos que guardam um certo conjunto de características básicas.

Figura 2 – Amplificação através da simplificação.



Fonte: What's Wrong With Being a Cartoon Character?¹

No que diz respeito ao estilo visual, os grandes estúdios de animação em suas produções para cinema, usam um conjunto de técnicas que, apesar de serem capazes de reproduzir o comportamento natural de fenômenos da natureza como luz, texturas e movimento, mostram seus personagens com uma composição visual cada vez mais caricata (VAN ROOIJ, 2019). Quando focados em representações hiper-realistas de personagens, as produções para cinema ainda não são capazes de criar um engajamento tão grande quanto nos filmes do gênero de animação, onde os personagens têm traços mais estilizados mesmo quando inseridos em um ambiente realista. Parece haver uma mistura exata de técnicas que

¹ Disponível em <<https://tinyurl.com/fh9c6fk2>> Acessado em 24/08/2021.

criam na audiência um pico de engajamento emocional com esses personagens (VAN ROOIJ, 2019).

Na indústria de jogos, especialmente em produções de estilo visual mais cartunesco onde e as expressões faciais dos personagens são em muitos casos pouco detalhadas, os jogos têm que expressar visualmente essas características de forma mais explícita do que em personagens desenvolvidos para outras mídias (VERKAAIK, 2015). As características visuais desses personagens, comunicadas através das cores, silhuetas, formas e acessórios, são propostas durante a etapa de desenvolvimento conceitual e definidas na etapa de desenvolvimento técnico, mas quando se trata de personagens jogáveis, elas só são consolidadas durante a etapa de animação, onde são trabalhados os ciclos de movimento de acordo com as interações com o ambiente do jogo ou através dos comandos do jogador (VALVE CORPORATION, 2017).

2.2. Desenvolvimento técnico e visual de personagem

O desenvolvimento de um personagem para jogos digitais normalmente passa por várias etapas bem definidas, principalmente na parte técnica que, diferente de animações, deve levar em conta algumas características específicas. Por exemplo, ao contrário de produções cinematográficas, os modelos tridimensionais em jogos precisam ser renderizados em tempo real. Além disso, embora os sistemas de computação gráfica estejam cada vez mais eficientes, eles têm limites para quantidade de polígonos que podem ser exibidos simultaneamente em tela sem que o processo de renderização comece a afetar o desempenho do sistema, causando travamentos e lentidão durante a execução de jogo. Portanto, é preciso que se construa modelos eficientes com uma baixa quantidade de polígonos (NOVAK, 2012).

A evolução das técnicas de *render* e a popularização tanto de ferramentas voltadas para indústria de efeitos visuais como de motores gráficos para jogos (*engines*), fez surgir nos últimos anos uma área de atuação inteiramente voltada para a representação visual mais realista de objetos tridimensionais, chamada de *look dev* (desenvolvimento de visualidade). O processo de *look dev* de um personagem diz respeito a toda representação visual capaz de torná-lo mais realista, mas possui foco específico em aspectos técnicos e artísticos de interação com o ambiente e o universo do jogo ou produto audiovisual em questão (THISTLEWOOD, 2019). Esses elementos podem ser representados por materiais, texturas e iluminação, aplicadas ao modelo tridimensional do personagem, evidenciando vestimentas,

cicatrizes, marcas de usos e desgaste de acessórios e armas que o personagem possua, ou quaisquer outros elementos que ajudem a compor visualmente esse personagem.

A Figura 3 mostra a representação visual fotorrealista de uma personagem de jogo digital, focada em simular de maneira fisicamente correta os materiais de pele, roupas e sujeira, de acordo com a iluminação do ambiente dentro do jogo.

Figura 3 – Representação visual realista de personagem em jogo digital



Fonte: site do jogo Hellblade: Senua's Sacrifice²

Na etapa de desenvolvimento visual de um personagem, o *look dev* pode ser conceituado às vezes com adições de etapas extras, abrangendo processos de *rigging* e animação, demandando um perfil profissional híbrido com conhecimento técnico e artístico que seja capaz de empregar conceitos que vão desde composição fotográfica até linguagens de programação (AMBROSE, 2019). Dentro do *workflow* de uma produção na indústria de entretenimento, a etapa de *look dev* é a responsável pela consolidação da proposta visual de um personagem (WHITEHURST, 2016).

Fica evidente que uma melhor representação visual de elementos em tela, causada pela evolução do poder de processamento dos sistemas atuais, ajuda a expandir as possibilidades narrativas a respeito dos personagens de um jogo. Personagens clássicos como Lara Croft do jogo de ação e exploração Tomb Raider, tem um desenvolvimento visual que possibilitou ao longo do tempo desdobramentos narrativos mais críveis e complexos, apesar de muitas vezes sutis. Nos primeiros jogos da série, lançados na década de 1990, não haviam artifícios

² Disponível em <<https://www.hellblade.com/gallery>> Acessado em 24/08/2021.

narrativos que abordassem a dificuldade física da personagem de superar os obstáculos propostos pelo *game*. Já nos jogos atuais, a representação visual mais realista da personagem permite que se acrescente uma camada narrativa extra, expressando essa dificuldade durante as sessões ativas de jogo. Essas melhorias na representação visual ajudam também a ampliar as nuances narrativas do próprio personagem, atribuindo-lhe mais complexidade ao mesmo tempo em que aumentam o engajamento emocional do jogador com ela, já que agora o game pode exibir texturas realistas dos ferimentos pelo corpo da protagonista quando ela toma dano durante a partida, fazendo o jogador temer pela sua segurança enquanto joga.

A Figura 4 mostra a personagem Lara Croft da franquia de jogos de ação Tomb Raider, representada com as limitações técnicas dos sistemas da época.

Figura 4 – Representação de personagem da década de 1990.



Fonte: Hard Core: A look at the original Tomb Raider games³

A Figura 5 mostra as texturas de um personagem tridimensional de jogo digital focado em estilo visual realista.

³ Disponível em <<https://tinyurl.com/9nrwm89m>> Acessado em 24/08/2021.

Figura 5 – Texturas de personagem tridimensional para game.



Fonte: There are Tomb Raiders and there are Tomb Raiders⁴

2.3. Pipeline de produção para jogos

Durante a realização de um produto audiovisual, é comum que as etapas de produção sejam divididas em 3 partes principais durante o projeto, sendo elas: pré-produção, produção e pós-produção (ou finalização no caso de jogos digitais) (DUNLOP, 2014). Cada uma dessas etapas são organizadas em sub etapas menores de acordo com o produto a ser desenvolvido, onde são aplicados os esforços de uma série de profissionais com perfis específicos em cada tipo de trabalho.

A produção de jogos possui várias semelhanças com os processos de produção para cinema e vídeo, especialmente no que diz respeito ao desenvolvimento de *assets*, que são artefatos que representam componentes específicos de um produto digital. Os *assets* podem ser trechos de áudio, imagens e texturas, modelos tridimensionais, dentre outros artefatos que representam uma parte do projeto, como um personagem ou objeto de cenário.

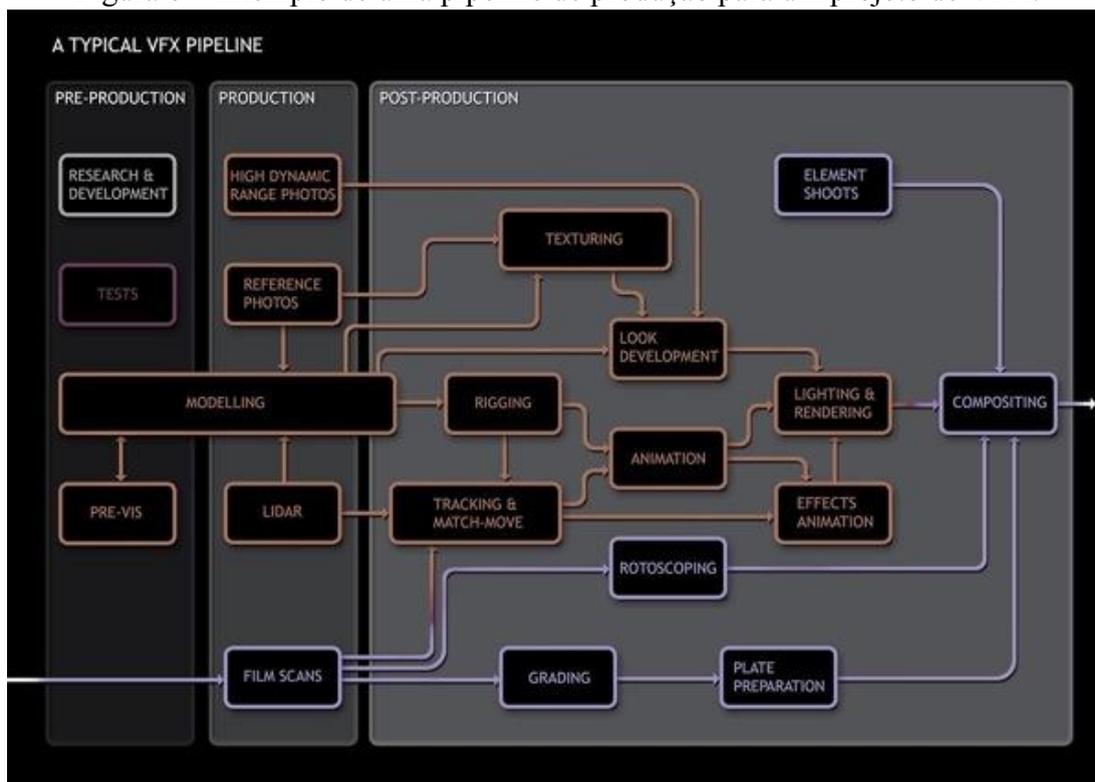
Os processos de criação e desenvolvimento de *assets* estão distribuídos entre as etapas de pré-produção e produção de um projeto, sendo a primeira focada em planejamento e ideação, e a segunda focada na execução. Ou seja, a etapa de produção é focada na criação de conteúdo (DUNLOP, 2014). Durante o produção, existem várias etapas específicas para cada tipo de *asset*, exigindo adaptações na *pipeline* de produção que podem abranger vários

⁴ Disponível em <<https://tinyurl.com/y3kub2ux>> Acessado em 24/08/2021.

processos específicos que se alternam e se sobrepõem de acordo com as particularidades de cada tipo de produção (Figura 6).

A Figura 6 mostra as etapas gerais e sub etapas dentro de uma pipeline de produção adaptada para um projeto de VFX.

Figura 6 – Exemplo de uma pipeline de produção para um projeto de VFX.



Fonte: site de Andrew Whitehurst⁵

No presente trabalho, focamos apenas nas etapas necessárias para a produção de um personagem para jogo tridimensional em 3ª pessoa, abrangendo as sub etapas de (1) conceituação visual; (2) modelagem tridimensional; (3) *Rigging* (aplicação de estruturas animáveis ao modelo); (4) *Look dev*; e (5) Animação, que serão apresentados a seguir. Essas etapas serão detalhadas a seguir.

2.3.1. *Conceituação visual*

Nesta etapa são concebidas e definidas as características visuais de personagens e cenários de acordo com os requisitos de um projeto. Nela são produzidas ilustrações conceituais (*i.e. concept art*) que expressam as características imagéticas dos assets. Essas

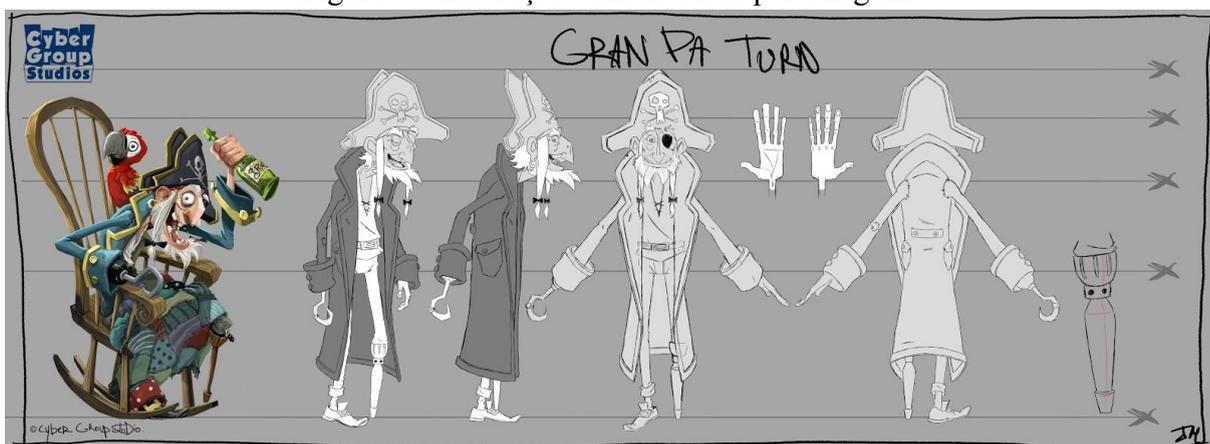
⁵ Disponível em <<http://www.andrew-whitehurst.net/pipeline.html>> Acessado em 24/08/2021.

ilustrações são usadas para orientar o processo de desenvolvimento técnico e artístico compostos pelas etapas seguintes (NOVAK, 2012).

As ilustrações conceituais serão produzidas com uso de técnicas diversas, a depender do artista, e tem o objetivo de testar estilos visuais, cores, formas, proporções de personagens e objetos e meios de interação destes assets com os cenários do jogo, como movimentos e ações específicas. As ilustrações usadas para as etapas seguintes de desenvolvimento devem mostrar as proporções do personagem, sua relação de escala com outros elementos do jogo, vestimentas que ele usa, objetos relacionados à sua caracterização ou utilizados por ele durante o jogo, além de estudo de expressões faciais que serão exibidas, sequência de ilustrações que mostra o personagem de corpo inteiro em pose neutra em ângulos diversos (*i.e. turn around*) e poses específicas que ajudem a expressar a sua personalidade. Também podem ser definidos estudos iniciais de texturas que devem ajudar a orientar o estilo visual geral do game.

A Figura 7 mostra o *turnaround* de um personagem para produção audiovisual, que normalmente o exibe em vistas frontal, posterior, lateral e em 3/4, que seria uma ou mais vistas em rotação de 45° com relação ao eixo vertical do personagem, partindo da vista frontal.

Figura 7 – Ilustração conceitual de personagem.



Fonte: The pirates next door- CyberGroupStuio⁶

2.3.2. Modelagem tridimensional

Um modelo tridimensional digital (*i.e.* modelo 3D), também chamado de malha (*i.e. mesh*), representa um conjunto de polígonos que dão a forma geométrica de um objeto tridimensional (LEVER, 2002). Os modelos 3D são desenvolvidos em softwares de criação

⁶ Disponível em <<https://www.artstation.com/artwork/6aOr1V>> Acessado em 24/08/201.

específicos como Blender⁷ e 3D Studio Max⁸, e podem ser feitos, mas não exclusivamente, em três etapas, compostas por (1) blocagem, (2) escultura digital (*i.e. sculpt*) e (3) retopologia.

No processo de blocagem, normalmente são utilizados primitivas 3D (*e.g.* cubos, esferas, cilindros, entre outras formas tridimensionais com representação de volume) para descrever de forma proporcional de cada parte de um objeto ou corpo de um personagem, como cabeça, dorso e membros, afim de montar uma figura tridimensional básica das formas que o compõem. No processo de *sculpt* são refinados os detalhes de um objeto a partir da mesclagem dos primitivos usados na blocagem, gerando assim um modelo com uma grande densidade de polígonos (*i.e. high poly*) e portanto maior possibilidade de representação de detalhes como amassados, ranhuras, cicatrizes, imperfeições de pele ou detalhes de vestimenta. Esse processo é feito com ferramentas gráficas que simulam o processo de escultura em argila, funcionando com adição e subtração de volume diretamente na malha do objeto tridimensional. Após a criação dessa escultura, é executado o processo de retopologia, que consiste na remodelagem da malha de um objeto ou personagem para criação de um modelo tridimensional mais otimizado com uma menor densidade de polígonos (*i.e. low poly*) e que possa ser animado e renderizado com uso de menos recursos computacionais (GUEVARRA, 2020).

Durante a etapa de modelagem, o modelo *high poly* é usado apenas como base para o modelo *low poly*, que é o modelo final do personagem e que de fato será exibido durante o jogo. Entretanto, ele não será descartado, pois servirá de base para as texturas produzidas mais à adiante no *workflow*, em um processo chamado *baking* de texturas, que será explicado mais adiante na seção 3.4.2.

A Figura 8 exibe quatro estágios da produção de um objeto tridimensional no *workflow* de produção de um jogo. Da direita para a esquerda: A primeira coluna representa o objeto em etapa de (1) modelagem *high poly*, a segunda coluna representa o (2) modelo *low poly*, a terceira representa o mesmo modelo *low poly*, mas com (3) aplicação de texturas geradas a partir do modelo *high poly* (*i.e. baking*), e a quarta coluna é o (4) objeto final, com modelo *low poly* e material acabado.

⁷ Disponível em <<https://www.blender.org/>> Acessado em 22/09/2021.

⁸ Disponível em <<https://www.autodesk.com.br/products/3ds-max/overview>> Acessado em 22/09/2021.

Figura 8 – Etapas de produção de objeto tridimensional.



Fonte: Props Gallery – Emilio Santoyo⁹

2.3.3. Rigging

Em animação digital, o *rig* representa uma estrutura de controladores que simulam o esqueleto de um personagem, ou qualquer outro tipo de mecanismo articulado para movimentação de objetos. O *rig* em si é formado por estruturas menores chamadas de *bones*, que representam as articulações (*i.e. joints*), e o processo de criação e adequação dessas estruturas ao modelo 3D é chamado de *rigging*.

Durante a etapa de *rigging* de um personagem, é criada uma estrutura de controladores para cada parte do seu corpo, que são responsáveis por movimentar os *bones*, e que por sua vez serão responsáveis por distorcer e animar cada parte do corpo do personagem através da malha *low poly* desenvolvida. Também é nessa etapa que, através dos controladores, os movimentos e expressões faciais do personagem são testados para identificar problemas como movimentos não naturais do personagem ou invasões de malha, que é quando os polígonos da malha acabam colidindo e se sobrepondo em algum ponto da superfície do objeto. Por isso é importante o uso de referências anatômicas corretas e que respeitem as poses e proporções criadas nas artes conceituais do personagem.

⁹ Disponível em <<http://emiliosantoyo.com/props.html>> Acessado em 24/08/2021.

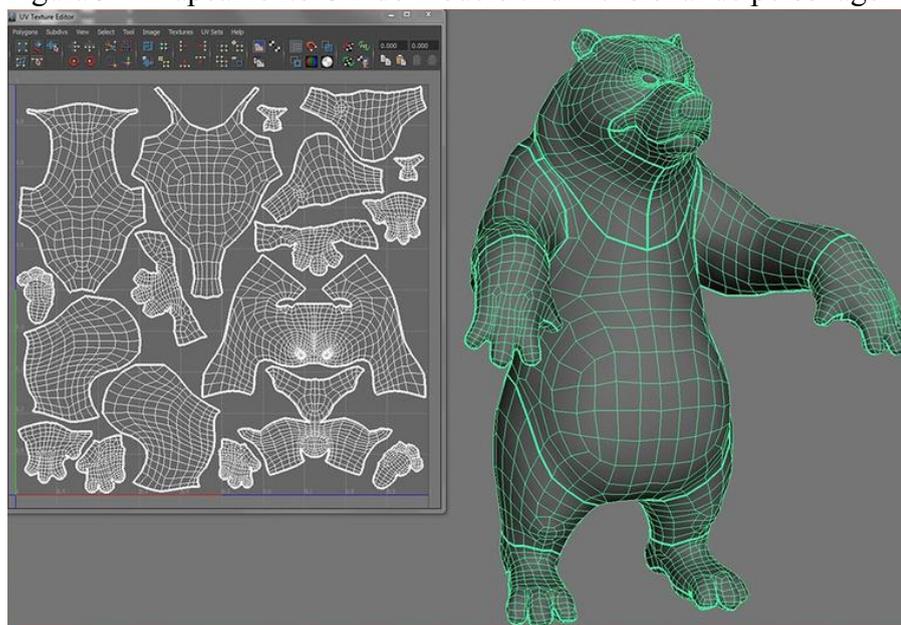
2.3.4. *Look dev*

O *look dev* ou desenvolvimento de visualidade é a etapa em que as características como texturas, cores e materiais definidos na conceituação visual do personagem serão transpostos para um ambiente digital tridimensional. O objetivo aqui é compor uma representação realista de como os objetos 3D vão se comportar dentro da *engine* do jogo, simulando a interação dos materiais sob a iluminação dos cenários. Os processos relacionados a essa etapa normalmente incluem (1) mapeamento UV, (2) criação de materiais e (3) iluminação.

O mapeamento UV é o processo em que a malha de um objeto é desdobrada (ou desenvolvida) e projetada sobre uma imagem 2D, em que cada área dessa imagem corresponde ao conjunto de polígonos que será sobreposto a ela (SORKINE; COHEN-OR, 2001). Dessa forma, os mapas criados (também chamados de texturas) sempre vão respeitar a mesma posição dos pixels em todas as imagens 2D geradas desde que sigam o mesmo mapeamento. Aqui, cada objeto 3D modelado é desdobrado a partir de uma marcação de arestas na malha chamada costura (*i.e. seam*) que serve para delimitar as bordas das áreas que serão mapeadas, também chamadas de ilhas de UV. As marcações aqui tem que ser feitas de forma a criar conjuntos de polígonos que sejam o mais planos possível, sem que haja sobreposição entre os vértices, caso contrário, um mesmo conjunto de pixels na imagem mapeada será exibido em mais de uma área diferentes na superfície do modelo 3D, causando efeitos indesejados no resultado final.

A Figura 9 mostra um modelo 3D com as marcações de abertura da malha destacadas para o mapeamento UV. A figura também exibe as ilhas de UV geradas, dispostas em um plano bidimensional.

Figura 9 – Mapeamento UV do modelo tridimensional de personagem.



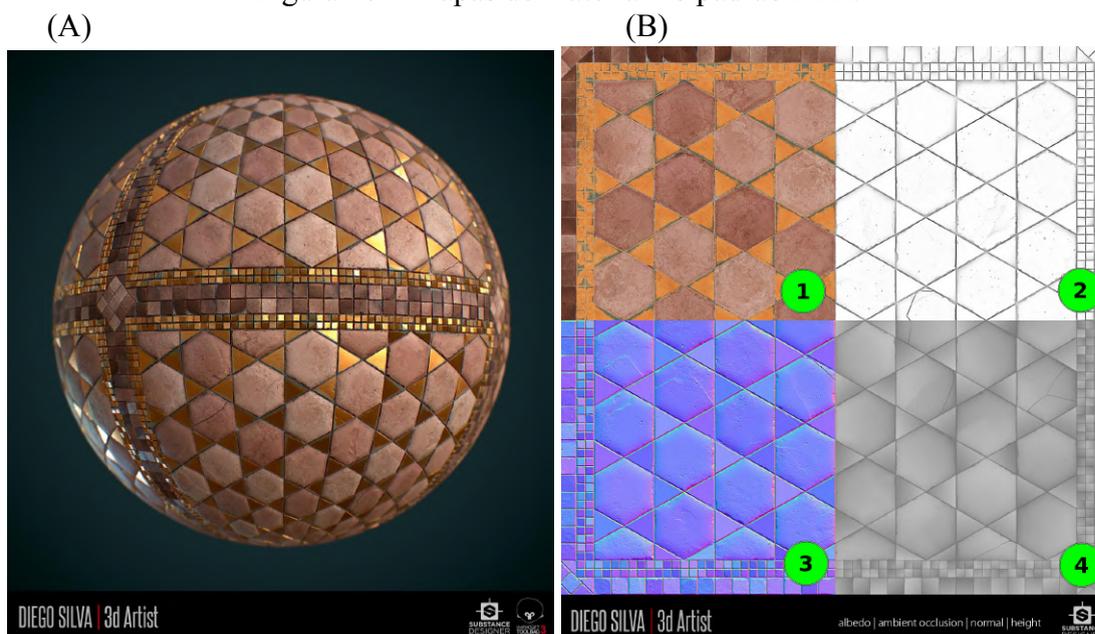
Fonte: The making of “Bear TV”¹⁰.

Com a malha devidamente mapeada, a criação de materiais será feita através de ferramentas de pintura 3D, em que texturas são criadas e combinadas em canais específicos para formar a representação digital dos materiais utilizados pela tecnologia PBR. A combinação desses canais é responsável por descrever as características físicas de cada superfície, evidenciando se ele é metálico ou não, se tem a superfície lisa ou rugosa, se reflete muita ou pouca luz, além de determinar sua cor base RGB (MCDERMOT, 2018). Outros canais também são usados para descrever características específicas de uma superfície, como o mapa de normal (*i.e. normal map*) que serve para representar detalhes de superfície de um objeto como arranhões e descascados ou cicatrizes, que não puderam ser modelados na malha *low poly*. Alguns desses mapas, como o de normal, são gerados a partir de um modelo *high poly* de um objeto, e mapeados para o seu modelo *low poly* correspondente, dessa forma, um modelo com baixa densidade de polígonos pode herdar uma quantidade considerável de detalhes do modelo original esculpido.

A Figura 10 mostra um material em padrão PBR, onde a imagem (A) mostra um material de ladrilhos aplicado à uma esfera em ambiente 3D, e a imagem (B) mostra os diferentes mapas utilizados no material para os canais de (1) cor (base color), (2) oclusão ambiente (AO), (3) normal e (4) altura (height).

¹⁰ Disponível em <<https://tinyurl.com/z4mf3usy>> Acessado em 24/08/2021.

Figura 10 – Mapas de material no padrão PBR.



Fonte: Ornament floor — Diego Silva¹¹

A etapa de iluminação geralmente é desenvolvida para validar os materiais utilizados nos objetos testando-os em situações de luminosidade e climas variados. O objetivo é simular a iluminação do jogo e averiguar se os objetos estão adequados à proposta visual do projeto. A iluminação pode ser feita a partir de pontos de luz posicionados em um ambiente tridimensional, ou com a ajuda de mapas HDRI (*i.e. High Dynamic Range Image*), que são imagens de um ambiente capturadas em 360°, que contém informação de luminosidade, capaz de simular fontes de luz diversas para iluminar uma cena 3D (PIERSON et al., 2017).

2.3.5. Animação

Algumas animações normalmente realizadas por personagens de jogos durante as sessões ativas de jogo, em que o jogador controla o personagem, incluem ciclos de corrida e/ou caminhada, interação com objetos com cenário, movimentação corporal relacionada a ataques e defesa, pulos e escaladas sobre obstáculos, dentre outros (COOPER, 2019). Os ciclos de animações são definidos na etapa de pré-produção do projeto, onde também são definidas as máquinas de estado do personagem que serão implementadas na *engine*.

A representação visual do personagem também se dá através dos seus movimentos durante o jogo, mas no caso de um game em modalidade de terceira pessoa, o jogador normalmente vê as costas do personagem, logo suas expressões faciais não precisam ser

¹¹ Disponível em <<https://www.artstation.com/artwork/QzEWX8>> Acessado em 24/08/2021.

usadas como único recurso para denotar o estado emocional e físico do personagem durante o jogo.

Para critérios de organização do projeto, as animações criadas para um personagem de jogo digital normalmente são divididas em três categorias complementares que podem variar durante a jogabilidade, chamadas de (1) ociosas (*i.e. idle*), (2) ação e (3) transição. A primeira vai agrupar os ciclos ociosos de animação em que não há comandos do jogador (*i.e. inputs*) sendo feitos naquele momento (ou seja, quando ele não interage com a interface de controle do personagem). A segunda engloba os ciclos de animação que serão executados para descrever uma ação do jogador, como executar o comando de pular ou correr. E a terceira vai agrupar as animações usadas para fazer a transição entre animações das outras duas categorias, a fim de evitar que haja uma quebra de fluxo na movimentação do personagem, como na mudança entre um ciclo de animação em que o personagem está parado e outro em que eles está correndo e vice-versa (COOPER, 2019).

Uma característica importante desses grupos de animação, é que os dois primeiros poderão ser interrompidos pelo *input* do jogador no momento em que ele o fizer, mas as animações de transição não devem ser interrompidas durante sua execução pelos *inputs*, a fim de manter a fluidez de movimentação do personagem. Entretanto, devem manter a responsividade dos comandos, sem prejuízo à jogabilidade, mantendo uma curta duração.

3. REALIZAÇÃO

Em acordo com os objetivos estabelecidos para o produto, a realização do projeto teve como foco a criação dos assets que compõe uma personagem interativamente manipulável, compatível com jogo tridimensional em terceira pessoa. Os controles deveriam permitir uma movimentação típica de um personagem característica desse tipo de jogo, como andar, correr, pular, escalar, etc... Nesse sentido, a personagem desenvolvida deveria permitir uma deformação abrangente da malha utilizada para o modelo tridimensional, que permitisse uma gama de movimentos suficientes para o gênero de jogo pretendido, além de apresentar texturas capazes de simular com fidelidade os materiais escolhidos para a caracterização da personagem através dos princípios de representação PBR.

Os recursos de hardware disponíveis para o projeto foram: desktop pessoal com placa mãe Asus Prime A320M-K/BR AM4 DDR4, processador AMD Ryzen 3 3200g, placa de vídeo AMD Radeon rx 5500 xt com 8Gb, 16Bb de memória RAM, um SSD de 240Gb, um HD de 500Gb; mesa digitalizadora Wacom Intuos Pen Small CTH480L. Já os recursos de software principais utilizados foram: sistema operacional de base Linux Pop!_OS 20.10; sistema operacional Windows 10 2021; Blender 2.92 para criação do modelo tridimensional; Substance Painter 2021.1.0 para criação de texturas e materiais PBR ; Unreal Engine 4 para testes do personagem e para a criação do Memorial Interativo. Outras ferramentas como complementos de softwares e plugins utilizados serão detalhadas durante a sessão de Realização do projeto.

Devido a familiaridade com o sistemas Pop!_OS e Windows, antes da etapa de modelagem foram feitos alguns testes de desempenho das ferramentas utilizando as próprias cenas padrão dos software, e foram levados em conta consumo de memória RAM, processador e de placa gráfica do computador. Assim, decidiu-se que todo o processo de desenvolvimento e testes da personagem na Unreal Engine seriam realizados no sistema operacional de base Linux, já que ele apresentou um melhor desempenho com o Blender e principalmente com o Substance Painter. Entretanto o seu desempenho foi pior na utilização da Unreal Engine quando comparado com o sistema Windows, apresentando falhas de renderização da interface do software e alguns travamentos. O sistema Windows foi utilizado apenas durante a etapa de desenvolvimento do Memorial Interativo, já que a Unreal Engine apresentou um desempenho melhor neste sistema operacional.

Seguindo a pipeline estabelecida para a criação da personagem, neste capítulo serão descritas as etapas de execução do produto desenvolvido, além das dificuldades encontradas durante o processo.

3.1. Conceituação visual

Durante a etapa de pesquisa para as referências visuais para o projeto, foi proposto um personagem que homenageia a figura de um vaqueiro, figura característica da região Nordeste do Brasil conhecida pela capacidade de adaptação e sobrevivência em áreas de sertão, onde normalmente desempenham funções de pastoreio e criação de animais. Como uma das representações do povo sertanejo, o vaqueiro (Figura 11) também é conhecido como um desbravador do interior do país, conhecido pela habilidade de montaria e captura de animais em meio a mata densa e espinhosa característica da caatinga. Por esse motivo, julgou-se adequada também em termos narrativos, a escolha dessa figura que, além de homenagear a região Nordeste, se encaixa no arquétipo de personagem aventureiro e de espírito explorador, normalmente associado ao gênero de jogos de ação e aventura.

Figura 11 – Foto de um vaqueiro.



Fonte: Germano Haiut¹².

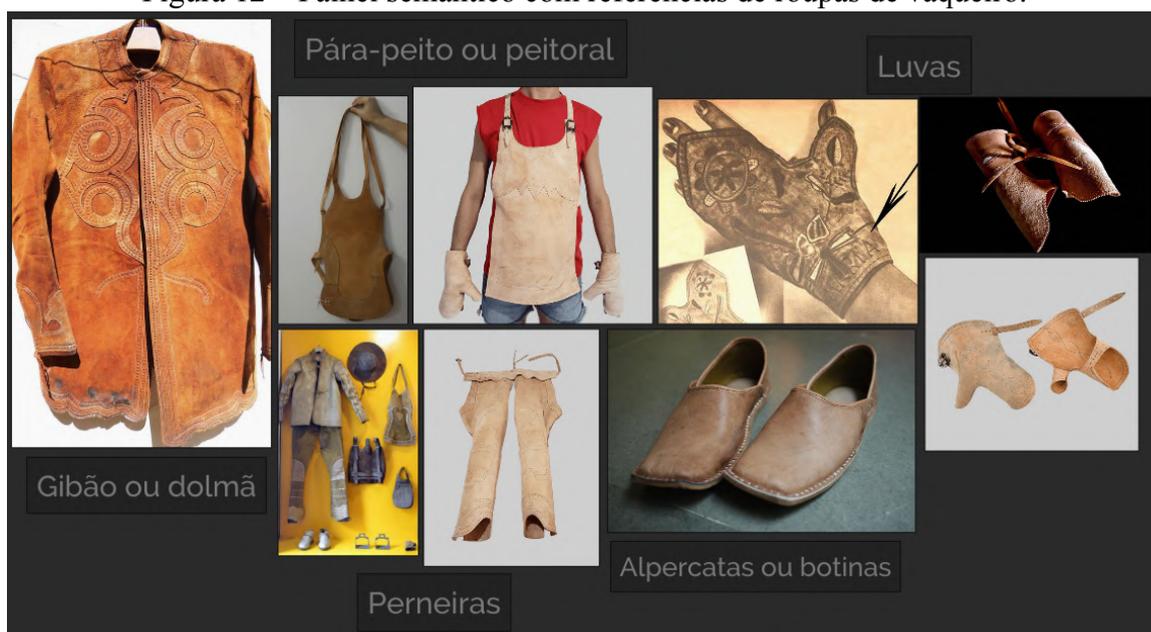
O processo de adaptação dos vaqueiros no sertão, levou ao desenvolvimento de um dos trajes mais característicos do Nordeste, composto por várias peças de roupa feitas de

¹² Disponível em <<https://tinyurl.com/vhkn9rb7>> Acessado em 22/09/2021.

couro que servem de proteção para o vaqueiro contra galhos secos e espinhos e também do sol característicos da região, compondo uma espécie de armadura para o desempenho de suas funções na mata. As principais peças de roupa que compõe o traje de um vaqueiro são o gibão ou dolmã, que é uma espécie de jaqueta presa ao corpo por cordões de couro; chapéu; peitoral ou pára-peito, que é uma peça amarrada no dorso e serve de proteção para impactos frontais no peitoral e abdômen; luvas, que normalmente são abertas nas palmas das mãos e dedos, facilitando o manejo da a montaria; perneiras, que são peças que envolvem as pernas e as vezes parte dos pés do vaqueiro e são amarradas à cintura; e por fim as botinas ou alpercatas, que são calçados para protegerem os pés (Figura 12).

A Figura 12 mostra as principais peças do traje típico de um vaqueiro, que compõem uma espécie de armadura para proteção contra impactos e contato com a vegetação seca da caatinga.

Figura 12 – Painel semântico com referências de roupas de vaqueiro.



Fonte: Elaborado pelo autor.

Durante as pesquisas iniciais para o projeto, a figura de um vaqueiro homem foi definida como representação visual do produto. Entretanto, foi notado que a grande maioria das referências imagéticas encontradas para os primeiros esboços representavam personagens de um universo muito masculino. Foi sugerido então que se pensasse a possibilidade de produzir uma protagonista feminina.

Uma análise sobre a representatividade feminina e masculina no jogo *Devil May Cry* aponta uma repetição de padrões da indústria de jogos na representação de gênero dentro do jogo, em que as personagens femininas são geralmente associadas a características de sensualidade, nudez e fragilidade (ROMANUS, 2012). E em uma análise mais recente dos 100 jogos mais populares da Steam, foi observado que apesar dos avanços feitos em vários jogos com representação feminina, os jogos de *Battle Arena* são os que pior representam este gênero. E ainda estão dentro de uma margem pequena (apenas 19 dentre os 100 mais populares) de jogos que sequer possuem personagens femininas (KOHLENER et al., 2021).

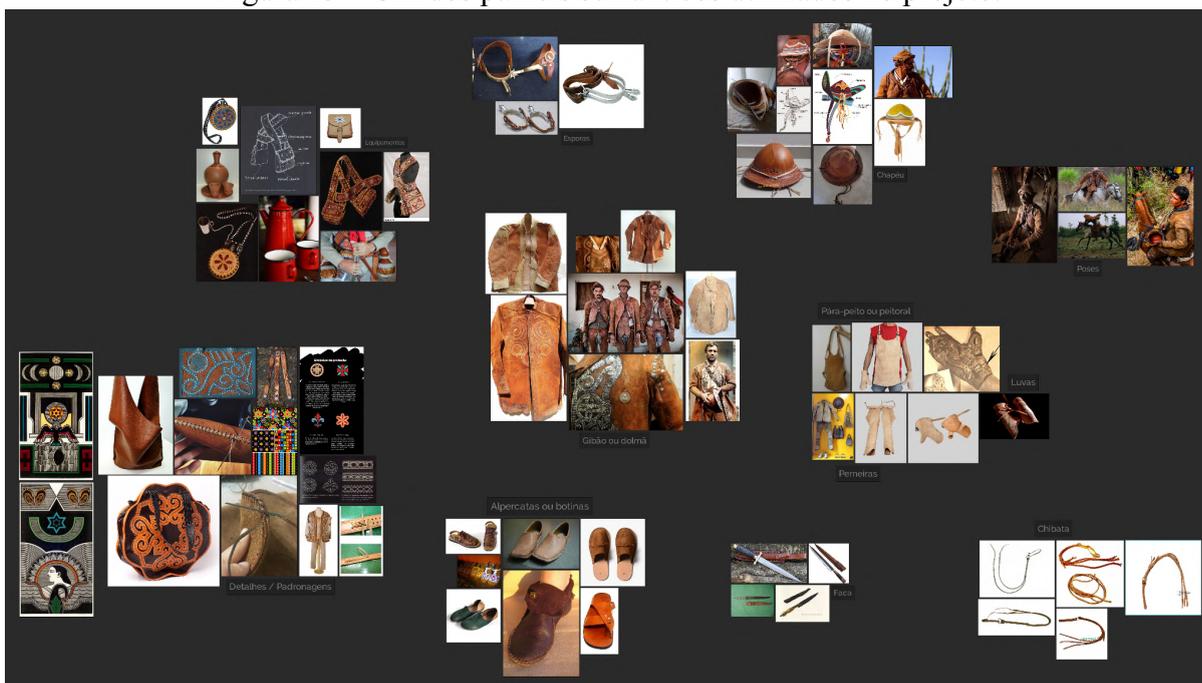
Outro problema da falta de representatividade feminina nos jogos pode ser observado quando, apesar da presença de personagens femininas em um jogo, há um direcionamento claro dele para uma parcela de consumidores majoritariamente masculino, como acontece em muitos jogos da franquia Tomb Raider, em que a protagonista é apresentada como uma personagem forte, inteligente e destemida, mas ao mesmo tempo é mostrada como símbolo sexual (BRISTOT et al., 2017).

Deste modo, além da representação da cultura regional, o projeto se abriu para as questões de representatividade de gênero, tão necessárias e urgentes hoje. Decidiu-se por fim alterar o gênero da personagem para o feminino, dando origem a Joana, personagem utilizada para o projeto.

A primeira etapa de desenvolvimento do conceito visual da personagem foi a criação de um painel semântico das características físicas e estilísticas previamente definidas. Um painel semântico é uma ferramenta que, através da organização de referências imagéticas em um painel, serve para orientar equipes de projetos (REIS; MERINO, 2020). Esse painel foi utilizado para organizar referências de cada parte individual relativa ao desenvolvimento da personagem (Figura 13), como cabelo, peças de roupas, estilo de volumetria do corpo, rosto e etc...

A Figura 13 mostra um dos painéis semânticos utilizados no projeto que contém referências visuais para o desenvolvimento das roupas da personagem.

Figura 13 – Um dos painéis semânticos utilizados no projeto.



Fonte: Elaborado pelo autor.

O maior objetivo da etapa de conceituação visual foi determinar a referência base para o desenvolvimento da modelagem tridimensional da personagem, definida como um *model sheet* (Figura 14), feito a partir de referências coletadas sobre vestimentas e estudos de miscigenação de povos da região do interior do nordeste. O *model sheet* contém estudos anatômicos, de estilo visual, e caracterização de uma personagem que atendesse os requisitos estabelecidos para o projeto. Ela deveria possuir traços biótipos que lembrassem povos característicos da região do sertão do Brasil denominados de caboclos, resultado da mistura de povos indígenas originários das região norte e nordeste do país, e povos africanos e europeus. Além de roupas de couro tradicionalmente usadas por vaqueiros da região, o *model sheet* apresenta alguns estudos de vestimenta casual, posturas e detalhes estilísticos referentes a caracterização cartoon da personagem.

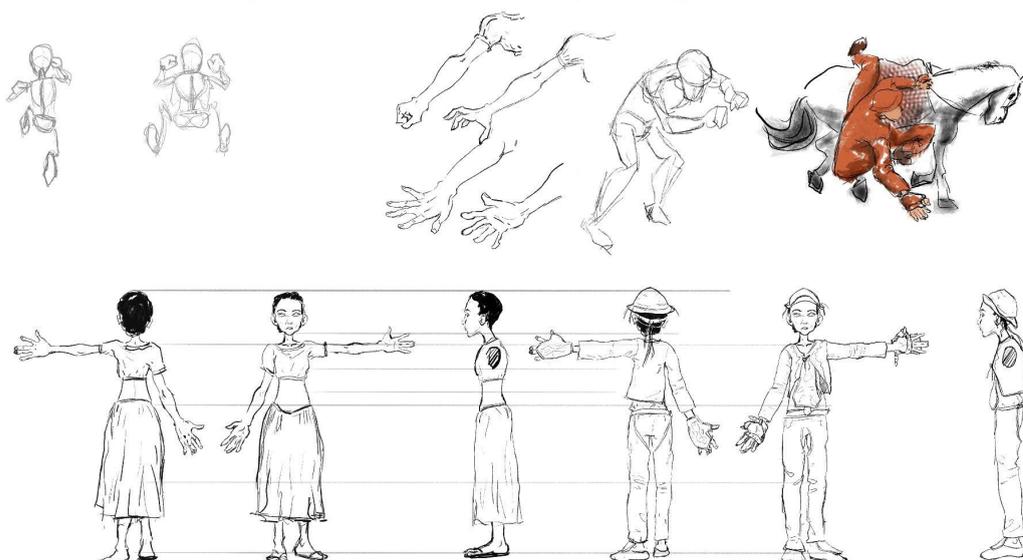
O estilo visual planejado para o projeto deveria contemplar uma estilização das características visuais da personagem, adotando um estilo que transita entre a estética cartunesca e realista, buscando uma mistura que trouxesse mais apelo, não só como o resultado do estudo de processos de desenvolvimento visual de personagem para fins acadêmicos, mas também como um produto de mídia com potencial comercial.

Com esse requisito em mente, ficou evidente a dificuldade de adaptação estilística para um personagem de estilo mais cartunesco já que muitos dos traços biótipos

característicos de um indivíduo percebido como caboclo acabariam, ou por se perder devida natureza da simplificação do próprio estilo cartoon, ou não seriam suficientes para representar adequadamente uma pessoa dessa região do Brasil, dado o caráter bastante sutil da representação do biótipo de um indivíduo de modo que não parecesse estereotipado. Então durante a etapa de desenvolvimento das artes conceituais, foi dado um foco maior às roupas da personagem como principal meio de caracterização, que ajudariam a evidenciar com maior facilidade os aspectos culturais mais representativos da região Nordeste, dado o recorte específico em que a personagem foi proposta.

A Figura 14 mostra um *model sheet* com uma série de ilustrações com estudos de poses da personagem, que serão usadas como referência direta para o seu desenvolvimento técnico.

Figura 14 – *Model sheet* da personagem Joana



Fonte: Elaborado pelo autor.

3.2. Modelagem tridimensional

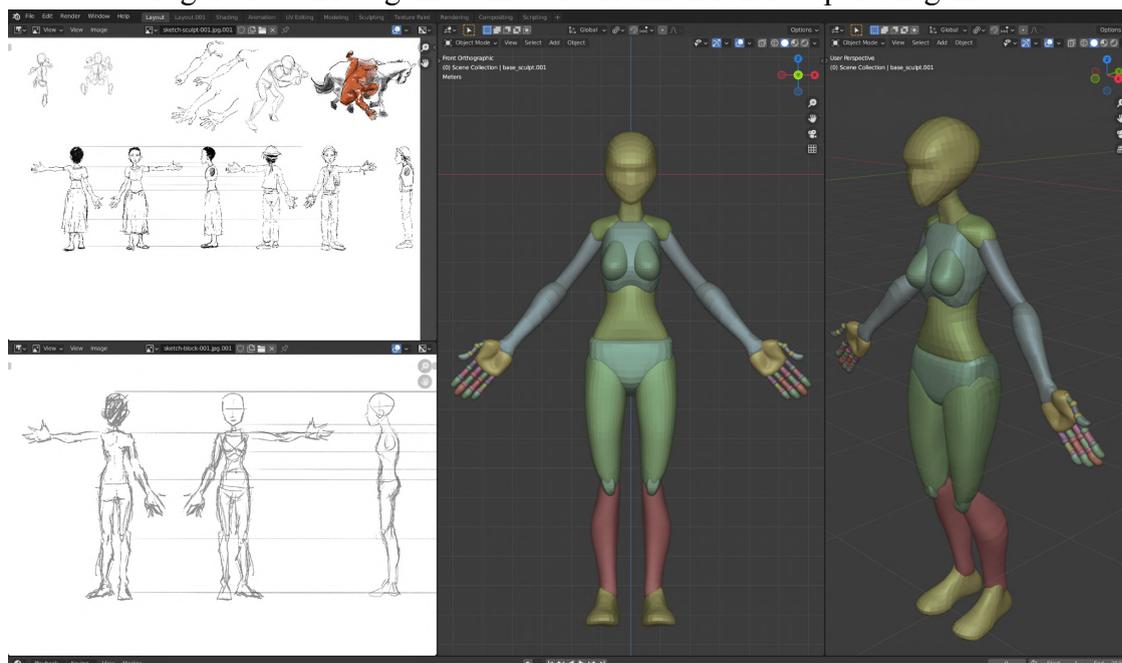
A etapa de modelagem da personagem seguiu um método de desenvolvimento baseado em dois processos específicos: o primeiro é a criação de um modelo *high poly* para definição dos detalhes de volumetria de um objeto ou personagem, e o segundo é a retopologia do modelo original, em que é feita uma segunda modelagem simplificada, chamada de *low poly*, que evidencia apenas a volumetria do modelo, permitindo uma manipulação mais rápida e de baixo custo computacional. Esse modelo *low poly* é que de fato será utilizado durante o processo de animação do personagem.

3.2.1. Escultura do corpo

O trabalho de escultura digital foi feito a partir de processo inicial chamado de blocagem, onde são organizados algumas formas tridimensionais básicas em uma volumetria baseadas nas proporções da personagem (Figura 15) estabelecidas no *model sheet*. A imagem do *model sheet* foi colocada dentro do software de modelagem Blender e usado como referência direta para o modelo, e depois que a blocagem é concluída, foi iniciado um processo de detalhamento chamado de escultura digital.

A figura 15 mostra a blocagem do corpo da personagem feita a partir de primitivos 3D. Os primitivos usados foram algumas esferas e majoritariamente cubos, com subdivisão de malha aplicada, o que permitiu que pudessem ter as quinas arredondadas e suas formas distorcidas para serem modelados com maior facilidade.

Figura 15 – Blocagem do modelo tridimensional da personagem.



Fonte: Elaborado pelo autor.

A escultura digital ou *sculpt* consiste na manipulação da malha de um objeto tridimensional, alterando sua forma de modo a adicionar detalhes ao modelo através da subdivisão dos polígonos de algumas regiões de sua superfície. Essa técnica de modelagem é extremamente útil para criar pequenos detalhes e imperfeições na superfície do objeto representado. Devido a grande quantidade de polígonos criados durante o processo de escultura digital, o modelo resultante desse tipo de processo também é chamado de *high poly*,

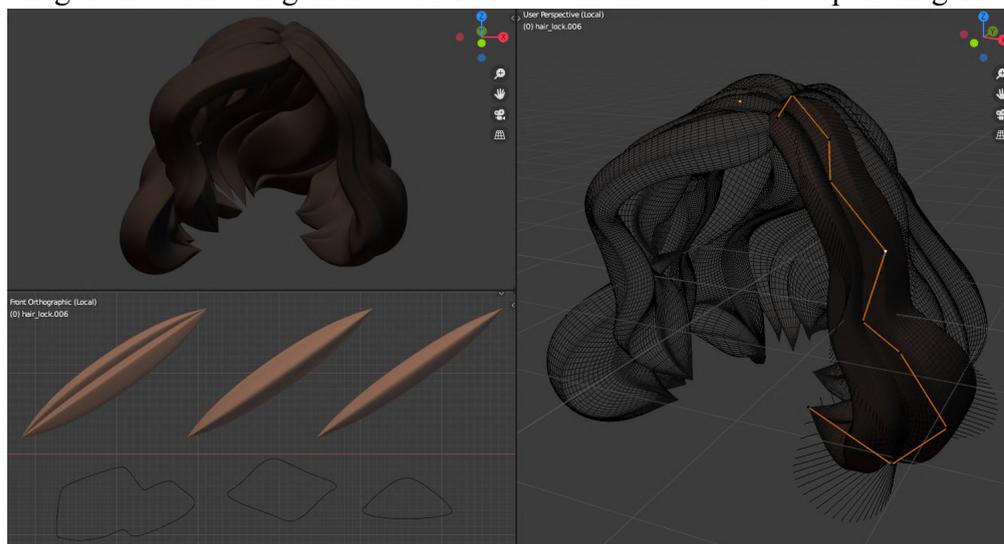
podendo facilmente atingir uma quantidade de milhões de vértices para cada objeto na cena, dependendo de sua complexidade e nível de detalhamento.

Algumas partes do modelo foram feitas com a utilização de técnicas não destrutivas de modelagem, como por exemplo o cabelo, que foi feito utilizando curvas como base que podem ter o formato de cada mecha ajustado, apenas modificando a posição de cada um dos nós da curva principal (Figura 16). Desse modo, não é preciso nenhum planejamento da quantidade de vértices ou organização das faces do modelo, já que esses parâmetros podem ser alterados a qualquer momento.

Além disso, a quantidade ajustável de vértices da malha a partir desse tipo de modelagem não destrutiva, permitiu que o mesmo modelo pudesse ser usado tanto como *high poly* quanto como *low poly* do cabelo, ajustando-se apenas a quantidade de vértices pretendida na hora da exportação de cada versão.

A Figura 16 mostra o objetos utilizados para modelagem do cabelo de forma não-destrutiva, em que um conjunto de curvas são usadas para definir o comprimento e ondulação das mechas de cabelo, e um outro conjunto de curvas é utilizado para definir o formato e volume de cada mecha.

Figura 16 – Modelagem não-destrutiva da malha do cabelo da personagem.



Fonte: Elaborado pelo autor.

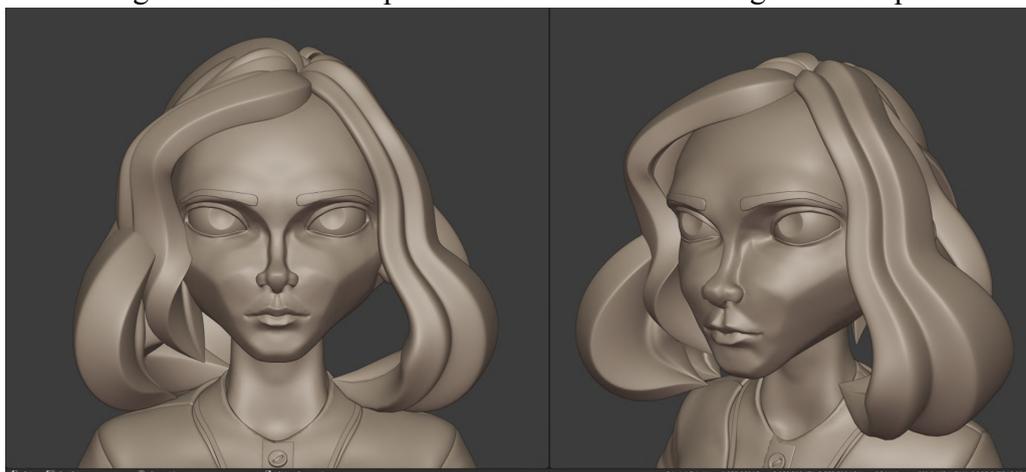
O maior desafio dessa etapa de modelagem foi determinar um equilíbrio visual entre proporções corporais cartunescas com detalhamento realista, que fizesse uma ponte entre esses estilos. O objetivo era que a personagem assumisse um estilo que possuísse um apelo

visual mais abrangente, podendo transitar com maior facilidade entre diversos estilos narrativos dentro do gênero de jogos de ação e aventura.

As primeiras versões da personagem foram esculpidas com uma volumetria mais pontiaguda de ângulos mais retos e curvas menos suaves (Figura 17). Mas essa decisão de design acabou dando à personagem uma característica muito impessoal e séria, fazendo com que ela não tivesse um apelo visual desejado de personagem empático, ao qual o jogador pudesse se apegar com mais facilidade.

A Figura 17 mostra as características faciais de uma das primeiras versões de Joana com formas mais retas e pontiagudas, o que acabava por deixar a personagem com uma representação facial mais séria e menos empática.

Figura 17 – Uma das primeiras versões da modelagem do corpo.



Fonte: Elaborado pelo autor.

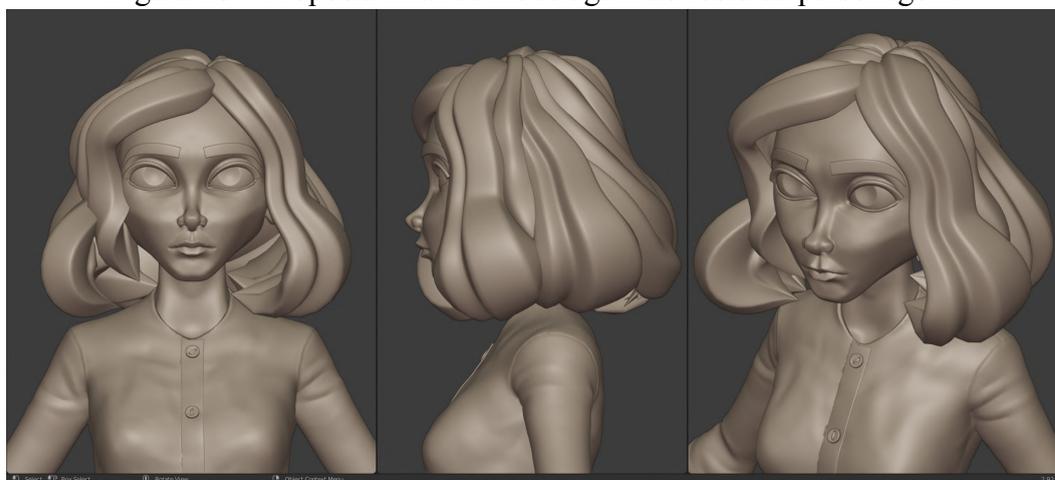
Nesse momento, por intermédio do orientador do projeto, foi estabelecido o contato com Camilo Cunha, que é um ilustrador e modelador 3D inserido no mercado nacional. O diálogo com o Camilo foi de grande importância durante o processo de revisão da modelagem da personagem, visto que várias sugestões de melhorias referentes a anatomia e estilização da estética cartoon foram feitas. Por exemplo: correção de postura, ajustes na volumetria e traços do rosto. A análise de um profissional mais experiente sobre os detalhes do projeto possibilitou que um conjunto de pequenas correções fizesse uma notável diferença na consolidação do estilo da personagem.

Após os ajustes baseados na análise de Camilo sobre a personagem, as últimas versões do modelo apresentaram formas mais suaves, com menos detalhamentos desnecessários, causando um apelo visual mais cartunesco e mais confortável visualmente (Figura 18).

Neste ponto do projeto foi definido um balanceamento visual mais assertivo, em que o estilo cartunescos ficaria evidenciado pelas formas suaves e proporções corporais levemente exageradas da personagem, e o estilo realista seria destacado pela representação fidedigna dos materiais e texturas utilizados nas superfícies do modelo, através das técnicas de representação PBR que foram trabalhadas mais adiante.

A Figura 18 mostra a modelagem final do rosto da personagem, apresentando um estilo mais cartunescos, destacando maior expressividade e equilíbrio entre as partes do rosto.

Figura 18 – Proposta final de modelagem do rosto da personagem.



Fonte: Elaborado pelo autor.

3.2.2. *Modelagem das roupas*

A modelagem *high poly* das roupas foi feita de forma separada para cada peça de roupa da personagem a fim de permitir maior detalhamento e flexibilidade durante a execução (Figuras 19 e 20). Nesta etapa da produção, os modificadores do Blender foram extensivamente utilizados para permitir uma modelagem não-destrutiva, facilitando o processo de ajustes e correções dos objetos (Figura 21).

Na modelagem do topo do chapéu, por exemplo, existe um modificador responsável apenas por criar os amassados presentes na peça, enquanto outro é responsável pela espessura do couro. A organização da malha durante o processo de modelagem, em conjunto com os modificadores utilizados, são responsáveis pela criação das costuras entre as peças, dando a impressão de que cada uma foi modelada individualmente, quando na verdade elas fazem parte de um único objeto tridimensional.

Posteriormente os modificadores são aplicados ao objeto, para criar de fato um modelo *high poly*, que será necessário durante a exportação desse objeto para criação das texturas. Nesse processo de aplicação dos modificadores, a possibilidade de ajustes dos modelos através de parâmetros não destrutivos acaba sendo perdida, e por esse motivo, todos os objetos que passaram por esse processo tiveram suas versões originais e ajustáveis guardada, como forma de backup, caso precisassem ser alterados mais adiante no projeto após a aplicação dos modificadores.

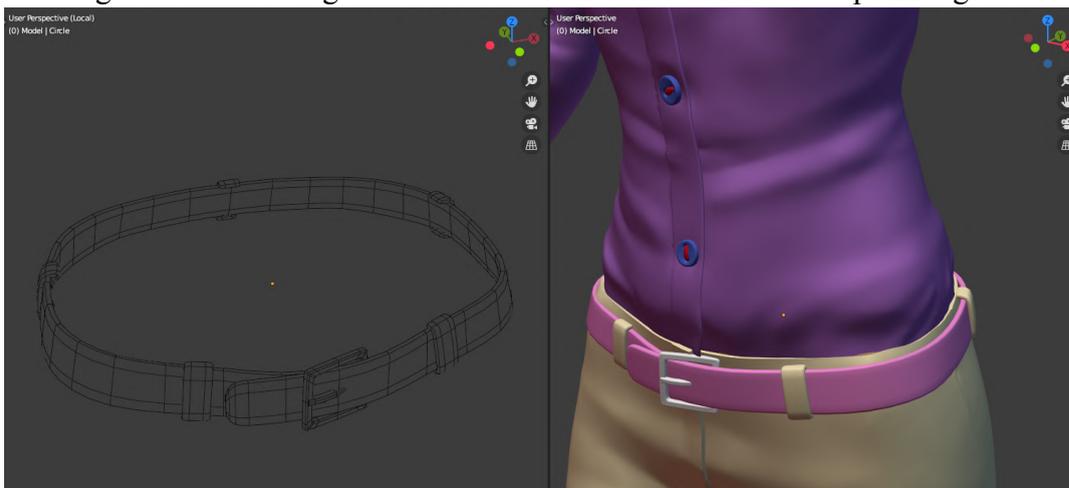
A Figura 19 destaca a modelo tridimensional do cinto da personagem, feito a partir de métodos não destrutivos de modelagem.

A Figura 20 mostra as peças de roupa do dorso da personagem (camisa, gibão, peitoral, calça e perneiras), que foram feitas com uma mistura de métodos não destrutivos e alguns ajustes feitos com a ferramenta de escultura do Blender.

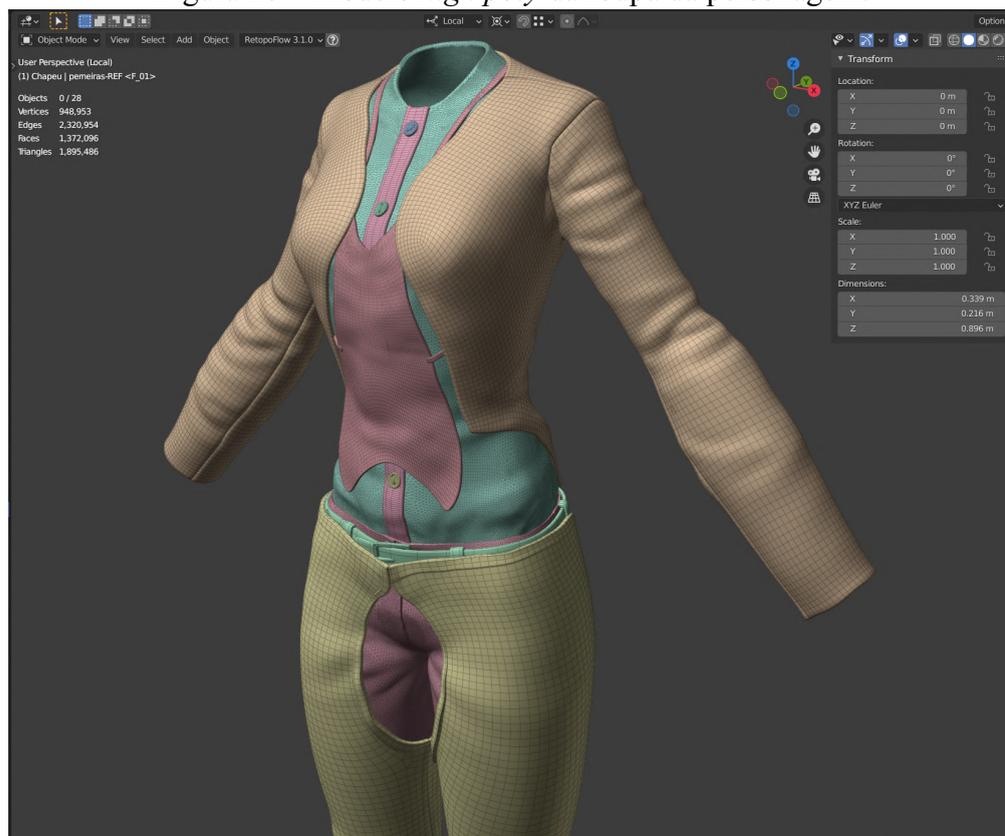
A Figura 21 mostra o chapéu da personagem em vários ângulos diferentes, feito a partir de objetos 3D simples e algumas curvas. Todos detalhados com métodos de modelagem não destrutiva.

A Figura 22 mostra quatro imagens de todos os objetos que compõem o modelo *high poly* da personagem organizados em uma sequência de *turnaround*.

Figura 19 – Modelagem não-destrutiva da malha do cinto da personagem.

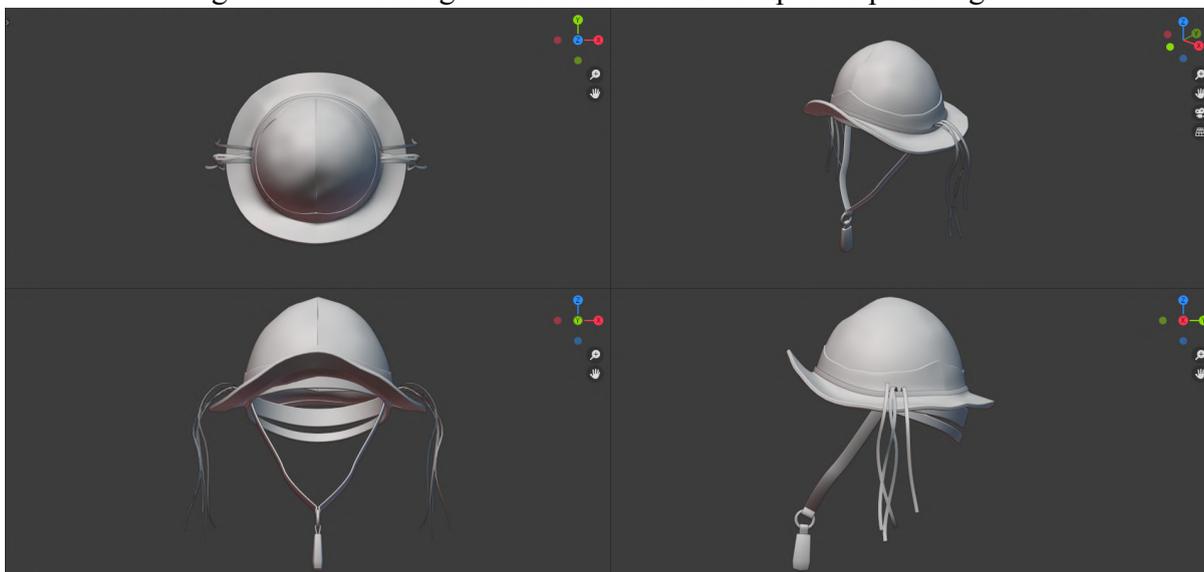


Fonte: Elaborado pelo autor.

Figura 20 – Modelo *high poly* da roupa da personagem.

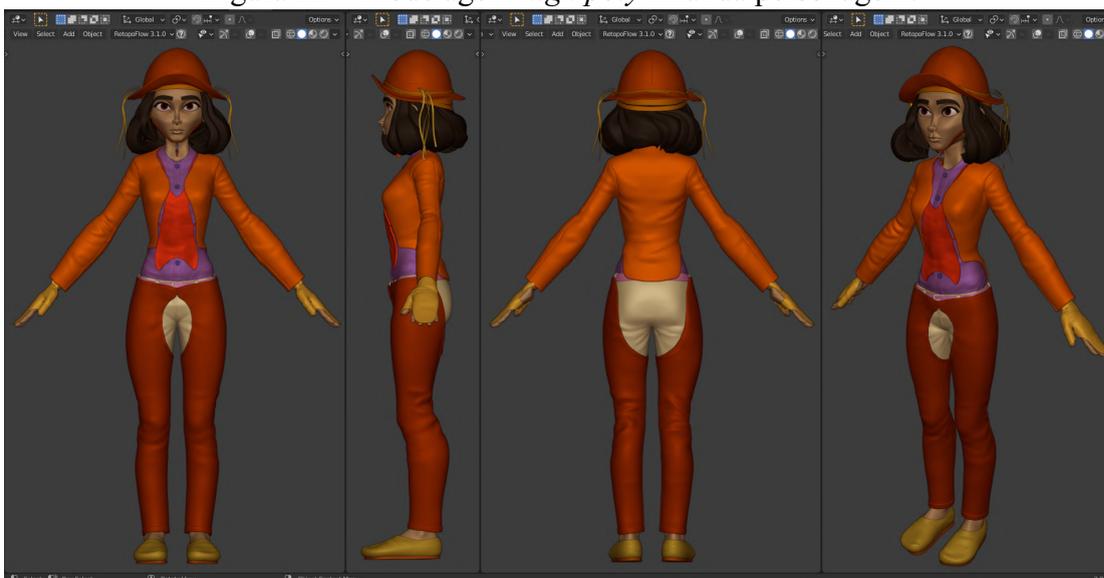
Fonte: Elaborado pelo autor.

Figura 21 – Modelagem não destrutiva do chapéu da personagem.



Fonte: Elaborado pelo autor.

Figura 22 – Modelagem *high poly* final da personagem.



Fonte: Elaborado pelo autor.

3.2.3. Retopologia do modelo

O processo de retopologia foi realizado com o auxílio de um addon para o Blender chamado RetopoFlow¹³, que adiciona uma interface específica no programa com diversas ferramentas voltadas especificamente para o processo de retopologia de modelos tridimensionais. O principal objetivo dessa etapa de produção é criar uma versão otimizada e organizada dos modelos trabalhados para que eles permitam uma deformação adequada de sua malha enquanto mantém um baixo custo computacional de renderização de polígonos (Figura 23).

Para termos de comparação, ao final do projeto, o arquivo em FBX (formato proprietário de arquivo para modelos tridimensionais) do modelo *low poly* da personagem que foi exportado do Blender tem um total de 27.385 vértices, com cerca de 1,6 Mb de tamanho. Já o arquivo do modelo *high poly* exportado tem 929.328 vértices, e possui cerca de 63,3 Mb de tamanho, o que representaria um salto gigantesco no custo computacional em qualquer sistema atual para renderizar um *asset* com essa densidade de polígonos, levando em conta apenas a malha da personagem.

Em produções que envolvem jogos digitais ou algum tipo de animação que necessite de um render em tempo real, a malha *low poly*, resultante do processo de retopologia, precisa ser planejada levando-se em conta o tipo de movimento que será executado pelo modelo

¹³ Disponível em <<https://github.com/CGCookie/retopoflow>> Acessado em 11/08/2021.

tridimensional e o tamanho que ele ocupará na tela. Para esse projeto, que se baseia em um jogo de terceira pessoa, em que a personagem não é vista tão de perto e vai aparecer majoritariamente de costas, optou-se por representar suas roupas em uma malha principal, sem distinção de peças de roupas no modelo, e que mantivesse uma densidade mediana de polígonos. desse modo uma mesma malha *low poly*, contendo as peças de roupa de camisa, gibão, peitoral, cinto, calça e perneiras, foi desenvolvida.

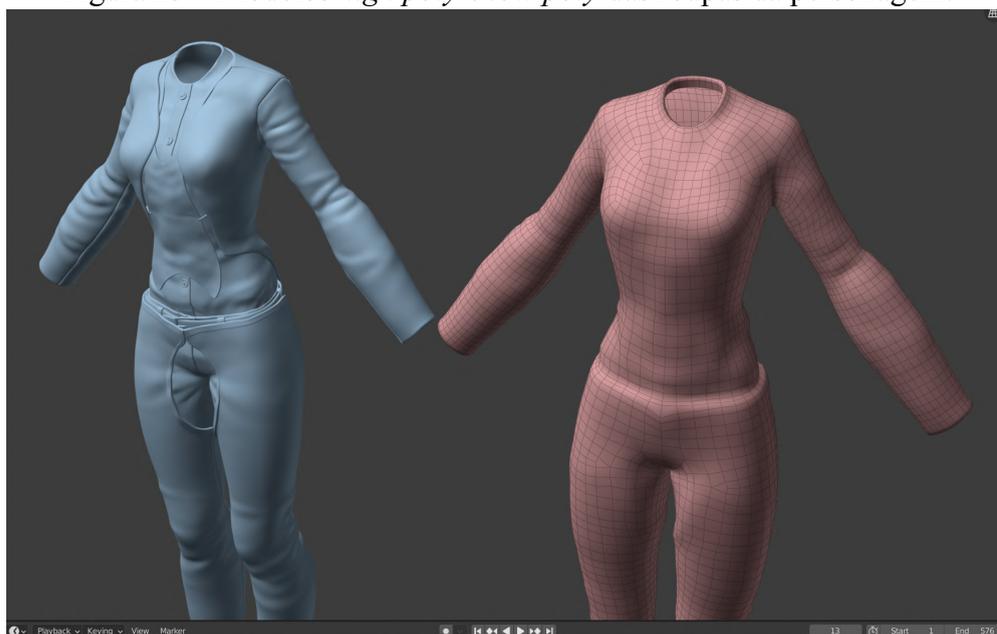
As principais vantagens obtidas com essa decisão consistem em uma maior otimização da personagem em termos de desempenho gráfico e uma facilidade maior de manipulação, *rigging* e animação do modelo. Em contrapartida, algumas limitações foram antecipadas, como a perda de flexibilidade da personagem durante as animações e uma perda de definição e detalhamento do modelo, principalmente na região frontal do corpo (Figura 23).

Outro fator que pesou para a decisão de tratar as principais peças de roupas como uma única malha, foi a complexidade de animação que seria enfrentada caso cada peça de roupa fosse tratada separadamente, já que o ajuste de pesos dos vértices seria mais complexo e a possibilidade de ocorrerem problemas de invasão de malha cada peça de roupa seria muito grande. Esse tipo de situação não inviabilizaria o projeto, mas certamente reduziria o tempo disponível para outras áreas de grande importância dentro da pipeline adotada.

Entretanto, apesar de simplificar o modelo, os critérios de retopologia adotados não excluíram o detalhamento feito na etapa de escultura digital com a malha *high poly*. Esses objetos foram utilizados mais adiante em outra etapa da produção responsável pelo detalhamento visual da personagem, logo, o processo de retopologia acabou sendo mais importante devido aspectos técnicos de otimização da produção de um personagem tridimensional desse tipo. Mesmo assim, a qualidade visual da personagem não fica prejudicada, já que essa simplificação do modelo *low poly* não define sozinha a caracterização estética do produto final.

A Figura 23 exibe os modelos *high poly* e *low poly* respectivamente da personagem, lado a lado, evidenciando a diferença na quantidade de detalhamento entre eles.

Figura 23 – Modelos *high poly* e *low poly* das roupas da personagem.



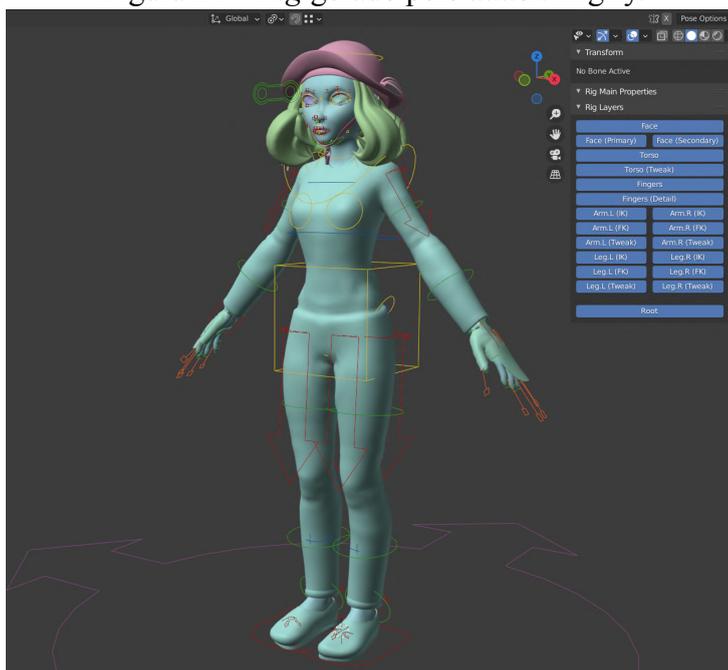
Fonte: Elaborado pelo autor.

3.3. Rigging

Em animação digital, o *rig* é uma estrutura de controladores que representam o esqueleto de um personagem. O *rig* em si é formado por estruturas menores chamadas de bones, que representam as articulações (*i.e. joints*) dos personagens, e o processo de criação e adequação dessas estruturas ao modelo 3D é chamado de *rigging*.

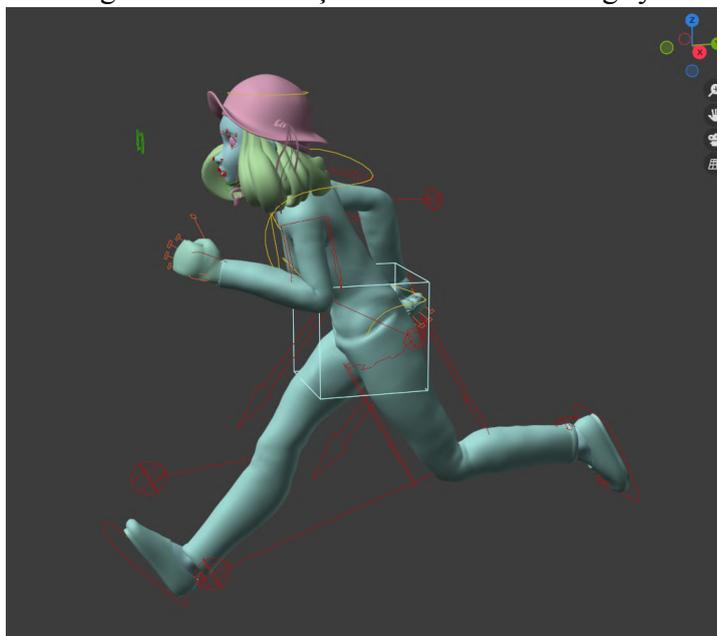
Neste projeto, a criação do *rig* foi feita com a ajuda de um *addon* do Blender chamado Rigify, que é uma ferramenta interna do programa que possui uma base para criação de personagens humanóides (Figuras 24 e 25). Na *pipeline* de produção utilizada, a etapa de *rigging* teve uma dupla função durante o projeto: a primeira foi criar de maneira rápida os controladores que vão permitir a animação do personagem, e a segunda foi testar a deformação da malha planejada durante a etapa de retopologia, orientando possíveis ajustes.

Figura 24 – Rig gerado pelo *addon* Rigify.



Fonte: Elaborado pelo autor.

Figura 25 – Animação feita através do Rigify.



Fonte: Elaborado pelo autor.

3.3.1. Ajustes de pesos dos vértices

O Blender possui um sistema que determina a quantidade de deformação que cada controlador deve exercer sobre cada ponto da malha de um modelo tridimensional, atribuindo para cada vértice um peso relativo a cada segmento (também chamado de *bone*) responsável pela deformação da malha. Em sistemas de animações complexos como Rigify, essa

atribuição é feita de forma automática, e organizada em camadas, mas dependendo da complexidade do modelo desenvolvido, esse sistema de pesos funciona apenas como ponto de partida para o processo de *rigging*, exigindo que algumas partes do modelo precisem de ajustes para que o *rig* deforme a malha dos objetos de forma adequada, evitando que ocorram problemas de deslocamento inadequado ou invasão de malha nos objetos (Figura 26).

Neste projeto o modelo precisou de vários ajustes na maior parte do corpo, principalmente nas regiões com maior deformação de malha e onde havia uma maior concentração de *bones* de deformação no *rig* (Figura 27). O método adotado para corrigir os problemas de peso dos vértices foi dividido em duas partes. (1) Primeiro, cada objeto teve a listagem de grupos de vértices atualizadas, eliminando os grupos desnecessários para cada um. Por exemplo, as sobrancelhas da personagem não precisam ser influenciadas pelos *bones* de deformação dos pés e mãos, logo, esses *bones* e de outras partes do corpo foram excluídos da lista de grupos de vértices das sobrancelhas (Figura 27). (2) Em seguida, foi utilizado o modo de pintura de pesos do Blender que permite que a influência de cada *bone* seja modificada para uma região específica da malha em cada objeto, através de uma redistribuição dos pesos de cada vértice ou conjunto de vértices para os *bones* dos quais aquela região sofre influência (Figuras 28 e 29). Como exemplo, temos a região do pescoço que, antes da correção dos pesos, sofria uma influência exagerada dos *bones* do queixo da personagem, o que fazia com que ele sofresse uma deformação muito grande sempre que ela estava em uma pose que mostrasse a boca aberta ou com a cabeça inclinada para trás (Figura 30).

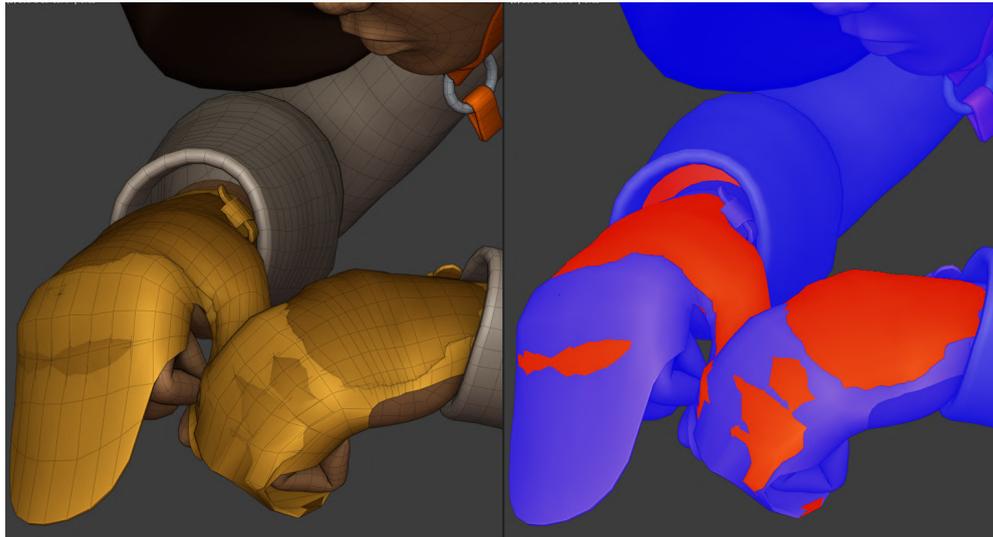
A Figura 26 mostra artefatos na malha das luvas, causadas por invasão de malha devido à espessura da luva ser muito fina nas primeiras versões do modelo, com os vértices problemáticos destacados em vermelho.

A Figura 27 compara o antes e depois dos ajustes feitos nos pesos de vértices das regiões da sobrancelha e boca da personagem. A Figura também mostra com destaque em vermelho os grupos de vértices que precisaram ser excluídos no processo de ajuste dos pesos.

As Figuras 28 e 29 mostram as áreas afetadas pelos grupos de vértices, destacados em vermelho, em duas regiões do modelo do chapéu.

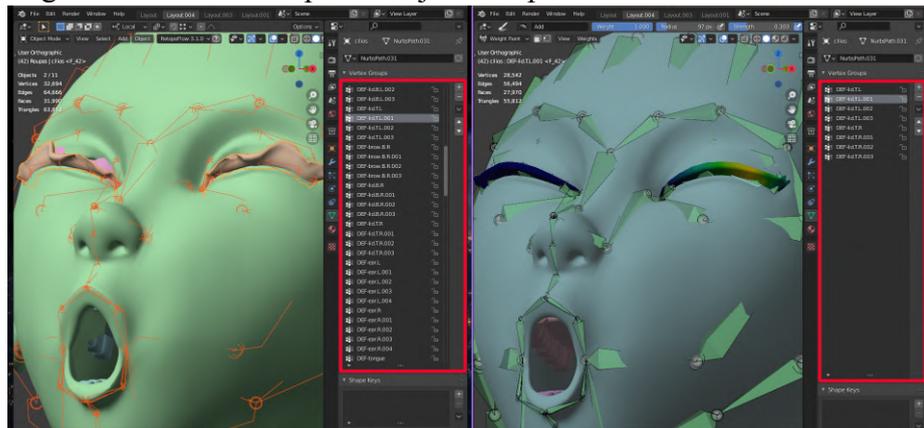
A Figura 30 compara o antes e depois dos ajustes feitos nos pesos de vértices da região do pescoço e mandíbula da personagem.

Figura 26 – Problemas na atribuição de pesos das luvas.



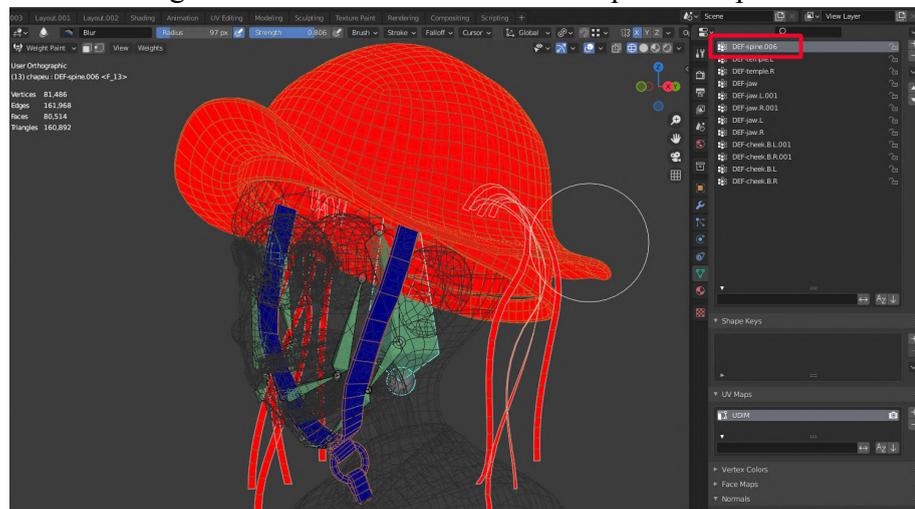
Fonte: Elaborado pelo autor.

Figura 27 – Antes e depois do ajuste de pesos das sobrancelhas e boca.



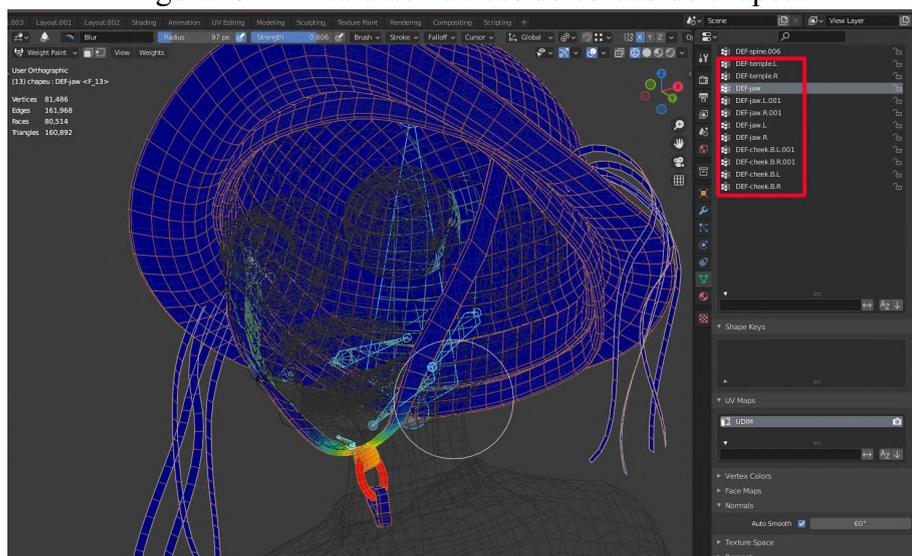
Fonte: Elaborado pelo autor.

Figura 28 – Pintura de vértices do topo do chapéu.



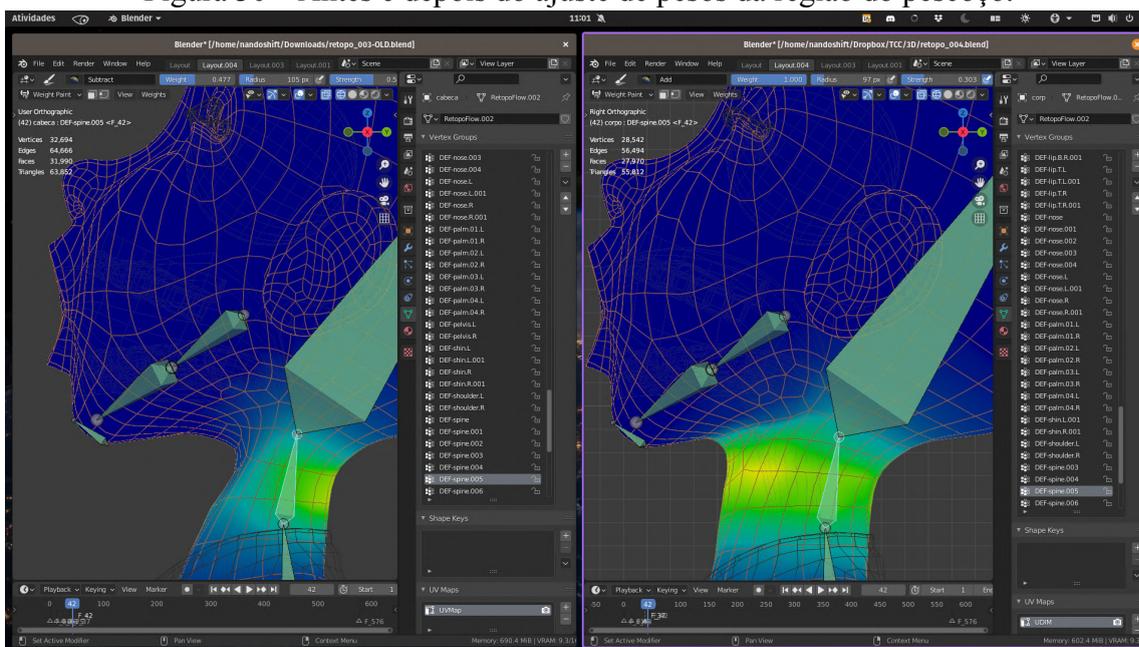
Fonte: Elaborado pelo autor.

Figura 29 – Pintura de vértices do cordão do chapéu.



Fonte: Elaborado pelo autor.

Figura 30 – Antes e depois do ajuste de pesos da região do pescoço.



Fonte: Elaborado pelo autor.

3.3.2. Ajustes de topologia

Antes da finalização do modelo, novamente foi buscado o contato com um profissional do mercado que pudesse fazer uma análise da topologia da personagem. Por intermédio do orientador do projeto, foi estabelecido contato com o profissional de *look dev* Lucas Falcão, que tem experiência em modelagem tridimensional, escultura digital, retopologia, iluminação e criação de materiais, dentre outras atividades relacionadas à criação de personagens.

A análise de Lucas apontou possíveis problemas que poderiam acontecer na topologia do rosto da personagem, principalmente na região da boca. As áreas apontadas como potencialmente problemáticas foram ajustadas para evitar problemas na deformação da malha: os *loops* de arestas em volta da boca, nariz e pescoço foram reorganizados (Figuras 31 e 32) e a quantidade de vértices do nariz foi reduzida (Figura 33). Outras áreas menos importantes da topologia da roupa também foram assinaladas (Figura 34), mas os testes de deformação realizados não apontaram problemas, logo essas áreas não foram alteradas.

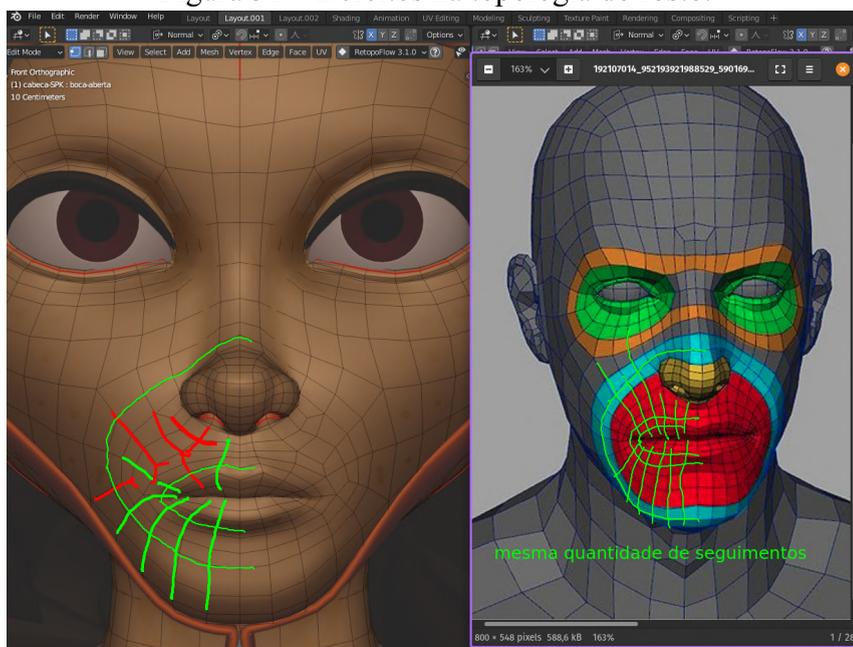
A Figura 31 destaca os segmentos de arestas corretos na cor verde, e os segmentos de arestas incorretos em vermelho. Ela também mostra um modelo de referência com os segmentos de arestas organizados.

A Figura 32 compara o antes e o depois dos ajustes feitos na topologia da região do pescoço e queixo da personagem.

A Figura 33 compara o antes e depois dos ajustes de topologia feitos no rosto da personagem. Ela também mostra um modelo de referência que destaca regiões do rosto com maior importância para organização da topologia de um modelo humanóide.

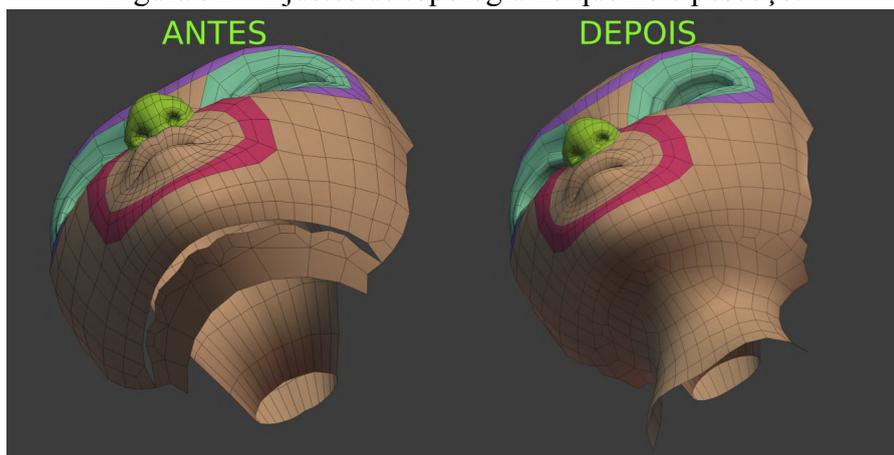
A Figura 34 destaca regiões da topologia da roupa que foram cogitadas para ajustes.

Figura 31 – Defeitos na topologia do rosto.



Fonte: Elaborado pelo autor.

Figura 32 – Ajustes de topologia no queixo e pescoço.



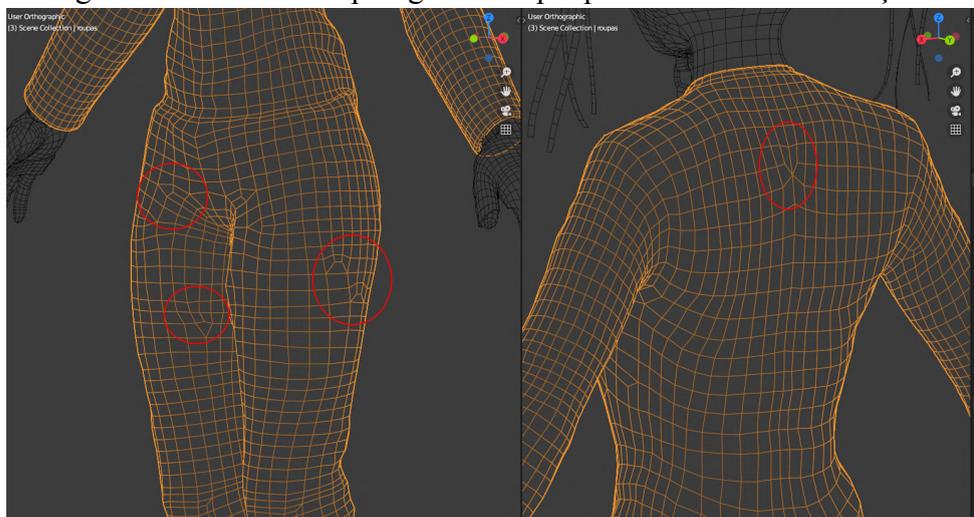
Fonte: Elaborado pelo autor.

Figura 33 – Ajuste de topologia no rosto.



Fonte: Elaborado pelo autor.

Figura 34 – Áreas da topologia da roupa que não sofreram alterações.



Fonte: Elaborado pelo autor.

Esse processo de ajuste acabou resultando na refacção de algumas partes do processo de *rigging*, já que a alteração na malha de algum objetos acabou causando uma redistribuição dos pesos de seus vértices, que por sua vez causaram novos problemas de deformação incorreta de malha.

Neste ponto do projeto também foram criadas, com a ajuda do *rig*, versões *high poly* e *low poly* da malha em uma pose da personagem de boca aberta. Essas versões seriam utilizadas mais tarde para facilitar o processo de pintura 3D do modelo, na etapa de *look dev*, já que seria mais fácil visualizar a parte interna da boca da personagem.

3.4. Look Dev

Após a etapa de *rigging*, os *assets* gerados normalmente são repassados para dois perfis de profissionais diferentes dentro da pipeline de produção, um responsável pela animação do modelo tridimensional e outro responsável pelo *look dev*. essas duas etapas podem ser executadas em paralelo, mas no contexto desse projeto, após os ajustes realizados no *rig* criado, a etapa de de look dev foi desenvolvida primeiro.

Nessa etapa foram desenvolvidos os materiais utilizados para a versão final do produto. O foco nesta etapa era criar materiais realistas que, além de representar superfícies de couro, tecido e pele, ajudassem a contar uma narrativa implícita sobre a personagem, que pudesse ser transmitida de forma visual e que ajudasse a compor um background básico da personagem. Essas informações são evidenciadas tanto através de marcas do tempo e desgaste das roupas da personagem quanto na própria pele, marcada pelo sol.

3.4.1. Mapeamento UV

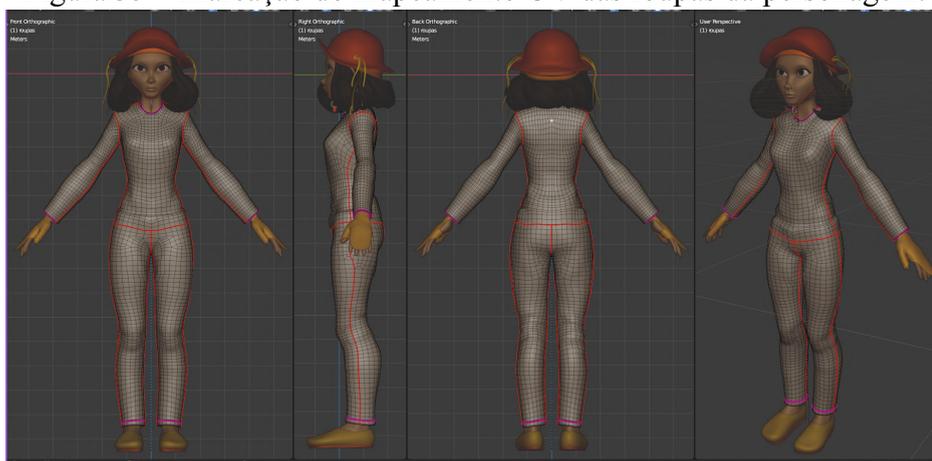
O mapeamento UV dos modelos das roupas da personagem foram feitos principalmente tomando como referência inicial o que seriam os pontos de costura de uma roupa real (Figura 35), aproveitando a modelagem que evidencia as costuras das peças no modelo *high poly*. E em seguida as áreas planificadas da malha, chamadas de ilhas, foram organizadas em um sistema de coordenadas UV chamada de UDIM ou *UV tiles*, que permite um fluxo de trabalho com diversas mapas UVs simultaneamente, permitindo que eles sejam exportados juntos, em um mesmo material, e que sejam armazenadas em tamanhos diferentes a depender do objeto mapeado.

No caso deste projeto, em que o modelo tridimensional é composto por vários modelos diferentes, eles compartilham um mesmo layout de UVs, que foi dividido e organizado em cinco regiões diferentes para cada parte do modelo (Figura 36). O primeiro mapa foi destinado para o rosto e corpo; o segundo mapa para os detalhes do rosto, boca e olhos; o terceiro foi destinado para a roupa; o quarto para o chapéu; e o quinto para o cabelo da personagem.

A Figura 35 mostra as marcações, em vermelho, das costuras usadas para o mapeamento UV da malha *low poly* das roupas.

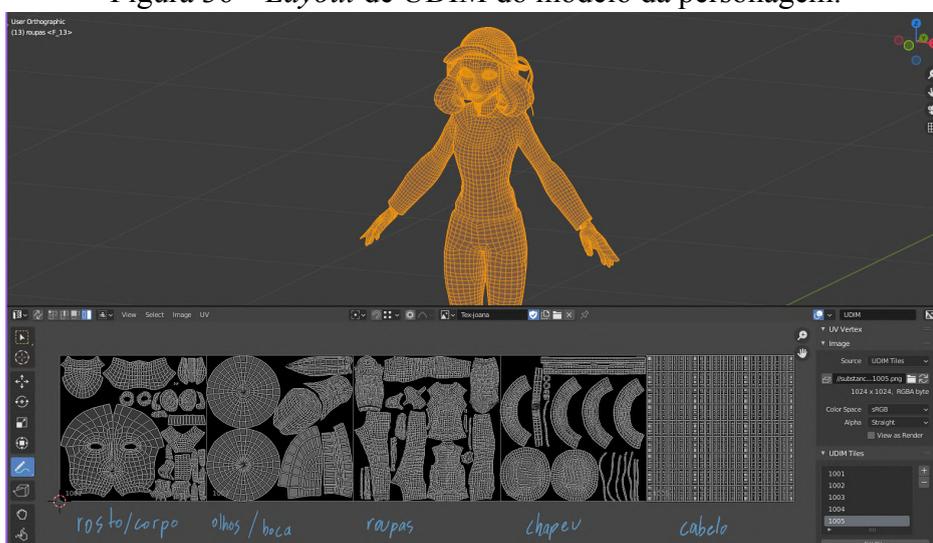
A Figura 36 mostra o modelo *low poly* da personagem e as ilhas de UV, organizadas no sistema de UDIM, de todas as partes do corpo e roupas.

Figura 35 – Marcação do mapeamento UV das roupas da personagem.



Fonte: Elaborado pelo autor

Figura 36 – Layout de UDIM do modelo da personagem.



Fonte: Elaborado pelo autor.

3.4.2. *Baking de texturas*

O *bake* das texturas de um modelo *high poly* permite que alguns dados da malha deste objeto possam ser convertidos para o modelo *low poly*, a fim de melhorar a qualidade deste último sem que sua topologia seja alterada. A informação de topologia do modelo *high poly* é convertida em imagens bitmap que depois são aplicadas em um modelo *low poly*, criando a ilusão de que essa malha possui uma topologia mais detalhada em sua superfície do que realmente tem.

Mas antes que esse processo fosse executado, os modelos *high poly* da personagem precisavam ser identificados, através de um processo chamado de pintura de vértices, em que cada vértice da malha dos objetos recebe uma cor (Figura 37). Esse processo permite que cada conjunto de vértices que possuem uma mesma cor armazenada, possam ser identificados posteriormente para ser manipulados juntos, tanto no Blender como em outros softwares de produção 3D.

O software empregado para a criação das texturas dos objetos foi o Substance Painter¹⁴ que é um dos softwares utilizados no mercado atualmente para essa finalidade, mas ele também possui uma ferramenta de *bake* de objetos, que também foi utilizada nesta etapa. Durante o *bake*, a pintura de vértices do modelo *high poly* foi especialmente útil para que fossem criadas texturas de identificação das peças de roupas através de um mapa de ID (Figura 38), que depois seriam utilizadas durante o processo de criação de materiais no Substance Painter, já que o objeto equivalente em *low poly* não possui densidade de polígonos suficientes em sua topologia para fazer a distinção de cada peça de roupa da sua contraparte *high poly*.

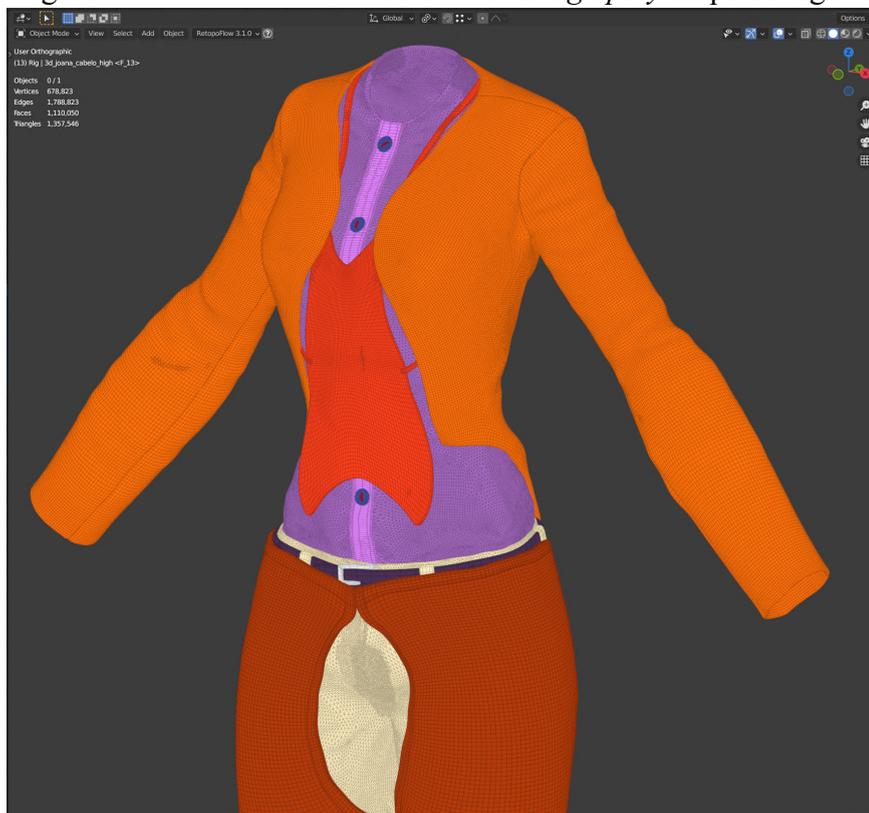
Outros mapas gerados pelo processo de *bake* e que são bastante utilizados no processo de criação de texturas no Substance Painter, são o mapa de Oclusão Ambiente (Figura 39), que é calculada a partir do contato entre objetos tridimensionais próximos, e o mapa de Normal, usado para calcular detalhamento de superfície em modelos *low poly* (Figura 40).

A Figura 37 mostra o modelo *high poly* das roupas da personagem com a pintura de vértices aplicada a cada peça de roupa, diferenciando-as através das cores.

¹⁴ Disponível em <<https://www.substance3d.com/>> Acessado em 11/08/2021

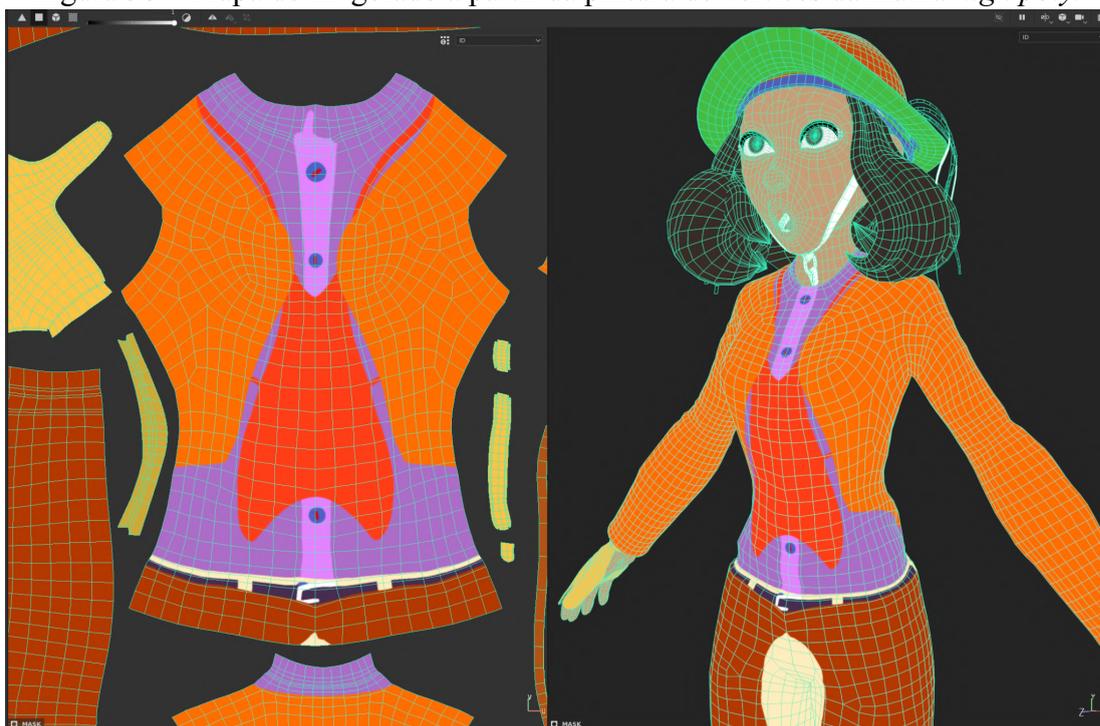
A Figuras 38, 39 e 40 mostram respectivamente os mapas de ID, oclusão ambiente e normal, provenientes do modelo *high poly*, aplicados à topologia do modelo *low poly* através do processo de *baking*.

Figura 37 – Pintura de vértices da malha *high poly* da personagem.



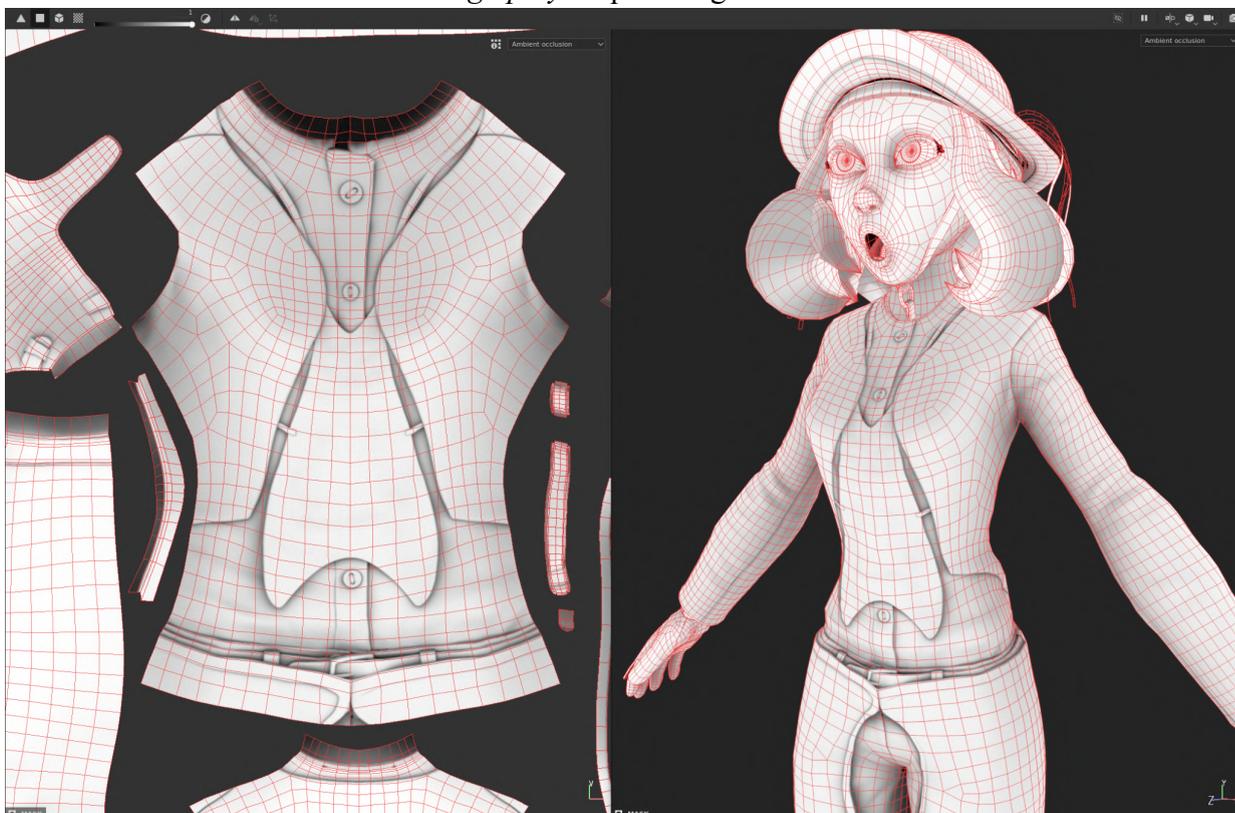
Fonte: Elaborado pelo autor.

Figura 38 – Mapa de ID gerado a partir da pintura de vértices da malha *high poly*.



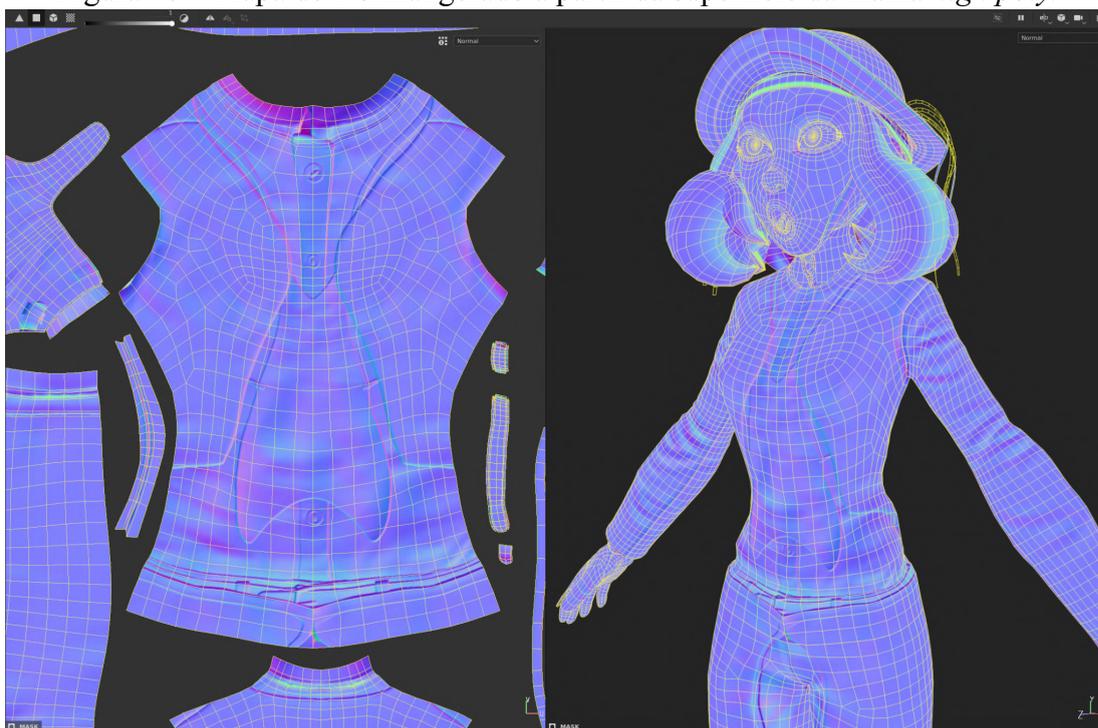
Fonte: Elaborado pelo autor.

Figura 39 – Mapa de Oclusão Ambiente gerado a partir do contato entre geometrias na malha *high poly* da personagem.



Fonte: Elaborado pelo autor.

Figura 40 – Mapa de Normal gerado a partir da superfície da malha *high poly*.



Fonte: Elaborado pelo autor.

O primeiro passo dentro do fluxo de trabalho do Substance Painter, foi a criação do *bake* a partir da modelagem *high poly* da personagem, mas depois de alguns testes, foram percebidos artefatos resultantes do processo (Figuras 41 e 42). O motivo para a criação desses artefatos era que todos os objetos utilizavam o mesmo material atribuído ao modelo durante o processo de criação de UDIMs, logo, eles compartilhavam as mesmas texturas, que sofriam influência de todos os objetos na cena. A solução encontrada foi separar os objetos a uma distância em que o *bake* das texturas fosse gerado sem que um objeto interferisse nos demais (Figura 43).

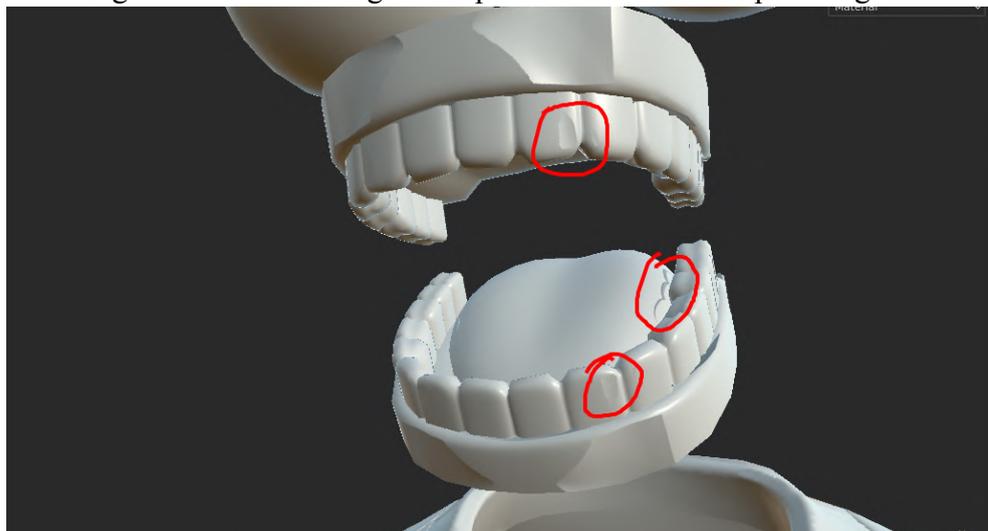
Neste momento, como parte do trabalho de criação de texturas já havia sido feito, o processo de correção de *bake* do modelo foi executado em um arquivo de projeto do Substance Painter separado. Os mapas bitmap gerados no novo arquivo de projeto (Figura 44) foram exportados para um diretório no computador de trabalho, e depois importados no arquivo anterior de projeto do Substance Painter, substituindo os mapas defeituosos gerados no primeiro *bake*.

A Figuras 41 e 42 mostram artefatos gerados durante o primeiro processo de *baking* aplicado ao modelo da personagem.

A Figura 43 mostra o modelo da personagem com várias partes da malha separadas, antes do processo de correção de *bake*.

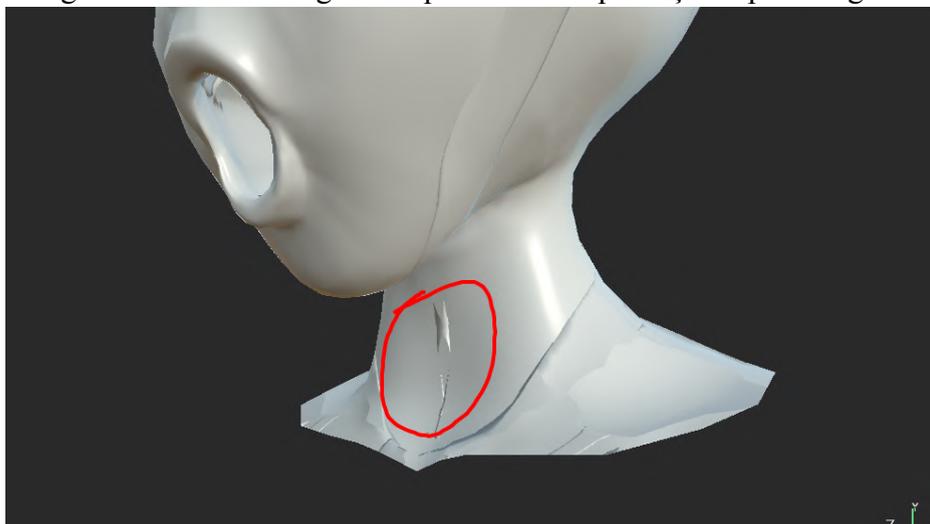
A Figura 44 mostra os mapas gerados a partir da correção do processo de *bake*, que foram posteriormente aplicados ao modelo *low poly* da personagem.

Figura 41 – Artefatos gerados pelo *bake* na boca da personagem.



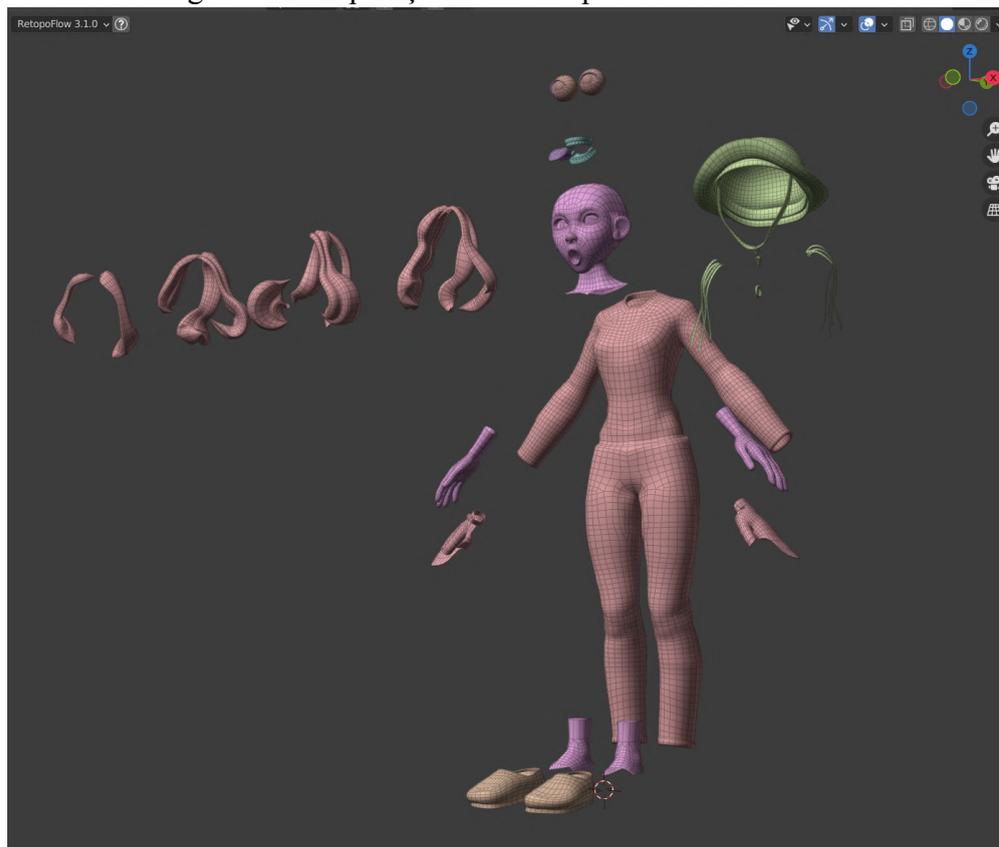
Fonte: Elaborado pelo autor.

Figura 42 – Artefatos gerados pelo *bake* no pescoço da personagem.



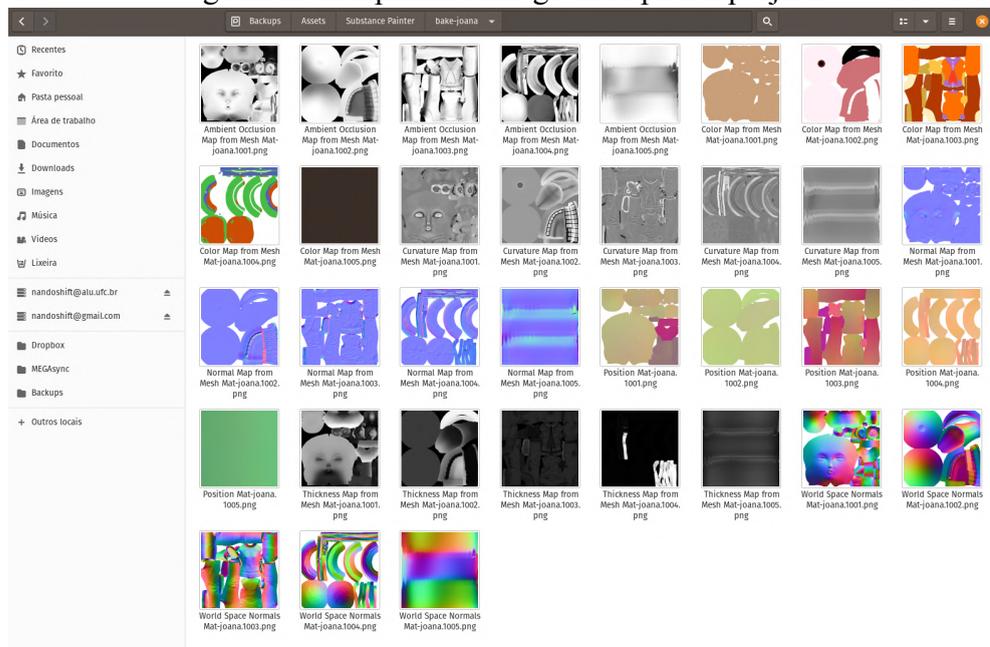
Fonte: Elaborado pelo autor.

Figura 43 – Separação da malha para *bake* do modelo.



Fonte: Elaborado pelo autor.

Figura 44 – Mapas de *bake* gerados para o projeto.



Fonte: Elaborado pelo autor.

3.4.3. *Shaders PBR e pintura 3D*

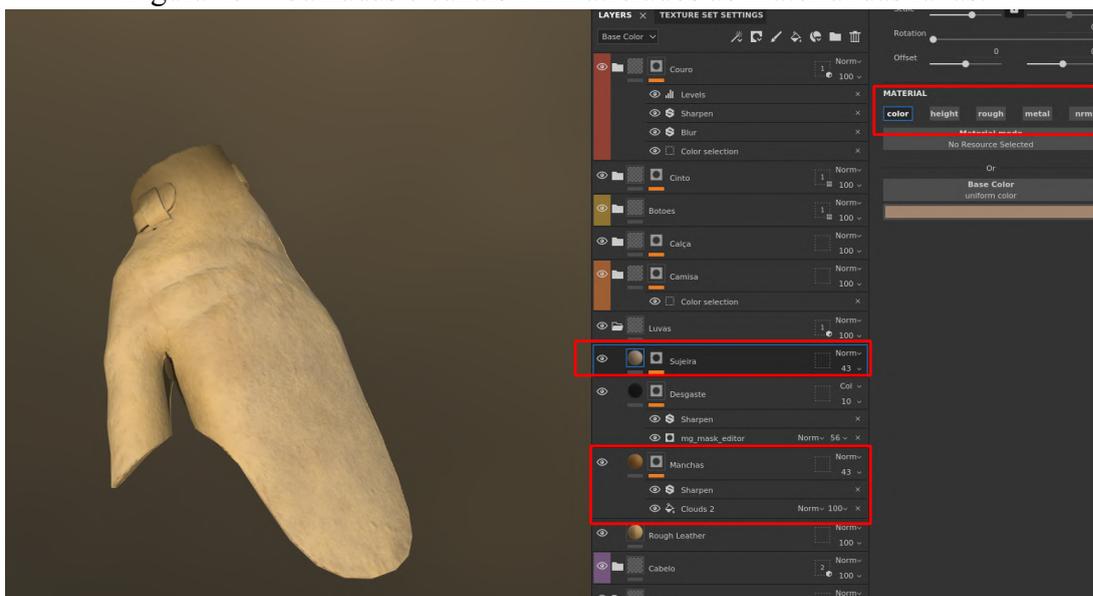
Após a exportação da malha e criação dos mapas de *bake* no Substance Painter, o trabalho de criação das texturas foi iniciado tendo em mente a lógica de produção para *shaders* PBR que representasse os materiais com maior fidelidade possível. Aqui é importante frisar que embora o objetivo para este projeto quanto ao estilo seja representar os materiais de forma realística, o fluxo de trabalho do Substance Painter não está restrito a esse tipo de representação, podendo ser usado para projetos com estilo visual mais cartunesco sem nenhuma limitação. Logo a escolha do software foi pautada pela familiaridade prévia com as ferramentas que ele possui.

No fluxo de trabalho dentro do Painter, através da lógica de produção PBR, a produção dos *shaders* foi primeiramente dividida em grupos contendo os principais materiais para cada uma das superfícies representadas: (1) caçados, (2) pele, (3) dentes, (4) superfícies internas da boca (língua e bochechas), (5) chapéu, (6) cabelo, (7) luvas, (8) camisa, (9) calça, (10) botões da camisa, (11) cinto, (12) peças de couro do dorso e membros (vestimenta de vaqueira), (13) cílios e (14) olhos.

Depois cada grupo foi subdividido em camadas específicas, que poderiam representar um material completo, como a sola dos calçados da personagem, ou um canal PBR isolado para um material que representasse uma particularidade específica sua (Figura 45), como cor da pele, metalicidade do fecho do chapéu ou porosidade do couro do gibão por exemplo. Essa subdivisão permitiu que fossem feitos ajustes específicos e correções em várias características dos materiais do projeto, o que permitiu um controle completo da forma como cada um deles foi representado.

A Figura 45 mostra a luva da personagem com material que simula camurça aplicado, com as camadas de material controlando aspectos específicos da representação gráfica da superfície.

Figura 45 – Camadas e canais PBR atrelados ao material das luvas.



Fonte: Elaborado pelo autor.

O processo de autorização dos materiais aconteceu de duas formas: a primeira foi a partir da criação de uma camada vazia para cada material, seguido de um processo iterativo de adição, teste e ajuste de canais PBR para representar as características físicas de cada um desses materiais, até que se chegasse a um resultado satisfatório baseado nas referências imagéticas do projeto (Figura 46). E a segunda foi a adição e ajuste de materiais prontos do próprio Substance Painter chamados de *Smart Materials*, que representam superfícies de materiais complexos ajustáveis à malhas diferentes, permitindo uma flexibilidade muito grande e um ótimo ponto de partida para a criação de uma material com características únicas, que no caso deste projeto, também foram guiadas pelas referências imagéticas coletadas na etapa de conceituação visual.

A Figura 46 exibe um painel semântico com referências de materiais para os calçados da personagem, cada um representando variações de superfícies de couro.

Figura 46 – Painel semântico com referências para o material dos calçados.



Fonte: Elaborado pelo autor.

Devido ao fato de quase todas as peças de roupa da personagem compartilharem a mesma malha, o processo de criação dos seus materiais foi um pouco mais complexo que nas demais superfícies. O primeiro problema encontrado foi que as texturas de algumas peças de roupa não estavam sendo projetadas da maneira correta sobre as UVs do objeto, o que causavam alguns artefatos. Felizmente o Substance Painter apresenta algumas opções de projeção das texturas sobre a malha, e o problema pode ser resolvido com a alteração da projeção padrão para outras formas, como uma esfera, por exemplo (Figura 47).

O segundo problema encontrado foram artefatos exibidos nas junções dos materiais que compartilhavam a mesma malha. Os artefatos consistiam em linhas brancas nos limites de cada material (figuras 48 e 49) causando um espaçamento estranho nas junções desses materiais. Depois de alguns testes, o problema foi resolvido com a aplicação de filtros para correção de contraste e suavização dos pixels no mapa de ID (Figura 50), que guarda as cores dos vértices da versão *high poly* das roupas. O problema não foi completamente resolvido, mas foi bastante reduzido até que quase não se percebe no material finalizado (Figura 51).

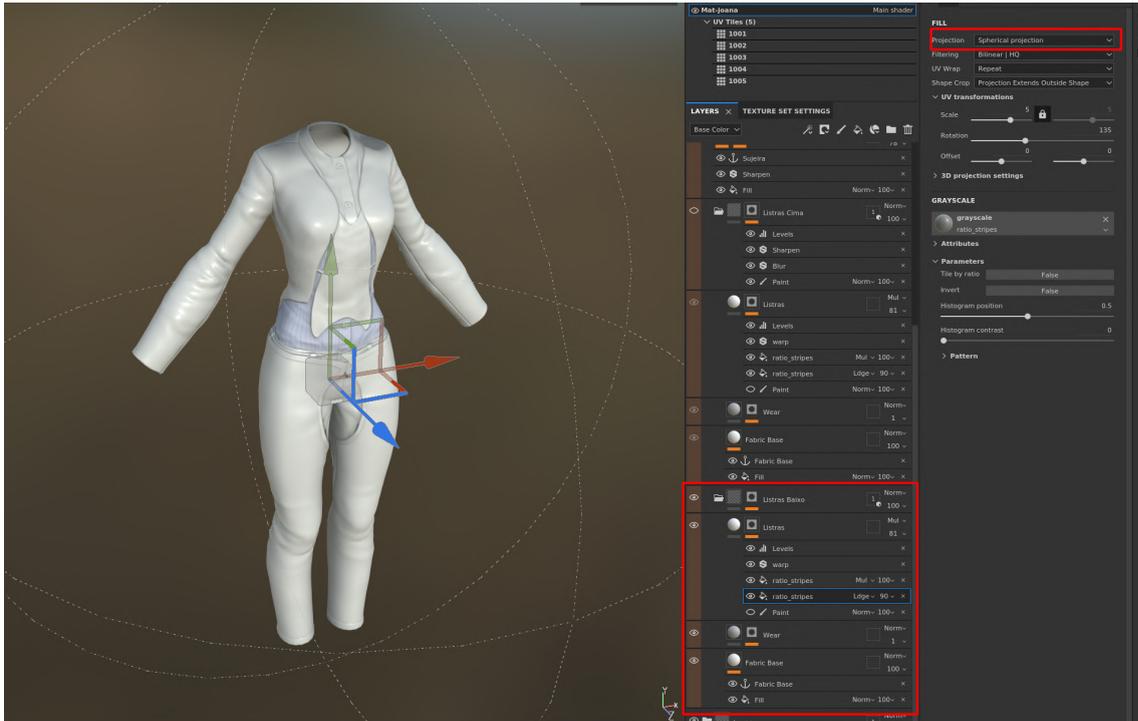
A Figura 47 mostra uma projeção esférica de textura sobre a malha das roupas da personagem.

As Figuras 48 e 49 mostram artefatos presentes nas junções dos materiais das roupas da personagem.

A Figura 50 mostra filtros de ajustes, destacados em vermelho, aplicados nas camadas que separam os materiais da roupa, amenizando os efeitos desses artefatos.

A Figura 51 mostra os materiais da roupa em suas versões finalizadas, com os artefatos corrigidos.

Figura 47 – Projeção esférica de uma textura sobre a malha das roupas.



Fonte: Elaborado pelo autor.

Figura 48 – Artefatos na divisão entre materiais.



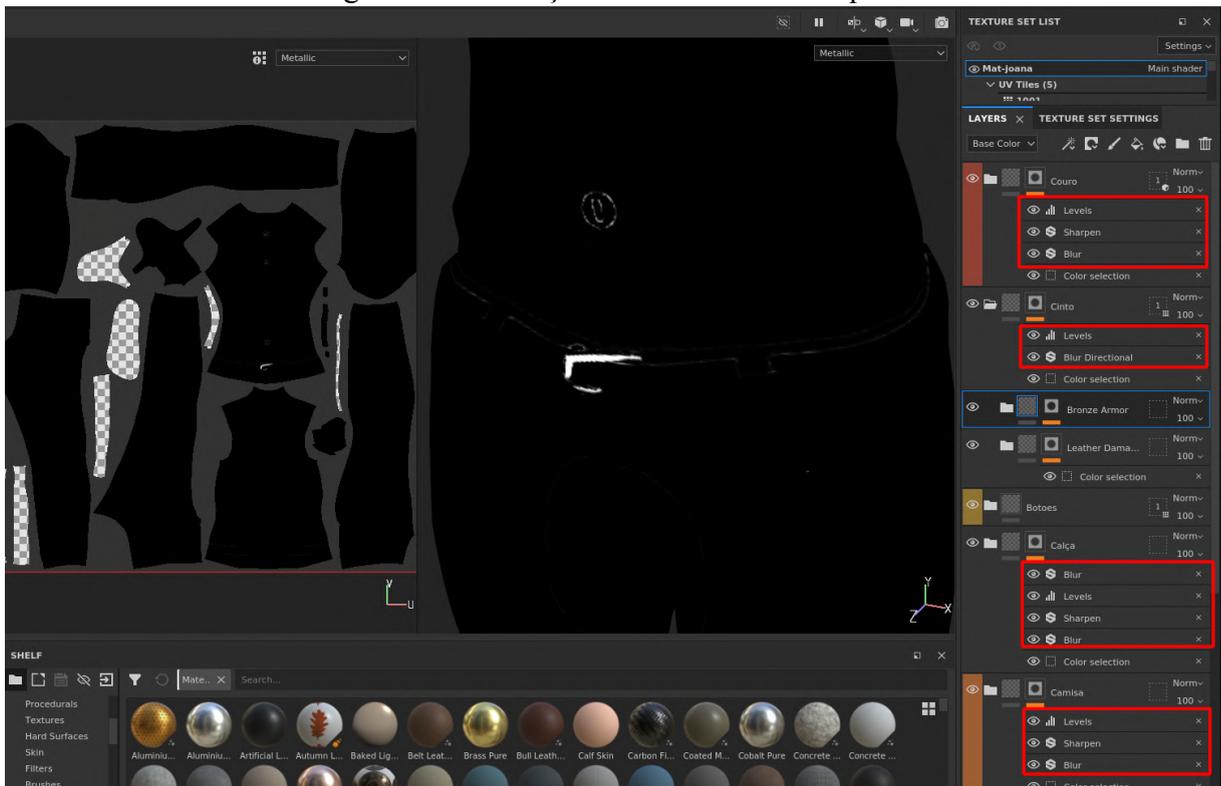
Fonte: Elaborado pelo autor.

Figura 49 – Artefatos isolados em uma máscara.



Fonte: Elaborado pelo autor.

Figura 50 – Correção de artefatos das roupas.



Fonte: Elaborado pelo autor.

Figura 51 – Resultado final dos materiais das roupas.



Fonte: Elaborado pelo autor.

No caso do material da pele, apesar de não ser muito complexo, foi o que demandou mais etapas de produção. Ele pôde ser dividido em poucas etapas principais, mas a quantidade de camadas de ajustes e de máscaras utilizadas foi a maior entre todos os materiais.

A primeira etapa foi a adição e ajuste de um material base para a pele, com a ajuda de um material padrão do próprio Substance Painter (Figura 52). Em seguida, a pele recebeu várias máscaras para correção de cor em áreas específicas do rosto (Figura 53), o que ajudou a acentuar os contornos de cada área, principalmente no nariz e nos olhos. A terceira etapa foi adicionar camadas de detalhamento e ajustes para controlar cada canal específico do material, como por exemplo aumentar ou diminuir a oleosidade da pele em áreas específicas, através do canal de rugosidade (*i.e. roughness*). E por último foram adicionadas imperfeições na pele, para ajudar a caracterizar a personagem em um ambiente mais hostil. Essa etapa foi a menos complexa, dividida em apenas uma camada para queimaduras de sol e outra para manchas e sardas da pele. Apesar de ter sido a etapa menos complexa, essas imperfeições foram o elemento de caracterização que mais se destacou no material da pele.

A Figura 52 mostra um material base de pele aplicado sobre a malha da personagem.

A Figura 53 mostra uma das máscaras de ajuste de cor aplicadas ao material da pele.

As Figuras 54 e 55 mostram respectivamente as texturas de normal e rugosidade aplicadas ao modelo da personagem.

A Figura 56 mostra uma camada criada para simular as queimaduras de sol sofridas na pele da personagem.

A Figura 57 mostra uma camada criada para simular manchas e sardas na pele da personagem.

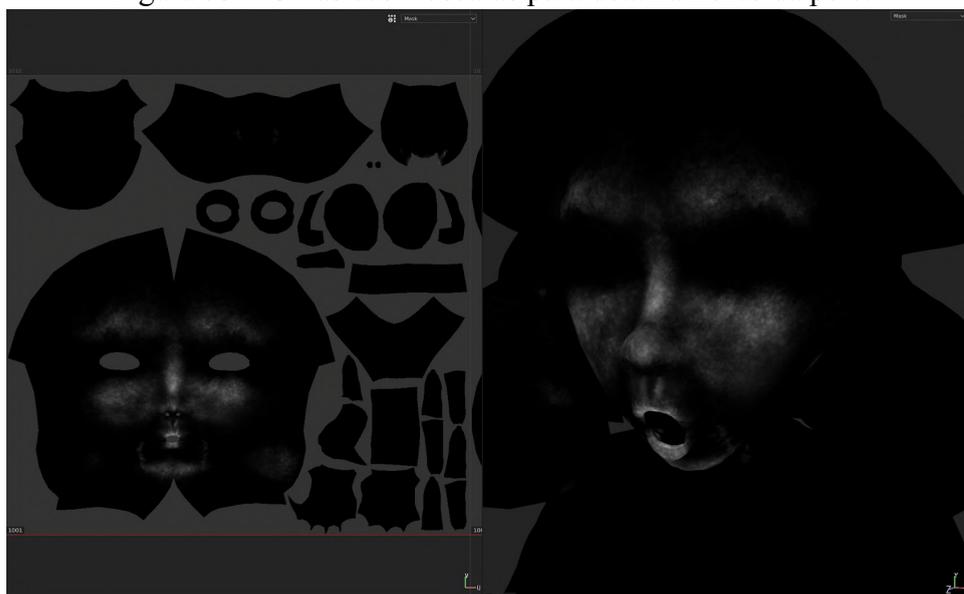
A figura 58 exibe uma comparação do rosto da protagonista entre a pele sem imperfeições ou cicatrizes, e a mesma pele com as imperfeições aplicadas.

Figura 52 – Material base para o rosto da personagem.



Fonte: Elaborado pelo autor.

Figura 53 – Umas das máscaras para detalhamento da pele.



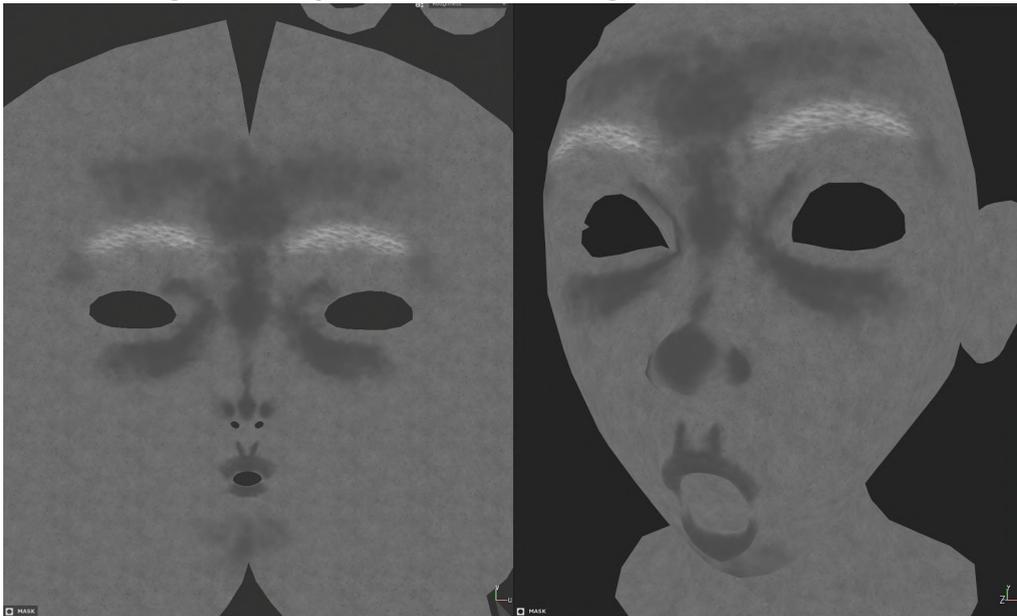
Fonte: Elaborado pelo autor.

Figura 54 – Ajustes nos canais de normal do material.



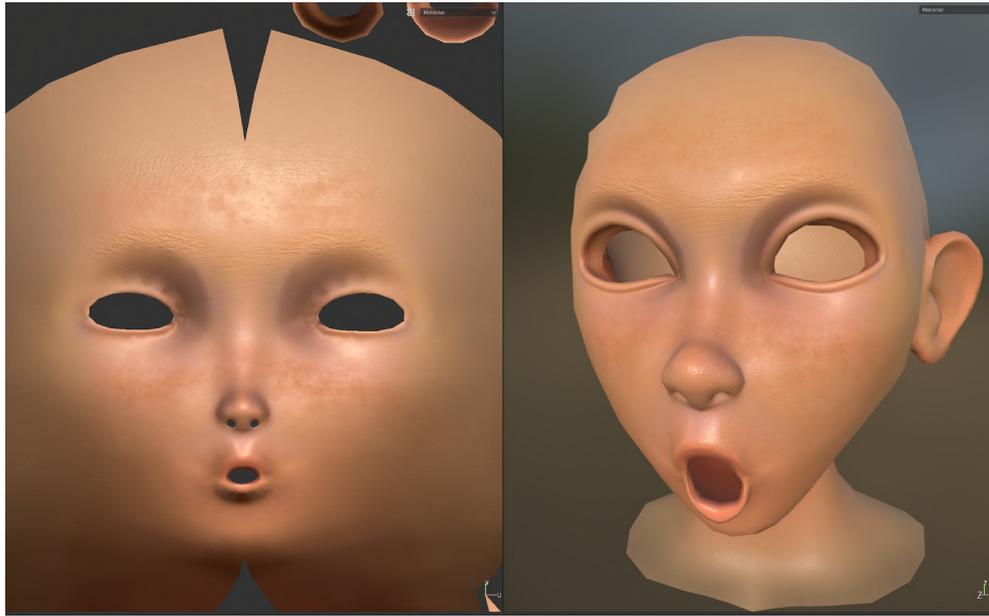
Fonte: Elaborado pelo autor.

Figura 55 – Ajustes nos canais de rugosidade do material.



Fonte: Elaborado pelo autor.

Figura 56 – Adição de queimaduras de sol na pele.



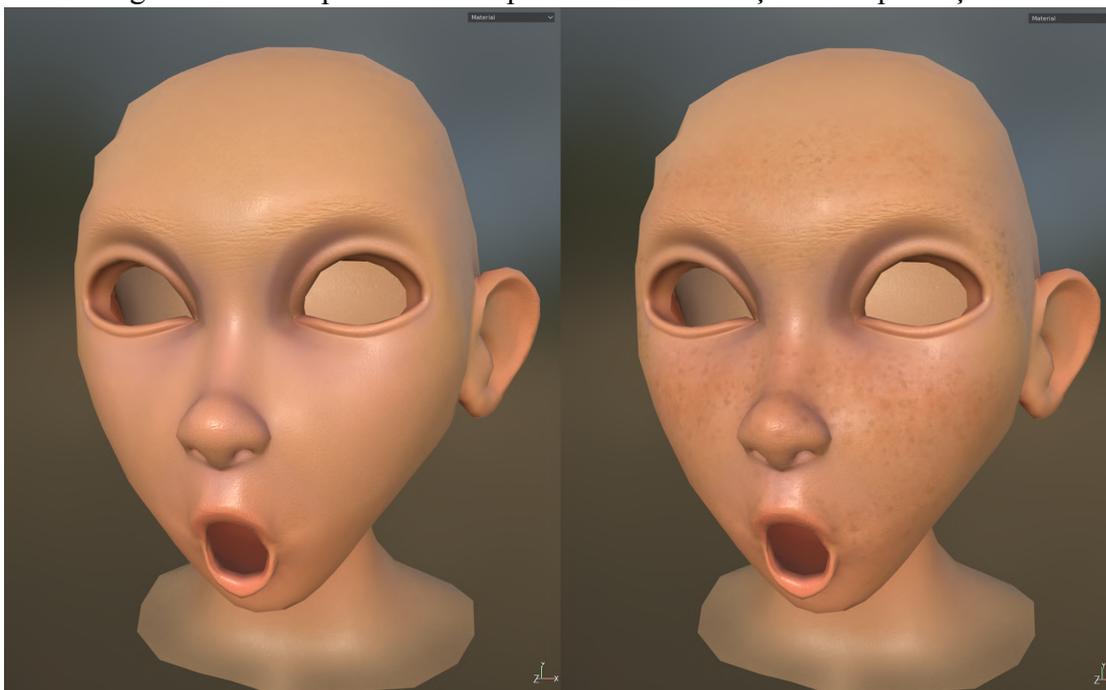
Fonte: Elaborado pelo autor.

Figura 57 – Adição de sardas na pele.



Fonte: Elaborado pelo autor.

Figura 58 – Comparativo entre pele sem e com adição de imperfeições.



Fonte: Elaborado pelo autor.

Para o material das peças de roupa principais de couro, que representam a “armadura” da personagem, foi feito um fluxo de trabalho dividido em quatro partes: (1) cor base do material, (2) textura de couro, que representa a rugosidade da superfície em seu estado natural, (3) um desgaste leve aplicado às dobras do material e (4) um desgaste pesado, aplicada às áreas que indicariam maior contato do material com outras superfícies e objetos que o danificam.

Na primeira parte foi definido o tom que seria utilizado para o couro, ajustando apenas o canal de *Base Color* do material, com uma cor de aspecto ferroso aplicada com pequenas variações de cor ajustadas de maneira procedural (Figura 59). Na segunda parte foi utilizado um *Smart Material* com padrão de textura de um material de couro já existente na biblioteca do programa, mas alterando só os canais de *Normal* e *Height* do material em questão (Figura 60). Na terceira parte a intenção foi de fazer um desgaste que desse a impressão que o couro já era velho, acentuando o efeito de muita exposição ao sol, mas com as partes altas das dobras já desgastadas por causa do manuseio e abrasão com outros objetos (Figura 61). E por último foram aplicadas camadas com desgaste ainda mais acentuado, para denotar dificuldades vividas pela personagem, como quedas e colisões com galhos (Figura 62).

É importante destacar o uso de máscaras específicas durante o processo de criação destas camadas. Para fazer o desgaste leve, foi utilizada uma camada procedural que é

baseada nos mapas gerados durante o *bake* do modelo 3D. Após aplicada, a máscara precisou apenas de alguns ajustes para que fosse atingido um resultado satisfatório (Figura 63). Mas para a criação do desgaste mais severo, a máscara utilizada precisou de mais ajustes, pois as áreas de desgaste acabaram ficando mais convincentes quando foram posicionadas de maneira arbitrária (Figura 64).

O restante dos materiais seguiu o mesmo fluxo de trabalho da “armadura” de couro, com pouca ou nenhuma alteração significativa. Sempre com uma divisão de material base, detalhamento, e aplicação de desgaste e/ou sujeira (Figura 63).

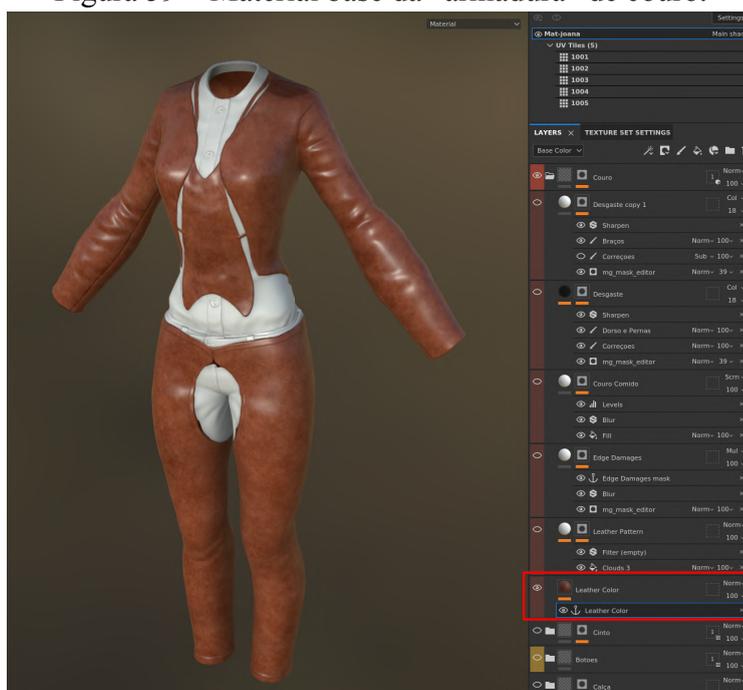
As Figuras de 59 a 62 exibem um passo-a-passo do desenvolvimento de um material de couro para o modelo das roupas da personagem.

As figuras 63 e 64 exibem respectivamente máscaras usadas para a aplicação de efeitos de desgaste leve e desgaste pesado nas roupas da personagem.

A Figura 65 exhibe o material da “armadura” finalizado, mostrando a textura usada para mapeamento UV das roupas e o próprio modelo tridimensional delas.

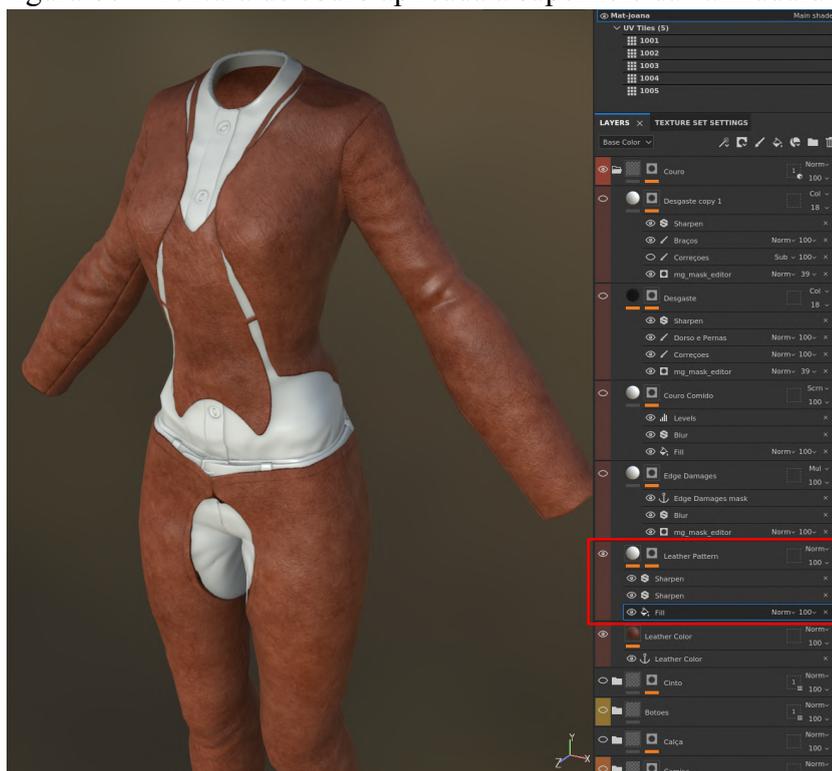
A Figura 66 Exibe o material do chapéu finalizado, mostrando a textura usada para o seu mapeamento UV, além do próprio modelo tridimensional do chapéu.

Figura 59 – Material base da “armadura” de couro.



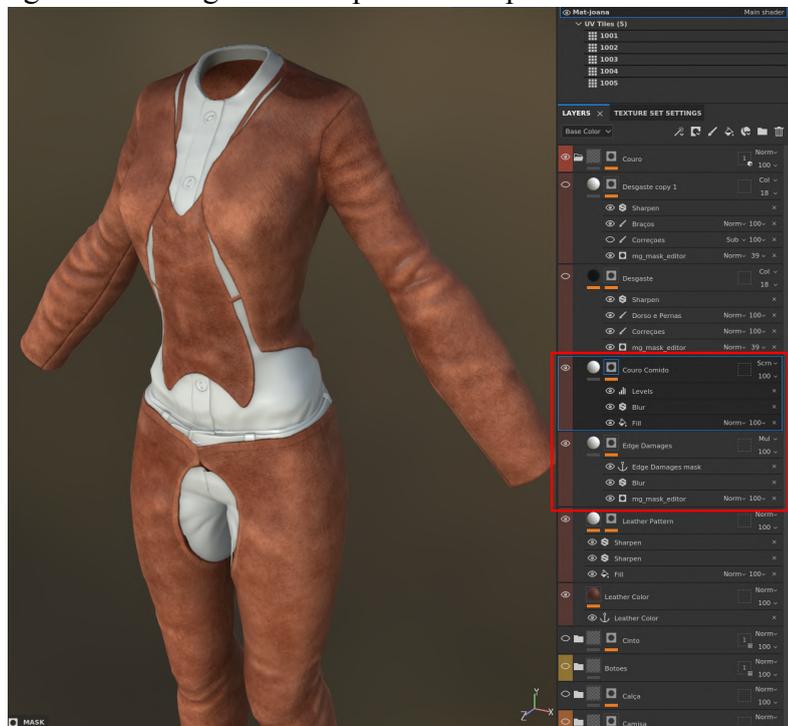
Fonte: Elaborado pelo autor.

Figura 60 – Textura de couro aplicada à superfície da “armadura”.



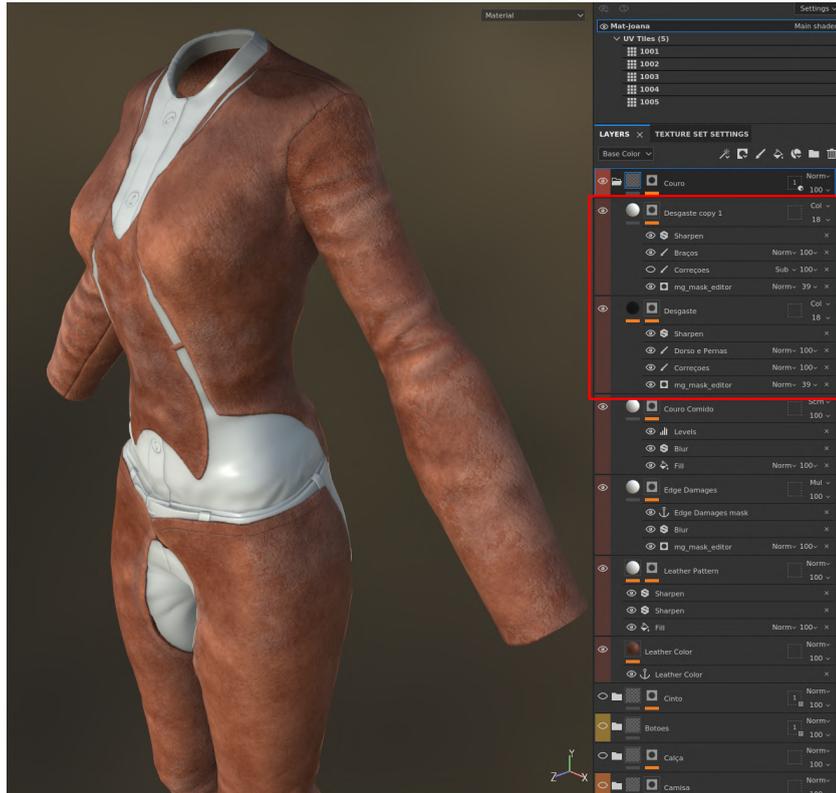
Fonte: Elaborado pelo autor.

Figura 61 – Desgaste leve aplicado à superfície da “armadura”.



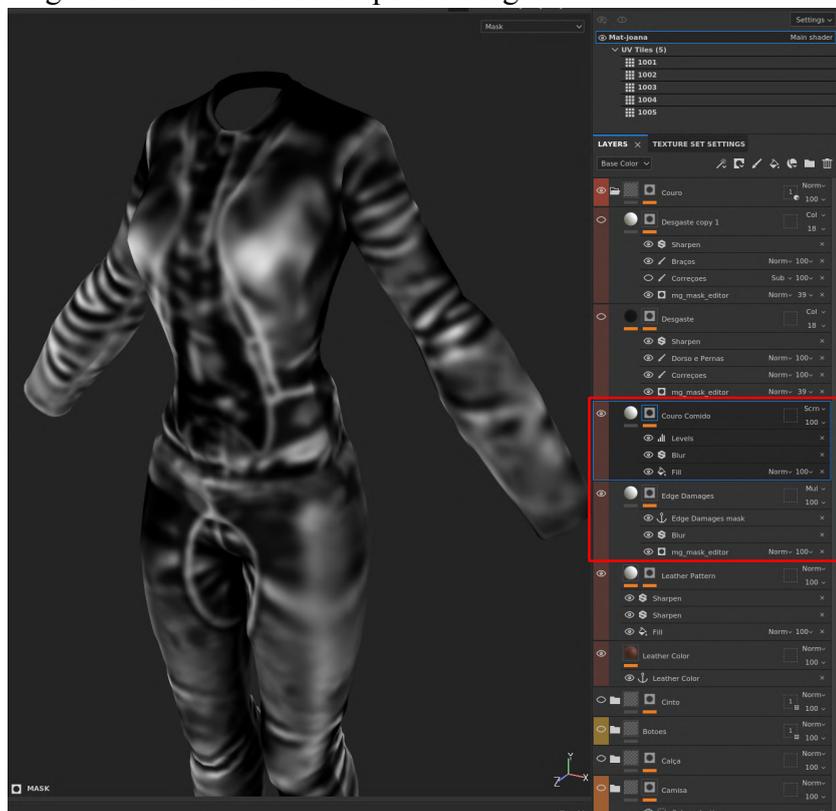
Fonte: Elaborado pelo autor.

Figura 62 – Desgaste acentuado aplicado à superfície da “armadura”.



Fonte: Elaborado pelo autor.

Figura 63 – Máscara usada para o desgaste leve da “armadura”.



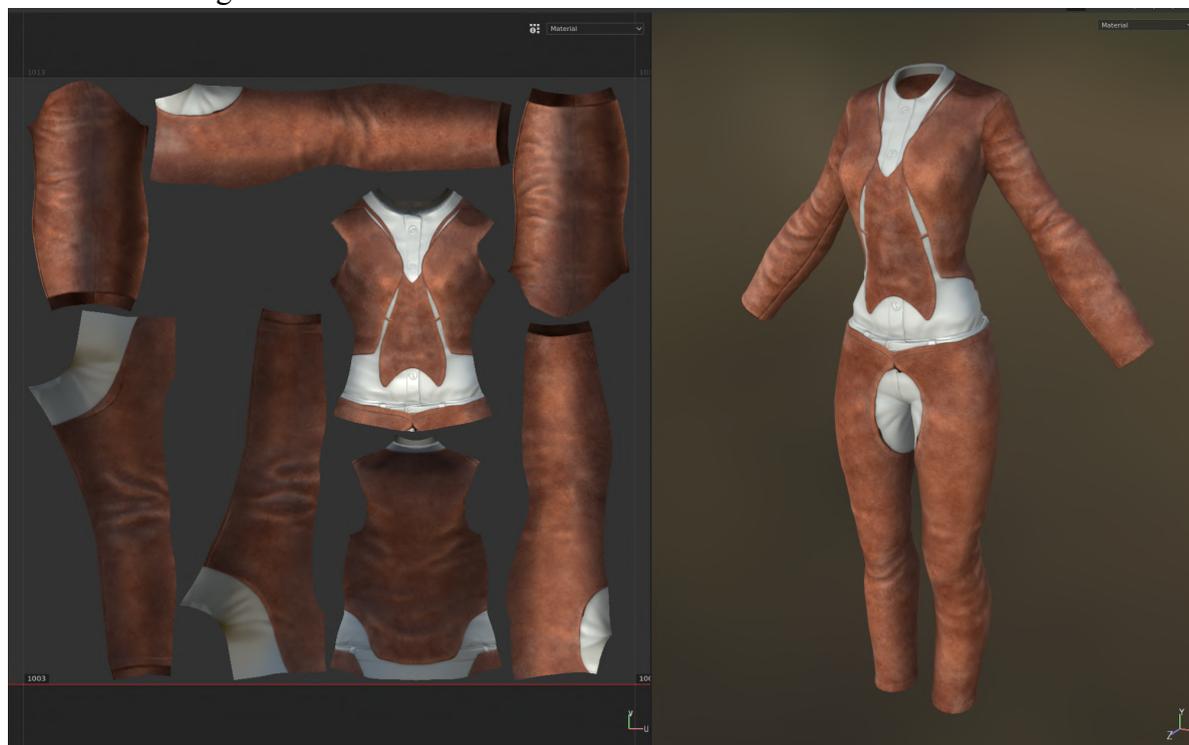
Fonte: Elaborado pelo autor.

Figura 64 – Máscaras usadas para o desgaste pesado da “armadura”.



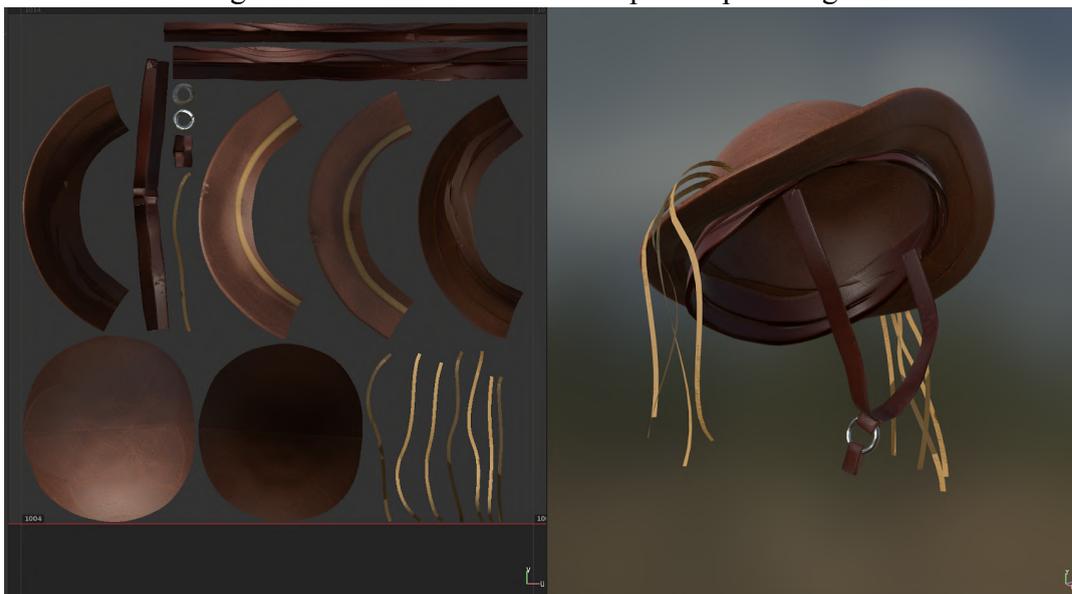
Fonte: Elaborado pelo autor.

Figura 65 – Resultado final do material de couro da “armadura”.



Fonte: Elaborado pelo autor.

Figura 66 – Material PBR do chapéu da personagem.



Fonte: Elaborado pelo autor.

Para o material dos cílios, foram identificados dois problemas devido a ajustes que precisaram ser feitos em sua malha e mapeamento UV do modelo: O primeiro foi que, após os ajustes, o processo de bake dos objetos precisou ser refeito, o que causou a perda de algumas das camadas criadas durante o processo de autoração de materiais, devido ao fato de vários parâmetros dessas camadas serem definidos a partir dos mapas de bake feitos antes.

E o segundo problema foi que com os ajustes, foi definida uma forma de representação dos cílios que precisaria que um canal de transparência fosse adicionado à uma das texturas exportadas para a UE4, o que aumentaria a quantidade de informação nessa textura, necessitando de um método de renderização específico para a representação correta dessa informação adicional de transparência no *shader* da personagem. E neste caso, já que o *shader* utilizado seria aplicado a malha inteira devido ao método de mapeamento UDIM (que unifica todos os materiais em um único *shader*), a utilização desse método de renderização causaria um desperdício de recursos computacionais, já que a informação referente à transparência do *shader* seria aplicada a cada pixel do corpo e roupas da personagem exibidos em tela, o que representa informação extra armazenada na memória do computador durante a renderização, mesmo que apenas os cílios fosse exibidos com o canal de transparência.

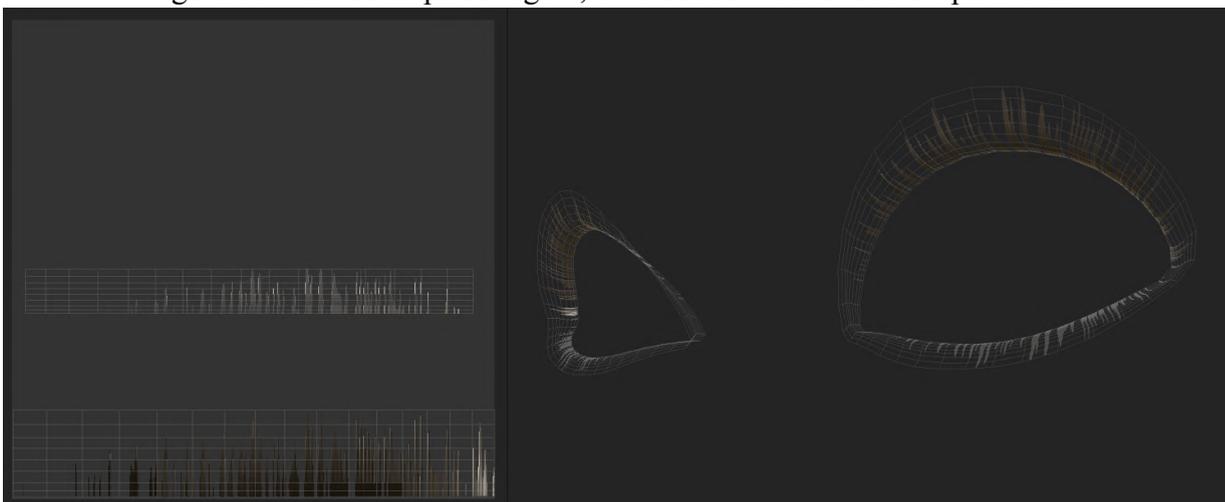
Este seria um problema de otimização muito difícil de ignorar, já que os cílios representam uma área da superfície muito pequena do modelo da personagem. Então, como forma de evitar um trabalho extra de refação de algumas das camadas perdidas para os materiais feito até essa etapa do projeto, e também, evitar o desperdício de recursos

computacionais e uma possível queda de desempenho do produto final, a solução adotada foi a de trabalhar o material dos cílios separado dos outros, exportando-o para a UE4 através de texturas específicas compatíveis com o canal de transparência.

Para o material dos olhos, optou-se pela utilização de um shader distribuído no projeto gratuito Digital Human¹⁵, que é feito na UE4 e possui assets preparados para importação em outros projetos. A Decisão de utilizar esse *shader* para os olhos se deu com o intuito de utilizar as ferramentas de personalização do shader, que possibilitam uma série de ajustes de personalizações para criação de materiais para olhos foto realistas.

A Figura 67 mostra os cílios da personagem e sua textura com a adição de um canal de transparência.

Figura 67 – Cílios da personagem, exibidos com canal de transparência.



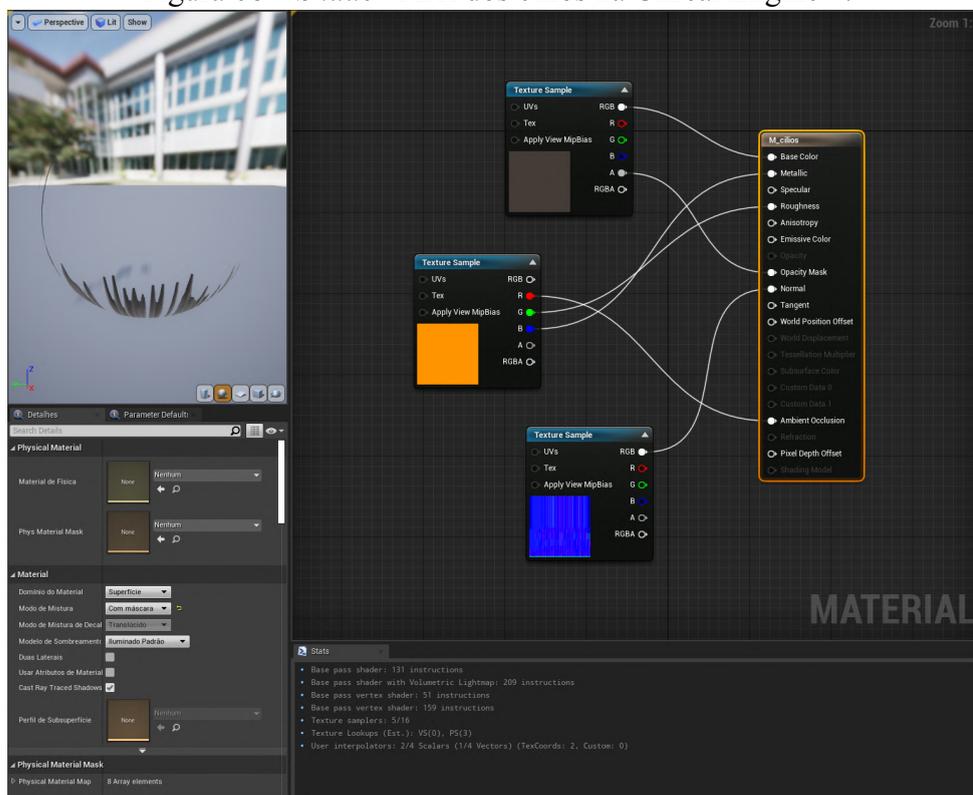
Fonte: Elaborado pelo autor.

Ao final do processo de criação dos materiais, cada um é exportado em texturas no formato de imagens bitmap PNG. No padrão de produção da Unreal Engine 4, cada material recebeu três imagens ao todo que contém os canais para renderização do shader PBR (Figura 68). A primeira textura contém, com sufixo (1) “BaseColor”, armazena o canal de cor do material no canal RGB da imagem, e no caso dos cílios, também armazena o canal alpha para transparência. A segunda textura, com o sufixo (2) “OcclusionRoughnessMetallic”, armazena os canais de oclusão ambiente, rugosidade e metalicidade do material, que são armazenados nos canais R, G e B da imagem respectivamente. A última textura contém apenas o canal de normal do material, que também fica armazenado no canal RGB da imagem e recebe o sufixo (3) “Normal”.

¹⁵ Disponível em <<https://tinyurl.com/3w58pvwe>> Acessado em 27/08/2021.

A Figura 68 mostra o shader dos cílios composto no editor de materiais da UE4, em que cada canal PBR que foi utilizado no shader está conectado aos canais das texturas geradas no Substance Painter.

Figura 68 – Shader PBR dos cílios na Unreal Engine 4.



Fonte: Elaborado pelo autor.

3.5. Animação

Existem vários *rigs* compatíveis com o Blender, e cada um tem uma organização específica que permite ajustes mais ou menos fáceis e focados em objetivos específicos. O Rigify foi a primeira escolha para este projeto por já vir inclusa na versão padrão do Blender, aproveitando os principais métodos de rig do software sem a necessidade de criar um rig inteiro e complexo do zero, ou instalar uma ferramenta externa. Mas o motivo principal foi que o Rigify automatiza várias etapas do rig além de fornecer uma grande quantidade de ajustes, o que permitiu tornar as etapas de teste e validação da malha low poly da personagem em um processo iterativo e muito mais rápido de se fazer correções.

Entretanto, o *rig* gerado pelo Rigify tem alguns problemas de compatibilidade com a UE4, que foi o motor gráfico utilizado, em sua versão 4.26, para validar o fluxo de produção da personagem. A UE4 permite uma compatibilidade com outros sistemas de animação utilizados na indústria de jogos, como por exemplo o Mixamo da Adobe, mas por possuir um

sistema de animação mais simples, além dos problemas de compatibilidade, acaba tornando o rig gerado pelo Rigify mal otimizado após ser exportado para o motor gráfico.

Então quando o projeto entrou na etapa de animação, em que o modelo e texturas já estavam concluídos, percebeu-se a necessidade de utilizar um sistema de animação que permitisse uma compatibilidade maior com os fluxos de trabalho da Unreal, e que permitisse uma exportação mais rápida para a engine. O Rigify acabou restrito apenas à fase de teste e validação da deformação da malha, e após alguns testes de possíveis candidatos durante a etapa de animação, acabou substituído pelo Auto-Rig Pro¹⁶, que tem melhor compatibilidade com a UE4, além de possuir uma ferramenta de exportação mais simples de se utilizar.

O processo de animação da personagem aconteceu em dois momentos diferentes, divididos em um fluxo de trabalho com o Rigify, durante a etapa de *rigging*, e um fluxo de trabalho dentro da UE4, que aconteceu com o modelo e texturas já finalizados após a etapa de *look dev*.

3.5.1. Workflow no Blender com Rigify

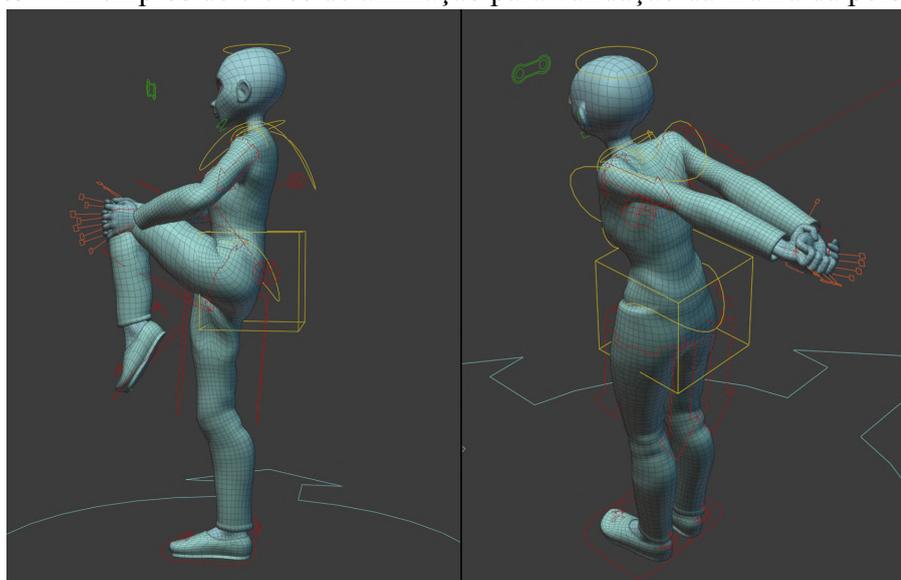
Durante o fluxo de trabalho com o Rigify, a intenção era fazer testes da personagem com os principais ciclos de animações utilizados em jogos de terceira pessoa, a fim de averiguar se a deformação da malha era adequada a mais de um tipo de animação sem que precisasse de ajustes. Durante o processo de validação, foram levados em conta a contração e estiramento de músculos das pernas e braços, principalmente na região dos ombros, e movimentos das articulações do corpo (joelhos, tornozelos, cotovelos, ombros, mandíbulas, pescoço e quadris), avaliando principalmente se haveriam artefatos na superfície do modelo, como invasão de malha e sobreposição de vértices.

Foram criados ainda ciclos de animação para corrida, caminhada, queda e aterrissagem, respiração, que representa os momentos ociosos, onde não há input do jogador, também chamado de ‘animação idle’. Além desses ciclos, foi criada uma animação que mostra a personagem fazendo um exercício de aquecimento muscular, feito especificamente para estressar as articulações do corpo durante a fase de teste e validação (Figura 69).

A Figura 69 exhibe a personagem em duas poses distintas preparadas para teste de topologia do modelo.

¹⁶ Disponível em <<https://blendermarket.com/products/auto-rig-pro>> Acessado em 11/08/2021.

Figura 69 – Exemplos de ciclos de animação para validação da malha da personagem.



Fonte: Elaborado pelo autor.

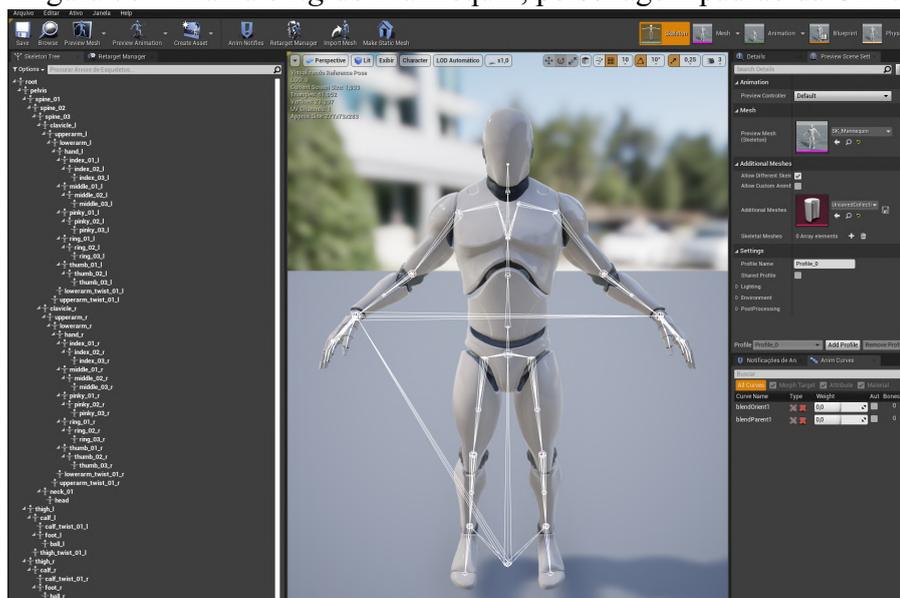
3.5.2. *Workflow na Unreal Engine*

O fluxo de trabalho com a personagem dentro da UE4 foi imaginado para possibilitar tanto uma maior facilidade do processo de animação, quanto para permitir uma integração maior com outras ferramentas de mercado que também sejam compatíveis com a Unreal. Para que essa integração fosse possível, foi preciso testar um método de exportação do *rig* da personagem que o tornasse compatível com o modelo de projeto padrão para jogos em terceira pessoa da Unreal, que possui um personagem próprio chamado Mannequin (Figura 70), com um *rig* e uma biblioteca de animações já implementadas e habilitados à inputs básicos com teclado, mouse e joystick.

O objetivo desse processo foi exportar a malha da personagem de modo que ela pudesse substituir a malha do Mannequin, aproveitando o *rig*, a bibliotecas de animações e os inputs já implementados para ele. Esse processo acaba impossibilitando o uso das animações já criadas para a personagem através do *Rigify*, mas a compatibilidade que resulta desse fluxo de trabalho acaba sendo mais vantajosa, já que permite a adição de vários ciclos de animação compatíveis com a Unreal que podem ser obtidos em bibliotecas online.

A Figura 70 mostra o Mannequin da UE4, que é um personagem padrão utilizado em vários dos seus projetos. Além disso, a Figura exhibe um conjunto de bones deste personagem.

Figura 70 – Malha e *rig* do Mannequin, personagem padrão da UE4.



Fonte: Elaborado pelo autor.

Foram testadas cinco ferramentas de exportação compatíveis com a Unreal dentro do Blender: (1) A ferramenta padrão de exportação de arquivos em FBX do Blender, o (2) Blender For Unreal Engine¹⁷, o (3) Send2Unreal¹⁸, o (4) Uefy 2¹⁹ e o (5) Auto-Rig Pro. Com exceção do Auto-Rig Pro, todas as ferramentas testadas tem um fluxo de trabalho baseado no *Rigify* e, durante o processo de exportação/importação da personagem para a Unreal, apresentaram algum tipo de incompatibilidade com o motor gráfico, demandando correções após o processo, ou necessitando de métodos específicos de preparação do *rig* antes que o processo de exportação fosse aplicado. Além disso, em algumas delas o aproveitamento do *rig* e animação do Mannequin não foi possível, obrigando que o trabalho de implementação de animações e inputs tivesse que ser feito do zero.

O Auto-Rig Pro entretanto, por ser uma ferramenta de *rig* completa que substituiu o *Rigify* (Figura 71), precisou de uma nova etapa de ajustes de pesos dos vértices da malha da personagem, mas em contrapartida apresentou um processo de criação do *rig* e uma compatibilidade com a Unreal que as outras ferramentas não possuíam, tornando a sua utilização mais vantajosa do que as ferramentas anteriores. Com o novo *rig* ajustado e exportado para a Unreal, a integração com o Mannequin e os sistemas implementados para ele ocorreu sem nenhum problema (Figuras 72 e 73), sem que nenhuma animação extra precisasse ser criada para o projeto.

¹⁷ Disponível em <<https://tinyurl.com/57m2fymr>> Acessado em 13/08/2021.

¹⁸ Disponível em <<https://tinyurl.com/4e3azn5j>> Acessado em 13/08/2021.

¹⁹ Disponível em <<https://www.rakiz.com/uefy/>> Acessado em 13/08/2021.

Após a exportação com o Auto-Rig Pro, os únicos ajustes que precisaram ser feitos com relação ao Mannequin foram o posicionamento da câmera, corrigindo sua altura e distância da personagem, e altura da própria personagem, que ficou deslocada do chão após a exportação (Figura 74).

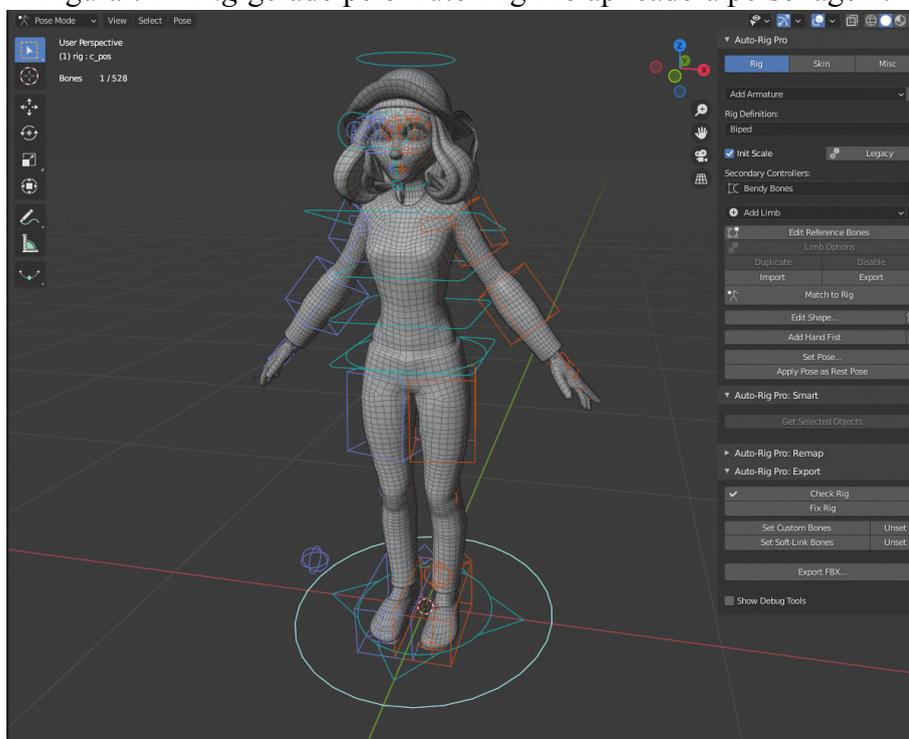
A Figura 71 exhibe o modelo da personagem em um ambiente tridimensional e com a utilização do sistema de controladores para animação chamado de Auto-Rig Pro e sua interface.

As Figuras 72 e 73 mostram respectivamente o modelo tridimensional da personagem já convertida para o padrão de projeto da UE4. Ambas Figuras também mostram a estrutura de animação padrão utilizada no programa.

A Figura 74 Mostra a comparação entre o Mannequin do projeto padrão da UE4 e o modelo de Joana após a exportação.

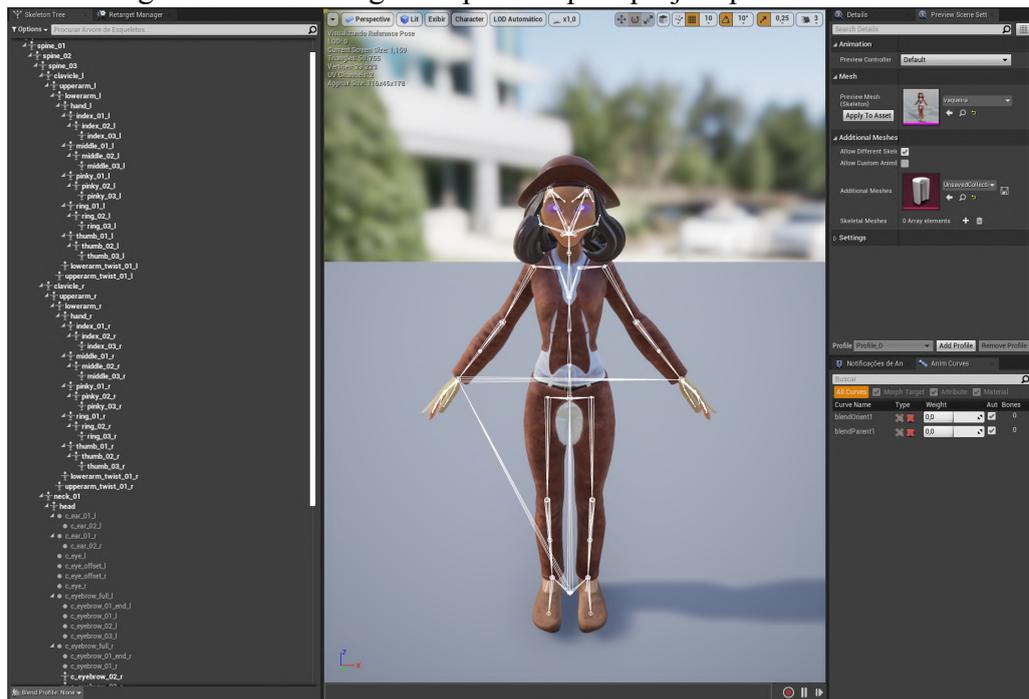
A Figura 75 mostra a comparação entre os modelos do Mannequin e Joana após os ajustes de câmera e posicionamento da personagem.

Figura 71 – Rig gerado pelo Auto-Rig Pro aplicado à personagem.



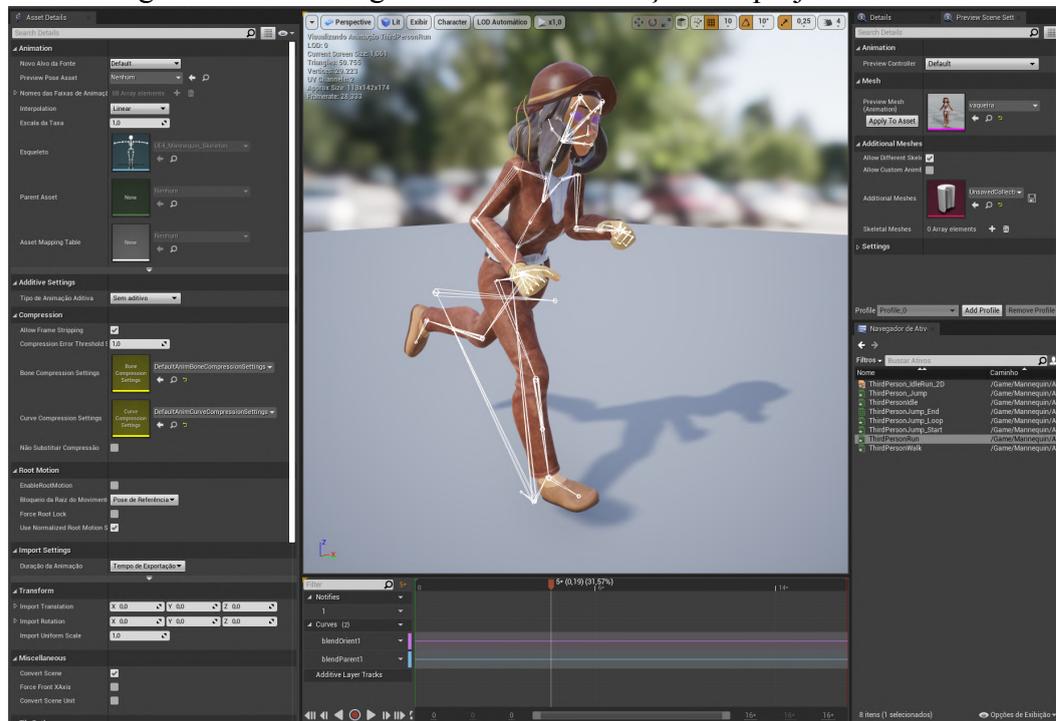
Fonte: Elaborado pelo autor.

Figura 72 – Personagem exportada para projeto padrão da UE4.



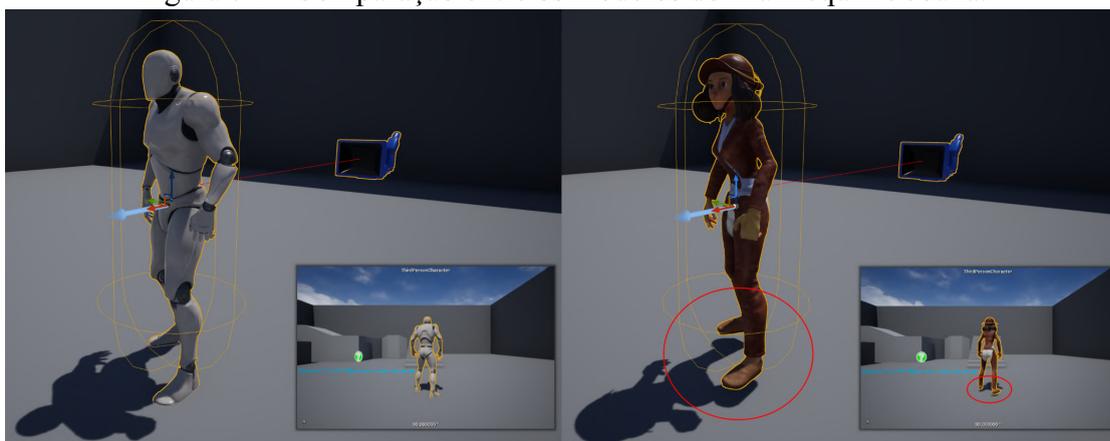
Fonte: Elaborado pelo autor.

Figura 73 – Personagem utilizando animações do projeto da UE4.



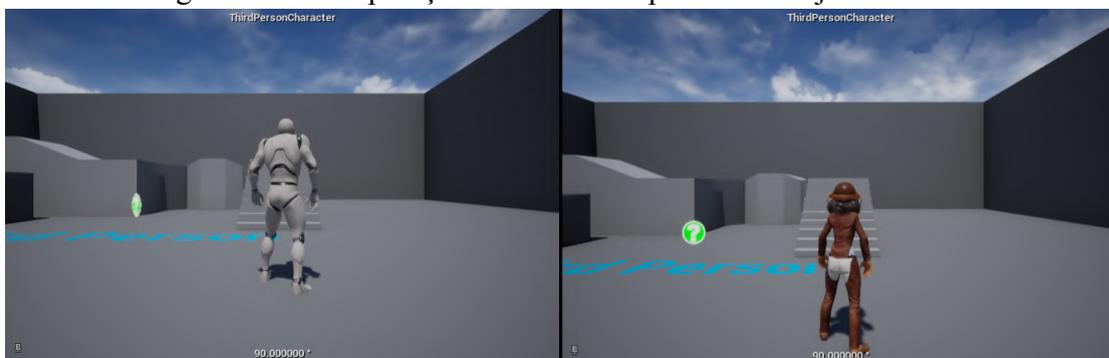
Fonte: Elaborado pelo autor.

Figura 74 – Comparação entre os modelos do Mannequin e Joana.



Fonte: Elaborado pelo autor.

Figura 75 – Comparação entre câmera padrão e os ajustes feitos.



Fonte: Elaborado pelo autor.

4. PRODUTO FINAL

O produto consiste em uma série de artefatos digitais dispostos desde modelos tridimensionais e estruturas de animação da personagem até as texturas aplicadas a ela, além de uma aplicação digital extra, chamada de Memorial Interativo²⁰, concebida como uma forma adicional de apresentar o trabalho desenvolvido. Todos os artefatos serão detalhados a seguir:

4.1. Personagem Joana

A personagem consiste em um conjunto de *assets* compatíveis com os padrões de projeto da UE4, podendo ser usados dentro de qualquer projeto que utilize personagens 3D compatíveis com um rig da *engine*. O próprio rig gerado neste trabalho apresenta alguns *bones* a mais na região da face, que acabam por não ser utilizados pelo rig do projeto escolhido na UE4. Embora esses *bones* não alteram a malha da personagem no projeto em questão, eles podem ser utilizados, bem como os outros *assets* produzidos, em outros projetos da Unreal através das ferramentas de exportação da própria *engine*, que permite a transferência de *assets* entre projetos.

Os *assets* finais gerados para UE4 que compõem o produto descrito neste trabalho (Figuras 76 a 81) são: A (1) malha da personagem Joana; (2) seis texturas para criação de materiais PBR do corpo da personagem, sendo três para mapeamento do corpo e roupas da personagem, e três para mapeamento dos cílios; seguidos de seus respectivos (3) materiais, sendo um para o corpo e roupas, e outro para os cílios da personagem.

Todos os *assets* descritos foram convertidos para o formato de arquivo ‘*uasset*’, que é o padrão de *assets* para os conteúdos de projeto da UE4. Desse modo, qualquer utilização em outras engines ou ferramentas precisam ser convertidas a partir destes formatos ou gerados novamente a partir dos arquivos originais criados durante o desenvolvimento do trabalho, que incluem: (1) Arquivo FBX que contém a malha 3D da personagem, criado a partir da modelagem original da personagem no Blender com a ferramenta de exportação do Auto-Rig Pro; e (2) texturas em bitmap no formato PNG, criadas a partir do *bake* e autoração de materiais no Substance Painter.

A Figura 76 exibe a personagem com o material PBR aplicado e renderizado em tempo real na UE4.

²⁰ Disponível em <<https://tinyurl.com/yzn2248t>>.

As Figuras 77 a 79 exibem respectivamente os canais PBR de cor, normal e rugosidade do material da personagem.

A Figura 80 exibe os bones da personagem convertidos para o padrão da UE4, sobrepostos sobre a malha da personagem.

A Figura 81 exibe os detalhes do material PBR do rosto e parte da roupa da personagem, comparando respectivamente a renderização completa do material, e os canais de cor, normal e rugosidade do material.

Figura 76 – Renderização em tempo real do material PBR da personagem.



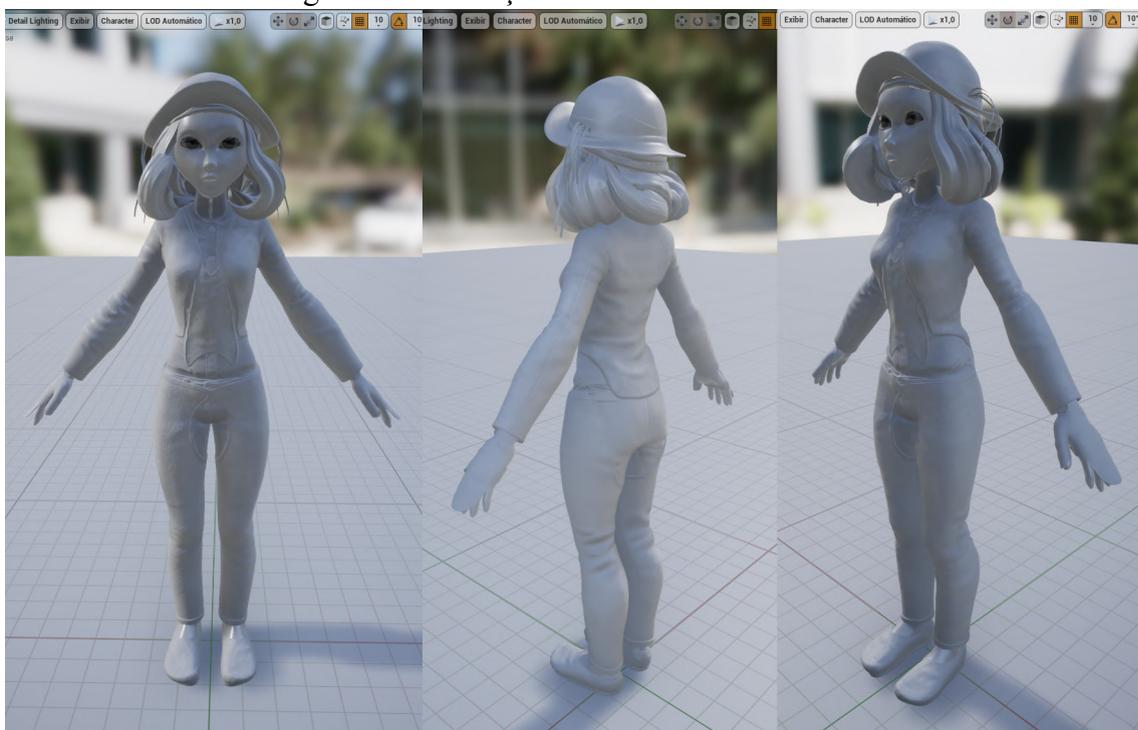
Fonte: Elaborado pelo autor.

Figura 77 – Exibição do canal PBR de cor.



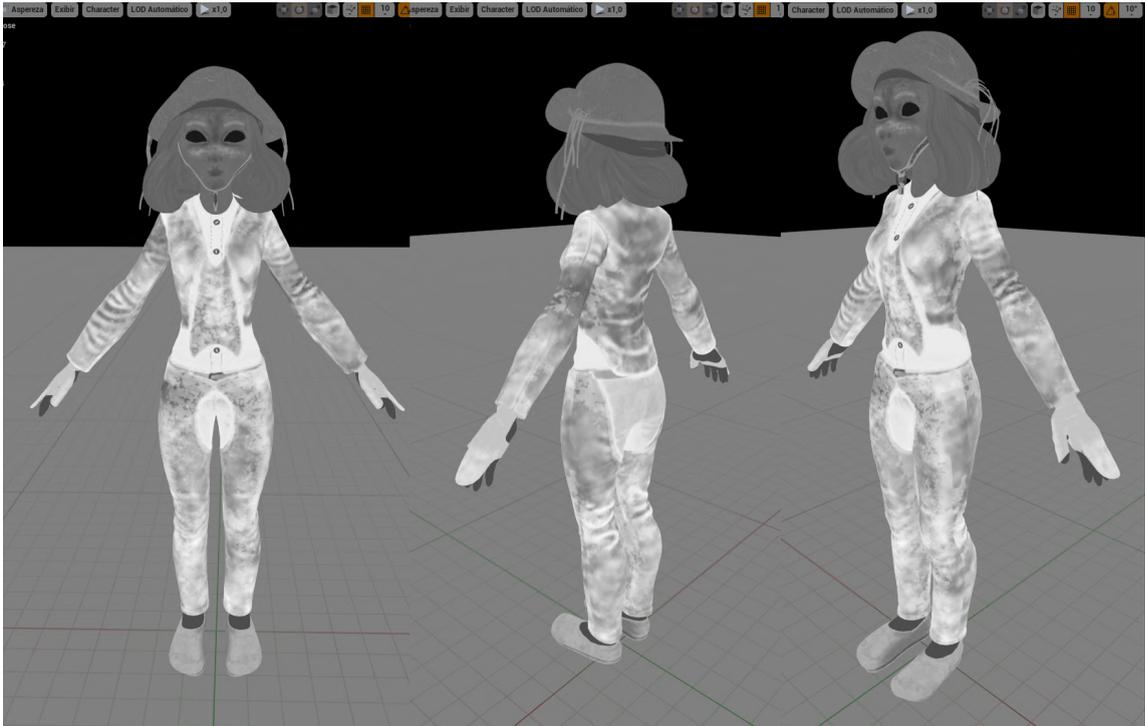
Fonte: Elaborado pelo autor.

Figura 78 – Exibição do canal PBR de normal.

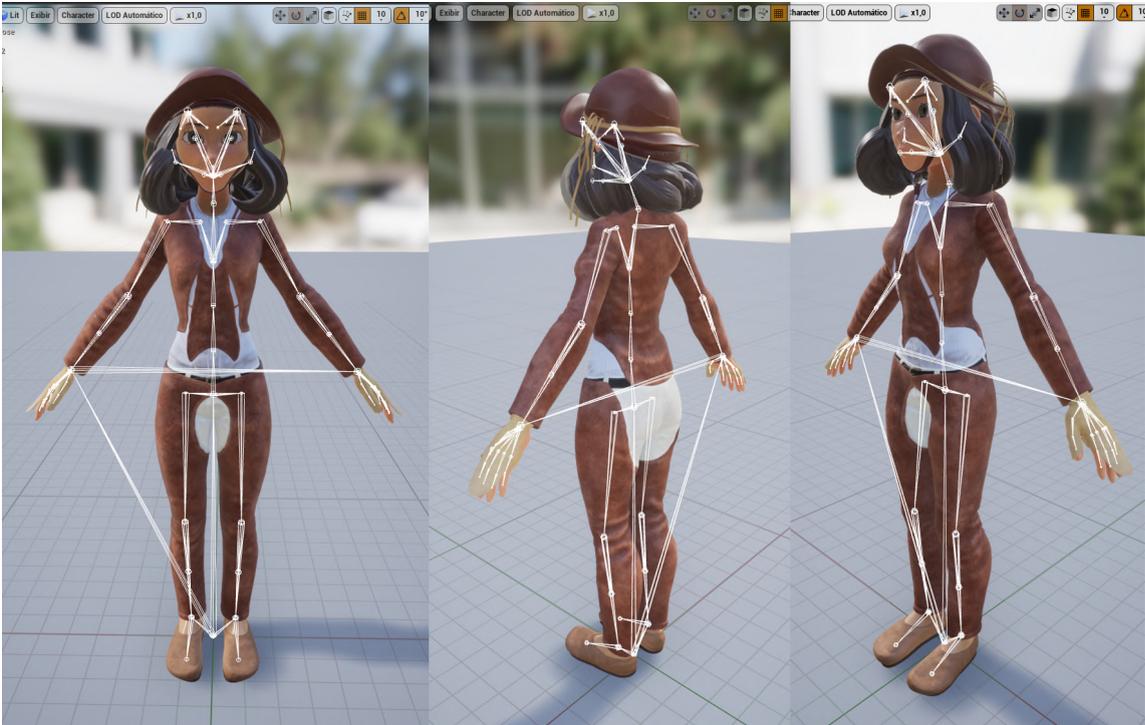


Fonte: Elaborado pelo autor.

Figura 79 – Exibição do canal PBR de rugosidade.



Fonte: Elaborado pelo autor.

Figura 80 – Exibição dos *bones* da personagem.

Fonte: Elaborado pelo autor.

Figura 81 – Comparação entre visualizações da personagem na Unreal.



Fonte: Elaborado pelo autor.

4.2. Memorial Interativo

Adicionalmente à produção da personagem Joana, foi desenvolvida uma peça digital como forma de um Memorial Interativo, pensado para documentar e apresentar as principais etapas de desenvolvimento do projeto e também como um campo de testes dos assets produzidos. O Memorial consiste em uma aplicação interativa executável para sistemas Windows, que funciona de forma similar a um jogo de computador, contendo um ambiente tridimensional organizado ao mesmo tempo como uma pista de obstáculos e como uma espécie de museu ao ar livre, que documenta as etapas de produção do projeto (Figuras 82 a 88).

As Figuras 82 a 84 exibem a personagem Joana explorando o ambiente virtual do Memorial Interativo. As imagens exibem um cenário com elementos naturais que referenciam o semiárido nordestino.

Figura 82 – Joana no ambiente virtual do Memorial Interativo.



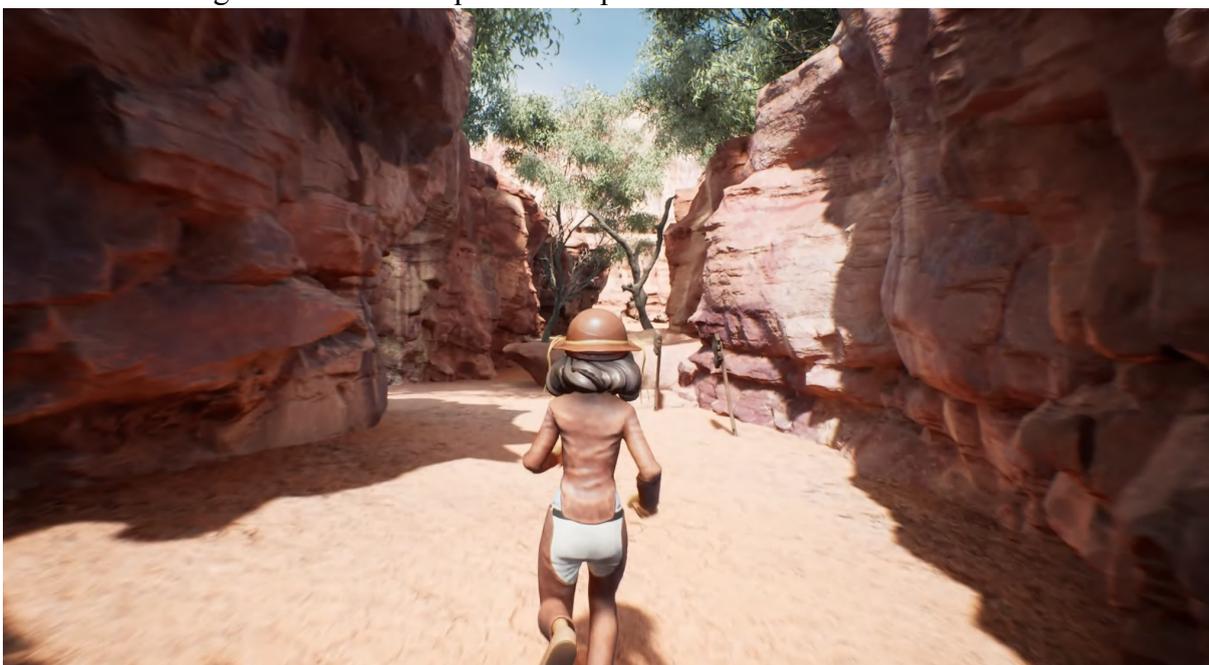
Fonte: Elaborado pelo autor.

Figura 83 – Joana avançando sobre um obstáculo.



Fonte: Elaborado pelo autor.

Figura 84 – Joana explorando o percurso do Memorial Interativo.



Fonte: Elaborado pelo autor.

Ao executar a aplicação o usuário assume o controle de Joana, e pode experimentar as possibilidades de movimentação da personagem e de interação com obstáculos característicos de um cenário de jogo tridimensional em terceira pessoa. A aplicação tem como cenário um ambiente com elementos geológicos e de vegetação que referenciam um ambiente semiárido da região nordeste do Brasil. O cenário virtual possui também áreas organizadas como salas

de exposição que apresentam as etapas do projeto através da exibição de vários dos artefatos produzidos em todas as fases de desenvolvimento da personagem, apresentando referências, esboços, versões dos modelos tridimensionais em várias etapas do fluxo de produção, materiais e animações utilizadas (Figuras 85 a 88).

As Figuras 85 a 88 mostram os assets expostos no percurso do Memorial Interativo, que servem para resumir as etapas de desenvolvimento do projeto através de conjuntos de artefatos gerados para cada uma.

Figura 85 – Joana em frente a um expositor da etapa conceitual do projeto.



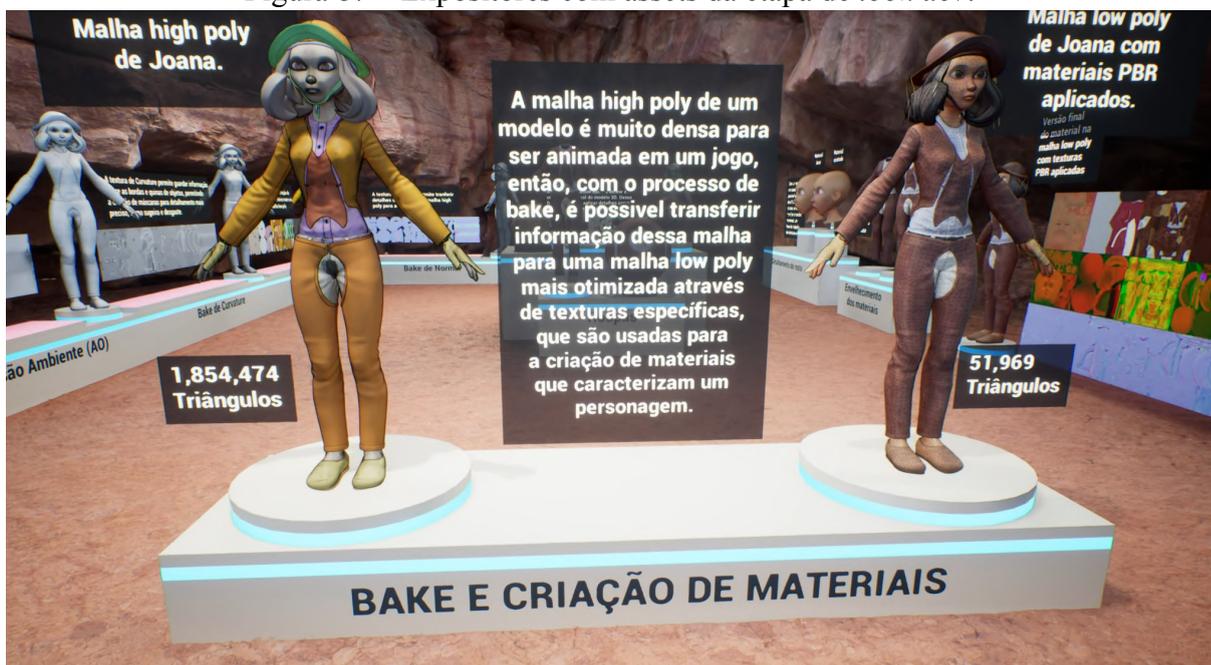
Fonte: Elaborado pelo autor.

Figura 86 – Joana explorando a sala da etapa de modelagem tridimensional.



Fonte: Elaborado pelo autor.

Figura 87 – Expositores com assets da etapa de *look dev*.



Fonte: Elaborado pelo autor.

Figura 88 – Exemplo de animação utilizada no Memorial Interativo.



Fonte: Elaborado pelo autor.

O Memorial Interativo foi desenvolvido dentro da UE4, e foram utilizadas ferramentas e assets disponibilizados gratuitamente que foram criados pela comunidade e pela própria Epic Games. Para o sistema de animação, controles e interface da aplicação foi utilizado o conjunto de ferramentas Advanced Locomotion System V4²¹ em sua versão gratuita, que funciona como um projeto da UE4 e possui ferramentas de desenvolvimento integradas ao ambiente de desenvolvimento da Engine. Para a montagem do cenário foram utilizadas bibliotecas de assets do Marketplace da Unreal Engine²² e Megascans²³ e também foram produzidos alguns assets no Blender para os expositores das salas do Memorial organizados no ambiente virtual utilizando as próprias ferramentas de adição da UE4.

²¹ Disponível em <tinyurl.com/4sdwh8cf> Acessado em 22/09/2021.

²² Disponível em <<https://www.unrealengine.com/marketplace/en-US/store>> Acessado em 22/09/2021.

²³ Disponível em <<https://quixel.com/megascans>> Acessado em 22/09/2021.

5. CONSIDERAÇÕES FINAIS

A pipeline de produção do projeto possibilitou que os recursos disponíveis – hardware, software e horas de trabalho – fossem melhor direcionados dentro de cada etapa do processo de desenvolvimento, ajudando a identificar e priorizar as tarefas mais importantes. Com uma estrutura organizacional definindo o fluxo de cada etapa, as tomadas de decisão acerca de ajustes e correções foram realizadas com maior assertividade em relação ao produto final, representando um ganho de eficiência durante todo o projeto. Entretanto, desafios específicos relacionados à integração de processos distintos dentro da pipeline, sobretudo a integração entre *softwares* diferentes, foram os que mais tomaram tempo durante sua execução.

Durante o processo de produção do trabalho foi percebido que, embora seja necessário seguir o planejamento inicial de um projeto com tão poucos recursos de pessoal, é muito importante estar aberto para mudanças e ajustes na aplicação da pipeline. Tendo em vista os recursos de hardware e software disponíveis, e principalmente se houver limitação de licenças de software, é fundamental planejar os métodos de produção mais adequados e equilibrar a importância de cada etapa, mas também é importante ter alguma margem para adaptar os fluxos de trabalho, já que uma pipeline de produção por si só já possui um certo grau de maleabilidade dependendo do projeto executado.

Etapas que se interligam e dependem de muitos processos encadeados, como a criação de um ciclo de animação, que depende de uma malha bem organizada, podem ser bastante prejudicadas caso não se mantenha durante o processo de desenvolvimento, uma visão geral de todo o projeto. Entretanto, o trabalho individual em um projeto com este, proporciona uma liberdade de adaptação bem maior das atividades do que se teria em um time com mais pessoas, em que uma única alteração precisaria passar pela avaliação de outros profissionais.

Embora o trabalho em um time reduzido a uma ou poucas pessoas seja menos burocrático, é perigoso para o resultado de um projeto com tantas etapas como este, confiar apenas nas habilidades e avaliação dos profissionais envolvidos. Então, sempre que possível, é importante que se busque algum tipo de avaliação de profissionais não envolvidos com os processos de produção empregados. O *feedback* dessas fontes, tem um valor estratégico altíssimo para o resultado final do projeto, já que não são presos a possíveis valores ou compromissos adquiridos pelas pessoas diretamente envolvidas nos processos de trabalho, o que torna as observações e sugestões mais livres de julgamento.

Em termos de expressão artística, foi muito interessante perceber como muitas tecnologias atuais são fáceis de se obter e de utilizar. Essa crescente acessibilidade de recursos possibilita um maior foco nos processos criativos das pessoas envolvidas, diminuindo as barreiras de conhecimento necessário para a produção de um projeto dessa natureza, que tem um apelo tão lúdico. Foi interessante notar como algumas das ferramentas que eu já utilizava tem uma gama de recursos técnicos e potencialidades criativas que eu ainda não conhecia. Esse tipo de conhecimento mais acessível, integrado e motivado por um projeto como o desenvolvido aqui, acabam estimulando a vontade de exploração de novas tecnologias e recursos disponíveis no mercado.

O processo de criação do trabalho com *look dev*, por exemplo, permite que uma narrativa seja contada apenas através dos detalhes de um objeto ou personagem, como manchas em superfícies, arranhões evidentes, queimaduras de sol na pele, etc... Todos esses detalhes ajudam a compor uma narrativa para o espectador em torno de um personagem ou objeto apenas com recursos visuais, sem que nenhuma linha de diálogo fosse escrita, já que os detalhes das texturas e materiais desenvolvidos inevitavelmente indicam elementos mínimos de um *background* para a personagem: Para uma pessoa com origens no Nordeste do Brasil, fica mais evidente que Joana possivelmente pertence a um ambiente de sertão, possibilitando, através da observação de suas roupas, imaginar sem muita dificuldade o ambiente ao qual ela pertence. Ainda assim, pela quantidade de roupas que ela usa, os materiais empregados e o estado dessas matérias, é possível posicionar minimamente Joana em um ambiente ensolarado e hostil ao corpo humano, pelo simples fato de que ela usa chapéu e roupas que cobrem praticamente o corpo inteiro, e possui no rosto, única área da pele exposta, algumas manchas e leves queimaduras, indicando exposição excessiva ao sol.

Adicionalmente ao desenvolvimento da personagem, a facilidade de se utilizar a Unreal Engine como ferramenta de apresentação do projeto, através do Memorial Interativo por exemplo, expande as possibilidades de experimentação criativa, representando também um ganho de tempo, já que a engine permite a utilização de bibliotecas de assets gratuitos com bastante facilidade, seja através do Megascans ou de projetos desenvolvidos pela comunidade.

Do ponto de vista técnico, é importante destacar que algumas decisões relacionadas à otimização do produto precisaram ser tomadas durante a execução do projeto para que ocorresse pouco ou nenhum prejuízo no resultado final. A escolha de criar uma malha única

para as peças de roupa do dorso e membros da personagem acabou por agilizar alguns processos de modelagem e ajustes, mas também demandou que algum tempo extra tivesse que ser despendido para resolver problemas na criação de materiais. Outra decisão importante foi o descarte das animações feitas com o sistema de animação Rigify, no Blender. A proposta de usar o Auto-Rig Pro para criação de um novo rig, ajudou bastante o fluxo de trabalho com a Unreal, possibilitando um ganho enorme na compatibilidade com a *engine* e representando um ganho de tempo expressivo que seria utilizado para ajustes das animações. Isso também abriu um leque de possibilidade de expansão do projeto para a integração com outros *assets*, como o suporte às animações do serviço Mixamo, que fornece uma biblioteca enorme com ciclos de animação prontos para uso na UE4. Entretanto, a escolha de utilizar o Auto-Rig Pro acabou por praticamente descartar as ferramentas de animação do Blender, restringindo o uso do Rigify aos estágios anteriores de testes de deformação da malha durante o *rigging*. Processo que inclusive poderia ser feito com o próprio Auto-Rig Pro.

Do ponto de vista de design, as decisões mais importantes foram tomadas durante a etapa de conceituação, como o uso de materiais realistas em contraponto com uma estética de formas mais cartunescas. Nesse aspecto pode se observar também que um maior investimento de tempo no processo de conceituação visual poderia orientar uma execução mais harmoniosa da relação de formas da personagem, como a proporções dos membros, anatomia e apelo visual para a silhueta da personagem, que apresenta um certo desequilíbrio quando notamos que o volume do cabelo e chapéu somados acaba por parecer exagerado em relação ao corpo magro da personagem.

Outras decisões de design acabaram limitadas pelo próprio escopo geral do projeto e disponibilidade de recursos. Dois exemplos são: A (1) falta de movimento do cabelo e dos enfeites do chapéu, que deveriam se mover durante as animações, causando uma quebra de expectativa quando observada a interação da personagem com o cenário. E (2) as peças da armadura de couro também se beneficiariam de um movimento mais fluido durante os ciclos de animação da personagem. Entretanto, tal artifício ficou inviável devido à decisão de usar uma única malha para as peças da armadura.

Esses tipos de decisão de design tomadas durante a execução do projeto podem acabar sendo um divisor de águas para a qualidade geral do produto, pois enquanto algumas podem ser atualizadas em versões futuras do produto com uma relativa facilidade, como os movimentos do cabelo, outras podem demandar um retrabalho muito grande e por vezes

desvantajoso do ponto de vista de tempo e recursos empregados para que sejam implementadas, como no caso das roupas, que demandaria uma revisão completa da malha da personagem, demandando uma grande quantidade de refações em várias etapas do projeto que são diretamente afetadas pela topologia do modelo 3D.

O projeto em questão aborda uma pequena parte, embora complexa, do universo de produções possíveis em mídia de jogos digitais, mas outros trabalhos futuros imediatamente possíveis podem ocorrer no aprofundamento narrativo em torno de Joana, através da criação de props para a personagem, como ferramentas e utensílios que indiquem seu estilo de vida. Ou ainda, outro trabalho pode ser desenvolvido na expansão de possibilidade de interação com a personagem, através do uso de novas animações, criação de um cenário com obstáculos mais diversos e desafiadores do ponto de vista da ludicidade de interação com a mídia jogo, ou experimentação de novos recursos tecnológicos, como realidade aumentada e realidade virtual. Uma terceira linha de trabalho poderia focar na análise visual da personagem do ponto de vista de representatividade cultural, investigando a relação dos elementos visuais desenvolvidos em torno de Joana com os arquétipos culturais dos quais a personagem se apropria. O que poderia ser bastante útil para propor adequações e melhorias na forma como a personagem poderia ajudar a representar melhor a cultura local dentro do cenário de desenvolvimento de jogos digitais no mercado nacional.

REFERÊNCIAS

- AMBROSE, D. Highly Creative Syntheses of the Arts and Technology: An Interview with Lead Look Development Artist Brian Kloc. **Roeper Review**, v. 41, n. 2, p. 143–146, 2019.
- BRISTOT, P. C.; POZZEBON, E.; FRIGO, L. B. A Representatividade das Mulheres nos Games. **XVI SBGames**, p. 862–871, 2017.
- COOPER, J. **Game Anim: Video Game Animation Explained**. 1. ed. Danvers: CRC Press, 2019.
- DUNLOP, R. **Production Pipeline Fundamentals for Film and Games**. 1. ed. Burlington: Focal Press, 2014.
- GUEVARRA, E. T. M. **Modeling and Animation Using Blender**. 1. ed. Berkeley, CA: Apress, 2020.
- KOHLER, L. P. DE A. et al. A representatividade feminina nos jogos digitais. p. 265–269, 2021.
- LEVER, N. **Real-time 3D Character Animation with Visual C++**. 1. ed. Woburn: Focal Press, 2002.
- MCDERMOT, W. **The PBR Guide**. 3. ed. [s.l.] Allegorithmic, 2018.
- MCCLOUD, S. **Understanding Comics: The Invisible Art**. 1. ed. New York, New York, USA: HarperCollins, 1993.
- NOVAK, J. **Game Development Essentials Third Edition**. 3. ed. Clifton Park: Delmar, 2012.
- PIERSON, C. et al. Luminance maps from High Dynamic Range imaging: calibrations and adjustments for visual comfort assessment. **Proceedings of the Lux Europa 2017**, p. 5. 147-151, 2017.
- REIS, M. R. DOS; MERINO, E. A. D. Painel semântico: revisão sistemática da literatura sobre uma ferramenta imagética de projeto voltada à definição estético-simbólica do produto. **Estudos em Design**, v. 28, n. 1, p. 178–190, 2020.

ROMANUS, Juliana Saldanha. **Gênero em jogo : Um Olhar Sobre Personagens e as Representações de Tipos de Feminilidades e Masculinidades nos Games de Ação Contemporâneos**. 2012. 58f. Monografia (Curso de Tecnologia em Design Gráfico) - Departamento Acadêmico de Desenho Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2012.

SMITH, M. **Engaging Characters: Fiction, Emotion, and the Cinema**. 1. ed. New York: Oxford University Press, 1995.

SORKINE, O.; COHEN-OR, D. Warped Textures for UV Mapping Encoding. **The Eurographics Association**, p. 2–6, 2001.

THISTLEWOOD, J. **How to Work Smarter in Look Development and Lighting**. Disponível em: <<https://magazine.artstation.com/2019/07/work-smarter/>>. Acesso em: 7 set. 2020.

VALVE CORPORATION. **Dota 2 Workshop - Character Art Guide**. Disponível em: <<https://support.steampowered.com/kb/9334-YDXV-8590/dota-2-workshop-character-art-guide>>. Acesso em: 6 set. 2020.

VAN ROOIJ, M. Carefully Constructed Yet Curiously Real: How Major American Animation Studios Generate Empathy Through a Shared Style of Character Design. **Animation**, v. 14, n. 3, p. 191–206, 12 nov. 2019.

VERKAAIK, E. **How Limitations of the Medium Influenced Video Game Art**. Disponível em: <<https://111426studio.wordpress.com/2015/12/01/how-limitations-of-the-medium-influenced-video-game-art/>>. Acesso em: 8 set. 2020.

VOGLER, C.; MACHADO, A. **A jornada do escritor: estruturas míticas para escritores**. 2. ed. Rio de Janeiro: Nova Fronteira, 2006.

WHITEHURST, A. **The Visual Effects Pipeline**. Disponível em: <<http://www.andrew-whitehurst.net/pipeline.html>>. Acesso em: 7 set. 2020.