

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE BELAS ARTES
CINEMA DE ANIMAÇÃO E ARTES DIGITAIS

Tiago Cornélio da Silva

DO DESENVOLVIMENTO DA MODELAGEM 3D AO RIGGING DE
PERSONAGEM NO BLENDER

Belo Horizonte

2023

Tiago Cornélio da Silva

**DO DESENVOLVIMENTO DA MODELAGEM 3D AO RIGGING DE
PERSONAGEM NO BLENDER**

Trabalho de conclusão de curso apresentado
como requisito parcial para obtenção do
título de Bacharel em Cinema de Animação e
Artes Digitais.

Orientador: Prof.: Pedro Aspahan

Belo Horizonte
Escola de Belas Artes / UFMG
2023

LISTA DE FIGURAS

Figura 1: Hexabo. Fonte: Autor.....	6
Figura 2: Bo ³ . Fonte: Autor.....	6
Figura 3: Esboço da ideia dos designers das locomoções. Fonte: Autor.....	7
Figura 4: Resultado final do design do corpo. Fonte: Autor.....	7
Figura 5: Blocagem das locomoções dentro do ambiente tridimensional. Fonte: Autor.....	8
Figura 6: Demonstração da Porta Lateral sendo aberta por seu controle. Fonte: Autor.....	9
Figura 7: Demonstração da Porta Topo sendo aberta por seu controle. Fonte: Autor.....	9
Figura 8: Modelo 3D do corpo finalizado. Fonte: Autor.....	10
Figura 9: Modelos 3D dos pistões da perna da locomoção do tipo aranha. Fonte: Autor.....	10
Figura 10: Modelo 3D da Locomoção tipo aranha finalizada. Fonte: Autor.....	11
Figura 11: Versões anteriores da Locomoção do tipo Duas Rodas. Fonte: Autor.....	11
Figura 12: Divisão dos blocos da Locomoção de Duas Rodas. Fonte: Autor.....	12
Figura 13: Modelo 3D da Locomoção do tipo Duas Rodas finalizado. Fonte: Autor.....	13
Figura 14: Conjunto de bones separados por camadas e funções. Fonte: Autor.....	15
Figura 15: Demonstração da modelagem e rig feito nos pistões da Locomoção do tipo aranha. Fonte: Autor.....	18
Figura 16: Conjuntos de bones da Locomoção de Duas Rodas, separadas por camadas. Fonte: Autor.....	19
Figura 17: Estrutura principal da movimentação da Locomoção do tipo Duas Rodas. Fonte: Autor.....	20
Figura 18: Layout de controles com design finalizado. Fonte: Autor.....	21

SUMÁRIO

LISTA DE FIGURAS.....	3
SUMÁRIO.....	3
RESUMO.....	4
ABSTRACT.....	4
1. INTRODUÇÃO.....	5
2. METODOLOGIA.....	5
3. DESENVOLVIMENTO DO MODELO TRIDIMENSIONAL.....	6
3.1. A IDEIA.....	6
3.2. DESENVOLVIMENTO DOS MODELOS.....	7
3.2.1. BLOCAGEM.....	7
3.2.2. MODELANDO O CORPO.....	9
3.2.3. MODELANDO A LOCOMOÇÃO TIPO ARANHA.....	10
3.2.4. MODELANDO A LOCOMOÇÃO TIPO DUAS RODAS.....	11
4. RIGIFY.....	12

4.1. LAYOUT DO RIGIFY.....	13
5. ELABORAÇÃO DO RIG.....	14
6. DESENVOLVIMENTO DO RIGGING.....	14
6.1. IDEALIZANDO O RIG DOS MODELOS.....	15
6.1.1. O CORPO.....	15
6.1.2. LOCOMOÇÃO DO TIPO ARANHA.....	17
6.1.3. LOCOMOÇÃO DO TIPO DE DUAS RODAS.....	19
6.2. LAYOUT DE INTERAÇÃO.....	20
6.2.1. DESENVOLVENDO O DESIGN.....	21
6.2.2. UTILIZAÇÃO DE VARIÁVEIS.....	22
7. CONSIDERAÇÕES FINAIS.....	23
8. BIBLIOGRAFIA.....	24

RESUMO

O artigo aborda o processo de desenvolvimento do *Rigging* do personagem robótico 3D *BO³*, composto de um cubo com quatro sistemas de locomoção. O trabalho descreve as etapas da modelagem até a etapa final do desenvolvimento do *rigging* no *Blender*, com o objetivo de deixar o personagem pronto para a etapa de animação. Com um foco maior no desenvolvimento do *rig*, apresentamos uma forma diferente de construir a interface de interação com o modelo tridimensional de cada sistema de locomoção e apresentamos uma estrutura de organização dos *bones*, desenvolvendo uma interface de controle simples, com uma estrutura levemente complexa por trás.

Palavras chave: rigging 3D digital, rigging 3D, Blender, robotic character

ABSTRACT

The article addresses the process of developing the Rigging for the 3D robotic character *BO³*, consisting of a cube with four locomotion systems. The work describes the stages from modeling to the final step of rigging development in Blender, aiming to prepare the character for the animation phase. With a greater focus on rig development, we present a different approach to building the interaction interface with the three-dimensional model of each locomotion system and introduce a bone organization structure, developing a simple control interface with a slightly complex underlying structure.

Keywords: rigging 3D digital, rigging 3D, Blender, robotic character

1.INTRODUÇÃO

Numa pipeline de animação 3D, há várias etapas, como a modelagem do modelo tridimensional (3D), o desenvolvimento do *rigging* 3D, a texturização do modelo, a animação do modelo, entre outras. Todas as etapas têm a sua importância, mas a etapa de *rigging* tem um grau de importância maior para o processo da animação, que precisa do modelo com *rig* pronto para começar a animar. Da mesma forma, para se desenvolver e aplicar o *rig*, o modelo precisa estar finalizado. Por ser uma etapa importante, o desenvolvimento do *rigging* tende a ser desafiador e assustador para artistas iniciantes, fica a cargo do artista *Rigger*¹. Levando em consideração que muitas das vezes o animador só recebe o modelo pronto, fica a cargo do *rigger* construir uma interface usuário intuitiva. Prezando que o animador, ao se deparar com o controle, logo entenda a sua função e como utilizá-la.

Tendo em vista todas as etapas na hora de se fazer uma animação, foi proposto como forma de desenvolvimento para portfólio, a criação de um personagem robótico que tivesse como foco a elaboração de seu *rig*. Indo da modelagem ao *rigging* e, posteriormente, para a fase de animação, onde o animador terá total liberdade de trabalhar com o personagem, utilizando os recursos disponibilizados. Assim, buscando a melhor forma de construir um *rig* do zero, buscamos encontrar algum sistema que possa auxiliar nesse desenvolvimento e de como estruturar e organizar esse esqueleto 3D.

2.METODOLOGIA

Como metodologia adotada propusemos o desenvolvimento de um personagem cúbico simples, mas cuja complexidade aparece em seus modos de locomoção com um foco maior no estudo e utilização da ferramenta para o desenvolvimento do *rig* para o personagem. Testamos a utilização do *addon Rigify* do *Blender* como uma das bases de pesquisa, onde propusemos estudar sua aplicação na hora de criar os *bones*, configurações e definir a sua organização dentro da hierarquia do *armature*. Utilizamos a base criada após o estudo do modo como o plugin *Rigify* organiza os *bones* entre as camadas e apresentação de controles, mas acabamos abandonando o uso do plugin e propusemos a elaboração de uma estrutura de *rig*, partindo do zero, sem auxílio de nenhum outro sistema, sendo todo desenvolvido no software *Blender*. Assim, vamos abordar as duas locomoções do tipo terrestre que desenvolvemos, enquanto as outras duas locomoções são uma repetição das ideias de funcionamento anteriores.

¹ Artista que trabalha diretamente com desenvolvimento de Rig, também conhecido como Diretor técnico do Personagem.

3.DESENVOLVIMENTO DO MODELO TRIDIMENSIONAL

3.1.A IDEIA

Ao iniciar meus estudos na área da animação 3D, percebi que uma das formas mais simples de modelar e imprimir uma animação sem se preocupar com a deformação da malha tridimensional, é por meio do uso dos objetos *Hard Surface*². Assim, ao escolher modelar um robô, busquei um formato hexagonal que não era muito convencional. A proposta por trás desse personagem, apelidado de *Hexabo* (ver figura 1), era que ele fosse visto de uma forma de hexágono de seis pontos de vista, sendo elas frente, lado, topo e as suas oposições.

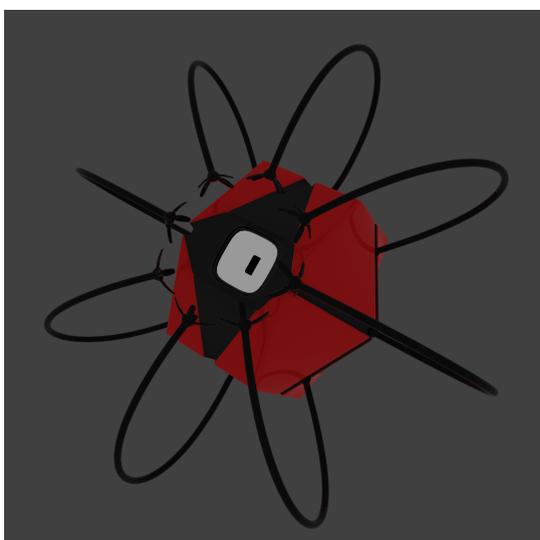


Figura 1: Hexabo. Fonte: Autor



Figura 2: Bo³. Fonte: Autor

Por mais que o design do Hexabo fosse interessante visualmente, para esse estudo precisava de um design mais simples e claro em seu formato, daí, desenvolvi um outro personagem para esse projeto. *Bo*³ (ver figura 2), um robô que tem um formato de cubo, tornando a leitura das suas ações muito mais claras, e facilitando o entendimento de como cada lado do cubo funciona. Com isso, o personagem apresenta seis lados: a frente, onde temos uma tela que demonstra a expressão do personagem; as costas, onde há uma tampa que ao ser retirada apresenta o compartilhamento de energia; além dos outros quatro lados, que possuem, cada um, um conjunto de portas. Essas portas possibilitam diferentes formas de locomoção ao personagem, duas são terrestres (uma imita patas de aranhas e a outra utiliza rodas de moto); uma é aérea (como hélices de drones); e uma aquática (como um barco).

² Estilo de modelagem que apresenta em seus modelos, uma aparência mais sólida e com ângulos mais retos, sendo comum em objetos mecânicos.

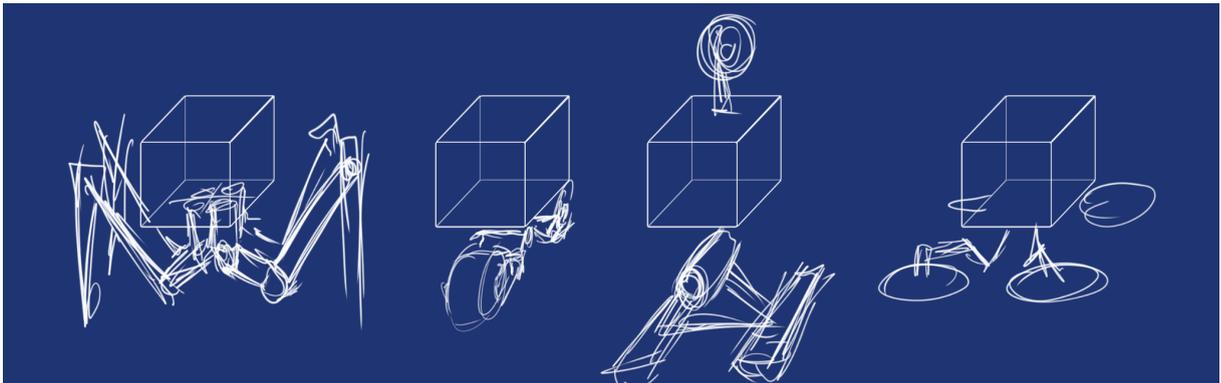


Figura 3: Esboço da ideia dos designers das locomoções. Fonte: Autor.

Ao desenvolver o design do personagem, uma das preocupações maiores era de deixar simples de se ler visualmente, mas com tom mais moderno, então a base sai de um quadrado para octógono que possui os seus vértices mais nas extremidades, assim, possibilitando que as faces do cubo ainda possuíssem os quadrados internos intactos, menos a frente e as costas que não tinham essa necessidade. Outra escolha foi separar a parte da frente e das costas, para que quando o robô girasse, o corpo ficasse no mesmo eixo, criando uma dinâmica maior na hora de trocar a locomoção, e facilitando a vida do animador.

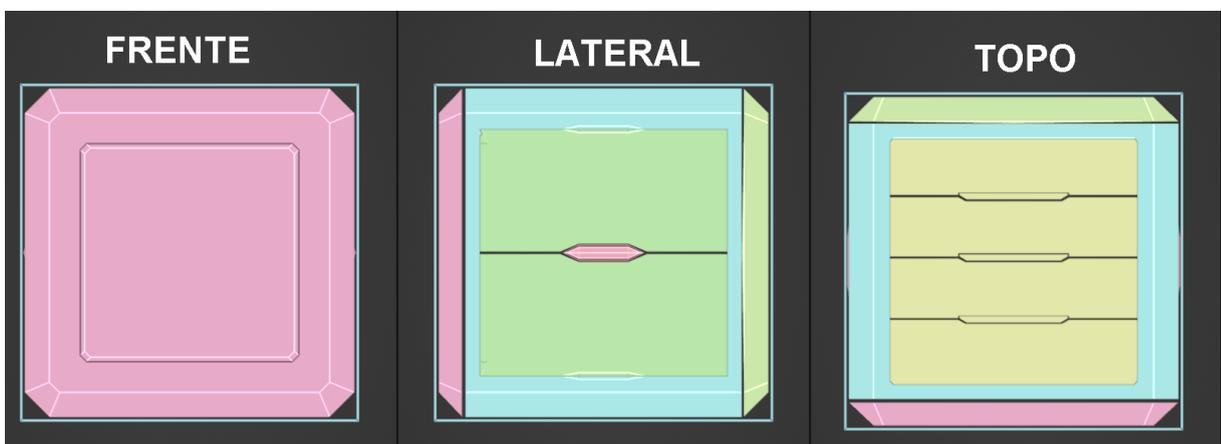


Figura 4: Resultado final do design do corpo. Fonte: Autor.

3.2.DESENVOLVIMENTO DOS MODELOS

3.2.1.BLOCAGEM

Por ser um personagem que possui outros quatro tipos de locomoção, após alguns testes, percebemos que a modelagem, ao partir do mesmo ponto de origem, poderia gerar

problemas no momento de realização de alterações futuras, e até mesmo na etapa da elaboração do *rig*. Optei por separar as locomoções do corpo dentro do ambiente tridimensional (chamando por *Viewport*³), assim, poderia visualizar todos objetos ao mesmo tempo e conseguir moldá-los sem preocupar com a interferência dos outros modelos. Como forma de organização, foi feita uma blocagem inicial para determinar o espaço ocupado por cada modelo. Com essa base pronta, partimos para as blocagens mais individuais dos modelos, priorizando as partes mais relevantes. Por exemplo, o corpo teve, como blocagem, a utilização de dois cubos, sendo um para definir a área geral ocupada pelo corpo, e o outro para definir a área interna.

Assim, a utilização dessa blocagem básica do corpo, foi replicada quatro vezes, para que cada locomoção pudesse definir o ponto de origem da sua estrutura e como ela iria se comportar perto da área do corpo. Partindo dessa base, a locomoção do tipo *Aranha* foi formada por quatro patas com três articulações, no entorno de seus membros, assim, a blocagem focou em determinar essas características base, utilizando paralelepípedos para definir o tamanho e comprimento dos membros e mantendo eles em ângulos retos, mesmo que naturalmente a estrutura não se comporte desta forma, facilitando na adição e organização dos *bones* na etapa de *rig*. Na outra locomoção terrestre, a do tipo de *Duas Rodas*, temos como ideia dois cubos que representam as rodas e definem a distância entre elas e o corpo, sendo o cubo da roda traseira maior que o da roda dianteira. Utilizamos um cubo no meio das rodas, abaixo do corpo, sendo a blocagem do eixo central, de onde partem os paralelepípedos que interligam os cubos das rodas com o corpo. Em particular, o paralelepípedo que vai em direção ao corpo se subdividiu em 2 paralelepípedos, replicando como funcionaria esse pistão que desloca as duas rodas para fora do corpo.

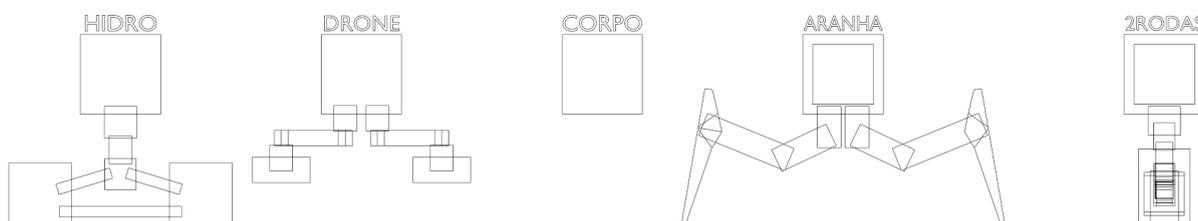


Figura 5: Blocagem das locomoções dentro do ambiente tridimensional. Fonte: Autor.

Após terminar a blocagem de todas as locomoções, por mais que as locomoções do tipo Hidromotor e do tipo Drone não iram ter sua modelagem e *rig* abordadas no texto, a sua blocagem teve uma importância para a organização dos modelos dentro *Viewport*.

³ Janela de interação com objetos 3D.

Organização essa que também poderia ser visto na forma que trabalhei com as *Collection*⁴, separando uma *collection* para cada modelo, e utilizando por definir cores para cada *collection* para facilitar na identificação.

Uma vez feita a blocagem, partimos para a etapa de modelagem das peças. Ao manter os objetos em ângulo reto, ficou mais fácil a modelagem das peças, por proporcionar uma posição e um ângulo de boa visualização. Por uma escolha não muito prática, o design das peças foi desenvolvido diretamente na modelagem, o que por um lado gerou uma extensão maior no tempo de desenvolvimento do design, mas apresentou uma forma de teste prévio do funcionamento das peças criadas. Assim, por mais que não tenha sido vantajosa essa forma de criação, ela possibilitou um adiantamento de como seria a elaboração do esqueleto do modelo.

3.2.2. MODELANDO O CORPO

Como idealizado anteriormente, a parte central do corpo deveria girar sobre seu próprio eixo, de modo a permitir que a porta lateral pudesse se abrir para ejetar a estrutura de locomoção. Optei por dividir o corpo em três partes, separadas por uma pequena fresta. As partes dianteira e traseira do corpo ficaram com o mesmo tamanho, enquanto a parte central ocupou o maior volume do corpo (ver figura 4).

Durante a elaboração das portas, para possibilitar esse giro do corpo, desenvolvemos dois tipos de porta (que seriam espelhadas). As duas portas laterais se abrem para fora (ver figura 6), enquanto as portas do topo e da base são portas de correr, divididas em quatro pedaços que deslizam, encaixando uma na outra, como se fossem a janela de um teto solar (ver figura 7).

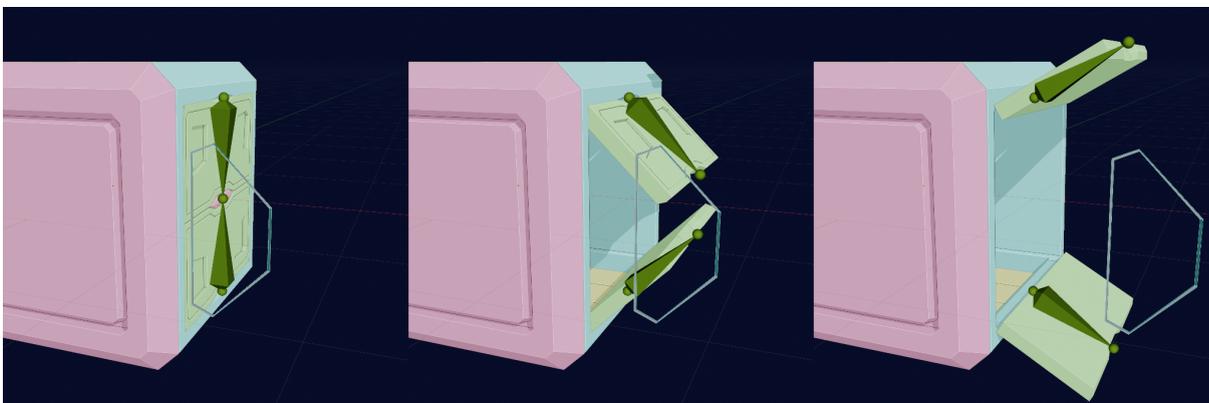


Figura 6: Demonstração da Porta Lateral sendo aberta por seu controle. Fonte: Autor.

⁴ Nomenclatura abordada pela Blender para um grupo de modelos.

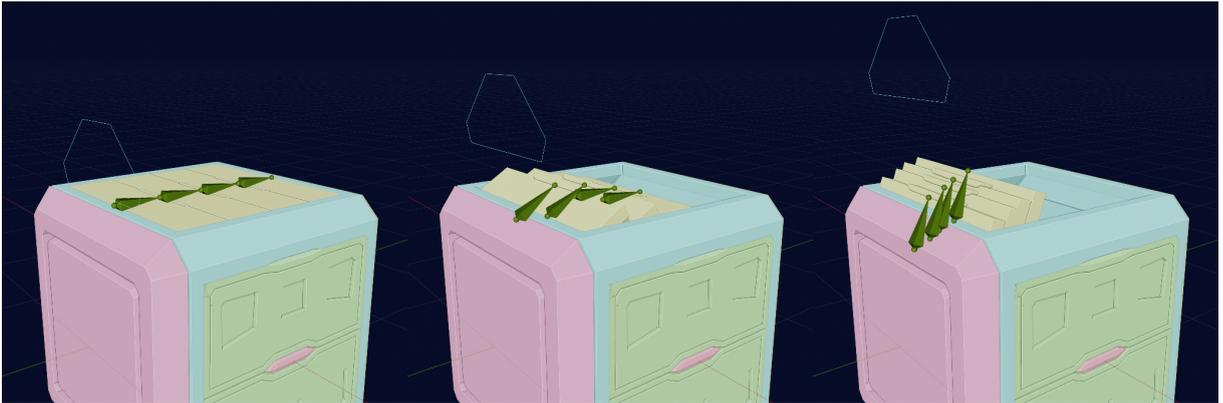


Figura 7: Demonstração da Porta Topo sendo aberta por seu controle. Fonte: Autor.

Durante a elaboração do design do personagem, o formato inicial de cubo foi se modificando para encaixar as portas. A separação da parte central em relação às outras foi uma escolha estética que fez o formato quadrado das faces se transformar num octógono.

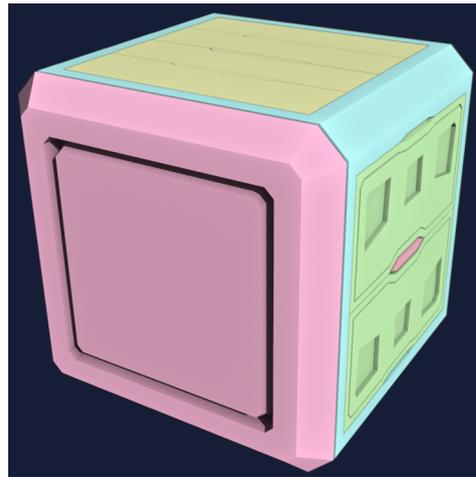


Figura 8: Modelo 3D do corpo finalizado. Fonte: Autor.

3.2.3.MODELANDO A LOCOMOÇÃO TIPO ARANHA

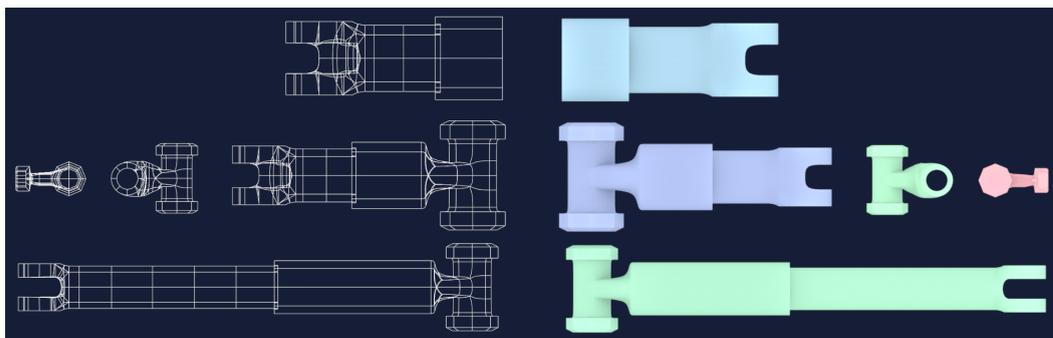


Figura 9: Modelos 3D dos pistões da perna da locomoção do tipo aranha. Fonte: Autor.

A locomoção do tipo aranha possui quatro pernas formadas por três pistões e uma pata em cada (ver figura 9). Pode-se dizer que a perna da aranha é formada por apenas dois tipos de objetos, a pata que tem a sua particularidade, e, diferente da pata, temos o objeto pistão que se repete para formar a perna. Ao observar a figura 9, pode-se ver as variações do pistão que se mantém tendo a mesma base, adicionando uma conexão ou outra para formar o pistão necessário para aquela parte da perna.

Sabendo que teria mais de um tipo de pistão e que ele acabaria tendo pequenas diferenças do outro, optei por modelar três objetos: a carcaça do pistão com o cilindro, a conexão fêmea e a conexão macho. Assim, utilizando esses três objetos fui conectando-os conforme a necessidade da peça. Criei assim três peças, um pistão com uma conexão fêmea, dois pistões que contavam com uma ponta com conexão fêmea e outra com conexão macho. Além dos pistões, tem a conexão macho/fêmea que serve para conectar a pata com a perna. Já a pata é dividida em duas partes, a superior e a inferior, sendo a parte inferior composta por dois pedaços. Como nos pistões tem uma parte que se retrai, a parte inferior da pata também pode se retrair, diminuindo o comprimento pela metade.

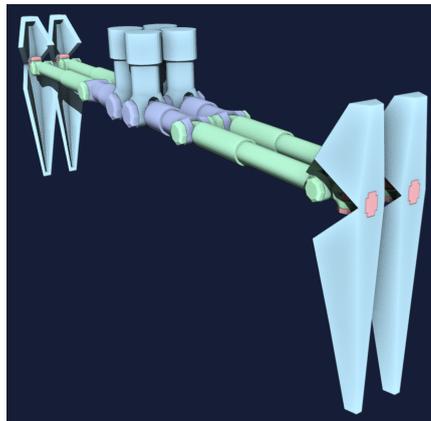


Figura 10: Modelo 3D da Locomoção tipo aranha finalizada. Fonte: Autor

3.2.4. MODELANDO A LOCOMOÇÃO TIPO DUAS RODAS

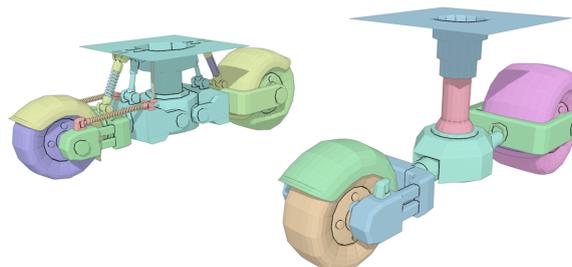


Figura 11: Versões anteriores da Locomoção do tipo Duas Rodas. Fonte: Autor.

Utilizamos da blocagem para exemplificar a forma de modelagem, no caso da locomoção do tipo Duas Rodas. Na figura 11, pode-se observar as versões anteriores da estrutura. Daí, realizamos o trabalho de *redesign* ao longo do processo de modelagem. Diferente das versões anteriores, o novo desenho (figura 12) focava na interação das conexões, utilizando várias formas de se conectar no mesmo eixo, expandindo, assim, as formas de se utilizar as conexões que foram pensadas no design da locomoção do tipo aranha, mantendo uma coesão estética para o design das locomoções. Mas antes de abordar a forma como foi feita a modelagem, resolvi dividi-la em três blocos partindo do eixo central, são eles: o bloco do corpo, o bloco dianteiro e o bloco traseiro, como veremos a seguir.

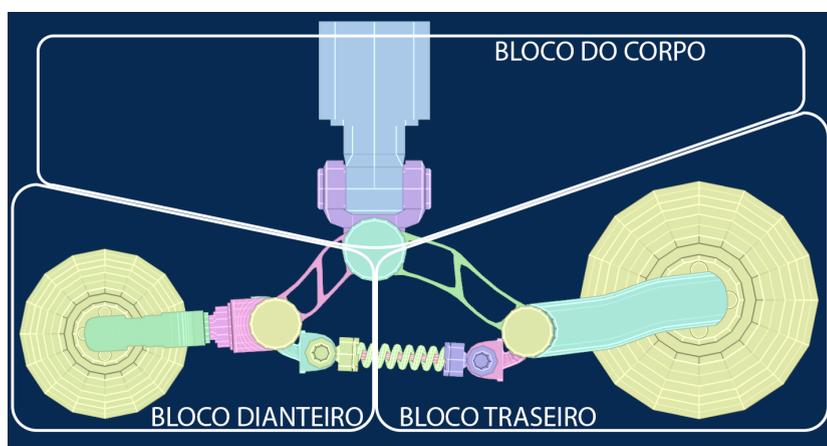


Figura 12: Divisão dos blocos da Locomoção de Duas Rodas. Fonte: Autor.

Bloco do Corpo: Brincando com a ideia apresentada anteriormente, utilizamos um pistão com uma conexão fêmea conectado a uma conexão macho/fêmea para assim conectar ao eixo central da locomoção. Diferente das versões anteriores, nessa nova forma, a parte mecânica de interação do corpo com a locomoção traz uma visão mais atrativa, por apresentar uma maior dinâmica em seu movimento.

Bloco Traseiro: Iniciando do eixo central, foram modeladas duas estruturas em paralelo que terminam em outro eixo que possui mais duas conexões vinculadas a ele. Uma delas é a estrutura da roda traseira que tem um design simples, com pouca curva, mantendo esse formato mais cúbico do personagem. E conectado a essa estrutura da roda traseira, está o eixo que tem a roda traseira vinculada a ele.

Bloco Dianteiro: tal qual o bloco traseiro, temos essa forma de estruturação acontecendo como se estivesse espelhado, mas quando chega na estrutura da roda dianteira, ela se torna totalmente diferente da traseira. Trazendo uma pré-estrutura da forma dianteira,

que era conectada por um pistão que permitiria que a estrutura da roda dianteira pudesse se projetar para frente. A estrutura da roda dianteira tem como design, um formato de “U”, onde nas pontas temos o eixo da roda dianteira conectado por conexão macho, permitindo com que o eixo possa se inclinar e fazer com que a roda gire além do próprio eixo. Pontas que na verdade são conectadas à base da estrutura por pistões que permitem com que elas se contraem ou expandam.

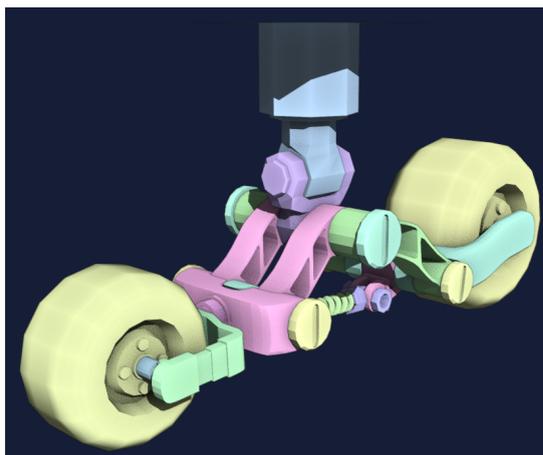


Figura 13: Modelo 3D da Locomoção do tipo Duas Rodas finalizado. Fonte: Autor.

4.RIGIFY

Como forma de ajuda na hora de construir um *Rig*, utilizamos plugins que auxiliassem na etapa de blocagem do *Rig*, deixando com que o artista *Rigger* tivesse acesso a uma biblioteca com pré-rigs que podem ser conectados para formar a estrutura do modelo 3D. Como exemplos temos *biped*, *CAT*, *auto-rig* e *RigiFy* do *Blender*, especificamente, que foi abordado durante a pesquisa para o desenvolvimento do projeto. Por não haver nenhum outro *addon* nativo do *Blender* que se assemelhe à funcionalidade de criação do *rig* como o *Rigify* para modelos não-orgânicos, foi decidido que a utilização do *RigiFy* se limitaria ao estudo de sua composição e organização do *bones* dentro dos modelos disponibilizado pelo *Rigify*.

4.1.LAYOUT DO RIGIFY

O processo de *Rigging* se utiliza de uma metáfora do corpo humano para a produção de suas estruturas. Temos o esqueleto composto por ossos, e partindo deles temos os músculos que vão se conectando, depois temos uma camada de gordura e por fim a pele. Os ossos sozinhos não conseguem mexer o corpo, mas os músculos fazem esse trabalho de contração e

expansão, assim trabalhando em conjunto com osso é possível ter a movimentação do corpo. Tendo isso em mente os *Bones* que são os ossos do modelo 3D, conseguem se movimentar sozinhos. Mas se levarmos em conta que uma musculatura, dependendo do movimento, mexe com muitos ossos, contrai estica vários músculos diferentes, animar isso manualmente demandaria muito tempo. Então criar um controle para interagir com esses conjuntos de *bones* facilita e economiza tempo no geral. E assim como o corpo humano tem camadas que vão dos ossos até a pele, pode-se imaginar a elaboração do *Rig* ocorrendo do mesmo modo, tendo uma camada responsável pela deformação da malha, outra que serviria como uma etapa mecânica, que a grosso modo seria responsável pela interação dos controles com os *Bones* responsáveis pela deformação, e por último, os *Bones* de controle, por meio dos quais o animador vai ter contato com *rig*.

No caso do *Rigify*, utilizamos o *Rig* humanoide que é gerado após a configuração das posições dos *bones* do modelo pré moldado, teremos a versão completa do *rig*. Após a análise das camadas do *Rig* geradas pelo *Rigify*, chegamos a cinco categorias: o *DEF*, de deformação; o *TWEAK*, serve para movimentar uma parte específica; o *MCH*, que são as ferramentas; o *ORG*, para a organização; e os de controle, que não ganharam uma nomenclatura específica. A estrutura de *bones* que serve para a deformação (*DEF*) não fica visível, e nenhum botão do menu gerado pelo *Rigify* lhe dá acesso a eles. Pode-se dizer que é uma forma de garantir que o animador não possa modificar a base, porque assim correria o risco de atrapalhar algum funcionamento dos controles. Partindo do *bone DEF* como base, temos os *bones* de ferramentas (*MCH*). Aqui utilizada a estrutura da perna, temos uma divisão no uso do *bone MCH*, trabalhando com uma versão *FK* e *IK* da perna. Diferente do *bone MCH*, o de Organização (*ORG*) trabalha de forma híbrida entre o *FK* e *IK*. Como o nome diz, ele serve para organizar quando o animador escolhe mudar de uma cinemática para outra. E por último, os *bones* de controle que se tornam a principal ferramenta do animador.

5. ELABORAÇÃO DO RIG

Aproveitando que durante a modelagem, as locomoções foram modeladas separadas do corpo, facilitou na hora de colocar todos os *bones* de deformação. Assim vimos, o personagem tem quatro estruturas de locomoção e um dos maiores trunfos dessa troca de locomoção está em como o *rig* é elaborado, e com isso em mente, um requisito que foi pensado e utilizado, foi de tratar cada locomoção como *rig* a partir do corpo principal. Dando autonomia para elaboração e desenvolvimento do *rigging* de cada locomoção sem precisar

interagir com corpo principal, inicialmente, pois posteriormente todas as locomoções serão integradas ao *rig* do Corpo.

Uma das coisas que se tem que levar em consideração na hora de se criar o *rig*, nesse caso estamos utilizando o software *Blender* como referência, é o *Edit mode* e o *Pose mode* que condiz com os dois modos do *rig*. No *Edit mode* temos o local para criação do *bone* e suas conexões. Mas, no entanto, no *Pose mode* temos outra forma de criar. É nesse espaço que a animação dos *bones* acontece, mas também é onde a configuração das funções também se manifesta. Funções essas chamadas de *Constraint*, que possibilitam várias modificações na funcionalidade dos *bones* além das suas hierarquias.

6. DESENVOLVIMENTO DO RIGGING

Por mais simples que seja na hora de definir como criar o *rig* para o personagem BO³, ele apresenta peças sólidas que não necessitam de deformação, tendo apenas algumas peças que sofreram com uma deformação da malha, mas a maior parte são objetos que não sofreram com a deformação. Assim, podemos separar as peças por dois grupos. Os que sofreram uma deformação da malha 3D do modelo, receberam um conjunto de *bones* que trabalharam na influência dos pontos em que a malha pode se deformar. Enquanto o outro grupo recebeu apenas um *bone* por modelo, por não ter uma necessidade de deformação em nenhum ponto específico da malha.

Por mais que essa etapa seja importante, por se tratar dos *bones* de deformação, que são os *bones* responsáveis por dar movimento ao objeto, eles são apenas mais uma etapa a se pensar na hora de elaborar o *rig*. Por mais importante que sejam, temos que levar em consideração a influência das outras camadas de *bones*. Por mais simples que seja a quantidade de camadas, é bom manter a ideia que o animador irá animar pelos *bones* de controles, não diretamente no *bone* de deformação, mas sempre existe uma excessão.

6.1.IDEALIZANDO O RIG DOS MODELOS

6.1.1.O CORPO

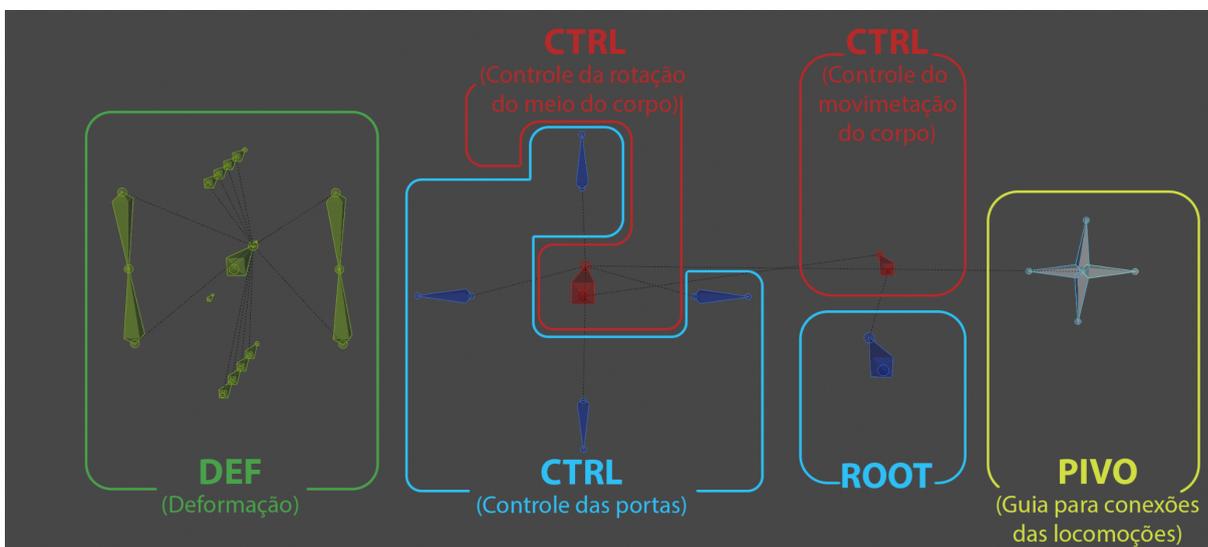


Figura 14: Conjunto de *bones* separados por camadas e funções. Fonte: Autor.

Por ter como forma básica um cubo, o corpo possui inicialmente a linguagem mais simples de se fazer, onde de início me preocupei em resolver o básico para que com essa etapa resolvida, eu pudesse me preocupar com a incrementações. Mas antes de começar a adicionar os *bones* e assim criar um esqueleto base para o corpo principal, como forma de planejamento, comecei a fazer um esboço das ações necessárias, como, por exemplo, abrir as portas, pegando tudo que foi idealizado na etapa de modelagem, chegamos num formato simples e com objetivo direto. Assim, consegui idealizar como seria a estrutura do esqueleto. Partindo da necessidade do personagem de girar para trocar a locomoção, foi pensado num controle que tivesse nele os controles da troca de locomoção. Seguindo com a ideia de controle, temos também um *bone* para movimentar o corpo como o todo, o *Bone ROOT* (que serve como o primeiro *bone* da hierarquia), aquele que tem controle sobre todos os outros *bones*. Tendo esboçado todos os controles necessários, parti para a elaboração da estrutura de *bones* específicos para Deformação (DEF), sendo bem metódico na hora de colocar os *bones* para cada parte que tem alguma movimentação. Por exemplo, as portas que possuem dois tipos diferentes, uma do tipo simples que é dividida ao meio e abre para fora. Enquanto o outro tipo, é uma porta dividida em quatro pedaços que se empilha no canto quando aberta. Assim, cada pedaço recebe um *bone* independente, podendo interagir com cada pedaço sem se preocupar com os outros sobre alguma transformação adjacente. Tirando os *bones* das portas, há apenas três *bones* do tipo deformação: a parte da frente do corpo, a parte de trás e o meio,

onde ficam as portas. Criamos então uma estrutura básica e simples para o corpo, que tem, como sua modelagem, a parte mais simples de todos os modelos 3D. Por ter essa forma simples, decidi não utilizar nenhum *bone* para fazer o intermédio entre o *bone* de deformação com os *bones* de controle, busquei utilizar os *bones* de controle para controlar diretamente os *bones* de deformação.

Assim, cada conjunto de *bones* de deformação referente a uma porta, recebeu um *bone* de controle, tendo como forma de interação apenas uma ação de locomover o *bone* no eixo Y (Local). Assim, foi aplicado um modificador (uma *constraint*) no *bone* de controle para limitar a distância que ele poderá percorrer, e com essa distância pré-determinada é possível aplicar um modificador de *Transformation*⁵ no conjunto de *bones* de deformação da porta. No tipo de porta dividida ao meio, a *constraint transformation* utiliza da locomoção do controle para provocar o giro da porta, abrindo as portas. Mais exatamente, utilizando a movimentação no eixo Y (Local), já tendo o seu intervalo pré-definido, faz com que o *bone* de deformação gire no eixo Z (Local), também tendo o seu intervalo pré-definido. Temos outro tipo de porta que é dividida em quatro partes, que possui como forma de funcionamento correr por um trilho até o final ou até a próxima parte. Assim, quando a parte chegar no final, ela rotaciona no próprio eixo, ficando numa posição que a próxima se encaixe com ela e no final teríamos todas as partes empilhadas numa extremidade.

Diferente de outro tipo de porta, na hora de pensar como funcionaria a interação do controle com as partes da porta, foram idealizadas quatro formas para se trabalhar com essa interação. Uma das opções era de se animar os *bones* de Deformação e por meio de uso de uma *constraint Action*⁶, que utilizaria o caminho feito pelos *bones* de controle para substituir a linha do tempo da animação. Outra opção seria a utilização de uma curva para criar um trilho em que os *bones* de deformação correriam. Por mais que a utilização do trilho fosse uma boa ideia, ela só resolveria a parte da locomoção do *bone*, enquanto a rotação teria que acontecer por intermédio da *Constraint Transformation*. Mesmo utilizando a junção das *constraint* para gerar a animação da porta, ainda apresentaria uma necessidade de refino para atrasar a rotação das partes.

Refino esse que utilizaria os *Drivers* para criar esse atraso, sendo também uma das opções para gerar a automação da porta controlada pelo *bone* de controle. Assim, aplicando o *driver* diretamente nos eixos que sofreram modificação, por exemplo no eixo Y (Local) da posição, que serviria para correr a peça no trilho, enquanto aplicava outro *drive* no eixo X

⁵ Uma Constraint do tipo Transform.

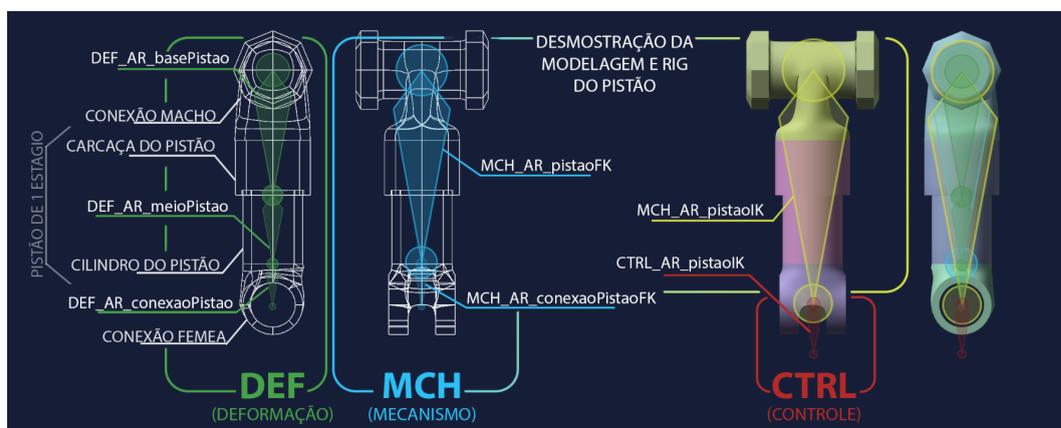
⁶ Uma Constraint do tipo Relationship.

(Local) da rotação, permitindo à peça se inclinar. E por meio do uso dos gráficos poderia refinar as transformações geradas pelo *drives*, como se estivesse animando.

A última opção de automação da porta, é pelo uso de um conjunto de *Constraint Transformation* para cada *bone* correspondente a uma parte. Por separar em até três modificações geradas pela *constraint*, podemos utilizar parâmetros diferentes para gerar a animação final. Por ter esse controle mais simples e fácil de alterar, caso a automação apresente alguma forma de irregularidade. Com essa praticidade, foi a opção escolhida para automatizar as portas.

Além dos controles das portas, temos o controle do corpo, responsável por três *bones* DEF, que juntos se tornam o corpo completo. Sendo um *bone* que fica ao centro do modelo, e utilizando da *constraint Copy Transforms*⁷, que permite o *bone* copiar o estado do *bone Target*⁸, fazendo com que o *bone Owner*⁹, responsável pela frente e o *bone* responsável pelas costas, fosse automaticamente para posição central, junto com o *bone* do corpo do central. Uma das formas que encontrei de resolver esse problema, foi alterar uma opção dentro das configurações da *constraint*. Alterando o tempo em que o modificador influencia o *bone*. Assim, podendo escolher entre a aplicação acontecer no momento, antes ou depois de alguma alteração. No menu *Mix* escolhi a opção *before*, que seria para o *bone* ser influenciado depois, deixando com que os *bones* da frente e das costas se mantivessem nos seus pontos de origem.

6.1.2.LOCOMOÇÃO DO TIPO ARANHA



⁷ *Constraint* responsável por copiar todos parâmetros (posição, rotação e escala) do *bone Target*, para o *bone Owner*.

⁸ Responsável por influenciar as alterações no *bone* com a *constraint*.

⁹ *Bone* proprietário, aquele que irá sofrer a influência gerada pelo *bone Target*.

Figura 15: Demonstração da modelagem e rig feito nos pistões da Locomoção do tipo aranha. Fonte: Autor.

A locomoção do tipo aranha é composta por quatro membros que se espelham, optei por comentar só um deles, ao invés de comentar sobre todos, por serem uma cópia. Assim, como foi abordado na parte de modelagem e com intenção de simplificar, abordarei o *rigging* de um pistão, demonstrando toda a sua base das camadas implementadas nesse rig. Podemos ver na figura 11 um pistão de um estágio e o *rig* com 4 camadas, partindo dos *bones* de deformação até o *bone* de controle. Sendo três *bones* para deformação, dois para mecanismo FK, um de mecanismo IK e um para controle. Dividindo a deformação em base, meio e conexão. Pude manter a parte do objeto que não pode sofrer com a deformação da malha como a base e a conexão, diferentemente do meio, que vai ter uma deformação na malha para expressar a forma de estiramento do pistão. Os *bones* de mecanismo do tipo FK, se tornam a primeira camada de interação com os *bones* de deformação. Dividido em dois, um responsável pelo pistão e outro pela conexão, fazendo que ao esticar o *bone* de deformação do meio ele não influencie o *bone* de deformação da conexão. E por meio das *constraints* temos uma hierarquia sendo construída, com os *bones* de mecanismo FK como pai e *bones* de deformação como os filhos, copiando a sua posição, rotação e tamanho (somente o *bone* de deformação do meio).

A próxima camada a ser incrementada foi do *bone* de mecanismo IK, por uma escolha pessoal (o fato de gostar de utilizar IK nas pernas dos personagens). Pelo fato de estarmos abordando apenas um objeto, não o seu conjunto todo, não é possível abordar direito como foi idealizado o mecanismo de IK da locomoção. Após habilitarmos as *constraints* que permitem o *bone* inverter a cinemática (IK) no *bone* de mecanismo IK, tive que definir qual seria o *bone* de controle desta cinemática, assim, se tornando necessário criar uma camada de *bones* de controle, que ficaria aparente para o animador no final do desenvolvimento. Após definir o controle e por ser parte de uma estrutura robótica, não podemos deixar com que os *bones* da hierarquia afetados pelo IK tivesse todos os eixos livres para sua movimentação, por se tratar de uma peça que rotaciona em um eixo. Assim, entrando nas configurações dos *bones*, na aba IK é possível restringir alguns eixos de sofrerem interações e é possível habilitar uma função (IK *Stretch*) que permite esticar o *bone*, de forma que ele não desconecte do *bone* de controle. Repetindo o mesmo processo feito com *bone* de deformação em relação aos *bones* de mecanismo FK, mas com os *bones* de mecanismo FK sendo o filho dos *bones* de mecanismo IK, copiando posição, rotação e escala, mas sempre mantendo os *bones* que interagem com as conexões sem deformação da malha.

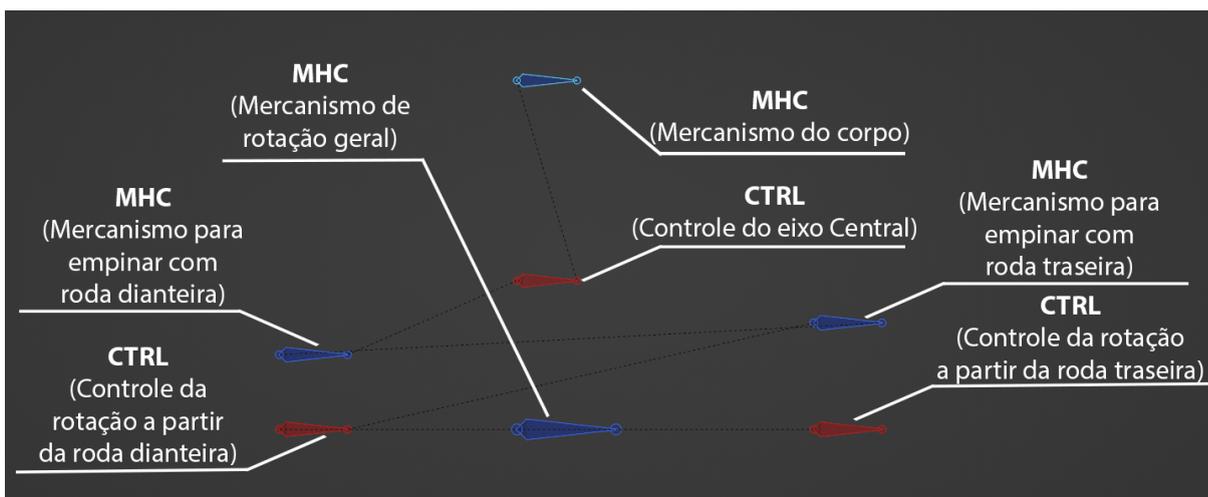


Figura 17: Estrutura principal da movimentação da Locomoção do tipo Duas Rodas. Fonte: Autor.

Ao estudar o *rigging* gerado pelo *Rigify*, mais precisamente a estrutura do pé do *rig* humanoide, percebi uma forma de adaptar as conexões entre os *bones*, para construir uma estrutura que permitisse ao modelo 3D não só andar e girar as rodas, mas poder se inclinar em seus dois eixos (sendo eles, a roda traseira e dianteira), permitindo uma representação de uma moto (objeto em que essa locomoção foi baseada). Estrutura essa que desenvolvi tendo três linhas de interação. Em sua base temos três *bones* de controles: sendo o do “meio” para inclinar no eixo Y (Local), além de ser o pai de toda hierarquia; o de “trás” não continua a estrutura da hierarquia, mas a sua forma de controle funciona por meio da *constraint*; o da “frente” continua com a hierarquia da estrutura. Assim, na segunda linha temos novamente mais três *bones*, sendo um de controle e os outros dois de mecanismo. O *bone* do mecanismo “traseiro” segue sendo o filho do *bone* de controle dianteiro e *pai* do *bone* de mecanismo “dianteiro”, mas tendo implementado o uso de *constraint* que sofrem influência do *bone* de controle traseiro. Enquanto o *bone* de mecanismo “dianteiro” é *filho* do *bone* de mecanismo “traseiro” e *pai* do *bone* de controle “central”, sem o uso de *constraint*. O *bone* de controle “central” acaba sendo o último *bone* pai da hierarquia, tendo como utilidade apenas de fazer leves correções na postura do personagem e em relação à locomoção. E na última linha temos um *bone* do tipo mecanismo, que serve para gerenciar a forma com que o corpo reagiria em relação a inclinação da locomoção.

6.2.LAYOUT DE INTERAÇÃO

Por mais que o design octogonal do *bone* seja funcional para a criação do *rig*, quando chegamos na etapa de animação o design pode se tornar poluído por apresentar tantos

controles na tela. Por isso o *Blender* disponibiliza outros designs para representação dos *bones*, deixando que o usuário escolha qual design se encaixa melhor a sua necessidade. Mesmo podendo variar o design dos *bones*, o animador ainda poderá ter uma possível poluição visual e um questionamento sobre o que cada controle faz. Pensando nesse fato que o design fornecido não é suficiente para expressar a sua necessidade, é possível customizar o *bone* com um objeto 3D. Assim, podendo customizar todos os *bones* de controle, para que sejam intuitivos na hora de expressar as suas possíveis ações.

6.2.1. DESENVOLVENDO O DESIGN

Após definir todos os *bones* de controle do personagem, iniciei a etapa de customização do design do controle, com a intenção de melhorar a interação do animador com os controles mais intuitivos para as suas ações repetitivamente. Como forma de exemplificar essa etapa, vou citar a elaboração do design dos controles do modelo do corpo, por ser a grande base inicial de todo modelo. Onde os seus controles são *ROOT_CP*, *CTRL_CP_corpo*, *CTRL_CP_rotLoc*, *CTRL_CP_abrePortaHD*, *CTRL_CP_abrePortaDR*, *CTRL_CP_abrePorta2R*, *CTRL_CP_abrePortaPT*.

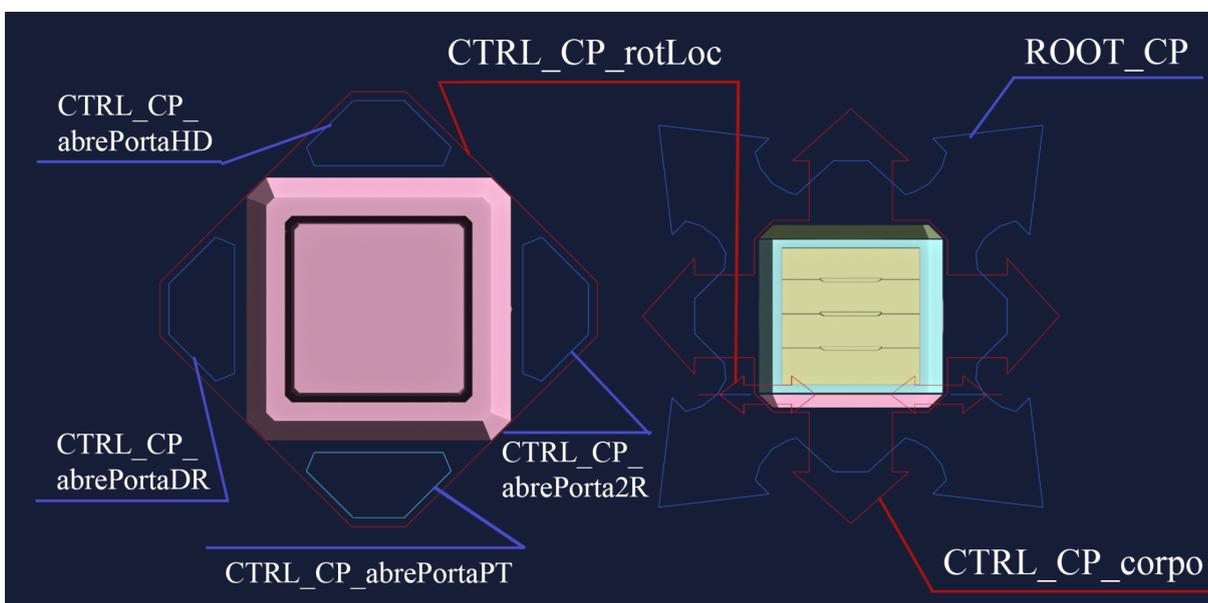


Figura 18: Layout de controles com design finalizado. Fonte: Autor.

ROOT_CP: tem a função de controlar o movimento geral, sem o pai da hierarquia de todos o *bones*, como proposta de design de ter essa forma de movimentação, foi idealizado em

um formato de quadro com seta saindo de suas arestas em sentido de X, para expressar essa sensação de liberdade.

CTRL_CP_corpo: por ter características bem parecidas com “ROOT_CP”, de movimentar o corpo, mas diferente dele esse *bone* é vinculado somente ao corpo e se tivesse alguma locomoção ativa ela não influenciaria nelas. Sabendo dessa diferença, segue quase o mesmo design pensado anteriormente com esse quadro que de suas arestas saem setas, mas agora no sentido de cruz. Usando a visão de cima do modelo, pode-se ver os dois desenhos das setas se complementando, assim, usando um pouco de ilusão de ótica, eu brinco com a ideia de que na verdade, tanto o controles “CTRL_CP_corpo” e “ROOT_CP” na verdade têm oitos setas indicando que ele podem se movimentar em todas as direções.

CTRL_CP_rotLoc: controle responsável por girar a parte do meio do corpo, assim, o design tinha que passar a ideia de rotação, mais ainda manter a estética de quadrados que tinha sido apresentado no design de controles anteriores. Na verdade, o quadrado é um octógono, mas durante o seu desenvolvimento a base utilizada foi um quadrado que veio se tornar um octógono por questões mais estéticas. Se olharmos de frente é possível ver esse octógono com um giro de 45° em comparação ao corpo do personagem, mas só essa distorção em relação ao corpo não é suficiente para especificar essa intenção de rotação do corpo. Pensando nisso, utilizando cada aresta de maior dimensão, foi adicionada uma seta de duas direções para exemplificar a rotação que pode seguir sentido horário e sentido anti-horário.

CTRL_CP_abrePorta[X] (sendo “X” qualquer abreviação de locomoção): os designs das quatro locomoções são os mesmos, o que muda é o sentido da rotação que ela adota no modelo. Tendo em mente que o controle vai estar em modo de repouso e ao ser movimentado para cima ele abre a porta. E tendo o design “CTRL_CP_rotLoc” pronto, utilizando com base para desenvolvimento do design escolhi um triângulo com forma base, que na visão de frente do modelo, é possível ver a base do triângulo um pouco acima do corpo, mas ainda mantendo a área toda dentro do limite criado pelo design do CTRL_CP_rotLoc. Assim, com esse formato básico, que posteriormente se tornou um hexágono, que ainda remetia à forma de um triângulo, e por estar dentro da área da rotação do corpo, tive a intenção de mostrar que ela estava vinculada às portas e que ao movimentar para cima, abriria a porta.

6.2.2.UTILIZAÇÃO DE VARIÁVEIS

Como o Rigify utiliza um painel dentro do menu quando está no *Pose Modo*, permite ao animador que controle a visibilidade dos controles visíveis na *viewport* e também

modificações de como algumas interações podem acontecer, como troca do FK para o IK. Por ser um sistema vinculado ao rigging gerados pelo Rigify, a reutilização fica precária, e por não ter conhecimento suficiente ainda para desenvolver um painel do zero. Optei por utilizar uma função do menu do Objeto que permite criar variáveis, permitindo que o animador interaja com números do tipo inteiro, número flutuantes, e até mesmo com texto. Ao criar e configurar a sua variável, na aba *Custom Properties*, a variável aparecerá para interação na menu *Properties* do objeto na *Viewport*, mas tem que estar ciente de que a variável que é ligada ao objeto onde ela foi criada, assim, não é possível visualizar essa variável em outro objeto. Tendo isso em mente, comecei a pensar quais os melhores *bones* para adicionar cada variável, com a ideia de deixar mais intuitivo qual *bone* permitiria a alteração da hierarquia de FK para IK, por exemplo. Na parte da aranha, a primeira peça dos membros que serve como base, carregaria essa função. No caso da função de troca entre IK e Fk, criei uma função do tipo inteiro que vai de 0 a 1 com opção, sendo 0 falso e 1 verdadeiro, por não ter uma opção de criação de variável do tipo verdadeiro ou falso, improvisei com uma interpretação de como funcionaria uma variável desse tipo.

Após criar as variáveis, era necessário transmitir o seu valor, para que assim a ação atribuída à interação se torne válida. Para fazer essa conexão o uso do *Drive* é mais que bem vindo, sendo bem versátil na hora de expressar a forma pensada para a utilização da variável criada. Por mais que no rig do personagem, a maioria das variáveis criada funcionasse como verdadeiro ou falso.

7. CONSIDERAÇÕES FINAIS

Ao longo do desenvolvimento desse texto percebi que enquanto o personagem estiver em construção, ele vai está sempre sendo reelaborado, redesenhado, seja na modelagem, ou até mesmo no rig, por abordar uma ideia que trabalhe melhor a proposta. E um dos fatores que permite essa fluidez nas alterações, se dá ao caso do personagem não ter uma finalidade a não ser possuir quatro locomoções distintas. Posteriormente, pode acontecer, ao se deparar com o personagem em outra mídia, talvez ele esteja diferente, por ganhar um objetivo, pode ocorrer do personagem amadurecer para uma versão mais idealizada, com o objetivo que será impresso.

No decorrer do desenvolvimento do personagem, por mais que tivesse a questão estética, priorizei o design do personagem e suas locomoções. O *rig* teve o papel fundamental nas decisões durante a modelagem, buscando uma forma mais prática de se aplicar o *rig*

posteriormente, a poder criar estruturas mais complexas para demonstrar formas de organizar as interações. Outro fato que podemos perceber, por não ter um norte definido de como o projeto precisava estar no final. O personagem sofreu com mudanças constantes, enquanto a pesquisa se desenrolasse, tendo como uma das formas de locomoção possíveis inicialmente um veículo de quatro rodas. Que deu lugar à locomoção do tipo Drone, que não foi abordado neste texto por ter como construção uma repetição da locomoção do tipo aranha, adaptada para um contexto diferente.

Uma das coisas que aprendi durante o desenvolvimento do *rig*, foi a forma de como pensar o *bone*. Se pensamos no *bone* em comparação ao ser humano, temos o pensamento do indivíduo em relação a si mesmo e em relação à sociedade. E utilizando esse contexto é uma forma de se pensar como o *bone* tem a sua individualidade, sendo a sua orientação (Local), enquanto dentro de uma hierarquia, o pai tem influência direta nas ações do filho ignorando completamente a vontade do filho. Com o passar do tempo, o filho começa a tomar a sua escolha, assim a influência de fora da sua hierarquia começa a provocar mudança. Assim o filho para de ser uma cópia do pai, moldando uma personalidade diferente. Com o *bone*, pode-se usar, em comparação, as *constraints*, que permitem com que outros *bones* influenciam características específicas, podendo cortar totalmente o vínculo com o pai da hierarquia ou usar de forma adicional. E com essa reflexão me ajudou a entender como o *bone* pode sofrer influência de outros *bones* e qual deles teria prioridade nessa influência.

8.BIBLIOGRAFIA

VEGDAHL, Nathan et al. *Rigify*. 3.4. [S. l.], 15 fev. 2023. Disponível em: <https://docs.blender.org/manual/en/3.4/addons/rigging/rigify/index.html>. Acesso em: 15 jun. 2023.

ARSHAD, Mohd Rosli et al. *Physical Rigging Procedures Based on CharacterType and Design in 3D Animation*. International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Número-3, September 2019. Disponível em: https://www.researchgate.net/publication/336306246_Physical_Rigging_Procedures_Based_on_Character_Type_and_Design_in_3D_Animation. Acesso em 07/06/2023.

ANTUNES, Diego de Paula, *Boas Práticas para a Construção de Rigs para Personagens Humanóides na Técnica 3D Digital*. Belo Horizonte: Escola Belas Artes, UFMG, 2014.

Trabalho de Conclusão de curso. Disponível em:

https://issuu.com/bibliobelas/docs/diego_depaula_tcc. Acesso em 15/06/2023.

PINHEIRO, Ricardo Miranda. *Desenvolvimento de rigging e movimento para cães em Animação 3D*. Escola das Artes da Universidade Católica Portuguesa, Mestrado em Som e

Imagem, 2015. Disponível em:

<https://repositorio.ucp.pt/bitstream/10400.14/19500/1/Dissertacao%20-%20Ricardo%20Pinheiro.pdf>. Acesso em 15/06/2023.