



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA**

Ivan Neves Silveira Bandeira de Souza

Posando Personagens 3D com Desenhos de Figuras de Palito

RECIFE

2024

**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA**

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ivan Neves Silveira Bandeira de Souza

Posando Personagens 3D com Desenhos de Figuras de Palito

Monografia apresentada ao Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE), como requisito parcial para conclusão do Curso de Ciência da Computação, orientada pelo(a) professor(a) Silvio de Barros Melo.

RECIFE

2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Souza, Ivan Neves Silveira Bandeira de.
Posando Personagens 3D com Desenhos de Figuras de Palito / Ivan Neves
Silveira Bandeira de Souza. - Recife, 2024.
36

Orientador(a): Silvio de Barros Melo
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado,
2024.

Inclui referências.

1. 3D. 2. 2D. 3. Animação 3D. 4. Desenho. 5. Computação Gráfica. I. Melo,
Silvio de Barros. (Orientação). II. Título.

000 CDD (22.ed.)

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ivan Neves Silveira Bandeira de Souza

Posando Personagens 3D com Desenhos de Figuras de Palito

Monografia submetida ao corpo docente da Universidade Federal de Pernambuco, defendida e aprovada em 21 de março de 2024.

Banca Examinadora:

Silvio de Barros Melo

Doutor(a)

Orientador(a)

Filipe C. A. Calegario

Doutor(a)

Examinador(a)

AGRADECIMENTOS

Queria agradecer especialmente aos meus amigos Henrique Mariz, Guilherme Augusto, Matheus Lima e Lucas Lin. Amizades que fiz neste curso, que sinto que durará a vida inteira e que tanto me ajudaram durante todo esse longo tempo de graduação, deixando essa jornada inteira mais leve e suportável

RESUMO

Artistas e animadores tentam combinar, à medida do possível, técnicas de animação de 2D e 3D no seu fluxo de trabalho de diversas maneiras para facilitar ou agilizar o processo de animação. Isso é possível pois o ponto fraco de um método pode ser o ponto forte de outro e essa mistura pode eliminar ou mitigar esses pontos fracos. Um exemplo destes pontos fortes e fracos complementares é o processo 2D e o 3D. No 2D, há uma facilidade e agilidade para se esboçar quadros de uma animação, porém o 2D requer muitos quadros a serem desenhados; No 3D, não se faz necessário muitos quadros pois quadros serão interpolados suavemente, mas posar um objeto é mais complexo, pois manipular o objeto 3D por uma interface 2D, a tela do usuário, possui limitações. Este trabalho tem como objetivo introduzir uma maneira de trazer a agilidade dos desenhos de baixa fidelidade 2D para o processo de animação 3D através de desenhos de figuras de palito para tornar o processo de criação de keyframes 3D mais rápido. Informação sobre o transform das figuras será recuperado e então aplicado às partes do modelo 3D ou ao seu osso equivalente em sua malha esquelética. A ferramenta criada então por esse método foi então avaliada por usuários e o tempo de geração de um keyframe com o método tradicional e o método proposto foi medido, a fim de comparar e confirmar se o método proposto realmente é mais eficiente. O artigo é finalizado ao se demonstrar que de fato há uma redução considerável no tempo para criar um keyframe, porém o desenho com figuras de palito provou-se não ser o mais adequado, pois retas não convém noção suficiente de perspectiva, o que causou certa confusão durante o processo de desenho dos usuários.

Palavras-chave: Animação; 2D; 3D; Espaço Projetivo; Figuras de Palito

ABSTRACT

Artists and animators try to combine, as much as possible, animation techniques from 2D and 3D in their workflow in different ways in order to facilitate or speed up the animation process. This is possible because one method's strength might be the other method weakness, and combining them might eliminate or mitigate a method weakness. One example of those complementary strengths and weaknesses is that 2D animations can be made with a rough sketch first to be made quickly, but it will always require huge amounts of frames. 3D animation does not require too much frames to be created, since keyframes can be interpolated, but producing a keyframe might be hard since mapping the 2D input from the user to the 3D object can have limitations. This work has the objective of introducing a method to bring the agility of rough 2D sketches to the 3D animation process through the drawing of stick figures to speed up the creation of keyframes for 3D objects. The 3D Transform information will then be recovered and then applied to the object part or the object skeletal mesh. The tool created through this method will then be tested by users and the time to create a keyframe with this method and the traditional one will be measured and then compared and then prove if the method is indeed faster or not. The article finishes demonstrating that indeed the method is capable of speeding up the creation of keyframes, but stick figures might not be the most adequate way to approach this method, since lines do not convey information about perspective well enough, which caused some confusion in the users in the testing phase.

Keywords: Animation; 2D; 3D; Projective Space; Stick Figures

Sumário

1. Introdução.....	8
2. A Proposta do Método.....	13
3. Desenvolvimento do Método.....	17
3.1. Resolvendo a Equação Unicamente.....	19
3.2. Resolvendo com um Ponto Conhecido.....	20
3.3. Extraíndo a Rotação e Translação.....	21
3.4. Solução Única para a Escala.....	23
3.5. Casos Especiais.....	23
4. Implementação.....	24
4.1. Corrigindo o Input do Usuário.....	26
4.2. Posando o Objeto.....	29
4.3. Testes com Usuários.....	33
5. Conclusões.....	36
6. Trabalhos Futuros.....	37
7. Bibliografia.....	39

Lista de Figuras

Figura 1 - Malha Esquelética de um Modelo 3D.....	8
Fonte: Google Imagens.....	9
Figura 2 - Motion Capture utilizado na indústria de filmes para captar expressões faciais.....	9
Fonte: BBC - Motion capture tech from Avatar Films used in Disease Research.....	9
Figura 3 - Cascadeur automaticamente movimenta os outros ossos da malha apenas ao movimentar o pé direito.....	10
Figura 4 - Uso híbrido do 3D e 2D na etapa de key-animation da obra “Attack on Titan”, produzido pelo Wit Studio.....	11
Figura 5 - Renderização final da cena de “Attack on Titan”.....	11
Figura 6 - Uso híbrido do 3D para gerar sprites em pixel art 2D em Dead Cells.....	12
Figura 7 - Cel Shading, técnica para renderizar objetos 3D como cartoons.....	12
Figura 8 - Rigging de um braço robótico ajustado para ser representado por círculos.....	14
Figura 9 - Exemplos de Renderização de uma malha de esqueleto.....	14
Figura 10 - Desenhos de figuras de palito.....	15
Figura 11 - Exemplos dos desenhos dos pares de retas para cada osso.....	16
Figura 12 - Esboço da figura de palito de uma malha esquelética de um ser humano a ser desenhada.....	16
Figura 13 - Segmento AB da figura de palito que representa um osso.....	17
Figura 14 - Representação do espaço projetivo.....	18
Figura 15 - Exemplo de representação dos palitos no espaço euclidiano.....	22
Figura 16 - Desenho de uma reta feita pelo usuário.....	24
Figura 17 - O objeto imitando uma cobra e as retas AB e CD geradas pelo script, parcialmente ocluídas pela malha triangular dos cubóides.....	25
Figura 18 - Os representantes dos ossos do objeto imitando uma cobra, sem a oclusão da malha triangular.....	25
Figura 19 - Segmento válido está destacado de verde.....	26
Figura 20 - Reta CD desenhada. Input do usuário em vermelho semi-transparente. A reta CD corrigida está pintada de azul.....	27
Figura 21 - A ferramenta por completo. À esquerda, um preview da cena. À direita, a lista de ossos do objeto que podem ser desenhados.....	28
Figura 22 - Desenho da reta atualiza o transform do objeto. À direita, aparece o novo osso que pode ser selecionado para ser desenhado.....	28
Figura 23 - Mudança no resultado do transform dado pelas duas soluções.....	29
Figura 24 - Preview, junto com a representação dos ossos, renderizados atrás do desenho das retas feitas pelo usuário.....	30
Figura 25 - Visualização do espelhamento das soluções, mantendo a reta AB mas mudando a reta CD e o ponto M.....	31
Figura 26 - Os três desenhos que os usuários de referência para posar o objeto de testes.....	32
Figura 27 - Gizmos que permitem a manipulação do objeto da forma mais tradicional.....	32

1. Introdução

Na animação 2D, cenários estáticos, como o plano de fundo, podem ser estáticos e usar-se de técnicas como paralaxe para dar a ilusão de movimentação, mas objetos dinâmicos, como personagens, que se movimentam numa cena, precisam ser desenhados quadro a quadro e uma animação precisa de milhares deles. Desta forma, desde o advento da animação 2D no século passado, através de nomes como Walt Disney, técnicas para a criação de animação foram estabelecidas e são usadas até hoje, como o uso de Extremes e In-Betweens(Williams, 2001, p. 48), animação Straight Ahead, Pose to Pose e uma combinação desses dois, além de se usar Storyboarding(WILLIAMS, 2001, p. 67) para saber como a cena irá se desenrolar e o desenho de keyframes ou key animation, para que os animadores de in-betweens possam trabalhar. Essas técnicas de planejamento são extremamente importantes, pois mudanças futuras, como trocar a posição da câmera da cena, grande parte ou todo o trabalho feito para a cena até então será perdido, mas o storyboarding e os key-frames podem ser feitos rapidamente por esboços de baixa fidelidade, para testar e então decidir como será a animação final.

No mundo 3D atual, a técnica predominante é o Skinning Animation ou Skeletal Animation(Gregory, 2019, p. 725).

Figura 1 - Malha Esquelética de um Modelo 3D



Fonte: Google Imagens

Com esta técnica, toda animação feita para um personagem pode ser aproveitada independente das mudanças que se faz com a câmera ou a cena, como se vê nos jogos(Gregory, 2019, p. 725), além de ser uma abordagem de animação diferente, já que é possível interpolar os ossos entre os keyframes(Gregory, 2019, p. 729). Todavia, posar os ossos da malha esquelética para produção de um keyframe não é simples. Isso pois o objeto e a cena estão todos em três dimensões, enquanto o input do usuário vêm de um mapeamento do 2D da tela para este 3D, tendo um grau de liberdade a menos. Desta forma, é necessário trocas constantes de ângulo da câmera para posicionar o osso no local correto, o que dificulta a produção de animações. Por isso, tecnologias como o Motion Capture foram desenvolvidas para aliviar esses problemas e facilitar a produção de animação 3D, funcionando de maneira excelente quando se necessita de animações realistas, como acontece na indústria de jogos e cinema.

Figura 2 - Motion Capture utilizado na indústria de filmes para captar expressões faciais



Fonte: BBC - Motion capture tech from Avatar Films used in Disease Research

Ferramentas também estão sendo desenvolvidas a fim de facilitar o processo de produção de keyframes para modelos 3D, como o caso do software Cascadeur, que utiliza IA para esse fim, quando ao mover um único osso da malha, todos os outros serão movidos automaticamente pela I.A numa espécie de auto-completar.

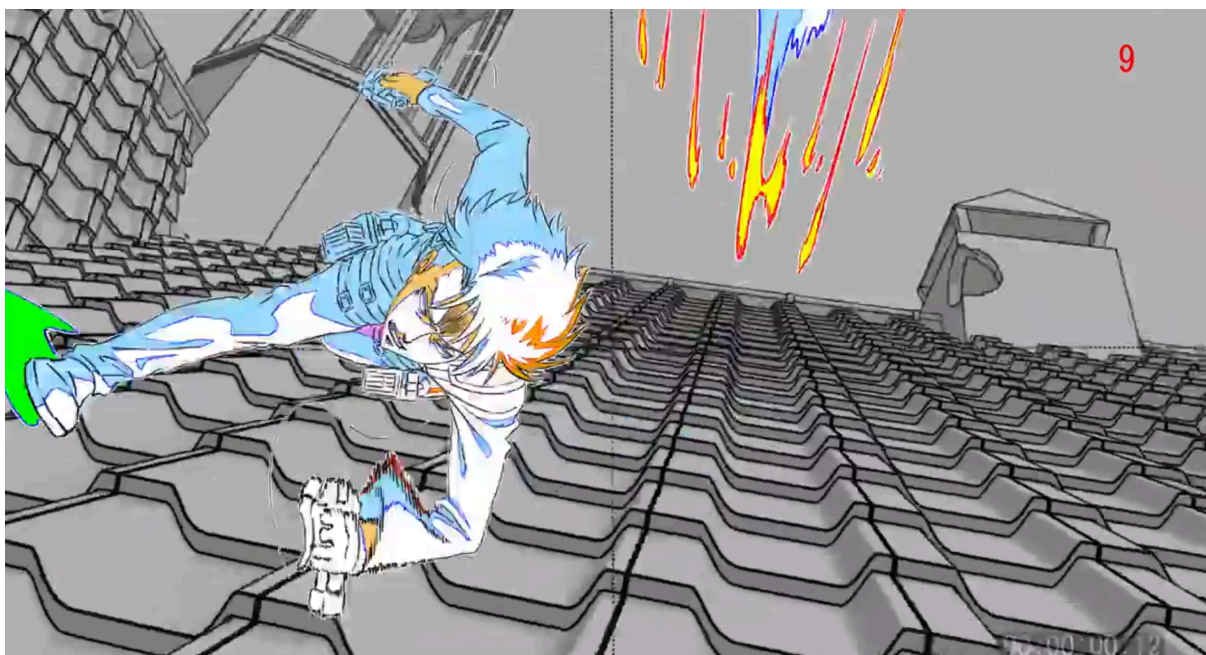
Figura 3 - Cascadeur automaticamente movimenta os outros ossos da malha apenas ao movimentar o pé direito.



Fonte: Canal oficial do Cascadeur no Youtube

O processo de animação 2D e 3D possui vantagens e desvantagens diferentes e da mesma forma que o 2D puro tende a fazer uma abordagem híbrida entre Straight Ahead e Pose to Pose a fim de eliminar as desvantagens de ambos os métodos(WILLIAMS, 2001, p. 63), abordagens híbridas entre 2D e 3D são comuns(MLYNARČÍKOVÁ, 2020) e existem diversas maneiras de combinar os dois métodos. Por exemplo, O estúdio Wit Studio, em 2019, combinou o 3D e o 2D para produzir um episódio de “Attack on Titan” ao utilizar um cenário renderizado em 3D com cel-shading e os personagens com 2D tradicional para esboçar os keyframes de uma cena de ação extremamente dinâmica e rápida. Na renderização final, o cenário recebeu texturas também e não foi desenhado à mão.

Figura 4 - Uso híbrido do 3D e 2D na etapa de key-animation da obra “Attack on Titan”, produzido pelo Wit Studio



Fonte: Twitter do Wit Studio - 19 de Setembro de 2019

Figura 5 - Renderização final da cena de “Attack on Titan”.

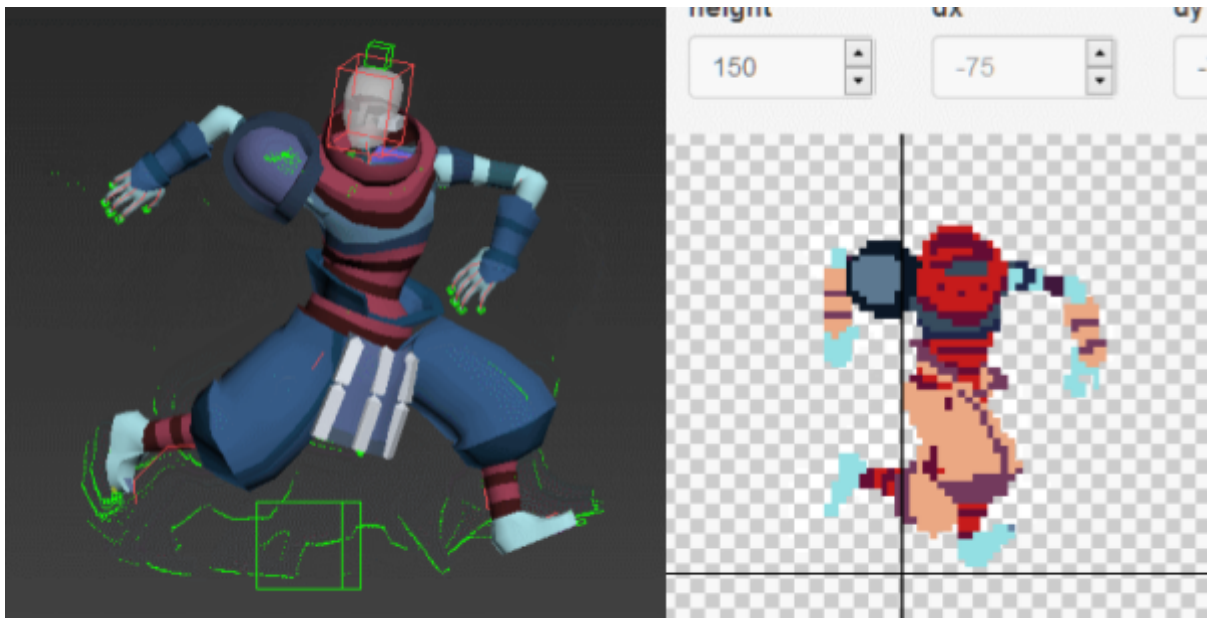


Fonte: Youtube

O jogo Dead Cells, produzido pela Motion Twin, também faz o uso de uma abordagem híbrida para gerar as sprites (Gregory, 2019, p. 722) do jogo, ao renderizar os personagens 3D em Pixel Art utilizando cel-shading, pois o animador era o único artista do

time e ele não tinha experiência suficiente com pixel art para produzir animações (VASSEUR, THOMAS 25; 2018, 2018).

Figura 6 - Uso híbrido do 3D para gerar sprites em pixel art 2D em Dead Cells



Fonte: Gamedeveloper.com

Figura 7 - Cel Shading, técnica para renderizar objetos 3D como cartoons



Fonte: Google Imagens

A abordagem do Dead Cells foi para produzir Pixel Art para jogos. Para produção de animações 2D tradicionais, existem trabalhos como “3A2A: A Character Animation Pipeline for 3D-Assisted 2D Animation”, onde se propõe uma interface 2D através de figuras de palito para controlá-las como objetos 3D, num pipeline para geração de animação 2D dadas as dificuldades existentes no 2D.

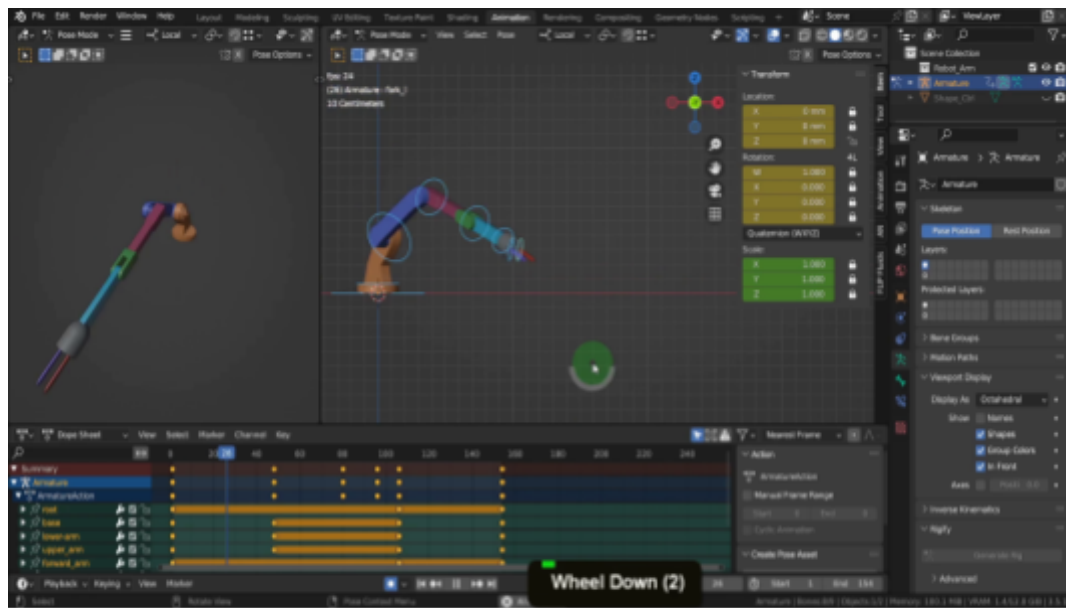
Vemos então que abordagens híbridas são frequentemente utilizadas na produção de animações, pois a mistura de técnicas permite extrair os pontos fortes de cada uma delas ao mesmo tempo em que diminuem ou eliminam seus pontos fracos. Anteriormente, vimos que uma das vantagens do 3D é a interpolação entre keyframes, diminuindo a necessidade de quadros a serem produzidos, mas que produzi-los pode ser lento. No 2D podemos criar esboços simples, de baixa fidelidade, para definir como será a animação final, mas sempre será necessário muitos quadros para uma animação. Desta forma, temos que alguns pontos fortes e fracos desses métodos são o que iremos chamar de complementares, ou seja, o ponto forte de um é o ponto fraco do outro.

Com isso, o objetivo deste trabalho é introduzir um novo método híbrido para produzir keyframes de modelos 3D, através de algum desenho 2D simples para que o desenho seja feito o mais rápido possível e então haja um mapeamento direto entre o desenho feito e o transform3D dos ossos de uma malha esquelética.

2. A Proposta do Método

O tipo de desenho escolhido foi o de figuras de palito. O motivo dessa escolha se dá pela natureza da ferramenta de animação 3D dominante, a animação esquelética. Existem diversas maneiras de se representar uma malha esquelética para manipulá-la. Softwares de modelagem como o Blender até possibilitam a troca da renderização padrão, por uma customizada, criada pelo usuário.

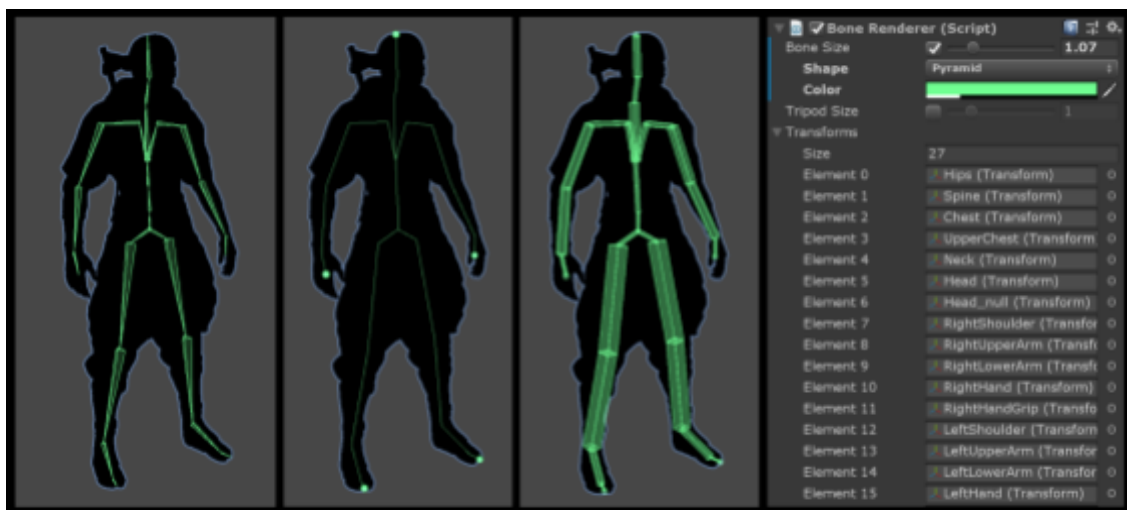
Figura 8 - Rigging de um braço robótico ajustado para ser representado por círculos



Fonte: Blender3D

Todavia, uma das maneiras de representar um osso da malha de esqueleto é simplesmente uma única reta, como visto em uma das renderizações da malha na figura 9 e essa representação, sendo bastante similar ao desenho de figuras de palito.

Figura 9 - Exemplos de Renderização de uma malha de esqueleto



Fonte: Unity Engine

Figura 10 - Desenhos de figuras de palito



Fonte: Google Imagens - Stick Figures

É pela semelhança entre a representação por retas da malha esquelética e as figuras de palito que este tipo de desenho foi o desenho escolhido. Contudo, percebe-se uma falha no desenho de uma única reta: É impossível extrair informações de rotação em torno do próprio eixo, pois falta informação. A reta é uma figura unidimensional e portanto não é possível inferir rotações em torno do próprio eixo da reta.

Desta forma, será utilizado o desenho de dois palitos para cada osso, onde o segundo terá como principal objetivo dar a informação de orientação em torno do próprio eixo para se extrair toda a transformação 3D. Por convenção, definimos que a segunda reta irá sempre intersectar a primeira em algum ponto M qualquer no espaço euclidiano e também que as duas retas serão sempre ortogonais no espaço euclidiano. Contudo, nota-se que essa ortogonalidade não será preservada nos desenhos feitos pelo usuário, pois o desenho deve ser feito considerando rotações, translações e distorções projetivas em perspectiva, os pontos de fuga e cabe à aplicação lidar com isso.

Com essas restrições, temos que os desenhos das retas ficarão em formatos de “T” ou “+”.

Figura 11 - Exemplos dos desenhos dos pares de retas para cada osso.



Figura 12 - Esboço da figura de palito de uma malha esquelética de um ser humano a ser desenhada



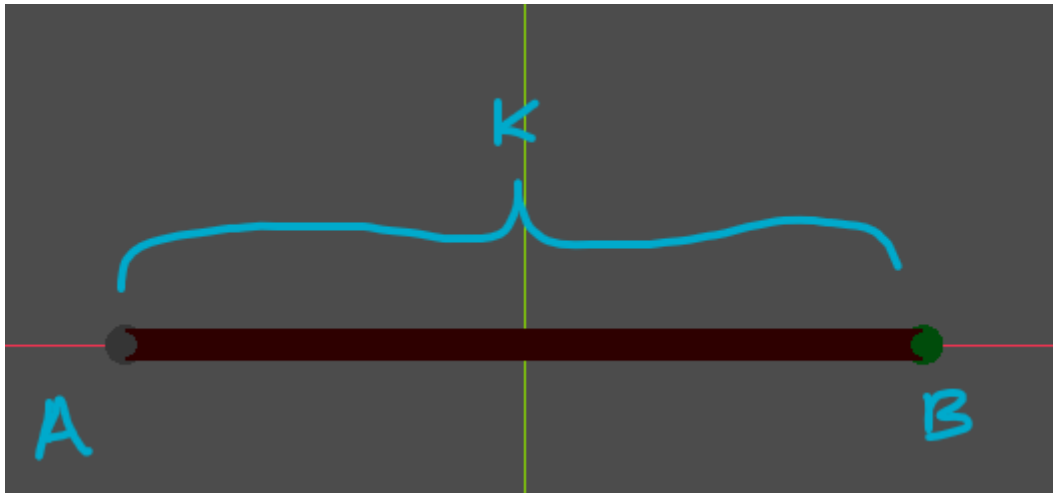
Desta maneira, o processo de animação consistirá em desenharmos duas retas para cada osso de uma malha esquelética. Então, a aplicação irá encontrar e aplicar o transform 3d encontrado pelo desenho ao osso equivalente e podemos repetir este processo para os ossos subsequentes, até terminar de posar o personagem. Ao finalizar o desenho de um osso, poderemos escolher o próximo a ser desenhado.

As retas serão desenhadas tal como em ferramentas como o Microsoft Paint, onde um clique do mouse define o primeiro ponto e ao soltar o botão do mouse, o segundo ponto da reta será definido. Desta forma, dada a facilidade em se desenharmos uma reta dessa maneira, espera-se que posar o personagem com essa abordagem híbrida seja mais eficiente e rápido que métodos tradicionais.

3. Desenvolvimento do Método

Um osso é representado por dois palitos, onde um deles representa a rotação em torno do próprio eixo. Com isso, precisamos desenhar duas retas. Cada reta terá um determinado tamanho k e suas duas extremidades num ponto A e B para a primeira reta e C e D para a segunda. A informação do tamanho k pode ser extraída do arquivo do modelo 3D que contém a malha de esqueleto ou até mesmo adicionado manualmente pelo usuário.

Figura 13 - Segmento AB da figura de palito que representa um osso.



A menos de uma transformação de escala, o tamanho de qualquer osso será constante. Sabendo que uma reta é definida por dois pontos e que a reta 2D desenhada representa um osso no espaço euclidiano 3D. Podemos então assumir que a solução para o problema será dado pela fórmula do tamanho de um vetor AB , dada por:

$$k = \sqrt{(Bx - Ax)^2 + (By - Ay)^2 + (Bz - Az)^2} \quad (1)$$

Onde k é o tamanho do osso e A e B são os pontos que representam o começo e o fim do osso, em coordenadas relativas ao objeto 3D.

Podemos eliminar a raiz da equação ao elevar ambos os lados ao quadrado.

$$k^2 = (Bx - Ax)^2 + (By - Ay)^2 + (Bz - Az)^2 \quad (2)$$

Contudo, encontrar os pontos A e B das retas é um problema difícil. Numa câmera em perspectiva, os objetos projetados tendem a ficar menores ou maiores, de acordo com a distância da câmera. No entanto, caso a transformação compense essa diminuição de tamanho

com um aumento de escala, a imagem final da reta do palito se manterá constante e há infinitas combinações de diminuição de tamanho pela distância e compensação pela escala.

Rotações também são capazes de alterar o tamanho da reta projetada tanto quanto movê-la para longe ou para mais perto da câmera o faria. Considerando o eixo Z como o eixo frontal da câmera, rotações em torno dos eixos X ou Y da câmera irão alterar o tamanho da reta. Existe uma infinita quantidade de combinações que diminuem ou aumentam o tamanho da reta através de uma rotação e uma compensação dessa diminuição ao mover o objeto para mais perto ou mais longe da câmera, podendo também compensar o tamanho com uma escala.

Deste modo, é necessário um método para encontrar uma solução única para encontrar A e B e então extrair o Transform 3D que reflita o desenho e intento do artista para ser aplicado ao modelo 3D.

3.1. Extraindo a Rotação e Translação

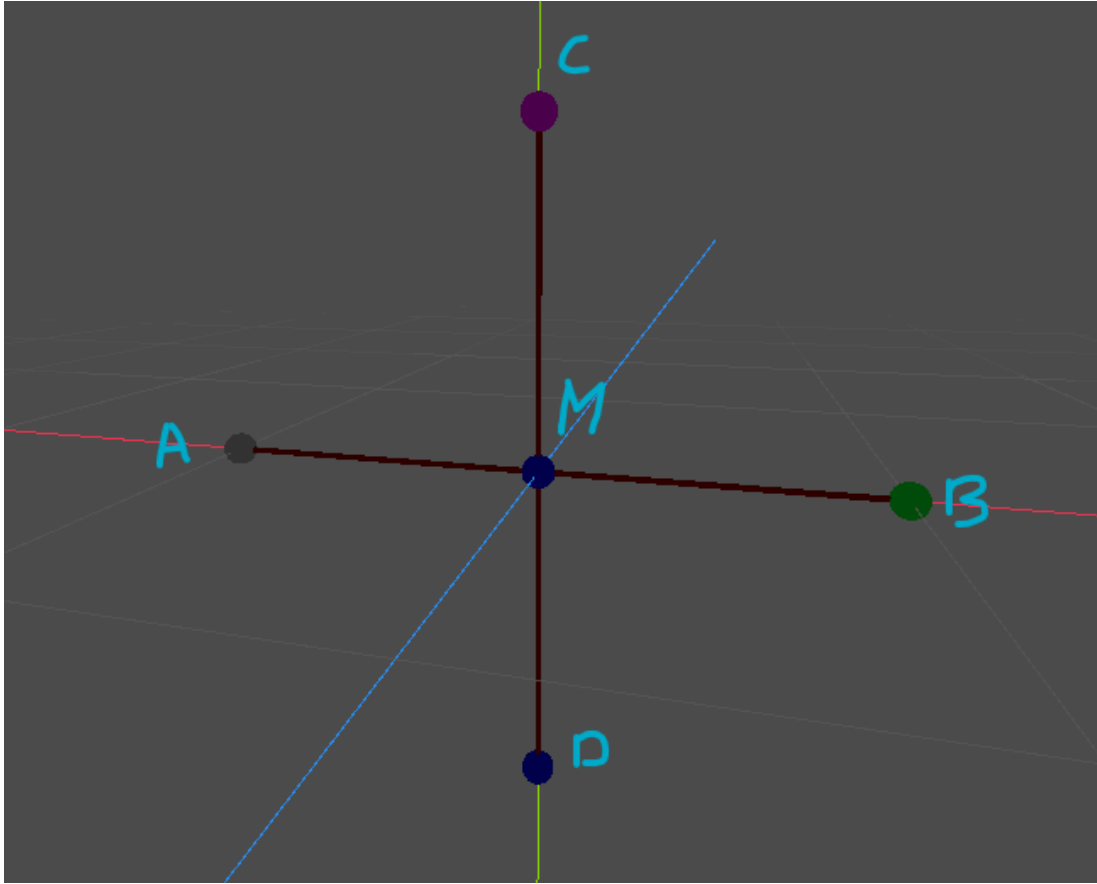
Conhecendo os pontos A e B da primeira reta e os pontos C e D da segunda reta e o ponto M compartilhado por ambas, podemos utilizá-los para extrair a rotação e a translação.

O primeiro passo é transformar todos os 5 pontos para que eles estejam em coordenadas globais, pois estes pontos encontrados estão em coordenadas de câmera.

Feito isso, encontrar a translação é bem simples. Sua translação será o ponto B do seu parente ou o seu ponto A, caso não tenha parente.

Para encontrar a rotação, utilizaremos os pontos A, B, C e o ponto M de interseção entre as retas AB e CD. Como definido anteriormente, os pares de retas que definem o osso do personagem são ortogonais no espaço euclidiano.

Figura 15 - Exemplo de representação dos palitos no espaço euclidiano.



Fonte: Godot Engine.

Com isso, conseguimos facilmente criar uma base utilizando os pontos A, C e M. Os vetores AM e CM já são ortogonais, pois definimos que as retas são ortogonais no início. Para conseguir o terceiro vetor ortogonal, que chamaremos de EM, basta fazer o produto vetorial entre AM e CM, que conseguiremos o terceiro vetor. Após isso, devemos normalizar todos os vetores, pois precisamos que eles tenham tamanho 1.

Podemos construir então um quatérnio a partir da matriz 3x3 que representa a base formada pelos vetores AM, CM e EM. Esse quatérnio será a rotação, porém em coordenadas da câmera e precisamos convertê-la com a multiplicação pela matriz de câmera-mundo para ter a rotação final do objeto.

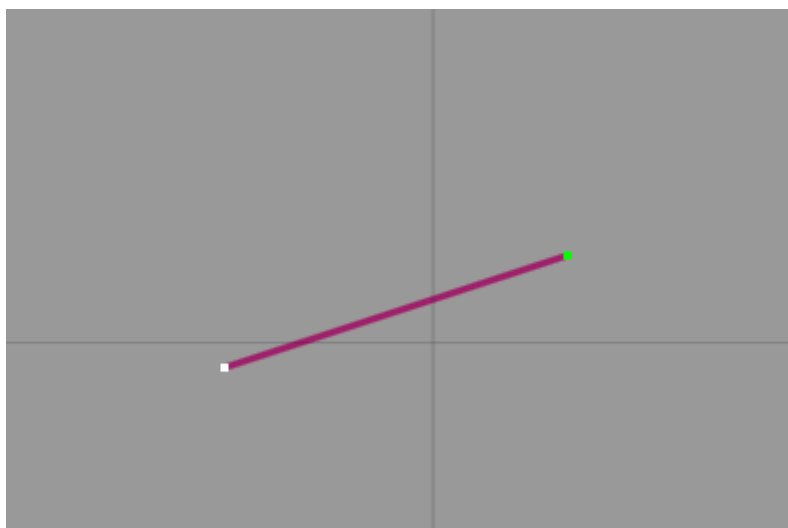
$$M = \begin{bmatrix} AMx & AMy & AMz \\ CMx & CM_y & CMz \\ EMx & EM_y & EMz \end{bmatrix}$$

4. Implementação

Para a implementação do método numa ferramenta, foi utilizado a engine open source Godot, pois possui uma API bastante adequada para programar desenhos num canvas.

O método escolhido para o desenho das retas foi um tradicional de desenho de retas em ferramentas como o Paint da Microsoft. O usuário irá clicar no canvas para escolher o primeiro ponto, depois irá arrastar o mouse pelo canvas, movimentando o segundo ponto dinamicamente e renderizando uma reta vermelha. Ao soltar o mouse, o segundo ponto será definido.

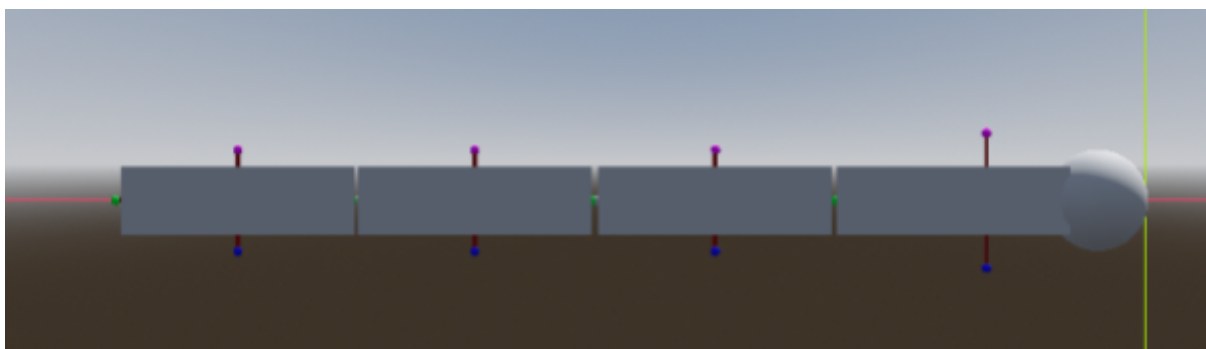
Figura 16 - Desenho de uma reta feita pelo usuário



O ponto A, B da primeira reta e C e D da segunda reta serão pintados de cores diferentes para facilitar a visualização de cada ponto.

Para fazer os testes, será utilizado um esboço de uma criatura como uma cobra, formada por cubóides em série. No osso que representa também sua cabeça, haverá um círculo para facilitar o entendimento.

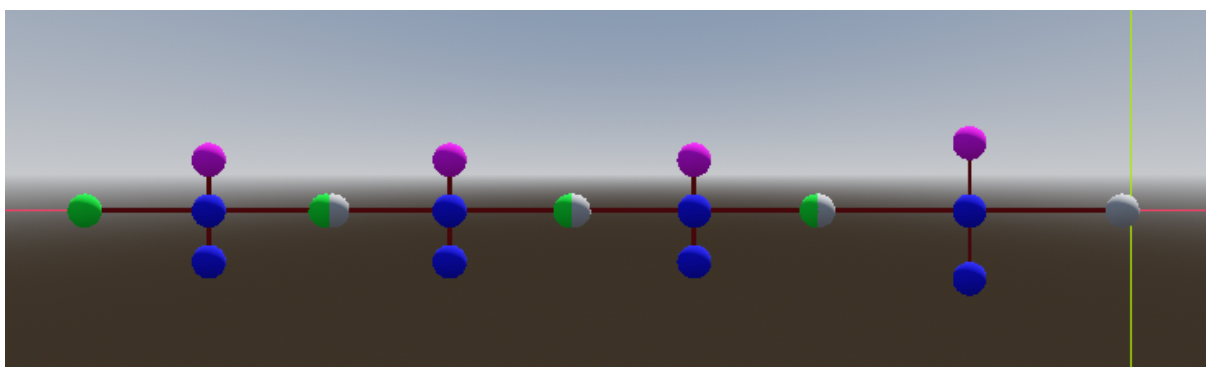
Figura 17 - O objeto imitando uma cobra e as retas AB e CD geradas pelo script, parcialmente ocluídas pela malha triangular dos cubóides



Fonte: Cena com o Objeto do Projeto no Godot Engine

Este objeto foi então configurado, lendo o tamanho do cubo para definir o valor do tamanho k da reta e ajustando o valor de α das retas AB e CD para cada segmento, com o valor de α sendo sempre 0.5 para todos os ossos, formando um par de retas com formato de “+”. Um script, então, utiliza os valores de k e α para gerar e posicionar os pontos A, B, C e D. Percebe-se que os pontos A, B, C e D, junto com uma reta para ligar cada ponto, está sendo desenhada, porém ocluída pela malha do cubóide.

Figura 18 - Os representantes dos ossos do objeto imitando uma cobra, sem a oclusão da malha triangular.



Contudo, é necessário de imediato lidar com o sistema de coordenadas do canvas. A origem do espaço do canvas 2D normalmente é o canto superior esquerdo da tela, enquanto a origem no espaço projetivo está em seu centro. Precisamos converter os pontos das retas, que estão em coordenadas de canvas, para um sistema de coordenadas em que a origem fica no centro antes de aplicar o método de solução da equação.

Além disso, para facilitar a visualização dos desenhos das retas, é necessário aumentar bastante seu tamanho. Então, precisamos remover a escala antes de enviar as retas para resolver o sistema de equação.

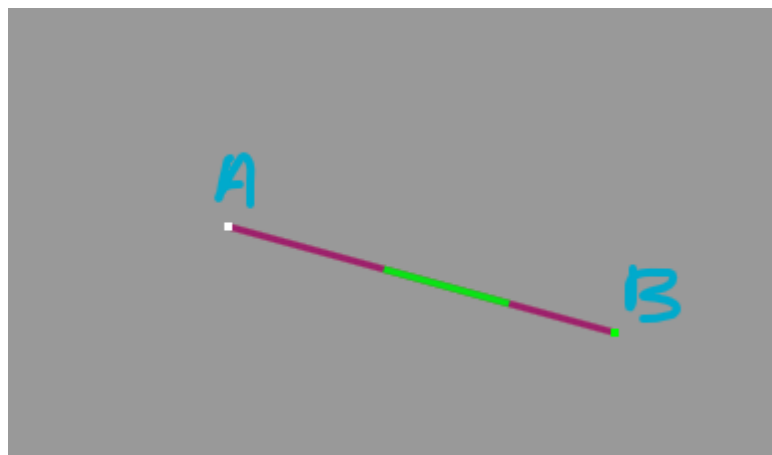
4.1. Corrigindo o Input do Usuário

Logo neste início, há um grave problema com o desenho das retas. É possível que o usuário desenhe uma reta inválida, grande demais. Empiricamente durante os testes, escolher uma reta AB válida por tentativa e erro não é tão difícil, mas para fazê-lo sempre corretamente seria necessário um desenho técnico, com medidas de retas e pontos de fuga. Todavia, desenhar corretamente o segundo segmento de reta sem um desenho técnico é impossível.

Para evitar que o usuário precise fazer um desenho técnico para conseguir posar o objeto, teremos que corrigir a reta desenhada por ele para que ela não tenha um tamanho maior que o necessário e iremos pintar a área válida para a interseção da segunda reta de outra cor, para destacá-la. O mesmo vale para o desenho da segunda reta, para corrigir seu tamanho e ponto final e inicial(C e D).

Primeiro, iremos resolver o problema de destacar a área válida. Isso é um processo bem simples. Subdividimos a reta em vários segmentos. Do ponto A ao ponto B' da subdivisão, verificamos se a equação produz uma solução válida, aquele ponto é válido e deve ser destacado. Empiricamente, percebe-se que a área válida para interseção da reta CD forma uma reta contínua.

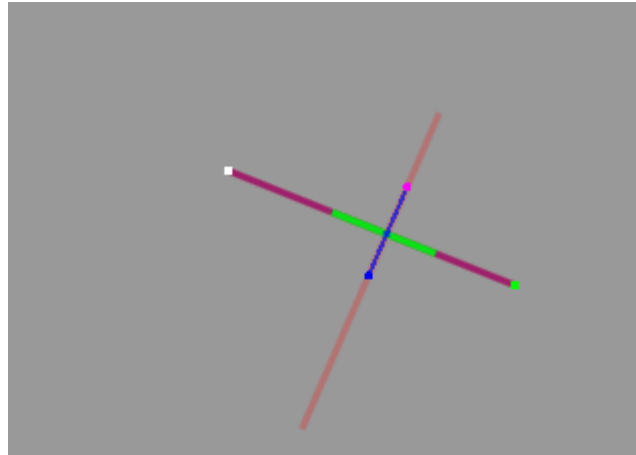
Figura 19 - Segmento válido está destacado de verde.



Definimos anteriormente que todos os valores de α serão 0.5. Desta forma, corrigir o input da reta CD torna-se muito mais fácil. Definimos também que a segunda reta é ortogonal à primeira no espaço euclidiano 3D. Como essa reta possui um tamanho fixo, preestabelecido, nós sabemos o tamanho do vetor MC e MD baseado no valor de α . Temos também que o

vetor AB é a normal de um plano que contém os pontos C e D e o ponto M também pertence a esse plano.

Figura 20 - Reta CD desenhada. Input do usuário em vermelho semi-transparente. A reta CD corrigida está pintada de azul.



Com a figura acima podemos entender visualmente porque desenhar corretamente a reta CD é quase impossível. Os pontos C e D podem estar em qualquer lugar deste segmento e escolher o ponto inicial corretamente sem fazer todo o processo com desenho técnico é quase impossível.

4.2. Posando o Objeto

Com as entradas corrigidas, desenhar as figuras de palito sem preocupações. Então, enquanto o usuário move a segunda reta com o mouse, ele verá o objeto sendo posado em tempo real até confirmar o input ao soltar o mouse.

A aplicação terá uma representação no lado esquerdo do canvas com o objeto a ser animado e ao lado direito uma lista com os ossos que podem ser escolhidos para serem desenhados. O primeiro osso, a raiz da árvore, será automaticamente escolhido, mas ao desenhá-lo, os próximos ossos devem ser escolhidos ao clicar em seus nomes nessa lista.

Figura 21 - A ferramenta por completo. À esquerda, um preview da cena. À direita, a lista de ossos do objeto que podem ser desenhados.

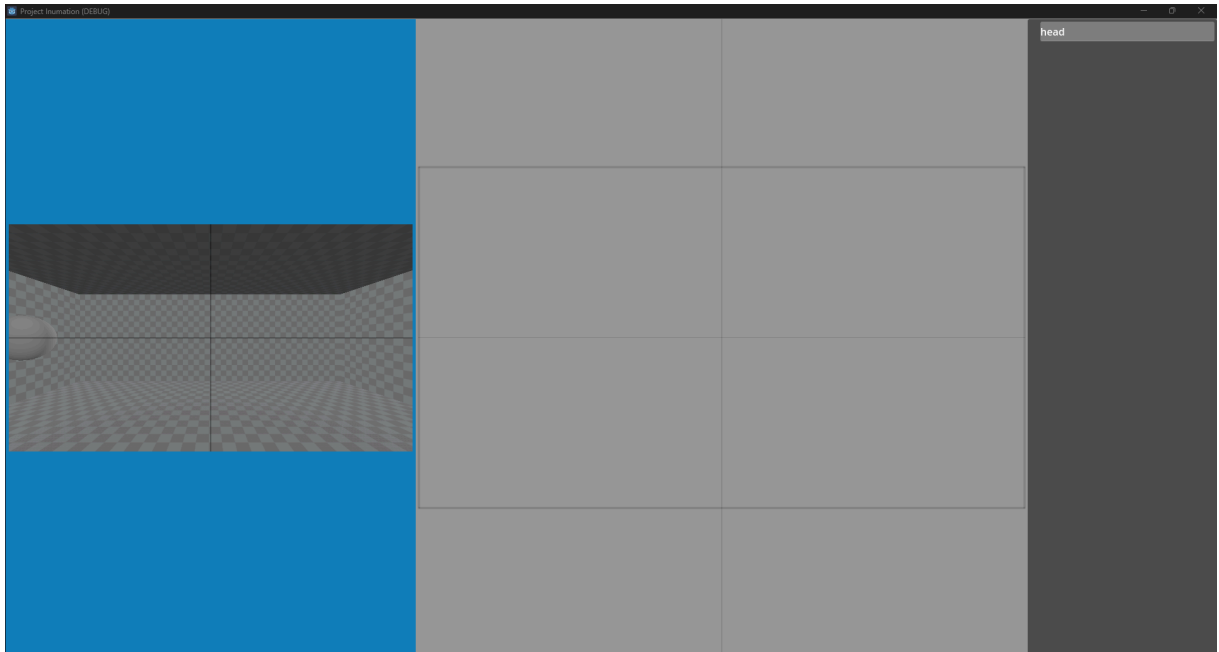
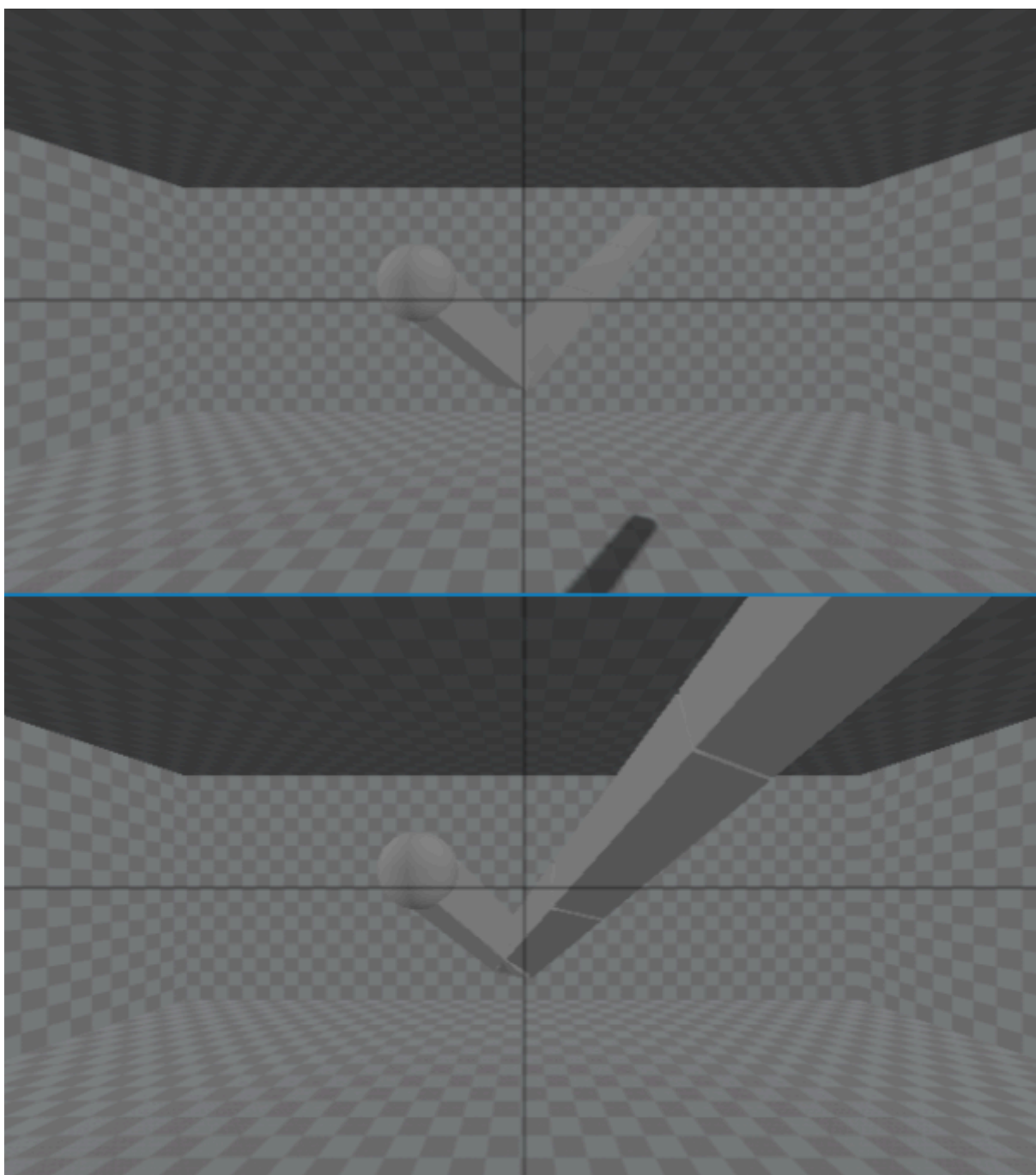


Figura 22 - Desenho da reta atualiza o transform do objeto. À direita, aparece o novo osso que pode ser selecionado para ser desenhado.



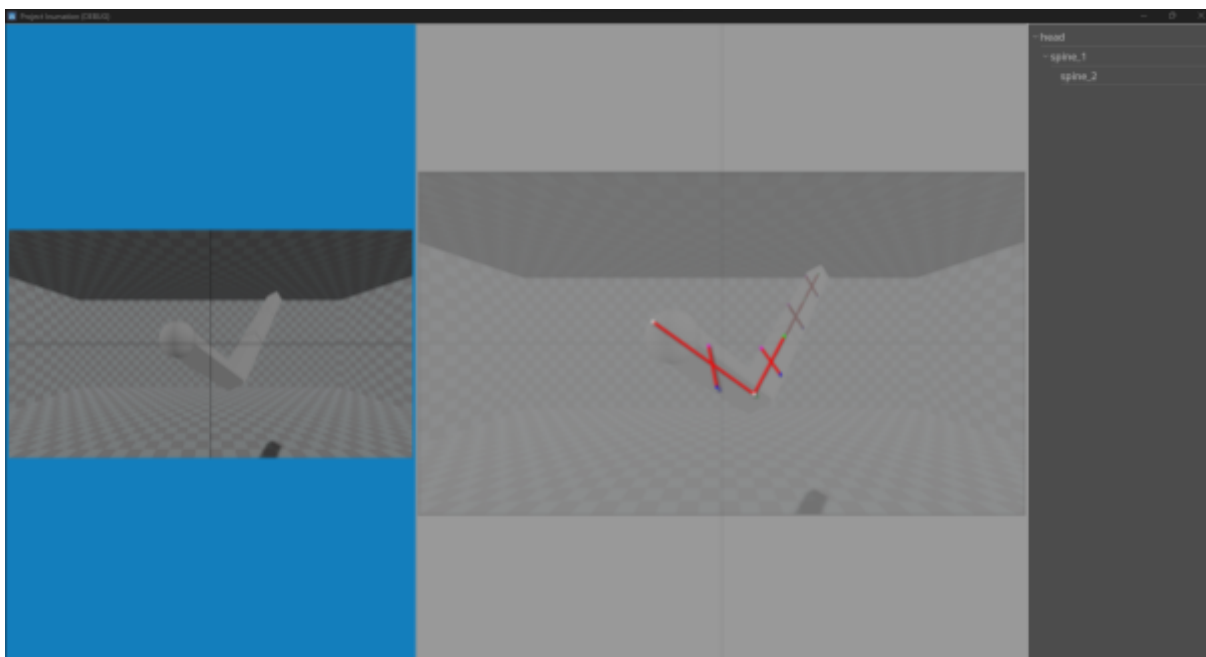
Quando vamos desenhar o segundo osso em diante, temos o caso em que já temos um ponto conhecido. A aplicação é interativa, então durante o desenho da reta CD, enquanto o usuário não soltar o mouse para definir o ponto D, o transform será calculado a cada quadro. Com isso, ao pressionar a tecla TAB, será feito uma troca entre a primeira solução encontrada e a segunda dinamicamente.

Figura 23 - Mudança no resultado do transform dado pelas duas soluções.



O preview do lado esquerdo foi adicionado para visualizar o resultado final e ele cumpre este papel. No entanto, o canvas e o preview não têm o mesmo tamanho, além de que o desenho das retas não dão qualquer indicação de noção de profundidade, causando em alguns momentos confusões com o resultado esperado e o obtido. Esse problema pode ser resolvido ao adicionar alguma visualização mais dinâmica dos pontos obtidos com a reta. Para corrigi-lo, foi feito com que os ossos gerados pelo script fossem desenhados abaixo do desenho das retas do usuário.

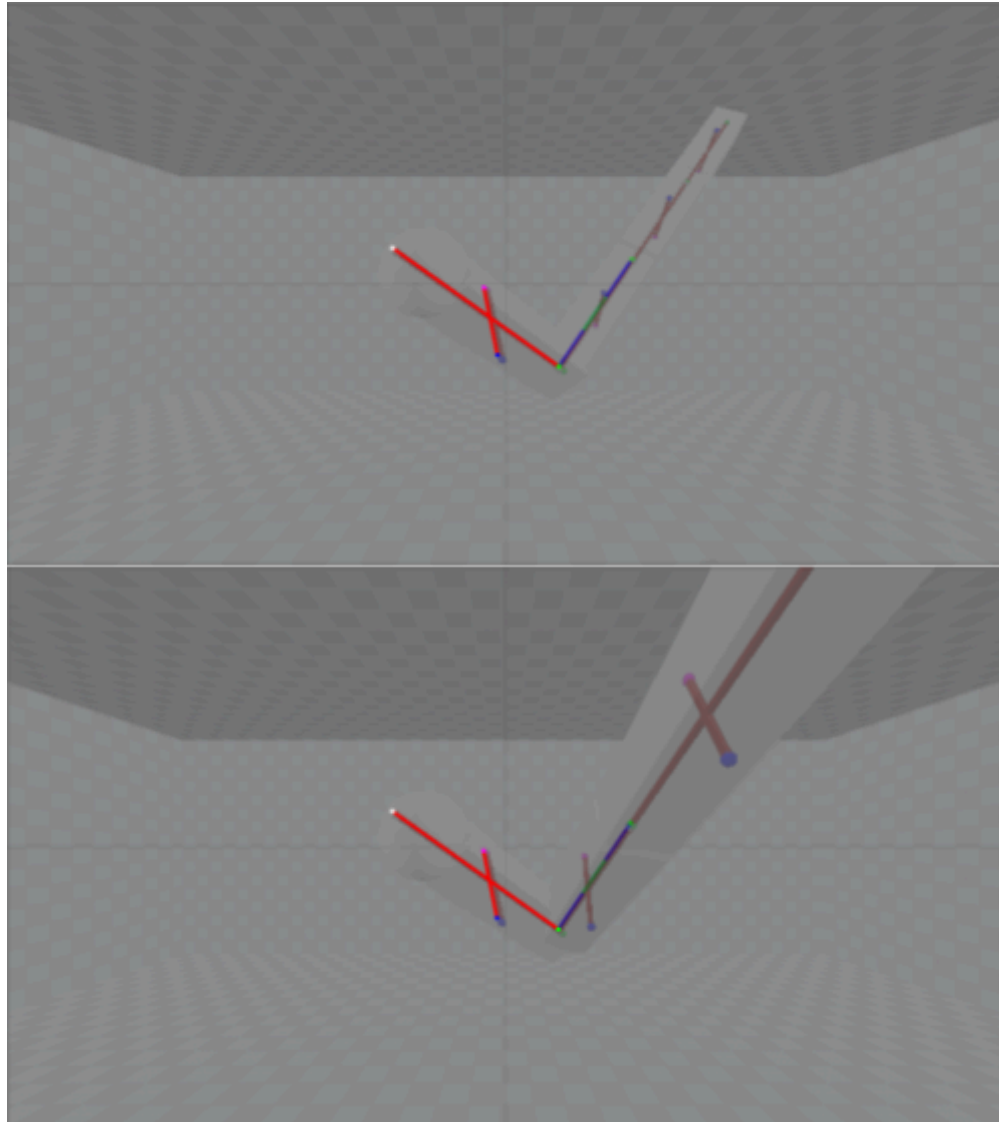
Figura 24 - Preview, junto com a representação dos ossos, renderizados atrás do desenho das retas feitas pelo usuário



Também foi adicionado um botão para trocar entre renderizar ou esconder a representação dos ossos gerados pelo script, mantendo apenas o preview atrás das retas. No entanto, essa funcionalidade foi esquecida durante os testes pois era difícil entender a relação entre o desenho do usuário e o transform gerado sem ela. Para tornar claro e resolver o problema por completo, ambos devem ser renderizados simultaneamente.

A renderização do preview e dos ossos também deixou mais claro o espelhamento quando se escolhe as duas possíveis soluções no caso em que há um ponto já conhecido, além de deixar claro que é possível escolher qual das duas soluções será utilizada com um método mais interativo.

Figura 25 - Visualização do espelhamento das soluções, mantendo a reta AB mas mudando a reta CD e o ponto M



No final, houve uma tentativa de ajustar os valores de α para os ossos. Embora a mudança desse valor possa ser proveitosa para um desenho final mais bonito, o valor 0.5 se provou mais consistente para os ajustes das retas. Isso pois o segmento válido de interseção ficou cada vez menor ao se ajustar o valor de α .

4.3. Testes com Usuários

Foram selecionados quatro usuários para realizar os testes. Todos eles possuíam experiência com o desenvolvimento de jogos 2D e 3D e já trabalharam na indústria e variados graus de experiência com desenhos 2D tradicionais.

Para cada um deles foi explicado como a ferramenta funciona e qual seu propósito e o que eles devem fazer para testar a aplicação, então foi dado a eles alguns minutos para brincar

e se familiarizar com a ferramenta. Foi necessário em torno de 5 a 10 minutos para cada um para se sentirem confortáveis com a ferramenta.

Foi pedido a eles para posar o objeto de testes de acordo com as imagens da figura 28 utilizando o método tradicional e depois posá-lo dentro da engine, através dos gizmos da engine, como mostra a figura 29, pois se assemelha ao método tradicional de animação.

Figura 26 - Os três desenhos que os usuários de referência para posar o objeto de testes

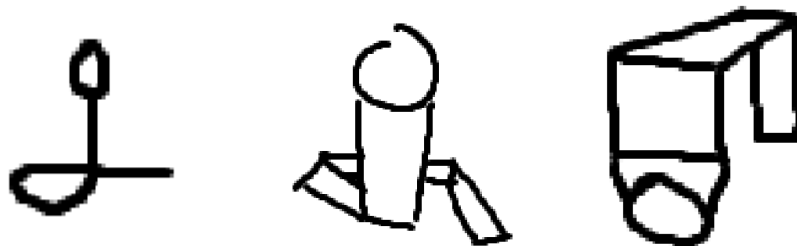
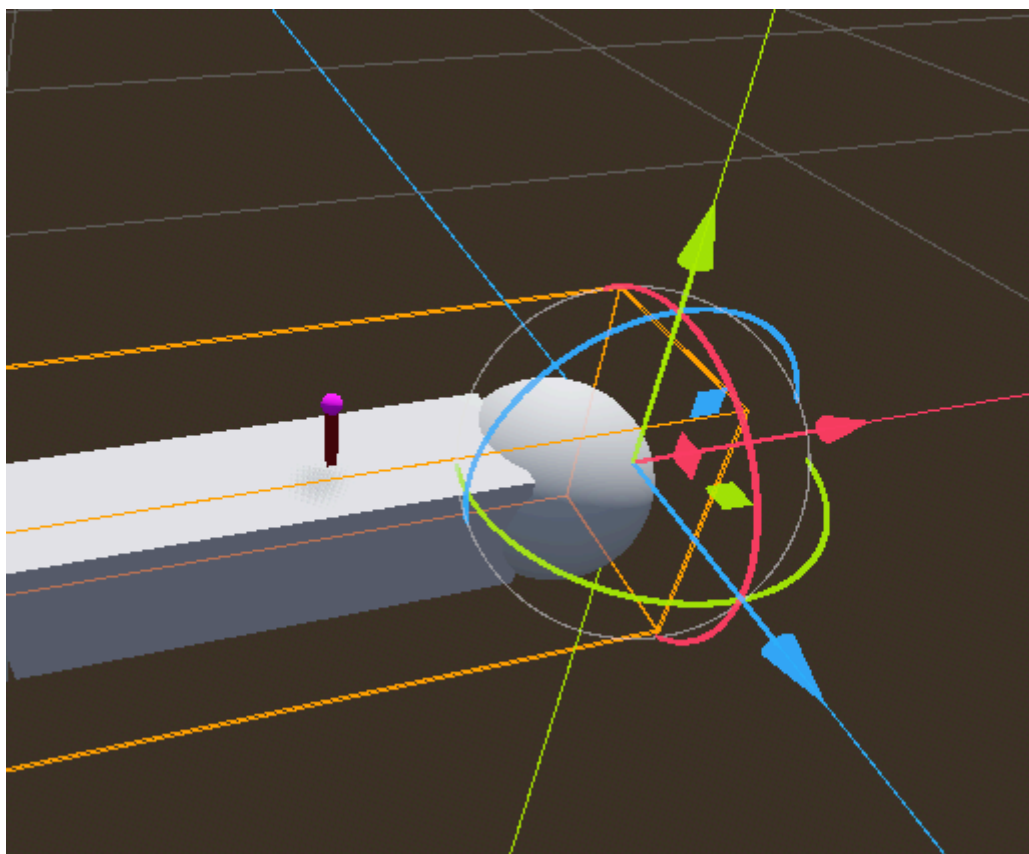
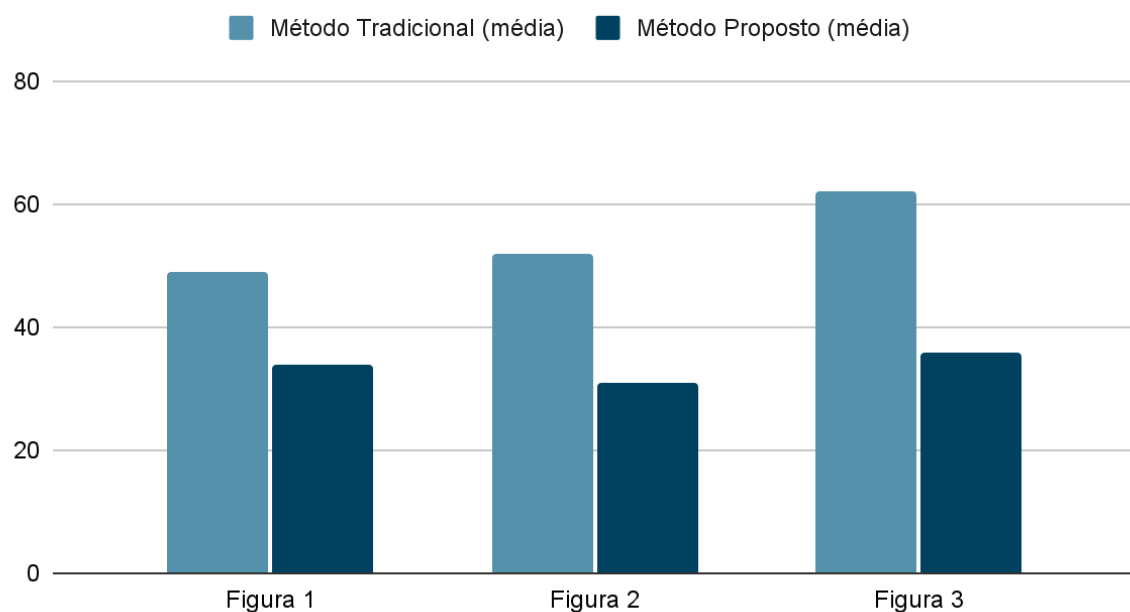


Figura 27 - Gizmos que permitem a manipulação do objeto da forma mais tradicional



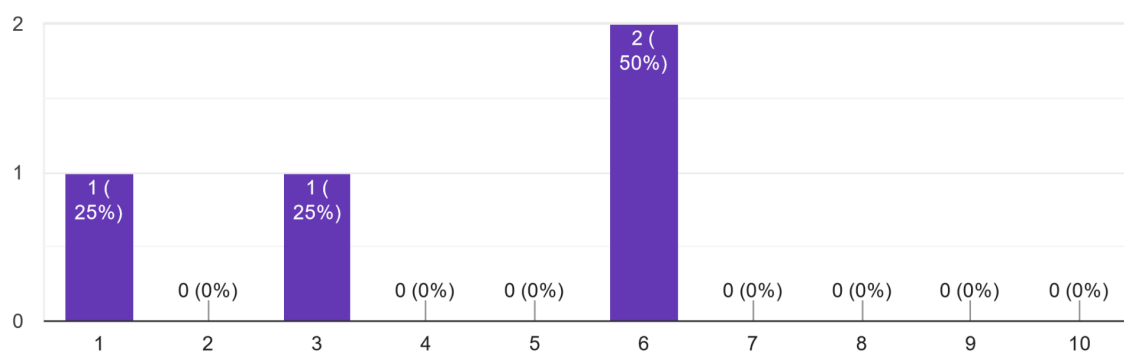
Durante os experimentos, foi cronometrado o tempo de execução para posar os objetos com ambos os métodos, para medir a eficiência em relação ao método tradicional. Após os testes terminarem, foi dado a eles um questionário com quatro perguntas, a fim de avaliar a ferramenta em si e se os problemas encontrados com o método de desenho pelas figuras de palito seriam sentidos pelos usuários.

Tempo Médio Para Gerar o Keyframe



A primeira pergunta pedia para avaliar de 1 a 10 se os usuários sentiram dificuldade para entender espacialmente o que os desenhos das retas estavam fazendo com o objeto 3D, sendo 1 para nenhuma dificuldade e 10 para muita dificuldade.

Você sentiu uma dificuldade para entender espacialmente o que estava desenhando com as retas?
4 respostas



A segunda pergunta era para o usuário explicar quais dificuldades ele enfrentou ou encontrou com a ferramenta durante seu uso e se ele teria alguma sugestão para saná-las.

Quais dificuldades você encontrou durante o uso do software e o que você acha que poderia fazer para melhorá-las?

4 respostas

O que mais me dificultou foi o baixo contraste entre o personagem e o fundo e também a falta de uma opção para mover a câmera para ter uma compreensão melhor da profundidade

Entender em qual passo da criação do esqueleto eu estava (podia ser mais claro)

Não conseguir dar zoom ou mudar o ângulo da câmera

A visualização espacial é uma dificuldade. Para resolver bastaria colocar mais câmeras com ângulos de visão diferentes.

Entender como funciona a profundidade para construção do primeiro segmento foi uma dificuldade.

Alguns bugs com o Ctrl+Z precisam ser resolvidos.

A terceira pergunta era uma também pergunta aberta, para saber se o usuário teria alguma outra sugestão melhorá-la, de maneira geral.

Você tem alguma outra sugestão para o software?

3 respostas

Indicativos de coordenadas esféricas (raio, ângulo theta, phi) e coordenadas retangulares (x,y,z)

Não, apenas mais liberdade para a câmera

Colocar mais ângulos de câmera para melhor visualização espacial.

Para otimizar ainda mais o desenho a rotação do seguimento poderia ser feita ao mesmo tempo que ele é direcionado usando o scroll do mouse.

A segunda pergunta era para o usuário explicar quais dificuldades ele enfrentou ou encontrou com a ferramenta durante seu uso e se ele teria alguma sugestão para saná-las.

Quais dificuldades você encontrou durante o uso do software e o que você acha que poderia fazer para melhorá-las?

4 respostas

O que mais me dificultou foi o baixo contraste entre o personagem e o fundo e também a falta de uma opção para mover a câmera para ter uma compreensão melhor da profundidade

Entender em qual passo da criação do esqueleto eu estava (podia ser mais claro)

Não conseguir dar zoom ou mudar o ângulo da câmera

A visualização espacial é uma dificuldade. Para resolver bastaria colocar mais câmeras com ângulos de visão diferentes.

Entender como funciona a profundidade para construção do primeiro segmento foi uma dificuldade.

Alguns bugs com o Ctrl+Z precisam ser resolvidos.

A terceira pergunta era uma também pergunta aberta, para saber se o usuário teria alguma outra sugestão melhorá-la, de maneira geral.

Você tem alguma outra sugestão para o software?

3 respostas

Indicativos de coordenadas esféricas (raio, ângulo theta, phi) e coordenadas retangulares (x,y,z)

Não, apenas mais liberdade para a câmera

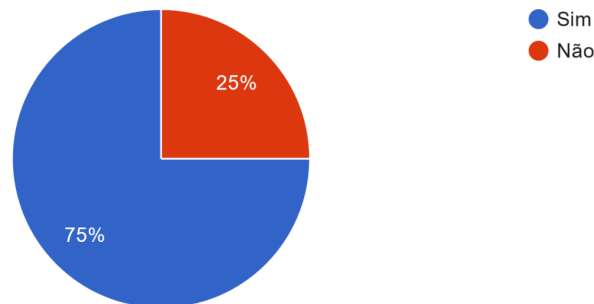
Colocar mais ângulos de câmera para melhor visualização espacial.

Para otimizar ainda mais o desenho a rotação do seguimento poderia ser feita ao mesmo tempo que ele é direcionado usando o scroll do mouse.

A quarta e última pergunta quis avaliar se os usuários achariam mais fácil visualizar espacialmente os desenhos caso os desenhos fossem feitos com retângulos ao invés de retas. Isso porque, no ensino tradicional de perspectiva e os pontos de fuga, retângulos e cubos são os objetos utilizados para o ensino.

Você acharia mais intuitivo visualizar os desenhos caso eles sejam feitos com retângulos ao invés de retas?

4 respostas



Como visto pelas queixas, apesar do tempo de desenho ter sido drasticamente reduzido, mesmo com pouco tempo de treino com a ferramenta e esta técnica inédita para os usuários, houve um bom grau de confusão com a noção espacial dos desenhos. Acredita-se que as funcionalidades de atualização em tempo real e correção de input ajudaram bastante em relação a isso, pois apesar dessas dificuldades, essa natureza dinâmica e que impedia o usuário de errar davam uma facilidade e agilidade, permitindo ao usuário tentativa e erro para se familiarizar rapidamente com a novidade.

Reclamações em relação a UX também ocorreram, como o contraste entre o objeto e o plano de fundo, que confundia o que era o que por terem exatamente a mesma cor e pouca iluminação para destacar um do outro.

5. Conclusões

O objetivo deste trabalho foi introduzir uma nova maneira de posar personagens 3D para criação de animações, numa abordagem híbrida de desenhos 2D para animar personagens 3D, através do desenho de figuras de palito, com o intuito de agilizar o processo de geração de keyframes para a animação desses objetos 3D.

O trabalho atinge este objetivo com relativo sucesso, como pode ser visto pelo resultado da pose dos personagens, onde a média de tempo se reduziu drasticamente apesar da pouca experiência com a técnica que era inédita para os usuários. Todavia, embora seja simples e rápido posar os objetos desta maneira, a escolha de retas para posar personagens 3D não foi tão adequada. Isso pois as retas não comunicam bem uma noção de perspectiva e profundidade, o que tornou o processo bastante confuso e talvez mais complexo do que poderia ser. As funcionalidades de correção de input e aplicação em tempo real ajudaram os

usuários a posar o objeto, mas a ideia de que poderia ser melhor esse processo de posar foi unânime.

Contudo, uma vantagem deste método é que apesar dele ter sido pensado considerando uma malha de esquelética, é possível animar qualquer objeto sem a utilização dela. Basta definir o tamanho k e o valor de α para os dois segmentos de reta, como foi feito na ferramenta. Isso seria útil para qualquer tipo de jogo, visto que ainda hoje, títulos como Minecraft são animados sem a utilização de uma malha de esqueleto. Um método como este poderia facilitar o desenvolvimento desses jogos e talvez ainda por cima eliminar a necessidade de rigging (o processo de criação de uma malha esquelética) de personagens, que se torna cada vez mais complexo quanto mais complexo é o personagem.

Um outro ponto positivo deste método é que ele abre a possibilidade para utilizar vídeos como referências de animação no processo de animação 3D e roscopia talvez possa ser aplicada neste contexto, a partir deste trabalho, servindo como uma alternativa barata e simples para o Motion Capture, para desenvolvedores independentes que não possuem recursos para arcar com equipamentos tão caros. No entanto, essa possibilidade ainda precisa ser estudada e validada.

6. Trabalhos Futuros

Este trabalho teve objetivos específicos. Todavia, ele foi um passo inicial para a produção de um conjunto de técnicas e ferramentas que tornam possível mesclar todo o processo de animação 2D com o 3D, numa abordagem híbrida para todo o processo de animação, com fim de aumentar a produtividade e capacidade de expressão artística de indivíduos e empresas.

Funcionalidades como um vídeo de referência, para dar suporte a roscopia, pode ser uma boa alternativa para desenvolvedores ou animadores independentes, com poucos recursos e que não podem pagar pelo Motion Capture.

Além disso, existem ferramentas no mercado que são capazes de rotacionar desenhos puramente 2D como se fossem 3D ao inferir informações de profundidade, excluindo a necessidade de desenhar o personagem quadro por quadro. Todavia, elas funcionam apenas com um estilo chamado pixel art. Essa técnica, funcionando para todo tipo de desenho e não só pixel art, em conjunto com o processo de animação proposto por este trabalho seria capaz

de praticamente unificar todo o processo de produção de arte 2D com 3D, juntando os pontos fortes de ambos e minimizando seus pontos fracos.

7. Bibliografia

- [1] HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. New York: Cambridge University Press, 2017.
- [2] GREGORY, J. **Game engine architecture**. Boca Raton, Fl: Crc Press, Taylor & Francis Group, 2019.
- [3] WILLIAMS, R. **The animator's survival kit**. London: Faber And Faber, 2001.
- [4] DADFAR, O.; POLLARD, N. S. 3A2A: A Character Animation Pipeline for 3D-Assisted 2D-Animation. **Lecture Notes in Computer Science**, p. 557–568, 1 jan. 2021.
- [5] MLYNARČÍKOVÁ, E. **Integration of 2D Animation into a 3D Pipeline**. Disponível em: <<https://theses.cz/id/yvafd7/?lang=en>>. Acesso em: 11 fev. 2024.
- [6] VASSEUR, THOMAS **Art Design Deep Dive: Using a 3D pipeline for 2D animation in Dead**. 25 DE JANEIRO de 2018. Disponível em: <<https://www.gamedeveloper.com/production/art-design-deep-dive-using-a-3d-pipeline-for-2d-animation-in-i-dead-cells-i->>. Acesso em 10 fev. 2024