

Visualização em tempo real da estrutura esquelética de uma mão humana usando Realidade Aumentada

Ivan Ademir dos Santos¹, Marcelo de Gomensoro Malheiros²

Resumo: Com o avanço e barateamento das câmeras de profundidade, novas aplicações são possíveis nas mais diversas áreas de conhecimento. Este artigo apresenta o desenvolvimento de um sistema de Realidade Aumentada para visualização em tempo real da estrutura esquelética de uma mão humana, utilizando para isso um sensor compacto de visão em profundidade: a câmera Leap Motion Controller. O objetivo é avaliar tanto a precisão como o tempo de resposta para a exibição da mão do usuário apenas com os ossos constituintes, sendo capaz de reproduzir tanto poses individual dos dedos como movimentos e posicionamento de toda a mão. Para garantir a eficiência da exibição do modelo em 3D, foi utilizado o motor de jogos Unity. O trabalho descreve como foi feito o mapeamento de cada uma das juntas sobre um modelo hierárquico de armadura de animação, que por sua vez altera a geometria de uma malha 3D de ossos humanos. Ao final, os resultados e as limitações técnicas desta abordagem são descritos, assim como possíveis trabalhos futuros.

Palavras-chave: Realidade Aumentada. Câmera de Profundidade. Modelos 3D. Animação. Armadura. Estrutura Esquelética.

1 Graduado em Engenharia da Computação pela Universidade do Vale do Taquari – UNIVATES.
iasivansantos@gmail.com

2 Professor na Universidade do Vale do Taquari – UNIVATES. Doutor em Computação na UFRGS – RS, mestre em Engenharia Elétrica na UNICAMP – SP, graduado em Engenharia da Computação na UNICAMP – SP.
mgm@univates.br

1 INTRODUÇÃO

A Realidade Virtual e Aumentada surgiu inicialmente para simular sentidos reais, e logo evoluiu para o campo de entretenimento, onde teve sua maior evolução e usabilidade. Conforme Martins e Guimarães (2012) seus usos são variados e já foram desenvolvidos em diversas áreas, como medicina, engenharia, química e física.

Atualmente é possível simular cenários e situações, igualmente ao que estamos acostumados a ver no mundo real, como também situações abstratas e imaginárias, que até então não eram possíveis. Existe também a possibilidade de interagir com mais indivíduos no mesmo espaço. Negroponte (1995) destaca que a ideia da realidade virtual é tornar o artificial tão realista que proporcione a sensação de estar dentro de uma situação, demonstrando como seria se realmente se estivesse lá. É o artificial se tornando real.

Essa dinamicidade tornou a Realidade Virtual e Aumentada extremamente difundida, e hoje está presente em diversos produtos existentes no mercado, tendo fácil acesso ao público em geral. Silva (2016) aponta características transformadoras no uso dessas tecnologias, permitindo diversos avanços, onde desse modo, áreas de extrema importância como saúde, educação e ciência, estão se desenvolvendo de forma mais rápida, trazendo inúmeros benefícios para todos.

No cenário de crescente evolução e expansão da Realidade Virtual e Aumentada, surgem uma infinidade de oportunidades e meios de exploração que até então não existiam. Dessa maneira é possível interagir e estudar mais a fundo uma série de situações e áreas, fazendo com que o entendimento e absorção do conhecimento, seja de forma mais rápida e eficiente. Há diversas áreas em que os métodos de estudo e ensino são aplicados em cenários controlados, e em muitos casos estão longe da realidade, fazendo com que exista uma diferença considerável entre a teoria e a aplicação prática do assunto.

Visando explorar essa versatilidade e flexibilidade este artigo apresenta o desenvolvimento de um sistema em tempo real, que permitem a captura e *feedback* em tempo real, com o objetivo de mostrar informações visuais que não seriam possíveis de outra forma. Em particular, deseja-se visualizar em tempo real o esqueleto da mão do usuário, o que se dá através da estruturação de um ambiente utilizando uma plataforma de desenvolvimento de jogos, tendo ligação conjunta com o dispositivo de captura de imagens em profundidade. No caso, utilizou-se o Leap Motion Controller, fabricado e usado especificamente para captura de movimentos e estrutura de mãos humanas.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados os principais conceitos pertinentes a este trabalho.

2.1 Realidade Virtual e Realidade Aumentada

A Realidade Virtual e a Realidade Aumentada, de acordo com Kirner e Siscoutto (2007), podem ser definidas como interfaces estendidas e avançadas para acessar e utilizar aplicações. A base para ambas está relacionada com a experiência de visualização e movimentação em tempo real com ambientes tridimensionais, e na interação com o mesmo. Um dos pontos que torna esses dois campos extremamente interessantes é a possibilidade de utilizar sentidos, como a audição e o tato para tornar a experiência ainda mais intensa.

As duas realidades possuem diversas características em comum, e que na sua maioria são conceitos iguais, ou semelhantes. O ponto que mais difere entre as duas e que torna a experiência de utilização diferente é relacionado ao ambiente onde as coisas acontecem. Conforme Kirner (2011) na Realidade Virtual, o usuário é levado para um ambiente computacional onde um cenário tridimensional é usado, fazendo a simulação do ambiente físico, e assim, possibilitando a interação. Já para a Realidade Aumentada esse conceito é totalmente inverso visto que o usuário permanece no seu ambiente real, e o ambiente virtual é levado para o físico, permitindo o seu uso.

Kirner e Siscoutto (2007) afirmam que existem algumas classificações que são feitas para categorizar os diferentes tipos de sistemas que compõem a Realidade Virtual e a Aumentada. Basicamente o que é levado em conta para chegar aos resultados é a forma como é realizada a interação do usuário. Na Realidade Virtual a classificação é feita de duas formas, imersiva ou não imersiva. A RV é imersiva quando o usuário é levado por completo para o ambiente virtual, e consegue ter contato com os elementos projetados. Ao contrário, a classificação é dita não imersiva quando o usuário é transportado de forma parcial para o ambiente virtual, assim ele consegue interagir com o ambiente tridimensional, mas continua a sentir-se no ambiente do mundo real, ou seja, com um envolvimento em escala menor.

Tori e Kirner (2006) afirmam que a Realidade Aumentada também possui duas formas de classificação, que pode ser visão direta ou imersiva, e visão indireta ou não imersiva. Quando o usuário enxerga seu ambiente físico misturado com o virtual, e entende que o virtual se torna também o real, é chamado de visão direta. Quando seu ambiente está misturado, porém é possível diferenciar a Realidade Virtual com a real, passa a ser de visão indireta. Nesses dois tipos também faz-se prevalecer o envolvimento do usuário, um modelo

utiliza esse conceito de forma completa e o outro em um âmbito menor, o que resulta em formas de uso diferentes, tendo duas maneiras distintas de usar essa tecnologia.

2.2 Câmeras de profundidade e infravermelho

Conforme Mitra e Acharya (2007), as câmeras de profundidade e infravermelho são bastante utilizadas em sistemas de RA e RV pois têm a capacidade de capturar e rastrear apenas o necessário para a aplicação, tornando assim o processamento dos dados mais simples e direto, facilitando o trabalho realizado pelos algoritmos e rotinas que processam os dados de saída. Para proporcionar esses recursos, os dispositivos existentes possuem sensores com capacidade de captação em três dimensões.

A diferença entre câmeras convencionais para as câmeras de profundidade e infravermelho está basicamente na capacidade de determinar a distância de qualquer objeto que está envolvido na cena capturada. Nas demais câmeras, como explica Figueiredo (2012), a captura possui apenas duas dimensões, sendo assim um dado limitado. Já para as câmeras com profundidade e infravermelho a captura é em três dimensões podendo assim fazer o cálculo da distância, e proporcionar dados de entrada mais precisos. Esse mapeamento da distância é possível pois uma combinação de sensores de emissão e recepção de iluminação infravermelho é executada em relação aos objetos envolvidos.

2.3 Animação de estruturas articuladas

A animação está presente em praticamente todas as aplicações que envolvem computação gráfica. Azevedo e Conci (2003) comentam que para a animação de estruturas articuladas o processo é feito em duas etapas, a primeira que trata a animação das articulações e movimentos, e a segunda que trabalha animações pontuais, que dão realismo ao trabalho. Para fazer representações de estruturas articuladas é indispensável que exista relação entre cada parte que compõe o modelo, assim evita-se quebras e falhas no processo de transformação. Os ossos, chamados de *bones*, são elementos individuais de estrutura, e fazem a ligação entre duas articulações. As articulações, chamadas de *joints* são os pivôs de rotação que fazem as conexões com os ossos, e permitem a rotação em até três eixos. Por fim temos o esqueleto, ou armadura, que é a retratação do conjunto formado por todos os ossos e articulações que compõem a estrutura de deformação.

O controle de movimentos tridimensionais que envolvem a animação de uma estrutura articulada é feito por três camadas segundo Azevedo e Conci (2003). A primeira camada é a

armadura de modo geral, que recebe os dados de entrada, e deve executar os movimentos de rotação e translação necessários. A segunda camada define a distorção do modelo geométrico, que trabalha diretamente com as deformações da malha. E a última camada é responsável pela aparência final da malha, ou seja, é a transformação visível para o usuário.

3 DESENVOLVIMENTO

Para atingir o objetivo principal de permitir a visualização de um esqueleto de mão humana, foi feita a captura da posição e pose da mão do usuário através do dispositivo Leap Motion Controller, que é então transferida e renderizada no motor de jogos Unity.

Foi utilizado um *setup* que envolve basicamente o dispositivo de captura, que está conectado a um computador pessoal, e que tem comunicação direta com a plataforma Unity. Dessa maneira é possível fazer a captura e rastreamento da mão do usuário e enviar para processamento em tempo real. Os resultados são mostrados diretamente na tela do computador, e têm atualizações em tempo real dos movimentos feitos pelo usuário.

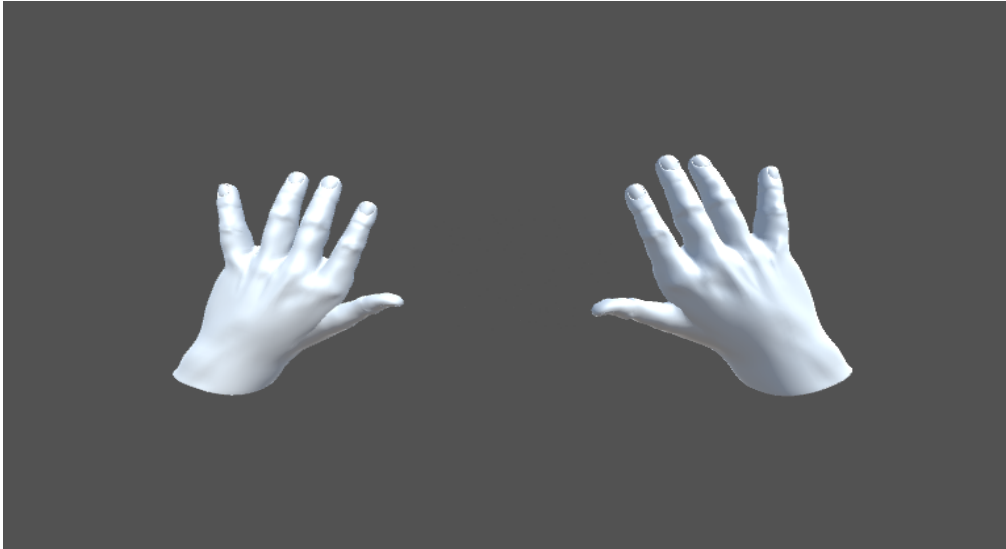
A modelagem do esqueleto humano é baseada em um modelo de mão humana disponível publicamente. Com base nele, foram feitas as devidas alterações e ajustes para contemplar a malha esquelética, atentando para todo o mapeamento de articulações e hierarquias necessárias, para que no momento da simulação, todos os movimentos ocorram de forma correta, simulando movimentos reais que o usuário fará em frente à câmera.

3.1 Modelagem

A tarefa inicial foi a de personalizar o modelo tridimensional de referência existente no SDK do Leap Motion Controller, a fim de transformá-lo em um modelo de esqueleto, contendo toda a estrutura existente em uma mão humana. O modelo base tinha como representação mãos humanas, com a malha de pele, conforme mostrada na Figura 1. Este modelo possuía toda a estrutura lógica e geométrica necessária para que todos os membros da mão fossem articulados, e tivessem a movimentação igualmente ao de uma mão real. Essa estrutura facilitou no entendimento e serviu de base para trabalhar com o modelo personalizado de esqueleto.

O modelo para o esqueleto de uma mão utilizado foi baseado em uma versão simplificada de um modelo de alta definição do projeto científico BodyParts3D. O modelo original foi obtido através da digitalização de ossos reais em um *scanner* 3D.

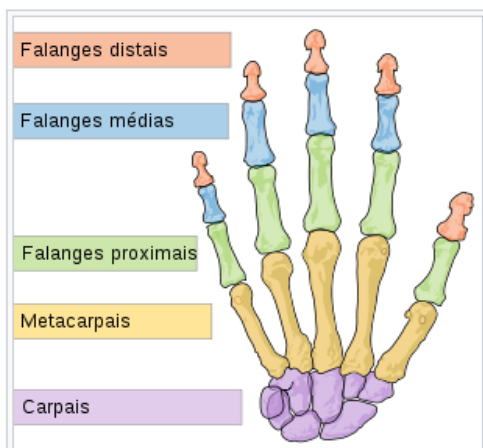
Figura 1 - Modelo base de mãos humanas



Fonte: Elaborado pelos autores (2019).

Foi utilizado o software de modelagem e animação Blender para ajustar o modelo esquelético, originalmente estático, em um modelo articulado. Para isso construiu-se a mesma hierarquia do modelo base, para que durante a execução dentro do Unity todas as partes tenham o mesmo comportamento em relação às entradas de dados.

Figura 2 - Ossos que compõem uma mão humana.



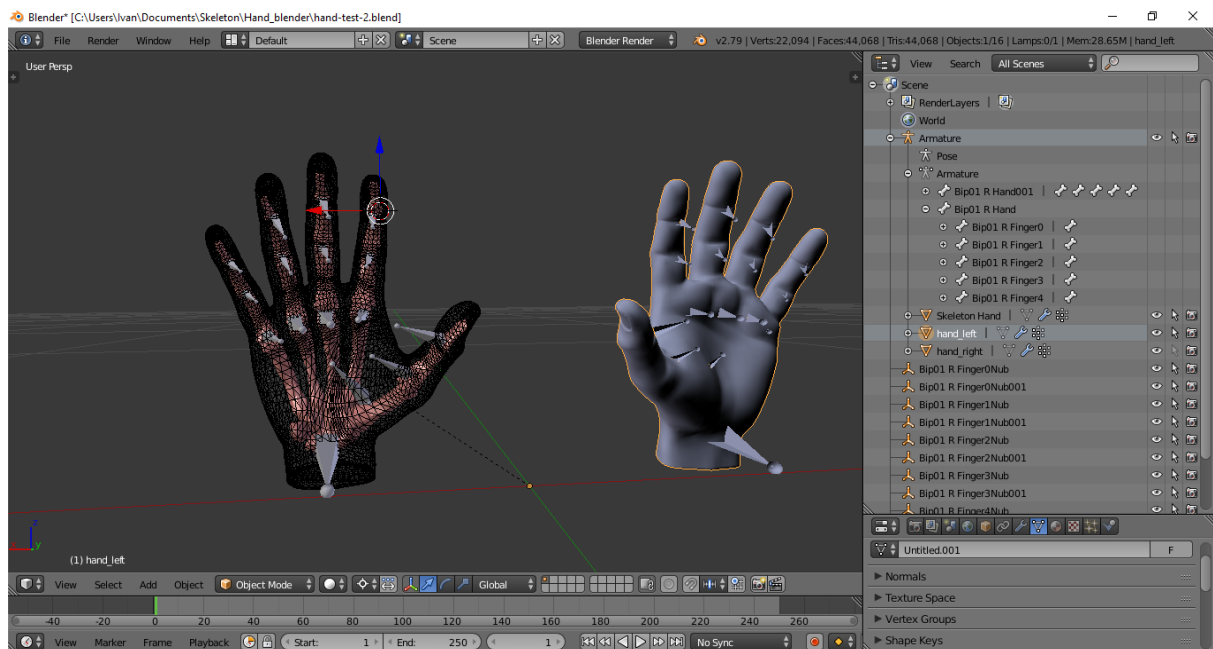
Fonte: Wikimedia Commons, 2007. Imagem no domínio público.

Nesse ponto houve grande preocupação em tratar todas as partes constantes nos dedos. Desde os dedos polegar, indicador, médio, anelar e mínimo, até as demais partes do esqueleto. Conforme a anatomia de uma mão, os ossos dispostos se dividem em três partes. A primeira é

o Carpo, a segunda o Metacarpo e a terceira as Falanges. O Carpo é formado por oito ossos, e compõe a base da mão. O Metacarpo é composto por cinco ossos que são na sua maioria iguais em tamanho, e formam a palma da mão, e por última a Falange, que são os dedos propriamente ditos. Os dedos, com exceção do polegar, são divididos em três partes, e são denominadas de Falange proximal, Falange média e Falange distal, como já falado, excepcionalmente o polegar não tem a Falange média. Na Figura 2 é possível identificar todas as partes de uma mão.

Como é possível ver na Figura 3, a malha base importada possuía toda a estrutura hierárquica, e o trabalho foi combinar os dois modelos. O processo foi feito diretamente no Blender, e possibilitou que o esqueleto fosse ajustado na posição correta, evitando a perda das articulações já mapeadas, sendo necessário ajustes pontuais. Na Figura 3, os modelos já estão combinados, porém a malha humana ainda faz parte da mão, para representar como o processo foi feito.

Figura 3 - Combinação de malhas.

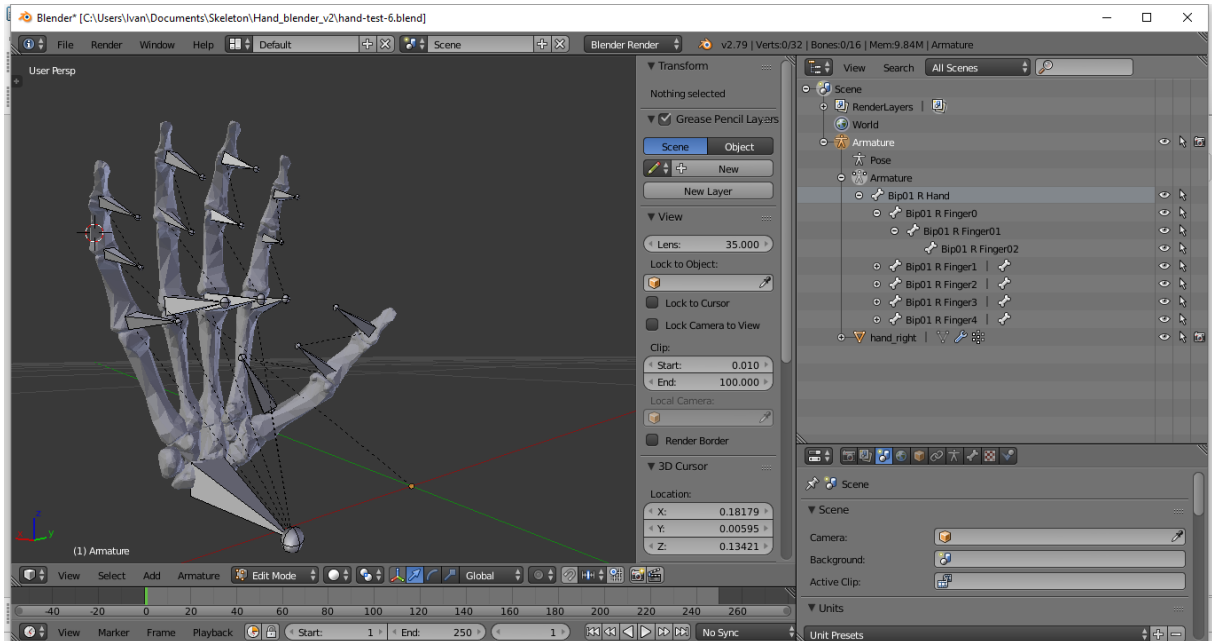


Fonte: Elaborado pelos autores (2019).

A versão final foi ajustada para que apenas a mão direita fosse representada, e para isso, os demais componentes foram removidos ou desabilitados, para evitar que prejudicasse o uso no Unity. Na Figura 4 é mostrado o modelo esquelético na forma final, com destaque para

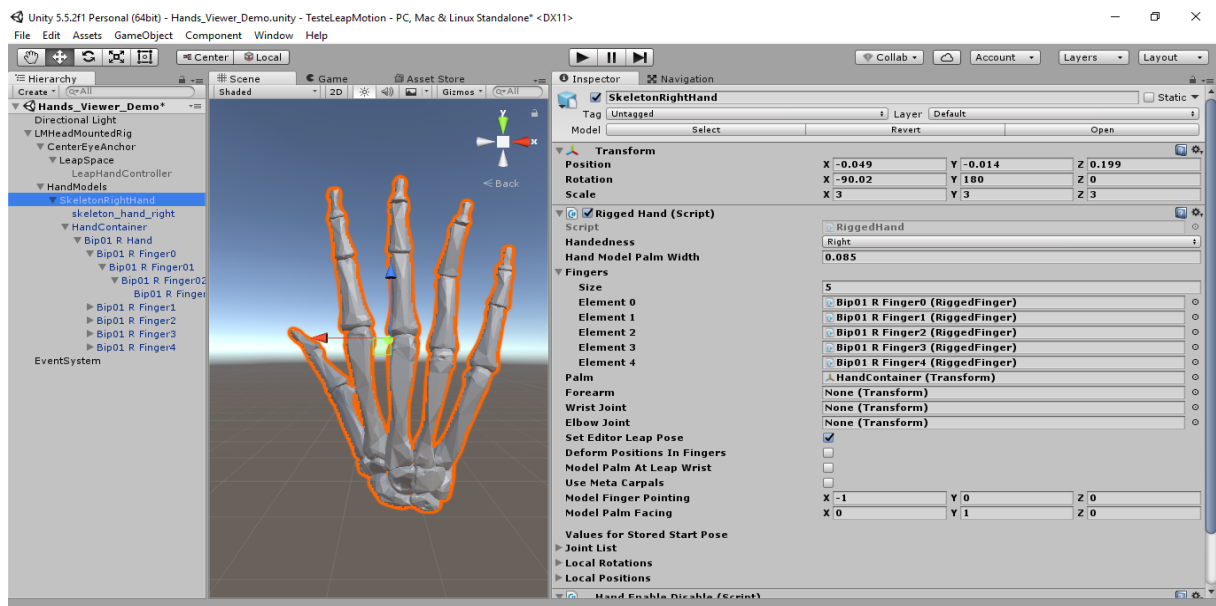
a distribuição hierárquica da armadura, que é composta pela base da mão e os cinco dedos com suas falanges vinculadas, conforme a descrição das partes já feitas no trabalho.

Figura 4 - Modelo esquelético finalizado e hierarquia da armadura



Fonte: Elaborado pelos autores (2019).

Figura 5 – Configurações do script de controle da mão



Fonte: Elaborado pelos autores (2019).

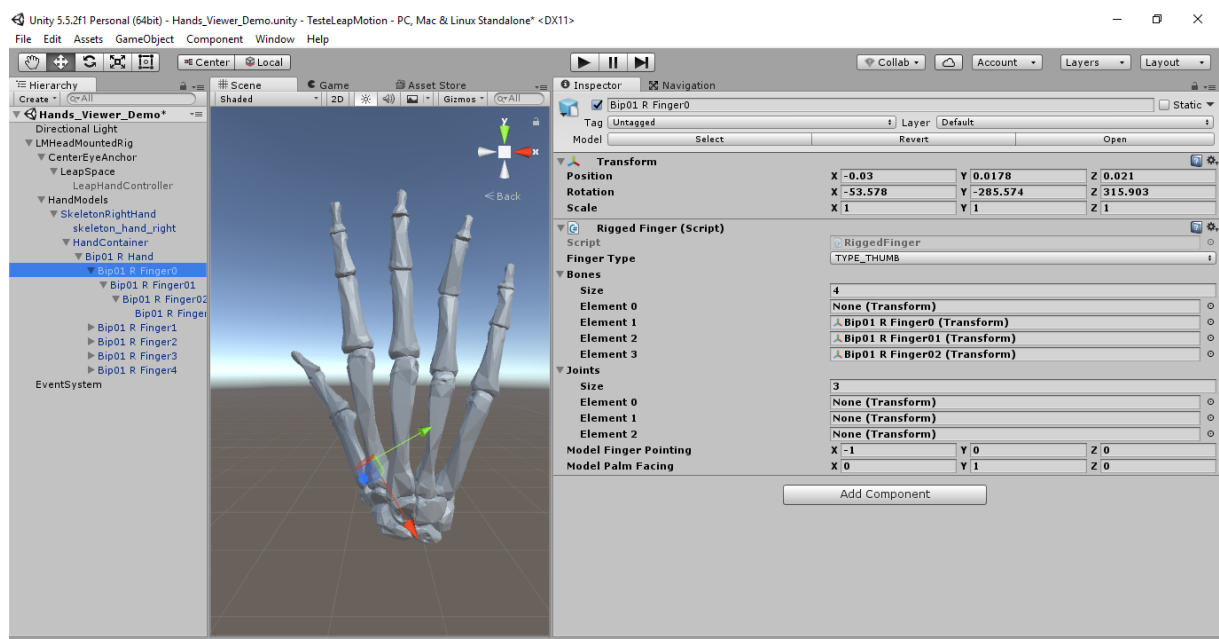
No projeto feito no Unity, o modelo 3D foi importado e acrescentado na cena, sendo necessário fazer todo o mapeamento das partes, visando contemplar cada membro. Igualmente

o que foi disposto na armadura feita no Blender, no Unity a hierarquia permaneceu, e cada parte foi animada de forma individual.

A Figura 5 mostra como o modelo ficou organizado na cena. A base é a malha esquelética, que tem amarrada a ela a base da mão, que é o Carpo, logo após a palma da mão, denominada de Metacarpo, e por últimos os dedos, que são as Falanges e suas subdivisões. Cada dedo possui três articulações, que são interligadas, seguindo a hierarquia lógica de uma mão humana. O painel de inspeção contempla diversos atributos de configurações para cada parte do modelo simulado, podendo dar destaque para as três camadas que envolvem a animação de modelos tridimensionais. É possível fazer o controle da armadura e de todos os ossos e articulações que compõem a mesma, contemplando assim a primeira camada do processo de animação. Há possibilidade de configurar as distorções do modelo que trabalham baseadas nos eixos x, y e z, que atendem a segunda camada, e por último é possível configurar a aparência da malha, que atende a terceira camada de animação.

Outro atributo importante é o *script* de controle, que permite vincular um código de programação, para que o determinado elemento tenha ações e controles feitos via linha de código. Destaca-se também o atributo que controla a malha que o componente utiliza, ou seja, é através dele que o objeto toma forma e é projetado na simulação.

Figura 6 – Configurações do *script* de controle dos dedos

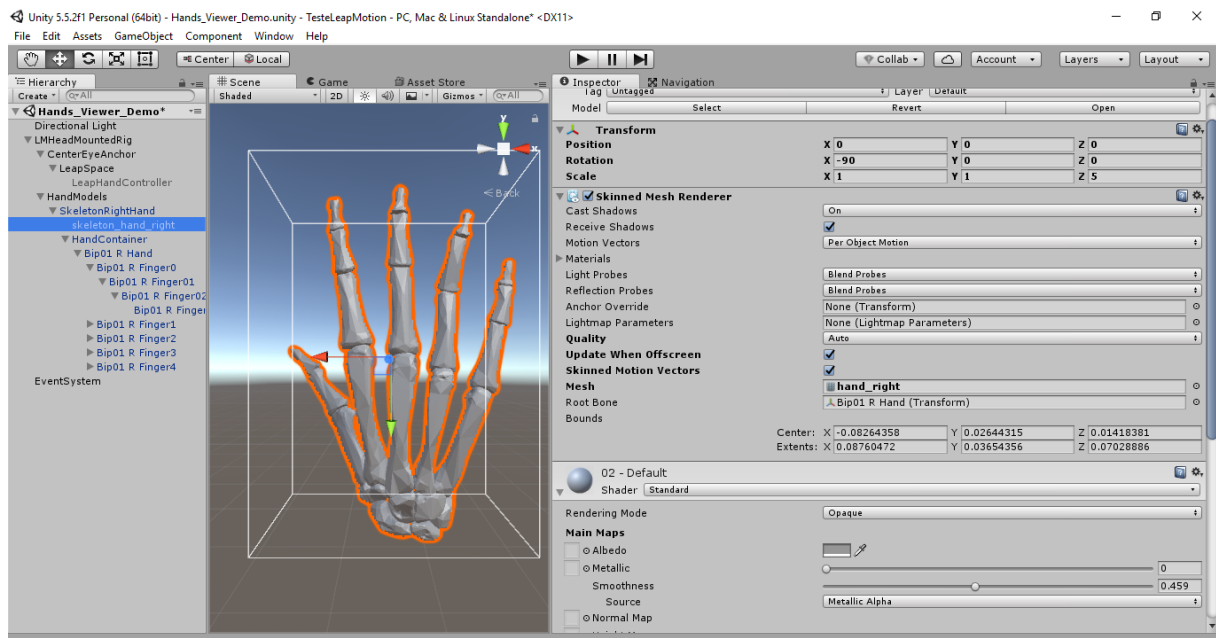


Fonte: Elaborado pelos autores (2019).

Para a mão de forma geral, há um *script* específico que controla toda a armadura, conforme visto na Figura 6. Este tem todas as Falanges mapeadas, e as mesmas seguem essa organização, no momento da execução e transformação da malha. Na mesma Figura 6 é possível ver a vinculação do *script* que controla a transformação de cada dedo. Da mesma forma há um vínculo de elementos em cada dedo, fazendo a composição de todas as subdivisões das Falanges.

A malha esquelética de forma geral também possui alguns atributos. Como ela é a base da armadura, tudo que está abaixo dela segue conforme o definido. Dentre as principais funções, mostrada na Figura 7, está o mapeado do tipo de malha e as definições de cores e estilo que a mesma pode ter, além de configurações pontuais de qualidade e modo de exibição que não foram exploradas na plenitude.

Figura 7 - Detalhamento de alguns atributos da malha esquelética



Fonte: Elaborado pelos autores (2019).

3.2 Configuração de software e hardware

Para criação e testes do trabalho foram utilizados recursos simples que permitiram as simulações. Os testes foram feitos em ambientes com um grau médio alto de luminosidade, e também em ambientes escuros, com pouca ou quase nenhuma luz. Para o dispositivo a luminosidade não é uma barreira, visto que ele utiliza recursos de emissão e recepção de infravermelho combinada com a câmera de profundidade para fazer as capturas.

Como base de toda a estrutura foi usado um notebook Asus X550C, que possui um processador Intel Core i5 de 1.7 GHz, com 6 GB de memória RAM e 500 GB de disco, com um sistema operacional Windows 10 Home Single Language de 64 bits. Para completar o ambiente de simulação foi utilizada a plataforma de desenvolvimento de jogos Unity, na sua versão 5.5.2f1 Personal de 64 bits com distribuição gratuita no plano básico para estudante. Foi utilizado também o dispositivo Leap Motion Controller V2, responsável por capturar os dados de entrada da aplicação.

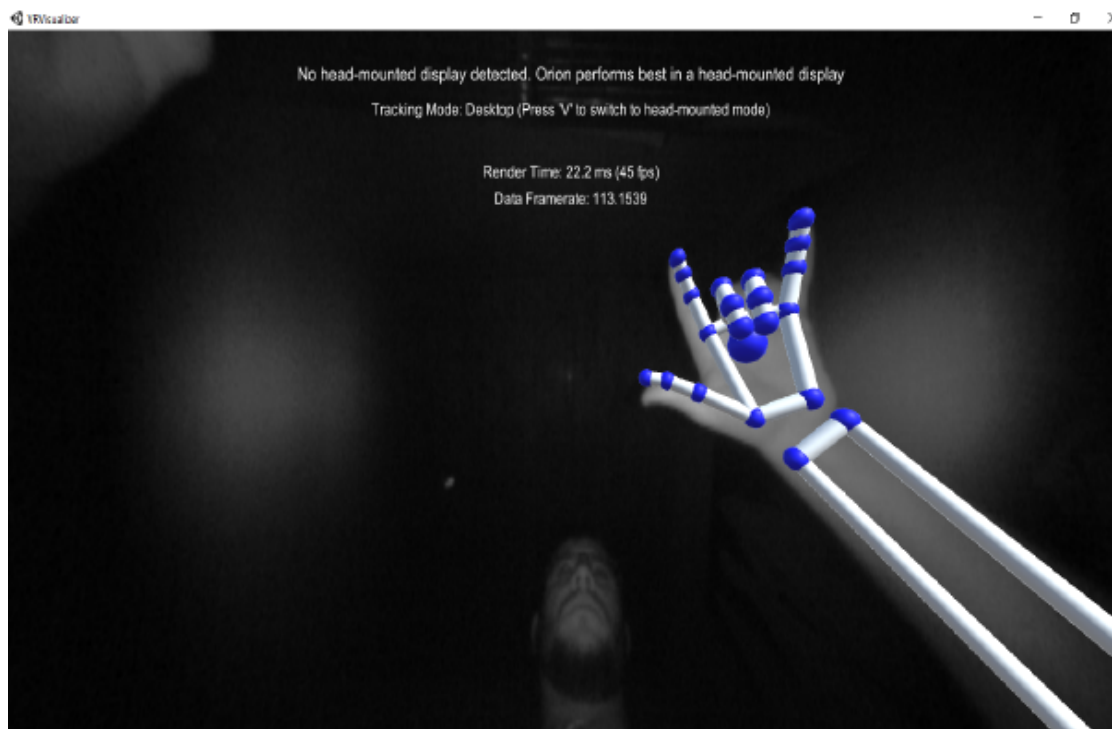
O SDK utilizado para o desenvolvimento foi a versão Orion Beta V2.3.1, com a integração do módulo adicional de mãos na versão 2.1.2, que além de possuir os recursos para integrar o sensor com plataforma Unity, conta também com um painel de controle para configurações e parametrizações gerais do dispositivo, um visualizador que proporciona a visão de captura do aparelho e também uma loja de aplicativos baseados e desenvolvidos exclusivamente para o Leap Motion Controller, proporcionando uma interação extremamente interessante com o hardware, podendo explorar diversos recursos e descobrir novas possibilidades.

O painel de controle conta com uma série de configurações, que permitem personalizar diversos atributos do hardware. Alguns ajustes foram especialmente necessários para garantir uma renderização com menos falhas durante os testes. O primeiro deles foi a possibilidade de permitir baixo desempenho nas capturas, pois deixando a configuração conforme a definição do fabricante se consegue uma renderização mais precisa, mesmo assim em diversos momentos o desempenho não foi adequado devido a um bloqueio, onde a imagem não foi capturada. Com essa configuração desligada a qualidade ficou inferior, porém as falhas pararam de acontecer, e a leitura dos movimentos do usuário tornou-se mais realista.

Outra funcionalidade muito importante foi a calibragem manual do dispositivo. Com ela foi possível obter mais precisão e melhor desempenho. O processo de calibragem sugere testar o ambiente em diversas superfícies, como espelhos, telas brilhantes, vidros, entre outros, fazendo assim, com que seu desempenho seja alto em diversos ambientes e condições diferentes, e também o tornando mais assertivo e preciso durante o rastreamento. Este processo foi realizado em diversos momentos durante os testes, pois houve a necessidade de alterações e atualizações de parâmetros importantes, resultando em uma captura mais precisa.

Para auxiliar no comportamento do sensor e facilitar os testes foi usado o visualizador de captura do Leap Motion Controller. Nele foi possível observar a captura feita pela câmera e como a mesma se comporta em diferentes situações. Na Figura 8 é possível ver um dos testes, com a captura da mão e de todos os dedos até o antebraço.

Figura 8 - Visualizador de captura do Leap Motion Controller



Fonte: Elaborado pelos autores (2019).

Com este visualizador foi possível testar todas as configurações feitas no painel de controle do dispositivo, e ver na prática como a câmera e os resultados de rastreamento se comportaram, facilitando assim, o desenvolvimento dos resultados esperados. Ainda com ele, foi possível entender os pontos de captura e comportamento em diversos ambientes, com luminosidade variada. O visualizador mostra também a taxa de quadros (*framerate*) por segundo que é feita pelo dispositivo. Em uma média, durante os testes foi conseguido uma taxa de 43 quadros, com um tempo médio de renderização de 19,9 milissegundos.

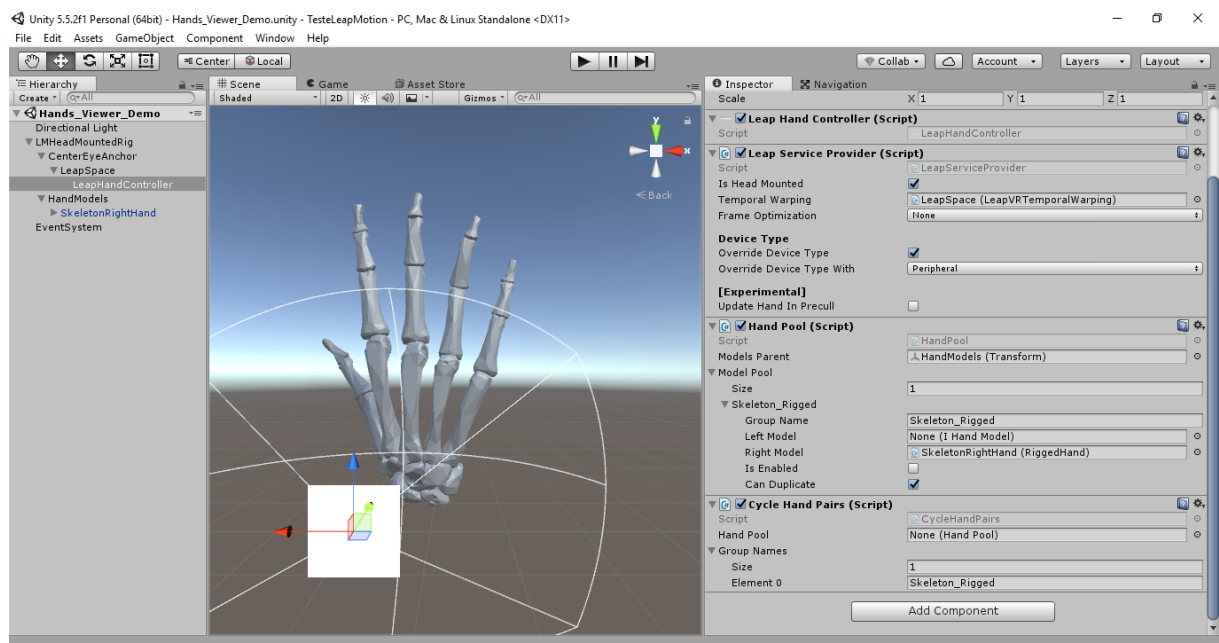
3.3 Integração com o Unity

A implementação do trabalho deu-se a partir da integração do dispositivo Leap Motion Controller com a plataforma de desenvolvimento de jogos Unity. O dispositivo fez a função de interpretar a mão do usuário e enviar todas as informações para a plataforma. No Unity, foi necessário processar esses dados e mostrar em vídeo o que estava sendo capturado pela câmera. Para a simulação foi utilizado o SDK nativo do Leap Motion Controller, com alguns módulos específicos, que possuem toda a estrutura necessária para integração com o Unity, bem como a comunicação com o sensor usado. A estrutura de projeto se repete para os

modelos de mão testados, desde o modelo mais simples do pacote até o modelo personalizado, já contendo o esqueleto na sua versão final.

Para possibilitar o uso das informações de entrada exclusivamente de captura de mão do usuário, foi importado junto ao projeto um módulo exclusivo para essa captura, já pronto para uso, projetado pelo fabricante. Nele, alguns modelos padrões fazem o acompanhamento, bem como o restante da estrutura que permite o devido uso e testes. A partir desse pacote, foi criado o modelo personalizado usado no desenvolvimento do trabalho. Na Figura 9 é possível ver como são dispostos os componentes que fazem parte da hierarquia da cena usada.

Figura 9 - Visão geral do painel de inspeção do dispositivo



Fonte: Elaborado pelos autores (2019).

O primeiro componente é a luz direcional (DirectionalLight), que possibilita iluminar o ambiente conforme a necessidade. Logo abaixo, encontra-se o principal componente da cena (LMHeadMountedRig), que trata basicamente do dispositivo usado, amarrado a ele, estão os demais componentes, como o direcionador da câmera (LeapSpace) que é a base de captura considerando o raio de precisão do sensor, e por último está disposto o modelo de esqueleto usado (SkeletonRightHand). A maioria dos componentes foram usados da mesma forma que o fabricante dispôs as configurações, visto que quase todos estavam ajustados para comportar o sensor de movimento. Apenas alguns ajustes foram necessários no modelo esquelético e no componente LeapHandController, que é responsável pela habilitação do modelo personalizado na cena.

4 VALIDAÇÃO

Para realização dos testes, buscou-se avaliar a precisão do sistema de captura em identificar diversas poses das mãos, tanto com movimentos suaves como em transições abruptas. Os testes utilizaram a mesma estrutura física do desenvolvimento. De maneira geral, a estrutura comportou-se dentro do esperado, tendo sucesso de reprodução da pose na maior parte das situações. Em alguns casos pontuais, houve problemas de renderização do modelo, distorções, transformações erradas, e até mesmo algumas falhas. Esses estão ligados em alguns ajustes principalmente relacionados ao sensor de captura, que tem interrupções na leitura dos movimentos do usuário, e que não puderam ser tratados, porém não prejudicam de forma expressiva o trabalho feito. Todas as figuras ilustradas foram feitas durante a execução do trabalho concluído, e capturadas nos testes.

Na Figura 10 é possível ver a representação do esqueleto de mão em sua posição neutra, tanto na visão frontal quanto de costas. Nelas são representadas a disposição de todas as partes que compõem o esqueleto, simulando os ossos igualmente aos existentes fisicamente na anatomia de uma mão humana real.

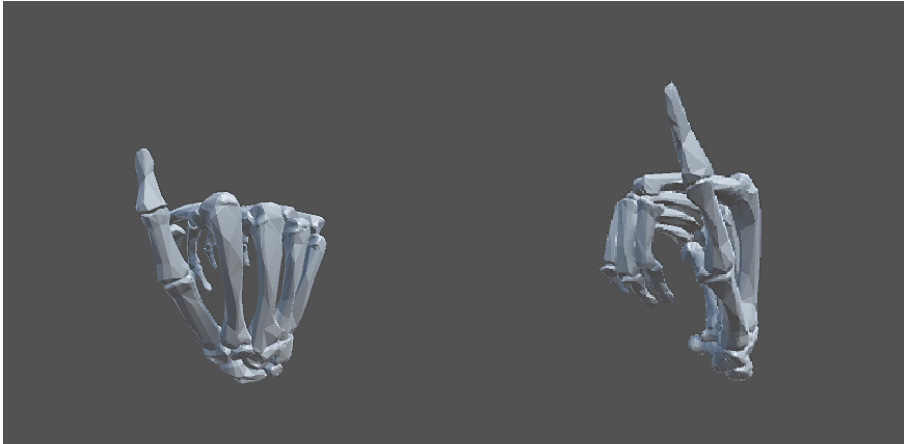
Figura 10 – Mão em pose neutra visto de costas (à esquerda) e de frente (à direita).



Fonte: Elaborado pelos autores (2019).

A simulação usando a mão fechada teve alguns problemas nas transformações, porém todos aceitáveis. A Figura 11 retrata o comportamento obtido, usando essa pose, e nos dois exemplos ilustrados o polegar não teve uma transformação tão eficiente quanto os demais dedos, apresentando dificuldades no posicionamento e orientação. De qualquer forma o resultado é muito próximo ao movimento real.

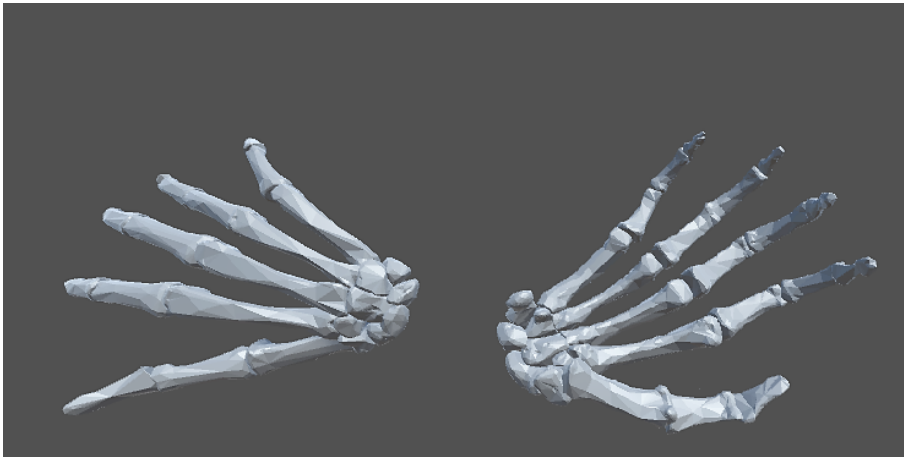
Figura 11 – Mão fechada visto de costas (à esquerda) e de frente (à direita).



Fonte: Elaborado pelos autores (2019).

As inclinações também foram testadas, visando ver o comportamento do modelo nas rotações dos eixos de trabalho. Nos dois casos simulados na Figura 12 os resultados foram satisfatórios, e não tiveram nenhuma deformação, ou transformação equivocada.

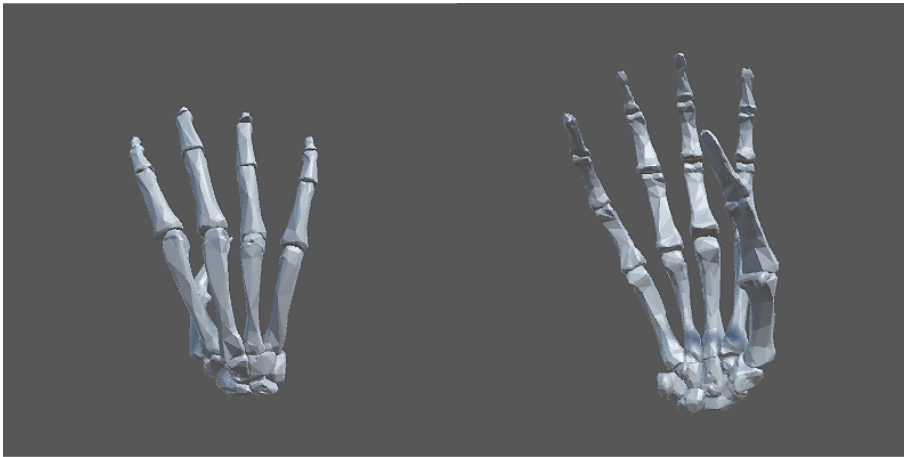
Figura 12 – Mão inclinada para esquerda (vista de costas) e para direita (vista de frente).



Fonte: Elaborado pelos autores (2019).

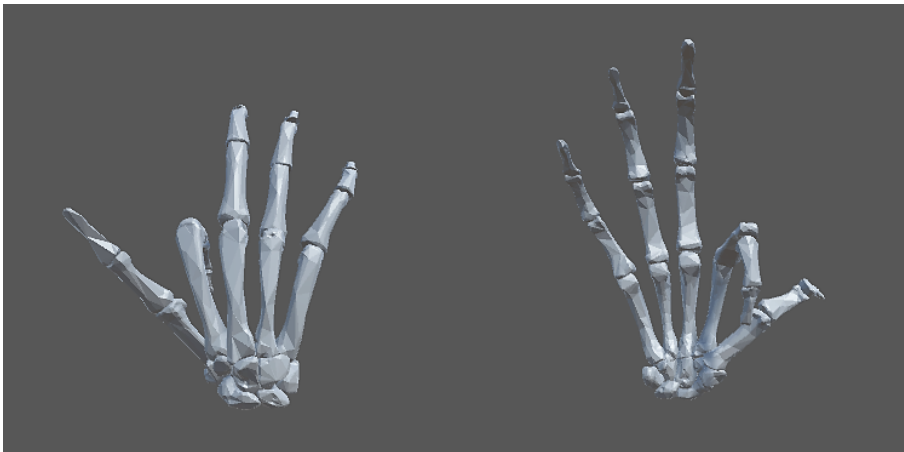
Conforme já citado para a Figura 11, novamente houve comportamentos não tão bons na pose simulada na Figura 13. Neste teste foi tentado simular o polegar fechado, porém a sua transformação e animação não ocorreu da melhor forma, mas é possível entender qual ação está sendo feita pelo usuário. Os demais dedos, indicador, médio, anelar e mínimo, tiveram seu comportamento extremamente aceitável, diferentemente do polegar. A Figura 14 ilustra um desses casos, em que todos os testes ocorreram praticamente iguais a uma mão humana.

Figura 13 - Dedo polegar fechado visto de costas (à esquerda) e de frente (à direita).



Fonte: Elaborado pelos autores (2019).

Figura 14 - Dedo indicador fechado visto de costas (à esquerda) e de frente (à direita).



Fonte: Elaborado pelos autores (2019).

Nos testes em que os demais dedos, além do polegar, estavam fechados, os resultados foram muito melhores do que havia se conseguido com esse dedo até o momento. Também foram testadas poses em que em alguns casos são difíceis de executar, como, por exemplo, abrir apenas o dedo anelar, sem interferir no comportamento dos demais, ainda assim as simulações foram satisfatórias. Em alguns casos pontuais novamente o dedo polegar foi alterado em função dos outros, mas nada que impedisse ou tornasse o resultado da ação feita em algo totalmente anormal.

O ponto mais crítico e de caso limite foi observado nos testes em que se tentou tocar os dedos ao polegar. Esse movimento não é de extrema dificuldade quando feito em uma mão real, porém nas simulações os resultados foram apenas medianos. As Figuras 15 e 16 mostram

os resultados onde é possível ver que mesmo o usuário tocando seu dedo ao polegar, a simulação não faz o mesmo, e permanece com uma certa distância entre um dedo e outro. É possível identificar qual movimento está sendo feito, mas poderia ter uma proximidade maior.

Figura 15 - Dedo indicador (à esquerda) e médio (à direita) tocando dedo polegar.



Fonte: Elaborado pelos autores (2019).

Figura 16 - Dedo anelar (à esquerda) e mínimo (à direita) tocando dedo polegar.



Fonte: Elaborado pelos autores (2019).

Outros testes foram feitos simulando outras ações e posições de mão e dedos. Nesses casos os resultados foram praticamente os mesmos já mostrados anteriormente, não tendo necessidade de apontamento e discussão. Em uma análise crítica, tendo em vista as limitações da estrutura do trabalho, os resultados foram extremamente satisfatórios. Os detalhes principais comentados, que basicamente são mais expressivos no comportamento do dedo polegar, podem ser corrigidos com ajustes na malha esquelética, e com isso o resultado ficaria mais preciso e próximo ao movimento humano.

4.1 Dificuldades encontradas

Ao longo do desenvolvimento, diversos problemas foram encontrados. A dificuldade inicial deu-se na obtenção de um modelo tridimensional de esqueleto humano. Os diversos modelos encontrados tinham baixa qualidade, poucos detalhes na composição, formatos incompatíveis ou de difícil alteração, e nos pagos, preços altos de aquisição.

Após a obtenção de um modelo adequado para o trabalho, o novo desafio foi transformá-lo em um modelo compatível com o modelo padrão de mão humana vindo do fabricante. Ou seja, fazer com que a malha humana fosse substituída pela malha esquelética, reconstruindo o modelo de hierarquia e armadura disponível no modelo original.

Outro obstáculo foi adaptar a estrutura do projeto proveniente do SDK, pois todas as configurações e parametrizações existentes eram específicas aos modelos do fabricante. A estrutura do projeto não contemplava modelos personalizados, mesmo que no *site* do fabricante estejam listadas diversas possibilidades de interações desse tipo. A forma simples e fácil tratada na documentação existente aplicava-se apenas para modelos de teste do próprio fabricante, tornando assim o ajuste bastante oneroso. Nesse estágio do desenvolvimento, os principais problemas foram relacionados à parametrizações do *setup* de integração e na configuração do modelo personalizado, fazendo com que a hierarquia desenvolvida fosse respeitada e trabalhasse de forma correta durante a execução.

Há também alguns problemas que ainda permanecem sem resolução, girando em torno da qualidade da renderização feita. Nas imagens capturadas nos testes é possível ver que em diversos momentos, principalmente nos críticos, o comportamento da malha não acontece de forma tão natural quanto o esperado. Há distorções nas transformações, e em alguns momentos, não respondem de forma clara aos movimentos feitos pelo usuário. As falhas de captura do dispositivo também afetam os resultados de saída, pois há interrupções nos dados de entrada. Foram testadas diversas possibilidades, inclusive utilizando os recursos do painel de controle do sensor, conforme já mencionado anteriormente. Não há garantia de estabilidade em todas as poses, o que pode prejudicar a experiência de uso.

5 CONCLUSÃO

A área da Realidade Virtual e Aumentada está em constante ascensão, proporcionando uma infinidade de possibilidades para os interessados. A partir disso o presente trabalho teve por finalidade o desenvolvimento da visualização esquelética de mão humana em tempo real, visando futuramente contribuir como ferramenta de apoio ao ensino.

Para seu desenvolvimento foram realizados estudos apresentados no referencial teórico, bem como um estudo mais aprofundado nos principais conceitos que envolvem animação de modelos 3D, criação e manutenção de projetos e cenas do Unity, configuração e organização de propriedades de modelos tridimensionais, reconhecimento e captura de câmeras de profundidade e infravermelho e integrações com plataformas e ambientes diversos relacionados a animação, visando uma maior abrangência e flexibilidade para o desenvolvimento do projeto proposto.

Com base nestes estudos foi possível desenvolver a visualização conforme os requisitos previamente mapeados e definidos no início do processo, tendo como resultado o modelo esquelético em funcionamento, atualizado em tempo real e baseado na captura dos movimentos feitos pelo usuário.

Os principais pontos do trabalho estão relacionados com a modificação e personalização do modelo tridimensional esquelético, tendo como base o modelo de malha padrão do fabricante. Através deste modelo foi possível definir as hierarquias vinculadas à armadura da malha, e fazer com que o modelo se comportasse de forma correta durante a maior parte da execução, sem quebras, ou transformações indesejadas, para evitar tornar a simulação uma experiência negativa. Igualmente importante foi o entendimento da plataforma de jogos Unity, bem com sua integração com o dispositivo Leap Motion Controller.

Apesar da Realidade Aumentada ser uma área já bem estudada, projetos e trabalhos envolvendo a estrutura desenvolvida não são comuns de encontrar, e ainda há pouco conteúdo para pesquisa. O próprio fabricante do Leap Motion Controller cita em alguns momentos que seus modelos e estruturas 3D não são simples de trabalhar, e exigem um trabalho adicional de desenvolvimento. A plataforma do fabricante disponibiliza um *setup* de testes com modelos personalizados, porém é bastante limitada e gira em torno de uma quantidade pequena de atributos para configuração nos *scripts* que envolvem os objetos, assim diversas possibilidades não podem ser exploradas e testadas.

Como continuidade deste trabalho pode-se explorar ainda mais o modelo 3D e torná-lo mais robusto, acrescentando outras estruturas, como por exemplo, os músculos ou nervos que compõem a mão. Há também a possibilidade de explorar partes do antebraço do usuário, visto que o sensor de captura tem condições de tratar esses membros. Com essa evolução, o modelo tornar-se-ia mais interessante para a área da Fisioterapia, ou outras áreas da Saúde.

REFERÊNCIAS

AZEVEDO, Eduardo. CONCI, Aura. **Computação gráfica: geração de imagens**. Rio de Janeiro: Campus, 2003.

FIGUEIREDO, L.; SOUZA, P.; SIMÕES, F.; ROBERTO, R.; TEICHRIEB, V. **Interação Natural a partir de Rastreamento de Mãos**. Niterói: SBC, 2012.

KIRNER, C. **Realidade Virtual e Aumentada**. Acesso em 28 de Maio de 2017. Disponível em <<http://www.realidadevirtual.com.br>>.

KIRNER, Cláudio; SISCOOTTO, Robson. **Realidade Virtual e Aumentada. Conceitos Projetos e Aplicações. IX Simpósio de RV e RA**. Petrópolis-RJ: SBC, 2007.

MARTINS, Valéria Farinazzo; GUIMARÃES, Marcelo de Paiva. **Desafios para o uso da Realidade Virtual e Aumentada de maneira efetiva no ensino**. Workshop de Desafios da Computação Aplicada à Educação. São Paulo, 2012.

MITRA, S. ACHARYA, T. **Gesture recognition: A surveys**. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and reviews, v37, n3, 2007.

NEGROPONTE, Nicholas. **A vida digital**. São Paulo: Companhia das Letras, 1995.

SILVA, Luciano. **Observatórios virtuais. In Tendências e Técnicas em Realidade Virtual e Aumentada**, v.6. Gramado: SBC, 2016.

TORI, Romero; KIRNER, Claudio. **Fundamentos da Realidade Virtual. Fundamentos e Tecnologia de Realidade Virtual e Aumentada Simpósio**. Belém-PA: SBC, 2006.