



**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
CURSO DE SISTEMAS DE INFORMAÇÃO**

**ITALO NEY SILVA PESSOA**

**INTERAÇÃO COM PERSONAGEM FISICAMENTE SIMULADO UTILIZANDO  
CAPTURA DE MOVIMENTOS COM O SENSOR KINECT**

**Quixadá  
2016**

ITALO NEY SILVA PESSOA

INTERAÇÃO COM PERSONAGEM FISICAMENTE SIMULADO UTILIZANDO  
CAPTURA DE MOVIMENTOS COM O SENSOR KINECT

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rubens Fernandes Nunes

Quixadá  
2016

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

P567i Pessoa, Italo Ney Silva  
Interação com personagem fisicamente simulado utilizando captura de movimentos com o sensor Kinect / Italo Ney Silva Pessoa. – 2016.  
41 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Bacharelado em Sistemas de Informação, Quixadá, 2016.  
Orientação: Prof. Dr. Rubens Fernandes Nunes  
Área de concentração: Computação

1. Computação gráfica 2. Interação homem-máquina 3. Videogames 4. Personagens de videogames I.  
Título.

**ITALO NEY SILVA PESSOA**

**INTERAÇÃO COM PERSONAGEM FISICAMENTE  
SIMULADO UTILIZANDO CAPTURA DE MOVIMENTOS  
COM O SENSOR KINECT**

Trabalho de Conclusão de Curso submetido à Co-ordenação do Curso de Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação. Área de concentração: Computação.

Aprovada em: \_\_ / \_\_ / \_\_\_\_

**BANCA EXAMINADORA**

---

Prof. Dr. Rubens Fernandes Nunes (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof Dr. Críston Pereira de Souza (Membro)  
Universidade Federal do Ceará (UFC)

---

Prof<sup>a</sup>. Dra. Paulyne Matthews Jucá (Membro)  
Universidade Federal do Ceará (UFC)

Dedico este trabalho a minha família  
que me criou, me educou e me ama.

## AGRADECIMENTOS

Primeiramente a meus pais, que mesmo em suas limitações sobre o conhecimento adquirido durante todos esses anos, sempre confiaram em mim, sabendo que o saber que busco não é apenas para crescimento próprio mas também para tentar um dia contribuir para criar um mundo melhor. Se estou aqui hoje é por causa deles.

A minha avó, Clarice e meu tio Célio, que cuidaram de mim por muito tempo durante a infância.

A meus amigos, Lucas Wendell, Daniel Araújo, Samuel Freitas, Felipe Freitas, Paulo Filipe, Damohill, Gerlan, Lucão, Matihara, Udy, Dr. Cleilson, Glauco, Anderson Marinho e Martinho que por muito tempo compartilharam seu conhecimento comigo colaborando para me tornar o que sou hoje. Um destaque para Daniel Araújo que sempre apoiou minhas ideias, mesmo quando não eram muito claras, me ajudando a perceber que aquele poderia não ser o melhor caminho. Glauco, me mostrou que os problemas podem ser grandes, mas quando se sabe lidar com eles, eles não passam de... problemas. Dr. Cleilson, que com sua famosa frase me ensinou que não importa o quando eu me esforce para atingir meus objetivos, concluir minhas tarefas, meus deveres. Eu sempre posso fazer melhor.

Marlon, Gustavo, Carlos Daniel e Maria Vilani, que hoje são apenas crianças mas que um dia perceberão que tiveram uma imensa contribuição na construção deste trabalho.

A meu irmão, que sempre foi uma referência para mim.

A todos os professores da Universidade Federal do Ceará - Campus Quixadá, que contribuem na mudança do cenário do sertão ao compartilhar o conhecimento que antes era luxo. Especialmente ao Professor Rubens Fernandes Nunes, que com paciência me orientou e sempre que necessário explicou o que precisava saber para concluir este trabalho. Sou realmente muito grato.

Não esqueci de Jefferson, Stephane e Diego. Futuros Arquitetos e Urbanistas que em pouco tempo abriram meus olhos para uma nova gama de conhecimento, e uma forma de viver a vida com mais alegria.

A D'avila, que me ajudou a entender melhor o que eu precisava para atingir meus objetivos, que apoiou minhas decisões e me chamou atenção quando passava dos limites. Que

apagou do quadro branco contas e cálculos que já não eram mais necessários, abrindo as portas para mais conhecimento. Que me atura, e que enquanto escrevo está aqui no seu sono de número 1565. Eternizo aqui a minha gratidão.

A aqueles que estão lendo e não foram citados, minhas sinceras desculpas. Em um momento como esse, sua mente está cheia de planos, sonhos, pensamentos. É muita informação misturada. Mas tenha certeza de que todos aqueles que me ajudaram, serão lembrados. Obrigado a todos.

“If you hear a voice within you say ‘you cannot paint’,  
then by all means paint, and that voice will be silenced.”

(Vincent Van Gogh)

## RESUMO

A utilização de personagens animados é um dos principais fatores de sucesso de jogos e simuladores. Personagens que reagem ao ambiente a seu redor proporcionam uma maior imersão ao usuário. Uma das formas de melhorar a imersão do usuário é proporcionar um maior controle sobre o personagem. Pensando nisso, foi criado um protótipo de personagem fisicamente simulado definido a partir das características físicas do usuário. Dessa forma o personagem utilizado na simulação se torna único para cada usuário, buscando assim melhorar a imersão do usuário ao ter um personagem que lhe representa, com suas vantagens e limitações físicas. Foi utilizado o dispositivo *Kinect* para capturar os movimentos do usuário em conjunto com a biblioteca de simulação física *Open Dynamics Engine*, responsável por criar os corpos que compõem o personagem e calcular os torques e forças necessárias para animar o personagem, através de um controlador PD. A biblioteca *OpenGL* foi utilizada para desenhar os elementos da simulação. Uma vez criado o personagem, o mesmo é inserido em cenários de interação previamente configurados para analisar a interação do usuário. O protótipo representou com um nível satisfatório de precisão os movimentos do usuário e suas reações a colisões no ambiente virtual. Os cenários configurados e a interface projetada para controlar a simulação foram eficazes nos testes realizados, comprovando que os personagens criados a partir deste trabalho possuem características dos usuários que o utilizaram.

**Palavras-chave:** Computação Gráfica. Simulação Física. Animação de Personagens.

## ABSTRACT

The use of animated characters is one of the main factors of successful games and simulators. Characters that react to the environment around them provide greater immersion to the user. One way to improve the user immersion is to provide greater control over the character. Thinking about it, a physically simulated character prototype created based on the physical characteristics of the user. Thus the character used in the simulation becomes unique for each user, seeking to improve the user immersion to have a character that represents him, with their advantages and physical limitations. A Kinect Sensor was used in order to capture user movement in conjunction with the physical simulation library Open Dynamics Engine, responsible for creating the bodies that make up the character, and calculating the torques and forces necessary to animate the character through a PD controller. The OpenGL library was used to design the simulation elements. Once created the character, the same is inserted into interaction scenarios previously configured to analyze user interaction. The prototype represented with a satisfactory level of accuracy the user's movements and reactions to external perturbations from the virtual environment. The configured scenarios and the interface designed to control the simulation were effective in the tests, proving that the characters created from this work have characteristics of users who used it.

**Keyword:** Computer Graphics. Physics simulation. Character animation.

## OVERZICHT

Het gebruik van geanimeerde karakters is een van de belangrijkste factoren van succesvolle games en simulatoren. Tekens die reageren op de omgeving om hen meer onderdompeling aan de gebruiker. een manier om de gebruiker onderdompeling verbeteren is om meer controle over het karakter verschaffen. het denken over, creëerde een fysiek gesimuleerd karakter prototype op basis van de fysieke kenmerken van de gebruiker. Waardoor de in de simulatie karakter wordt uniek voor elke gebruiker, die ter verbetering van de gebruiker onderdompeling een teken dat hij vertegenwoordigt, met hun voordelen en fysieke beperkingen. Een Kinect Sensor werd gebruikt om de gebruiker de beweging vast te leggen in samen met de fysieke simulatie bibliotheek Open Dynamics Engine, verantwoordelijk voor het creëren de organen die deel uitmaken van het karakter, en de berekening van de koppels en krachten die nodig zijn om animeren het karakter door middel van een PD-controller. De OpenGL bibliotheek werd gebruikt voor het ontwerpen simulatie elementen. Zodra het personage, wordt daarvan in wisselwerking's ingevoegd eerder geconfigureerd om interactie te analyseren. Het prototype vertegenwoordigde een bevredigende nauwkeurigheid bewegingen van de gebruiker en reacties op externe perturbations uit de virtuele milieu. De geconfigureerde scenario's en de interface ontworpen voor de simulatie waren controle effectief in de tests, waaruit blijkt dat de personages uit dit werk hebben kenmerken van gebruikers die het gebruikt.

**Trefwoorden:** Computer beelden. Physics simulatie. Motion Capture.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Sensor 3D PrimeSense . . . . .	20
Figura 2 – Componentes do <i>Kinect</i> . . . . .	20
Figura 3 – Articulações identificadas pelo <i>Kinect</i> . . . . .	20
Figura 4 – Pose PSI para identificar personagem. . . . .	22
Figura 5 – Ruído causado por superfície reflexiva. . . . .	22
Figura 6 – Erro no momento de criar o personagem. . . . .	23
Figura 7 – Quanto pesa cada parte do corpo. . . . .	24
Figura 8 – Exemplo de colisões com e sem controlador PD e juntas. . . . .	25
Figura 9 – Correção do posicionamento da geometria no ODE . . . . .	26
Figura 10 – Junta <i>ball and socket</i> . . . . .	27
Figura 11 – Representação de uma mola do controlador PD . . . . .	28
Figura 12 – Mola com amortecimento agindo em uma junta. . . . .	28
Figura 13 – Interface de controle. . . . .	29
Figura 14 – Configuração de usabilidade (acesso com uma tecla). . . . .	31
Figura 15 – Configuração de usabilidade (tooltip). . . . .	31
Figura 16 – Todos os objetos disponíveis no cenário de interação. . . . .	32
Figura 17 – Gol de letra. . . . .	32
Figura 18 – Personagem interagindo com a caixa. . . . .	33
Figura 19 – João Teimoso. . . . .	33
Figura 20 – Saco de pancadas. . . . .	33
Figura 21 – Saco de pancadas. . . . .	34
Figura 22 – Comparação de personagens diferentes. . . . .	35

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>TRABALHOS RELACIONADOS</b>	<b>15</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
<b>3.1</b>	<b>Computação Gráfica</b>	<b>17</b>
<b>3.2</b>	<b>Animação</b>	<b>18</b>
<b>3.3</b>	<b>Simulação Física</b>	<b>18</b>
<b>3.4</b>	<b>Captura de Movimentos</b>	<b>18</b>
<b>4</b>	<b>PROCEDIMENTOS METODOLÓGICOS</b>	<b>21</b>
<b>4.1</b>	<b>Captura de movimentos do usuário utilizando o Kinect</b>	<b>21</b>
<b>4.2</b>	<b>Personagem fisicamente simulado</b>	<b>23</b>
<b>4.3</b>	<b>Movimentação do humanoide</b>	<b>25</b>
<b>4.4</b>	<b>Interface de controle</b>	<b>28</b>
<b>4.5</b>	<b>Ambiente virtual de interação</b>	<b>31</b>
4.5.1	Pênalti	32
4.5.2	Caixa	32
4.5.3	João Teimoso	33
4.5.4	Saco de Pancadas	33
4.5.5	Bola de Socos	34
<b>5</b>	<b>RESULTADOS</b>	<b>35</b>
<b>5.1</b>	<b>Protótipo de personagem fisicamente simulado</b>	<b>35</b>
<b>5.2</b>	<b>Cenários de interação</b>	<b>36</b>
<b>5.3</b>	<b>Interface de controle</b>	<b>36</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>37</b>
	<b>REFERÊNCIAS</b>	<b>39</b>
	<b>ANEXO A - DIAGRAMA DE ATIVIDADES</b>	<b>41</b>
	<b>ANEXO B - DIAGRAMA DE CASO DE USO</b>	<b>42</b>
	<b>ANEXO C - DIAGRAMA DE SEQUÊNCIA PARA CRIAR UM PERSONAGEM</b>	<b>43</b>

## 1 INTRODUÇÃO

Na tentativa de criar sistemas capazes de proporcionar ao usuário uma melhor imersão em ambientes virtuais, tecnologias cada vez mais avançadas, relacionadas à Computação Gráfica, têm sido exigidas. Azevedo (2004, p. 3) define computação gráfica como “um conjunto de ferramentas e técnicas para converter dados para ou de um dispositivo gráfico através de um computador.”. Em 1964, ao defender sua tese, Ivan Sutherland apresenta o que se tornaria uma das mais importantes criações para a computação gráfica, um sistema inovador que permitia a interação homem-computador através de uma caneta equipada com uma luz, no qual era possível desenhar círculos, linhas e formas arbitrárias diretamente na tela. O sistema atraiu o interesse de companhias automobilísticas, aeronáuticas, mecânicas e elétricas que viram naquele sistema uma forma de aprimorar e inovar seus produtos. Desde então novas tecnologias surgiram tornando a computação gráfica cada vez mais próxima do que temos hoje (gráficos ricos em detalhe e realistas). Entre algumas das tecnologias podemos citar: o algoritmo para remoção de superfícies ocultas de Robert W. Lindeman, o algoritmo Z-Buffer de Edwin Catmull, lançamento da primeira placa de vídeo pela NVIDIA e a criação da OpenGL (AZEVEDO, 2004; SUTHERLAND, 1964).

Mesmo sem perceber, utilizamos recursos da computação gráfica diariamente, seja em um vídeo game, assistindo a um programa de TV, em um banner de publicidade ou até mesmo editando um arquivo de texto no computador. Com o crescimento desta tecnologia, empresas de montagem, construção civil, robótica, multimídia, jogos e saúde investem cada vez mais para aumentar a qualidade do seu produto final, utilizando computação gráfica e áreas afins para criar novas soluções (MANSSOUR; COHEN, 2006). Com o avanço da computação gráfica diversas áreas de estudo surgiram, dentre elas: Processamento de imagens, Visão computacional, Animação, Captura de Movimentos e Simulação.

Dentre as áreas citadas anteriormente, captura de movimentos e simulação são utilizadas pelo serviço militar, aeronáutica e astrônomos para treinamentos e simulação de cenários reais. Treinar pilotos em um novo modelo de avião, treinar soldados em zonas de conflito, analisar objetos espaciais que estejam em rota de colisão com o planeta ou analisar a estrutura de um prédio são alguns dos casos em que a computação gráfica pode diminuir os riscos de um resultado negativo ao utilizar ambientes virtuais para treinar pilotos ou soldados, ou para analisar objetos espaciais ou o efeito de fenômenos naturais sobre um prédio. Mas para que isso possa ser utilizado de forma que os resultados sejam o mais próximo possíveis do mundo real, é necessário que características reais sejam adicionadas a todos os elementos envolvidos no ambiente virtual.

Para proporcionar a interação do usuário com um ambiente virtual é necessário um intermediador que forneça as informações necessárias para interagir com o mesmo. Neste trabalho é utilizada a técnica de captura de movimento para inserir o usuário em um ambiente virtual, para que o mesmo a partir de seus movimentos possa controlar um personagem fisicamente simulado que represente algumas características do usuário (peso e altura) e possa simular reações. Essas informações serão fornecidas através do sensor *Microsoft Kinect*, por ser prático, possuir preço acessível e possuir os recursos necessários para este trabalho. Utilizando o *Kinect*, os movimentos do corpo inteiro do usuário podem ser capturados e utilizados em tempo real para controlar um personagem virtual.

Dessa forma, espera-se que interações com o ambiente virtual no qual ele está inserido sejam mais naturais. Na medida em que o usuário controla esse personagem através do movimento do seu próprio corpo, é possível interagir com o ambiente em torno do personagem simulado. Realizamos também uma discussão sobre as características de animação que, quando aplicadas a jogos ou simulações que possuam realismo e boa interação como requisitos, possam melhorar a experiência do usuário final.

Na Seção 2, são apresentados os trabalhos relacionados. Na Seção 3, são descritos os conceitos importantes para o entendimento desse estudo, na Seção 4 são apresentados os procedimentos, na Seção 5 são detalhados os resultados, e por fim na Seção 6 são feitas as considerações finais.

## 2 TRABALHOS RELACIONADOS

Nesta seção são descritos trabalhos utilizados como base para o desenvolvimento deste, apresentando suas contribuições, pontos distintos e pontos em comum.

Shum e Ho (2012) apresentam um *framework* para resolver um problema comum ao utilizar o *Kinect* para captura de movimentos: os ruídos e a oclusão de pontos. O *framework* apresentado utiliza uma base de dados de movimentos para encontrar uma posição que possa ser utilizada para substituir um ponto com ruído, a fim de manter a continuidade da animação. O trabalho de Shum e Ho (2012) assemelha-se a este por utilizar o *Kinect* para capturar os movimentos e a biblioteca Open Dynamics Engine para gerar os corpos. Porém se diferencia do trabalho proposto por utilizar um único modelo para todos os usuários, ajustando as informações obtidas do *Kinect* para controlar o personagem.

Sousa (2014) utilizam visão computacional para controlar objetos e interagir com um personagem fisicamente simulado. Eles se baseiam no trabalho de (NUNES et al., 2008) para inserir um objeto controlado por marcadores, comparar as bibliotecas *ARToolKit* e *OpenCV* e avaliar qual das duas proporciona uma melhor imersão quando utilizadas para interagir com um personagem fisicamente simulado. Entretanto, são rastreados apenas objetos ou marcadores simples, que fornecem basicamente uma posição e uma orientação, limitando assim as opções de interação com o ambiente virtual. Este trabalho estende as opções de interação testadas, permitindo que o corpo inteiro do usuário seja rastreado. Apesar do rastreamento mais limitado, o trabalho de Sousa (2014) possui a vantagem de utilizar apenas uma *webcam*, tornando o sistema mais prático e barato.

Zuher e Romero (2012) criam uma interface de comunicação homem-robô, no intuito de fazer com que o robô possa reproduzir alguns movimentos do humano que o controla, assim como realizar algumas ações pré-definidas através do reconhecimento de gestos. Para possibilitar que qualquer pessoa seja capaz de controlar o robô foi utilizado o *Kinect*, que permite capturar os movimentos do usuário de forma não-intrusiva, ou seja, sem a necessidade de adicionar sensores ao corpo do usuário, permitindo um maior conforto no momento da captura de movimentos. Foi utilizado o robô humanoide NAO<sup>1</sup> para reproduzir os movimentos do usuário. O trabalho de Zuher e Romero (2012) assemelha-se a este por utilizar o mesmo dispositivo para captura de movimentos e a mesma biblioteca para obter os dados do mesmo. O trabalho de Zuher e Romero (2012) esclarece alguns conceitos básicos sobre as informações disponíveis na biblioteca

---

<sup>1</sup> Para mais informações acesse: <https://www.aldebaran.com/en/humanoid-robot/nao-robot>

*OpenNI*, utilizada para captura de movimentos. Assim como em Shum e Ho (2012), um único personagem é utilizado para reproduzir os movimentos.

Tabela 1 – Trabalhos relacionados

<b>Trabalho</b>	<b>CM<sup>2</sup></b>	<b>RA<sup>3</sup></b>	<b>Tempo real</b>	<b>Interação</b>
(SHUM; HO, 2012)	Kinect		Sim	Virtual
(SOUSA, 2014)	OpenCV <sup>4</sup>	ARToolkit	Sim	Virtual
(ZUHER; ROMERO, 2012)	Kinect		Sim	Robô
Este trabalho	Kinect		Sim	Virtual

**Fonte:** Elaborada pelo autor.

<sup>2</sup> Captura de Movimentos

<sup>3</sup> Realidade Aumentada

<sup>4</sup> Não faz a captura de movimentos completa, apenas marcadores.

### 3 FUNDAMENTAÇÃO TEÓRICA

Nesta Seção são apresentados conceitos e definições que serão utilizadas no decorrer do desenvolvimento deste trabalho. Na Seção 3.1 são descritos os conceitos básicos de Computação Gráfica necessários para o entendimento deste trabalho. A Seção 3.2 descreve brevemente o que é animação de personagens. Na Seção 3.3 é descrito o conceito de simulação física com corpos rígidos. A Seção 3.4 descreve o que é captura de movimentos e lista ferramentas e dispositivos necessários para a execução da mesma.

#### 3.1 Computação Gráfica

Evoluindo com as demais áreas da tecnologia, a computação gráfica adquiriu um papel importante no mundo da computação ao possibilitar, com a utilização de dispositivos gráficos, a visualização de informações antes abstratas e de difícil compreensão, transformando dados em imagens e possibilitando uma melhor interpretação de problemas, acontecimentos e comportamentos do mundo em que vivemos (GOMES; VELHO, 2008). Azevedo (2004, p. 3) descreve computação gráfica como “Matemática e arte”. Englobando outras áreas de estudo, tais como modelagem, visão computacional e animação, a computação gráfica antes restrita a pesquisas militares teve seu crescimento com a criação de placas e bibliotecas gráficas que permitem criar desenhos utilizando formas geométricas.

Um exemplo de biblioteca gráfica bastante utilizada é a OpenGL, uma *Application Programming Interface* (API) que possibilita o acesso a funções de dispositivos gráficos, adotada por grandes empresas como Microsoft, IBM, Intel, Mitsubishi entre outras (AZEVEDO, 2004; SHREINER et al., 2013). De código aberto, multiplataforma e com suporte a diversas linguagens, a OpenGL é amplamente utilizada para criação de imagens que vão de simples desenhos 2D a Jogos e Filmes, como: *Jurassic Park*, *Star Wars*, *Quake* e *Half-Life* (AZEVEDO, 2004). A OpenGL não possui funções nativas para criar janelas onde as imagens serão exibidas, por isso foi utilizada a ferramenta *Qt Design* para criar a interface de controle com janelas, botões e componentes que possibilitem a manipulação de eventos e objetos no ambiente virtual. Para gerenciar os eventos de desenho da *OpenGL* foi utilizada a classe *QGLWidget*, disponível após a instalação da ferramenta *QtCreator*.

#### 3.2 Animação

Amplamente utilizada por produtoras de filmes e jogos, os primeiros registros do que hoje chamamos de Animação foram encontrados em paredes egípcias, onde eram retratadas

as cenas de luta e adoração. Paul Roget Frenchman definiu em 1828 a animação como: a persistência da visão (AZEVEDO, 2004).

Liu e Zordan (2011, p. 1, tradução nossa) dizem que “A qualidade da experiência sentida por aqueles que participam de uma animação depende da credibilidade dos movimentos do personagem no contexto do ambiente virtual”<sup>1</sup>. Seguindo a descrição de Liu e Zordan, movimentar um personagem é mais que simplesmente modificar as coordenadas X,Y e Z de um corpo em um ambiente 3D. Neste trabalho, foi utilizada uma *engine*<sup>2</sup> de simulação física, capaz de gerar movimentos e reações contínuas, obedecendo às leis da física de maneira automática (GEIJTENBEEK; PRONOST, 2012).

### 3.3 Simulação Física

Neste trabalho foi utilizada a engine física *Open Dynamics Engine* (ODE), uma biblioteca gratuita desenvolvida por Russell Smith e outros colaboradores, que utiliza o conceito de corpos rígidos articulados para simular corpos dinamicamente. A ODE é rápida, robusta, flexível e com seu próprio detector de colisões e é utilizada para simular veículos terrestres, criaturas com pernas e mover objetos em ambientes virtuais (SMITH et al., 2005). Os corpos rígidos são representados por um corpo (*dBodyId*), utilizado para a dinâmica, e uma geometria (*dGeomId*), utilizada para colisões, sendo que este último possui diversos tipos disponíveis (caixa, cápsula, esfera, cilindro). Implementando separadamente a dinâmica e a colisão, ODE tem a vantagem de permitir que outra biblioteca de detecção de colisão seja utilizada. Mesmo sendo elementos independentes, é possível conectar uma geometria a um corpo. Assim, a posição e orientação da geometria vinculada é automaticamente atualizada de acordo com o corpo. É possível modificar diversas características de um corpo e/ou de uma geometria durante a execução da simulação.

### 3.4 Captura de Movimentos

A Captura de Movimentos (*Motion Capture*) permite que os movimentos de um ator sejam analisados e processados para produzir uma animação. Os movimentos podem ser armazenados e reproduzidos em um personagem (robô, braço mecânico ou um personagem virtual) independente do ator. Bastante utilizada na indústria de filmes e jogos, a captura de movimentos além de permitir o estudo dos movimentos e um maior realismo em simulações, pode diminuir riscos e custos quando aplicada em certos cenários como simulação de colisões ou movimentação de personagens de jogos ou filmes que contenham animação (AZEVEDO, 2004;

<sup>1</sup> The quality of the experience felt by the media participants often depends on the believability of the avatar’s motion in the context of the virtual environment.

<sup>2</sup> Conjunto de bibliotecas para abstrair a criação de aplicações gráficas em tempo real.

MENACHE, 2010).

Abaixo são descritos alguns conceitos complementares para o entendimento da Captura de Movimento neste trabalho.

- **Interface Natural:** Com o avanço da tecnologia tem-se tentado cada vez mais criar formas de interação homem-computador que possibilitem uma maior liberdade para o usuário de forma não-intrusiva. “Interface Natural é a capacidade de interagir com uma máquina utilizando nada além do corpo humano.”<sup>3</sup> (OVADIA, 2012, tradução nossa).
- **Microsoft Kinect:** Sensor inicialmente projetado para jogos, criado a partir do dispositivo PrimeSense (ver Figura 1). Ele posteriormente foi adquirido pela Microsoft para ser utilizado no Projeto Natal, e assim se tornou o que hoje conhecemos apenas como *Kinect*. Apesar de ter sido criado para ser utilizado em jogos da plataforma Xbox 360, não demorou para que pesquisadores percebessem o potencial que o dispositivo possuía para ser utilizado em pesquisas (CARDOSO, 2013). Em 2011, alguns meses após o seu lançamento, a Microsoft anunciou a primeira versão do SDK (*Software Development Kit*) para o *Kinect*, que podia ser obtido gratuitamente com suporte às linguagens C++ e C#, para a plataforma Windows (CARDOSO, 2013). Isso limitou por um certo tempo aqueles que utilizavam outros sistemas operacionais a terem acesso às funcionalidades do *Kinect*. Contando com um sensor infravermelho, é possível identificar uma pessoa e obter informações sobre suas articulações como: posição, distância e orientação em relação ao *Kinect*. Possui também uma câmera RGB, microfones (com os quais é possível identificar quem e de onde está falando), uma base que pode ter seu ângulo ajustado e um acelerômetro (ver Figuras 2 e 3).
- **Open Natural Interaction (OpenNI):** Desenvolvida pela mesma empresa que criou o dispositivo que mais tarde se tornaria o Kinect, a API OpenNI foi escolhida para ser utilizada neste projeto por suportar vários dispositivos de captura de movimento, inclusive o Kinect. Além do dispositivo, é possível utilizar um arquivo de vídeo no formato *.oni*<sup>4</sup> para reproduzir os movimentos, tornando o teste da aplicação mais prático ao remover a necessidade de um usuário modelo para captura de movimentos. Através da API é possível acessar as informações de posicionamento dos membros do corpo do usuário em tempo real permitindo controlar o humanoide no ambiente virtual. Atualmente é mantida

<sup>3</sup> Natural User Interface (NUI) is the ability to interact with a machine using nothing but the human body.

<sup>4</sup> Arquivo com gravação de movimentos utilizado para substituir o Kinect.

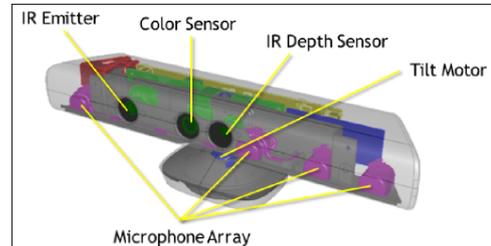
pela empresa Occipital, que conta com alguns dos originais parceiros da PrimeSense. Continuando como software livre, encontra-se na segunda versão (OpenNI, 2012).

Figura 1 – Sensor 3D PrimeSense



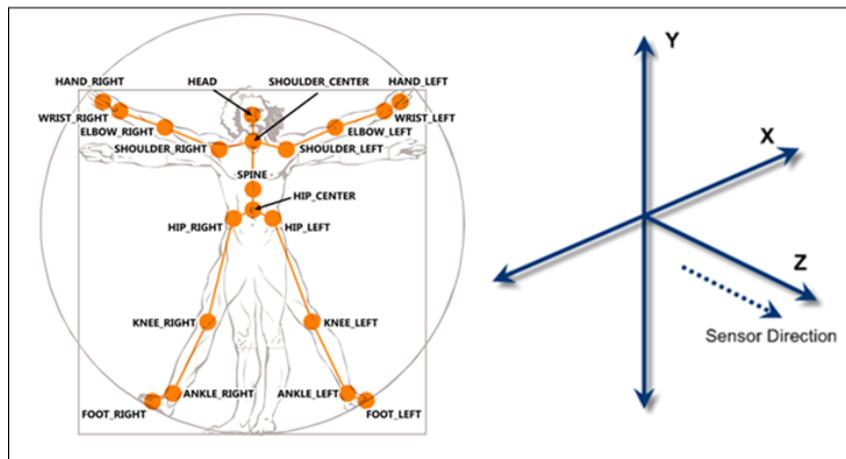
Fonte: <http://photos2.appleinsidern.com>

Figura 2 – Componentes do *Kinect*



Fonte: <https://msdn.microsoft.com>

Figura 3 – Articulações identificadas pelo *Kinect*



Fonte: <http://archive.renauddumont.be>

## 4 PROCEDIMENTOS METODOLÓGICOS

Nesta seção são descritas as etapas para o desenvolvimento deste trabalho. Na Seção 4.1 é descrito todo o processo de captura de movimentos do usuário. Na Seção 4.2 detalhes sobre a criação do personagem são esclarecidos. Na Seção 4.3 é descrito o processo de utilização de forças e torques para movimentar o personagem. Na Seção 4.4 é feita uma breve descrição dos elementos da tela utilizados no controle da simulação. Na Seção 4.5 são descritos os ambientes de interação utilizados para testar o personagem.

### 4.1 Captura de movimentos do usuário utilizando o Kinect

A captura de movimentos consiste em identificar o usuário, calibrar o usuário (medir todos os membros do corpo e calcular o peso estimado) e inserir o personagem no ambiente virtual. Neste projeto, foi utilizado o reconhecimento de pose, pois no momento da identificação do usuário, utilizar uma pose é importante para evitar que usuários não desejados (pessoas que venham a passar na frente do *Kinect* no momento da captura) sejam adicionados na interface como personagens válidos e disponíveis para serem calibrados. Foi utilizada a pose PSI (ver Figura 4).

Após ser identificado e adicionado como um usuário válido, é necessário calibrar o personagem. A versão utilizada do *Kinect* possui uma grande sensibilidade a ruídos na imagem da captura de movimentos, sendo necessário escolher bem o ambiente, de preferência sem superfícies reflexivas (ver Figura 5) e posicioná-lo em um ângulo paralelo ao chão. Um mau posicionamento do usuário no momento de calibrar ou uma grande quantidade de ruídos pode resultar na criação de um personagem deformado (ver Figura 6). A fim de evitar esse problema, o usuário deve permanecer em uma posição fixa para que o personagem possa ser criado com a representação adequada (sem deformações de tamanho dos membros, por exemplo). Após ser criado, o personagem é adicionado no ambiente de simulação como *inativo* e todos os membros<sup>1</sup> são posicionados lado a lado no chão, mas ainda não estão sendo desenhados.

Para atualizar o posicionamento do personagem no ambiente virtual, a comunicação entre *Kinect* e ODE foi implementada utilizando funções *callback* da plataforma *QT Creator*. Dessa forma as informações de posicionamento do personagem serão constantemente atualizadas na simulação, que se encarrega de todos os procedimentos necessários para que o personagem se movimente.

---

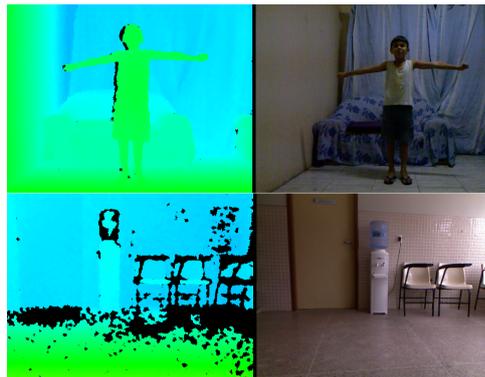
<sup>1</sup> Geometrias que compõem um personagem.

Figura 4 – Pose PSI para identificar personagem.



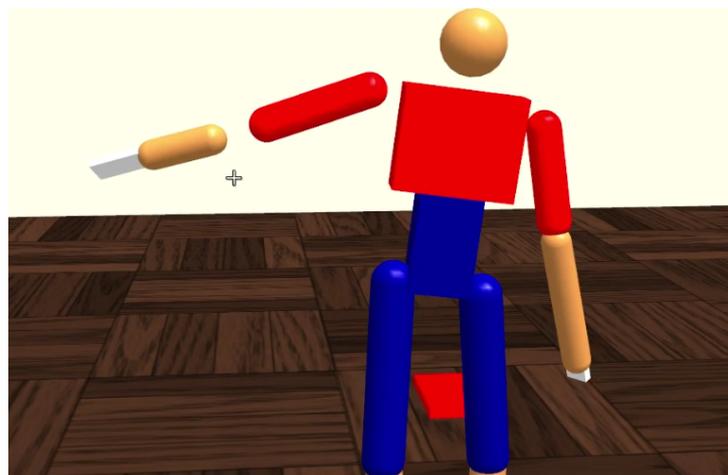
**Fonte:** Elaborada pelo autor.

Figura 5 – Ruído causado por superfície reflexiva.



**Fonte:** Elaborada pelo autor.

Figura 6 – Erro no momento de criar o personagem.



**Fonte:** Elaborada pelo autor.

## 4.2 Personagem fisicamente simulado

Para criar o personagem foi necessário inicialmente escolher as geometrias utilizadas na representação de cada membro, sendo assim escolhidas: esfera para a cabeça; caixa para tronco, mãos e pés; cápsula para braço, antebraço, coxa e perna. Uma vez definidas as geometrias, se tornou necessário encontrar uma forma simples de estimar o peso do personagem para que o personagem seguisse a proposta de representar características únicas do usuário. Neste projeto, foram utilizadas as fórmulas de Devine e Robinson, para homem e mulher, respectivamente (HALLS, 2016):

$$\text{Homem} = 50\text{kg} + 2,3\text{kg para cada } 2,54 \text{ cm acima dos } 152,4 \text{ cm} \quad (4.1)$$

$$\text{Mulher} = 49\text{kg} + 1,7\text{kg para cada } 2,54 \text{ cm acima dos } 152,4 \text{ cm} \quad (4.2)$$

As equações apresentam um cálculo do peso ideal para homens e mulheres. Contando com um único parâmetro (altura do usuário), e este sendo comprometido pelo ruído na captura de movimentos, foram ainda necessários alguns ajustes empíricos nas constantes das fórmulas na tentativa de melhorar a estimativa final. Os valores  $2,54 \text{ cm}$  e  $152,4 \text{ cm}$  foram simplificados para  $2,5 \text{ cm}$  e  $150 \text{ cm}$  respectivamente e a altura (obtida através do medição da distância entre a cabeça e o pé do personagem utilizando os dados do *Kinect*) teve um acréscimo de  $25 \text{ cm}$ , na tentativa de corrigir a falta de precisão do *Kinect*.

Os valores de posicionamento obtidos do *Kinect* precisam ser utilizados em escala reduzida para que sejam mostrados de maneira adequada na tela, e são mapeados através da expressão:

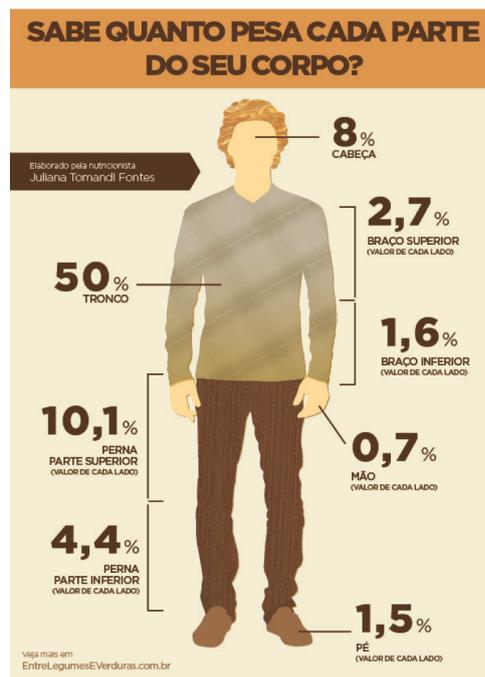
$$O_c = O_{cmin} + \left( \frac{K_c - K_{cmin}}{K_{cmax} - K_{cmin}} \right) (O_{cmax} - O_{cmin}) \quad (4.3)$$

na qual são definidos valores de máximo e de mínimo para os eixos  $X$ ,  $Y$  e  $Z$ , correspondentes aos limites dos volumes a serem considerados em cada ambiente (real e virtual). Esses volumes foram usados para definir o mapeamento. Considerando que as coordenadas 3D reais são fornecidas pelo *Kinect* e que as coordenadas 3D virtuais são utilizadas pelo ODE e pela OpenGL, diferenciamos essas informações pelas letras  $K$  e  $O$ , respectivamente. A letra  $c$  subscrita representa cada coordenada  $X$ ,  $Y$  e  $Z$ . A equação então mapeia as coordenadas reais  $K_c$  em coordenadas virtuais  $O_c$ . Um ponto real  $(K_x, K_y, K_z)$  é então mapeado em um ponto virtual  $(O_x, O_y, O_z)$ .

Para representar os membros do corpo na simulação física, foi utilizada a estrutura de corpos rígidos do ODE que possibilita ao personagem reagir a eventos externos do ambiente virtual. Neste ponto, todos os membros do usuário são medidos, e essas medidas são utilizadas

na criação de cada membro do personagem, exceto o raio, que foi definido empiricamente como um valor fixo. Uma vez que todos os corpos possuem suas dimensões, seu peso individual é calculado utilizando as proporções listadas na Figura 7. O personagem é então inserido no ambiente com o status **inativo**, esperando até que a captura de movimentos altere a posição dos corpos.

Figura 7 – Quanto pesa cada parte do corpo.

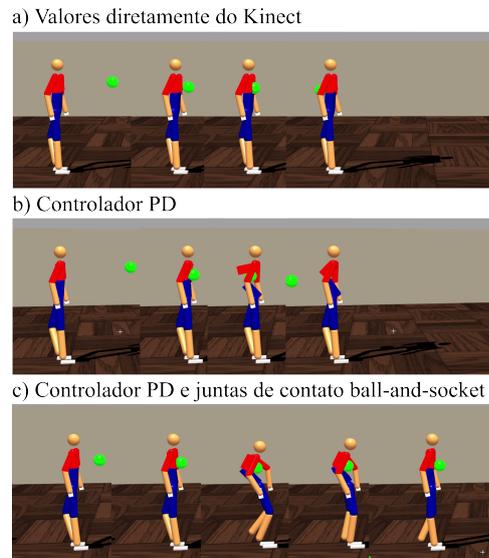


Fonte: Juliana Tomandi Fontes.

### 4.3 Movimentação do humanoide

Uma vez que o personagem está criado com suas medidas e massa ajustadas é possível inseri-lo no ambiente virtual para que possa ser desenhado, e os valores do *Kinect* passarão a ser utilizados para movimentá-lo. A maneira mais direta de movimentar o personagem é através da atualização direta da posição e da orientação de cada membro do personagem simulado, usando os valores obtidos da captura de movimentos. Essa abordagem cinemática (que não leva em conta as leis da dinâmica) no entanto, não permite que o personagem reaja a eventos externos do ambiente, não sendo assim adequada para testes de colisão e reação (ver item **a** da Figura 8). Por outro lado, por utilizar os valores originais da captura de movimentos, garantem uma maior precisão, ideal para situações como a configuração das juntas de contato, abordadas mais a frente, onde a precisão do posicionamento influencia a experiência final da simulação.

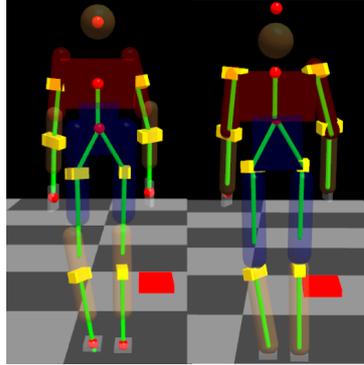
Figura 8 – Exemplo de colisões com e sem controlador PD e juntas.



**Fonte:** Elaborada pelo autor.

Como mencionado anteriormente os valores de posicionamento precisam ser mapeados antes de serem utilizados no personagem. Além desse mapeamento é necessário mencionar outro detalhe, agora relacionado a manter a consistência entre as informações do *Kinect* e do ODE. Uma vez que uma geometria é criada no ODE o centro da mesma passa a ser referência para translação e escala, ou seja, ao criar uma capsula com o valor 2 (dois) para o comprimento, a mesma vai medir a partir do centro 1(um) para cada lado. Já as informações de posição vindas do *Kinect* correspondem às posições das articulações, as quais se encontram nas extremidades dos corpos, o que causou uma certa inconsistência ao atualizar as posições dos corpos na simulação. O problema é que, mesmo sem resolver essas inconsistências, o desenho aparentava estar correto em várias circunstâncias. Ao tentar reproduzir o trabalho de (NUNES et al., 2008), foram notados esses problemas e aplicadas as devidas correções nas funções de desenho. Essa inconsistência foi corrigida ao atualizar cada corpo simulado no ODE com o ponto médio entre as duas juntas vizinhas vinculadas ao mesmo, em vez de usar a posição de uma única junta diretamente para atualizar o centro do corpo, como pode ser visto na Figura 9. Para os corpos nas extremidades, os quais não possuem duas juntas vinculadas (por exemplo, mãos e pés), um deslocamento a partir da única junta, escolhido empiricamente, foi acrescentado. Inconsistências entre *Kinect* e ODE relacionadas às orientações dos corpos também foram encontradas ao utilizar cápsulas como geometrias. Para corrigi-las, foi suficiente adicionar rotações de 90 graus às informações vindas do *Kinect*.

Figura 9 – Posicionamento direto (à esquerda), ponto médio (à direita).



**Fonte:** Elaborada pelo autor.

Como já mencionado, utilizar os valores do *Kinect* diretamente no personagem não é suficiente para criar um personagem que possa interagir com um ambiente virtual. Para que as reações físicas sejam simuladas automaticamente, as posições e orientações dos corpos devem ser atualizadas indiretamente de acordo com forças e torques aplicados com o objetivo de alcançar uma pose específica desejada. Para calcular essas forças e torques, foram utilizados controladores PD (*Proportional-Derivative*). Controladores PD lineares são responsáveis pela movimentação de um corpo da posição atual à posição desejada, corrigindo o erro entre estes estados atuando como uma mola linear (ver Figura 11) através da expressão:

$$f = \phi \cdot [k_s \cdot (P_d - P) - k_d \cdot (v)] \quad (4.4)$$

onde  $f$  é a força aplicada ao corpo,  $P$  é a posição atual,  $P_d$  é a posição desejada,  $v$  é a velocidade linear atual,  $k_s$  e  $k_d$  são as constantes de rigidez e amortecimento (neste projeto foi utilizado um par de valores para cada corpo),  $\phi$  é um fator escalar global utilizado para ter controle sobre todas as forças lineares dos corpos do personagem. De maneira semelhante, controladores PD angulares são responsáveis pelo controle da orientação (ver Figura 12) dos corpos através da expressão:

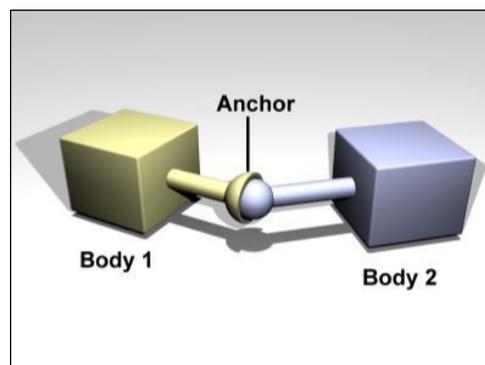
$$\tau = \psi \cdot [k_s \cdot (\theta_d - \theta) - k_d \cdot (\omega)] \quad (4.5)$$

onde  $\tau$  é o torque aplicado ao corpo,  $\psi$  é um fator escalar global utilizado para ter controle sobre todos os torques dos corpos do personagem,  $\theta_d$  é a orientação desejada,  $\theta$  a orientação atual e  $\omega$  a velocidade angular atual do corpo. Dessa forma o personagem já consegue interagir com o ambiente, embora de maneira ainda não apropriada, como mostra o item **b** da Figura 8. Como exemplificado na figura, ainda existe uma falha na reação do personagem: ele permanece sendo atravessado por objetos lançados contra seu corpo.

O ODE dispõe de uma variedade de juntas utilizadas para conectar corpos, dentre elas

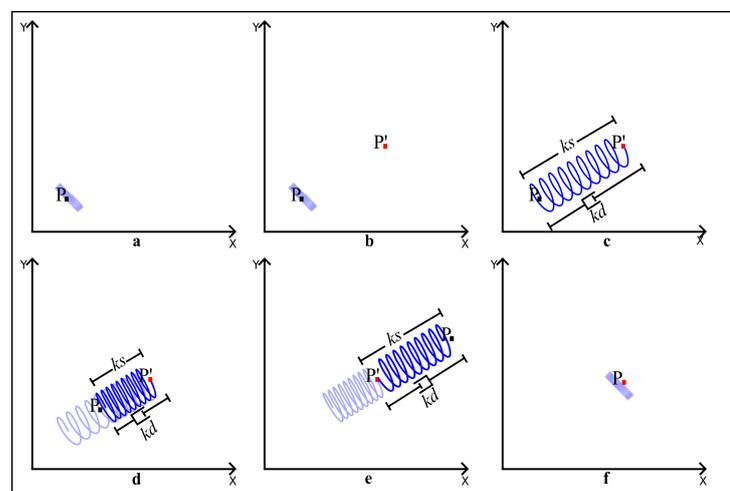
foi utilizada a junta do tipo *ball and socket* (ver Figura 10). Ao adicionar uma junta desse tipo conectando dois corpos a simulação garante que os mesmos permaneçam conectados. Isso possibilita a reprodução de um movimento mais realista quando o personagem é atingido por um objeto, como mostra o item **c** da Figura 8. Para melhorar a precisão do posicionamento das juntas, suas propriedades são atribuídas diretamente com os valores do *Kinect* enquanto o usuário permanece imóvel em uma posição que todos os membros estejam bem alinhados. Os valores  $k_s$  e  $k_d$  podem ser alterados na interface para tornar o personagem mais rígido ou mais flexível, mas é importante ressaltar que o sistema utiliza valores para personagens com diferentes distribuições de massa, e possibilita que os valores sejam salvos para reuso dos valores calibrados, dessa forma cada usuário tem seu próprio personagem (NUNES et al., 2008; NUNES, 2012).

Figura 10 – Junta *ball and socket*.



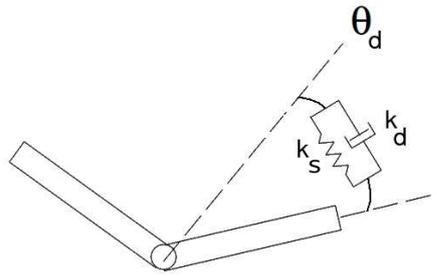
Fonte: (SMITH et al., 2005).

Figura 11 – Representação de uma mola do controlador PD



Fonte: Elaborada pelo autor.

Figura 12 – Mola com amortecimento agindo em uma junta.

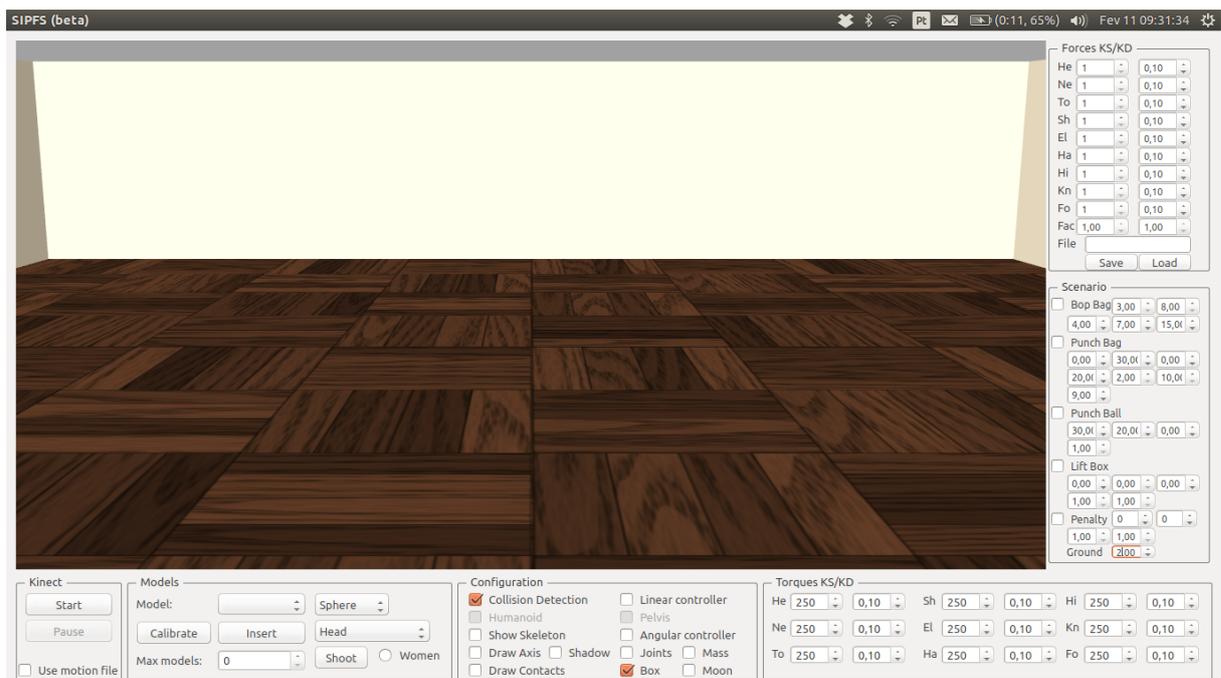


Fonte: (NUNES, 2012)

#### 4.4 Interface de controle

Para controlar o ambiente de interação e configurar o personagem, foi necessário criar uma interface rica em parâmetros de controle, possibilitando assim que cada usuário tenha seu personagem com os ajustes necessários para que a simulação seja a mais realista possível ou atingir determinado nível de personalização, de acordo com o usuário ou com a necessidade (ver Figura 13). O sistema foi todo implementado na linguagem C++ utilizando a IDE (*Integrated Development Enviroment*) QT Creator.

Figura 13 – Interface de controle.



Fonte: Elaborada pelo autor

Pensando na acessibilidade, os componentes da interface foram ordenados de maneira que podem ser acessados em ordem apenas utilizando a tecla TAB. Ideal, por exemplo, para ajustar os valores do controlador PD (ver Figura 14). Por uma limitação de espaço, alguns campos que indicam o nome do parâmetro foram removidos, outros apenas compactados, mas todos possuem uma mensagem (tooltip) indicando o nome do parâmetro (ver Figura 15). Também pensando no melhor uso da interface, os parâmetros de configuração dos objetos dos cenários foram dispostos de forma padronizada (Posição, Massa, Raio e Comprimento).

**Tela de desenho** Utilizada para projetar os desenhos de todos os objetos do cenário a partir do seu posicionamento e orientação no ambiente de simulação física.

**Kinect** Componentes para controlar o estado da captura de movimentos, podendo Iniciar, Pausar, Resumir e Parar. Também é possível substituir a captura de movimentos em tempo real por um arquivo .oni.

**Models** Componentes utilizados para controlar quantos personagens podem ser inseridos na simulação e para criar e inserir novos personagens. Com estes controles é possível selecionar um personagem para interagir lançando objetos em seu corpo para observar as reações.

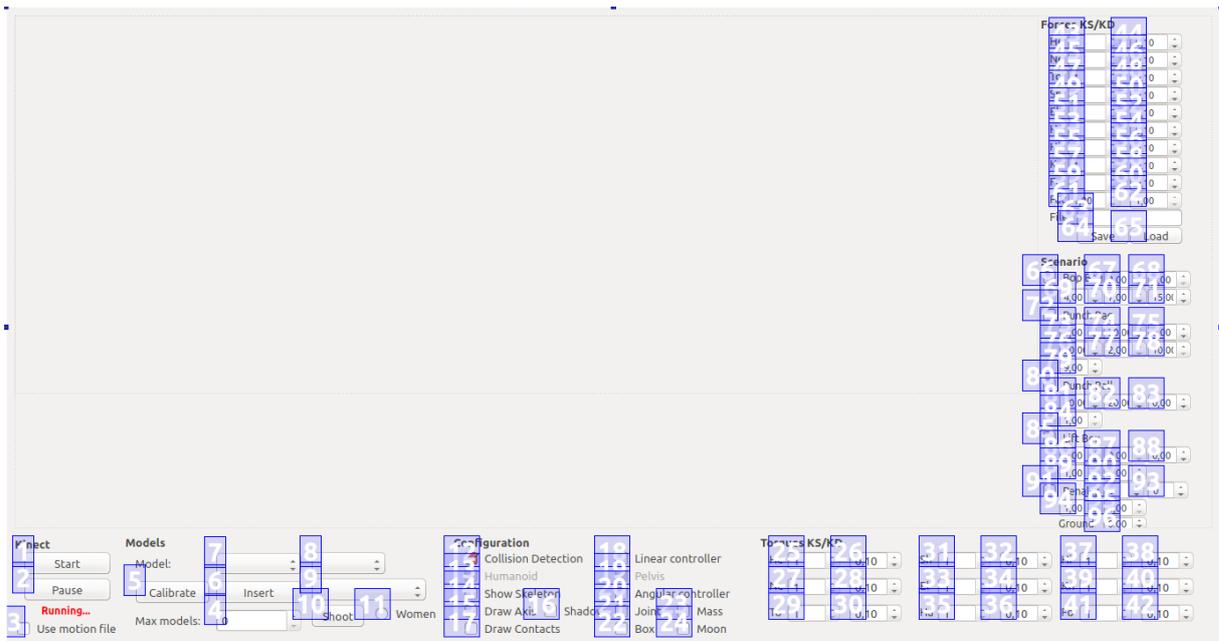
**Configuration** Configurações relacionadas ao ambiente de interação. É possível desativar a detecção de colisão, ativar os controladores PD, criar as juntas de contato para o personagem selecionado, alterar a distribuição de massa do personagem, mudar a gravidade, desenhar sombras, esqueleto, eixos e pontos de contato.

**Forces KS/KD** Aqui é possível configurar os valores de rigidez e amortecimento lineares de cada corpo de um personagem. Foi optado por adicionar um par de valores para cada par de corpos semelhantes, por exemplo, uma configuração de  $k_s$  e  $k_d$  para braço direito e esquerdo. Também é possível alterar o valor escalar  $\phi$  e salvar a configuração de todos os valores da interface em um arquivo no formato json permitindo assim que o ambiente seja totalmente configurado para um usuário.

**Torques KS/KD** Aqui é possível configurar os valores de rigidez e amortecimento angulares de cada corpo de um personagem. Foi optado por adicionar um par de valores para cada par de corpos semelhantes, por exemplo, uma configuração de  $k_s$  e  $k_d$  para braço direito e esquerdo.

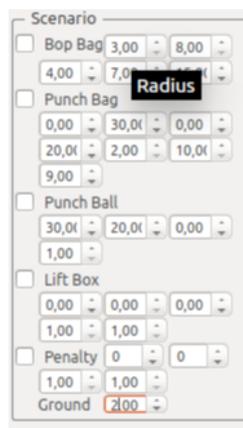
**Scenario** Por conter diversos cenários de interação, este grupo de controles foi adicionado para permitir ao usuário escolher os objetos de interação e modificar seus parâmetros para que melhor se adequem ao personagem durante a simulação.

Figura 14 – Configuração de usabilidade (acesso com uma tecla).



**Fonte:** Elaborada pelo autor

Figura 15 – Configuração de usabilidade (tooltip).



**Fonte:** Elaborada pelo autor

#### 4.5 Ambiente virtual de interação

Foram criados alguns cenários para testar a interação do personagem com o ambiente. Todos os objetos podem ter seus parâmetros alterados de acordo com a preferência do usuário

(ver Figura 16). Todos os cenários contam com uma implementação de câmera em primeira pessoa para proporcionar uma melhor imersão.

Figura 16 – Todos os objetos disponíveis no cenário de interação.



**Fonte:** Elaborada pelo autor

#### 4.5.1 Pênalti

Cenário composto por uma bola e uma trave. É possível ajustar as dimensões da bola para melhor se ajustarem ao personagem. O usuário pode interagir com os pés tentando chutar a bola e fazer o gol. A trave não está fixa, sendo assim o usuário pode derrubar a trave chutando a bola (ver Figura 17).

Figura 17 – Gol de letra.

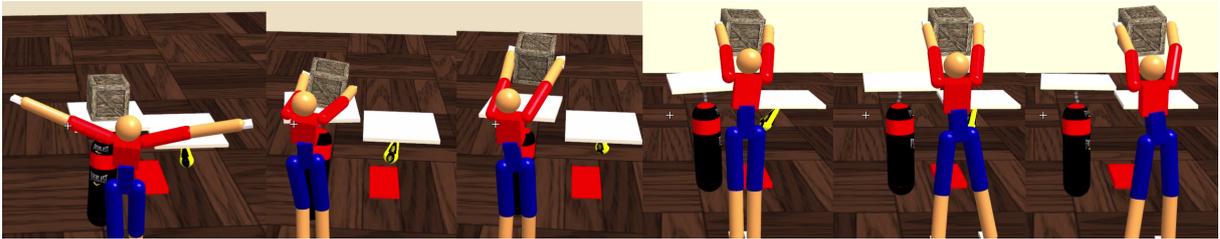


**Fonte:** Elaborada pelo autor

#### 4.5.2 Caixa

Cenário composto apenas por uma caixa. Neste cenário o usuário é livre para interagir com a caixa, tentando levá-la, movê-la de um ponto a outro, arremessá-la, etc (ver Figura 18).

Figura 18 – Personagem interagindo com a caixa.

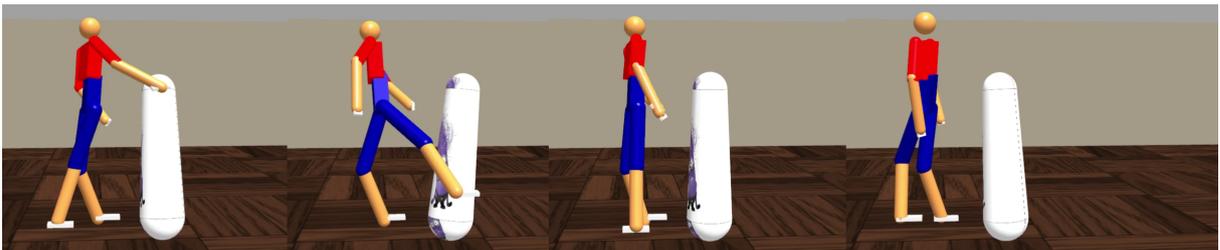


**Fonte:** Elaborada pelo autor

#### 4.5.3 João Teimoso

Inspirado no famoso brinquedo para crianças, este cenário conta com uma capsula posicionada na posição vertical. Foi utilizado um torque para mantê-la sempre em pé após receber um impacto (ver Figura 19).

Figura 19 – João Teimoso.

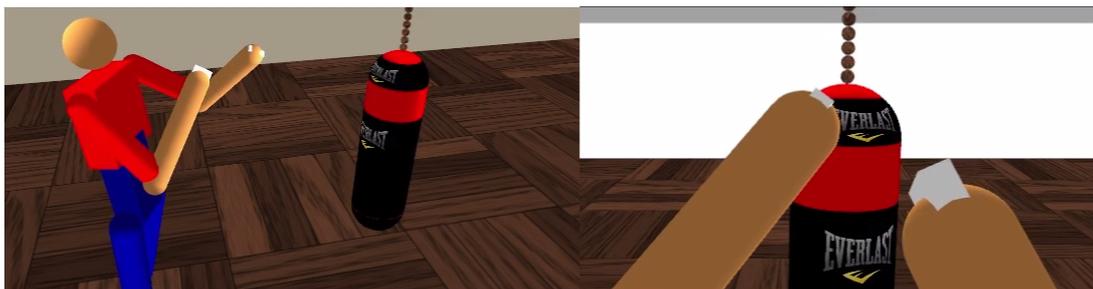


**Fonte:** Elaborada pelo autor

#### 4.5.4 Saco de Pancadas

Composto por uma base móvel, uma corrente formada por esferas ligadas por juntas *ball and socket* e uma cápsula. Este cenário consegue exemplificar a mudança na reação de um corpo com diferentes configurações de massa. É possível também alterar o tamanho da corrente e as dimensões do saco de pancadas (ver Figura 20).

Figura 20 – Saco de pancadas.



**Fonte:** Elaborada pelo autor

#### 4.5.5 Bola de Socos

Semelhante ao Saco de Pancadas, é composta por uma base móvel e uma esfera que a conecta à cápsula. Este cenário é ideal para testar a coordenação do usuário no controle do personagem para conseguir executar uma sequência de golpes sem perder o ritmo (ver Figura 21).

Figura 21 – Saco de pancadas.



**Fonte:** Elaborada pelo autor

## 5 RESULTADOS

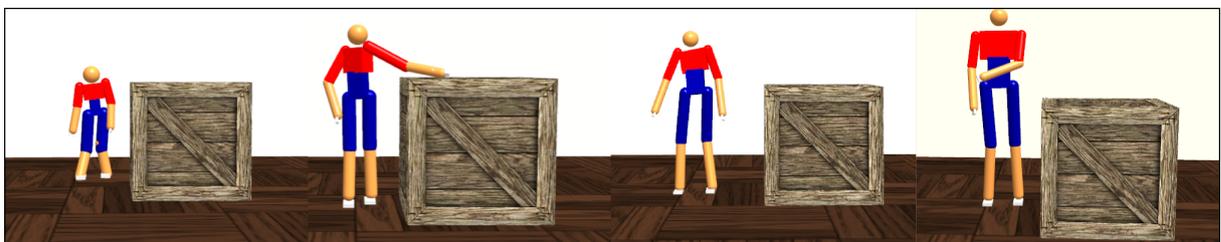
Nesta seção estão todos os resultados obtidos através da observação dos voluntários enquanto os mesmos utilizavam o sistema. Na Seção 5.1 são mostrados os resultados referentes ao protótipo de personagem fisicamente simulado. Na Seção 5.2 é explicado como os cenários de interação definidos foram utilizados pelos usuários. Na Seção 5.3 os resultados referentes a interface de controle e sua contribuição no processo de interação com o ambiente virtual.

### 5.1 Protótipo de personagem fisicamente simulado

O personagem criado se mostrou eficiente na representação das características do usuário. Foi fácil perceber as diferenças na estatura dos personagens mesmo quando não há um referencial próximo a eles. A ideia de que utilizar características únicas de cada usuário teria um impacto na simulação foi confirmada uma vez que ao utilizar a massa estimada através do cálculo da massa ideal, ou usuários de estaturas diferentes, se torna necessário ajustar os valores do controlador PD do personagem para que o mesmo possa ser controlado sem erros.

Os ajustes dos controladores do personagem foram eficientes até certo ponto, pois a configuração do passo da simulação<sup>1</sup> fez com que os valores de rigidez e amortecimento tivessem que ser muito elevados dificultando a precisão do ajuste. Por exemplo, em (SOUSA, 2014) o valor de rigidez dos membros do personagem permanece em um intervalo entre 60 e 250. Neste trabalho, os valores ultrapassam o valor de 1000. Sendo assim foi optado por diminuir a precisão do acompanhamento da captura de movimentos do mantendo uma boa representação do usuário no ambiente virtual através do personagem fisicamente simulado.

Figura 22 – Comparação de personagens diferentes.



**Fonte:** Elaborada pelo autor.

<sup>1</sup> Velocidade de atualização do personagem

## **5.2 Cenários de interação**

Os cenários descritos foram eficientes na sua proposta de possibilitar a interação e imersão do usuário. Cenários mais simples como Pênalti e João Teimoso foram utilizados de uma forma mais recreativa, sempre buscando o entretenimento. Os demais, exigindo um pouco mais de concentração, foram encarados como um jogo de desafios, como mover a caixa de uma plataforma para outra ou bater na bola de socos no momento correto para manter o ritmo.

Todos os cenários são flexíveis (dentro de suas limitações) para que possam ser ajustados a diferentes personagens ou de acordo com a preferência do usuário.

## **5.3 Interface de controle**

A interface teve um papel fundamental na execução do projeto. Seria difícil imaginar o controle e ajustes de um ambiente complexo como o apresentado sendo feito de maneira improvisada (utilizando o teclado, por exemplo), a quantidade de combinações possíveis de configurações teria um resultado totalmente negativo em uma simulação de tal nível. A disposição dos elementos na tela criou um fluxo de passos a serem seguidos, desde a criação até o ajuste dos valores de controle do personagem e do ambiente. Sendo também considerada como um componente da interface de controle por auxiliar no controle do personagem, a câmera em primeira pessoa possibilitou ao usuário um melhor referencial de espaço do seu personagem dentro do ambiente virtual.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho procurou demonstrar que uma vez que se utiliza características físicas de cada personagem para produzir um protótipo de interação fisicamente simulado, é possível observar diferentes reações do personagem (assim como do ambiente a sua volta) às forças aplicadas ao seu corpo, ao mesmo tempo em que ele segue movimentos capturados. Também foi identificado que as configurações de forças, rigidez e amortecimento de cada personagem são únicas, salvos os casos em que os usuários possuem um perfil físico semelhante, possibilitando assim o compartilhamento das configurações. Em qualquer outro caso o uso de configurações diferentes pode resultar no descontrole do personagem ou até mesmo na explosão da simulação (destruição de elementos do ambiente causado por erros numéricos).

Devido a limitações de espaço e uma certa imprecisão na captura dos movimentos ao utilizar o *Kinect*, os movimentos do personagem assim, como suas formas, nem sempre foram representados adequadamente. O valor utilizado para definir o tamanho do passo da simulação foi definido empiricamente, o que resultou em alguns erros, identificados somente em um estado avançado do desenvolvimento, mas que não impactaram diretamente na continuação do mesmo. Algumas funções utilizadas no desenho dos elementos da simulação não estavam de acordo com os parâmetros do ODE, e se fez necessário uma correção individual, para não impactar o uso das funções fora do escopo da simulação. A ausência de limitação do ângulo de movimento das juntas, assim como o tipo de junta escolhida para conectar os corpos permitiu que em alguns momentos os membros do corpo do personagem fossem dispostos em ângulos impossíveis de se atingir no corpo do usuário (joelho dobrado para trás, por exemplo).

Para realizar os testes foram selecionados apenas 4 voluntários entre homens e mulheres (crianças e adultos, altos e baixos) pois os ajustes feitos em cada personagem ainda não estavam como desejado, podendo assim gerar um falso resultado em relação ao controle do personagem. Durante os testes realizados na casa dos voluntários foi observado que, o protótipo abrange todos os requisitos definidos nos objetivos deste trabalho, uma vez que com pouca instrução todos os usuários conseguiram realizar as tarefas propostas, como por exemplo: mover a caixa, chutar a bola e bater no saco de pancadas. Assim como as reações do personagem e suas características físicas baseadas no usuário mostraram ter efeito direto sobre o ambiente em torno do personagem, afetando a interação e aumentando a imersão do usuário. Restando apenas o refinamento de sua implementação.

A seguir são mencionadas algumas ideias que ficaram como trabalhos futuros. Implemen-

tar a identificação do diâmetro dos membros do corpo para que se possa criar um personagem mais fiel à fisionomia do usuário. Identificar um valor de passo do ODE que possa ser utilizado para diminuir o valor de controle ( $k_s$  e  $k_d$ ) dos corpos. Utilizar juntas do tipo *hinge* (dobradiça) para proporcionar uma maior limitação nos movimentos do personagem e consequentemente melhorar o realismo. Implementar o controle do personagem utilizando a pélvis para compensar todos os torques gerados nas juntas. Identificar automaticamente os valores de rigidez e amortecimento de um personagem. Adicionar mais personagens na mesma simulação permitindo a interação entre si. Utilizar realidade aumentada para criar um cenário mais imersivo. Automatizar o processo de criação do personagem. Utilizar o *Kinect One* para obter uma captura de movimentos mais precisa. Salvar os movimentos do usuário em um arquivo e reproduzi-los posteriormente utilizando um arquivo *.mot* e otimizar a interface para permitir utilizar mais opções de controle. Realizar testes com mais usuários e aplicar um questionário sobre a usabilidade do sistema e a interação com o ambiente virtual através do mesmo.

O código fonte deste trabalho encontra-se disponível para todos aqueles que desejem estudá-lo: <https://bitbucket.org/italopessoa/ipfs/>

## REFERÊNCIAS

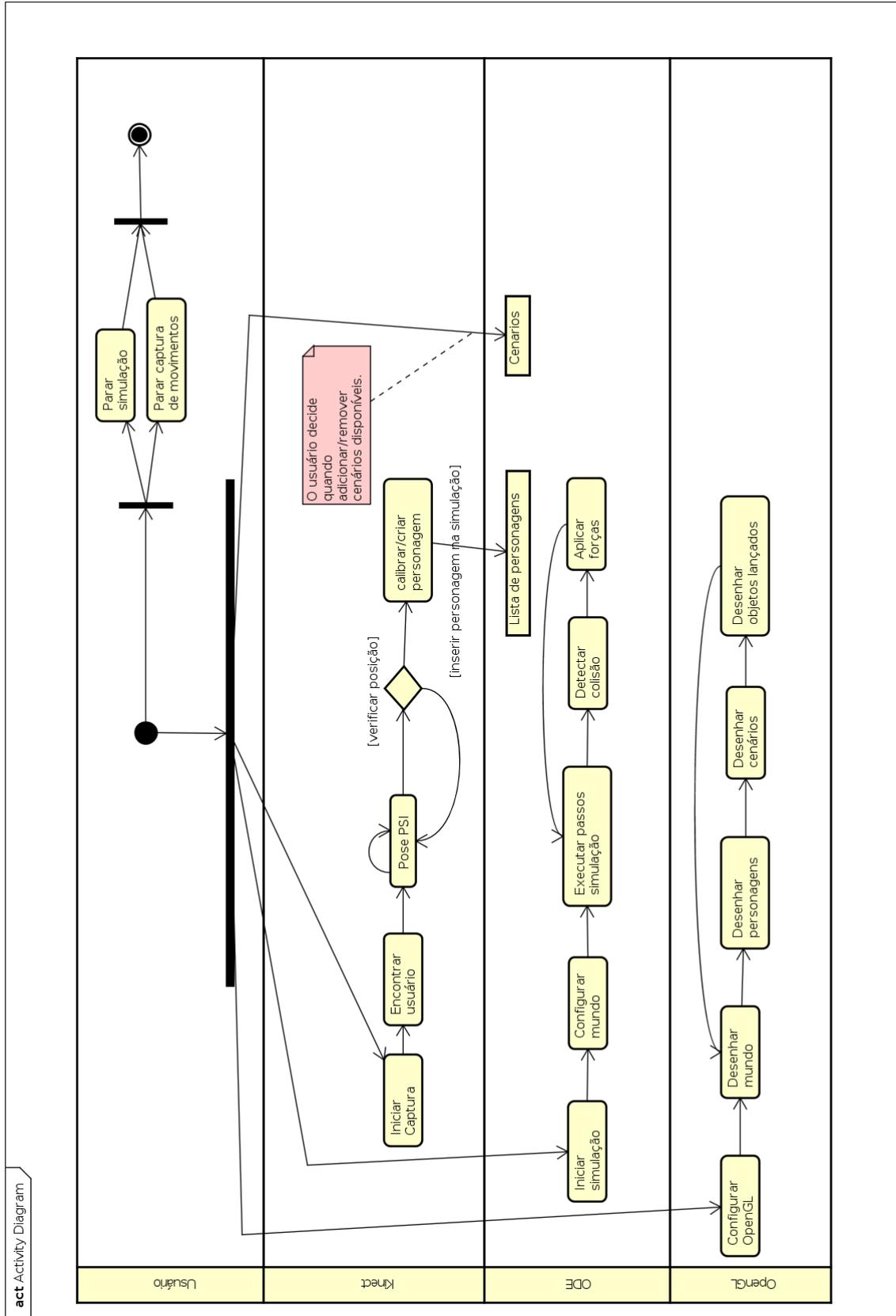
- AZEVEDO, E. *Computação Gráfica*. Elsevier Acadêmico, 2004. ISBN 8535212523. Disponível em: <<http://www.elsevier.com.br/site/produtos/Detalhe-produto.aspx?evp=VolAulas15.1&tid=1484>>.
- CARDOSO, G. S. *Microsoft Kinect. Criando aplicações interativas com o Microsoft Kinect*. [S.l.]: Casa do Código, 2013.
- GEIJTENBEEK, T.; PRONOST, N. Interactive character animation using simulated physics: A state-of-the-art review. *Comp. Graph. Forum*, John Wiley & Sons, Inc., New York, NY, USA, v. 31, n. 8, p. 2492–2515, 2012. ISSN 0167-7055. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2012.03189.x>>.
- GOMES, J. de M.; VELHO, L. *Fundamentos da computação gráfica*. [S.l.]: Impa, 2008.
- HALLS, D. *Ideal weight formulas by Broca and Devine (About arithmetic formulas for calculating ideal weight)*. 2016. Disponível em: <<http://halls.md/ideal-weight-formulas-broca-devine/>>.
- LIU, C.; ZORDAN, V. Natural user interface for physics-based character animation. In: ALLBECK, J.; FALOUTSOS, P. (Ed.). *Motion in Games*. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 7060). p. 1–14. ISBN 978-3-642-25089-7. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-25090-3\\_1](http://dx.doi.org/10.1007/978-3-642-25090-3_1)>.
- MANSSOUR, I. H.; COHEN, M. Introdução à computação gráfica. *RITA*, v. 13, n. 2, p. 43–68, 2006.
- MENACHE, A. *Understanding Motion Capture for Computer Animation, Second Edition (Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 2010. ISBN 0123814960. Disponível em: <<http://www.elsevier.com/books/understanding-motion-capture-for-computer-animation/menache/978-0-12-381496-8>>.
- NUNES, R. F. *Usando Vibrações Naturais na Descrição e no Controle de Locomoções Fisicamente Simuladas de Personagens Articulados Arbitrários*. Tese (Doutorado) — Universidade Federal do Ceará, 2012.
- NUNES, R. F. et al. Simulando reações flexíveis em movimentos capturados. In: *X Symposium on Virtual and Augmented Reality*. [S.l.: s.n.], 2008. p. 241–250.
- OpenNI. 2012. <<http://en.wikipedia.org/wiki/OpenNI>>. Acesso em: 15 abr 2015.
- OVADIA, D. *Natural User Interface (NUI) in a Nutshell*. 2012. <<http://www.onysus.com/natural-user-interface-nui>>. Acesso em: 14 abr 2015.
- SHREINER, D. et al. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 (8th Edition)*. [S.l.]: Addison-Wesley Professional, 2013. ISBN 978-0-321-77303-6.
- SHUM, H.; HO, E. S. Real-time physical modelling of character movements with microsoft kinect. In: *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*. New York, NY, USA: ACM, 2012. (VRST '12), p. 17–24. ISBN 978-1-4503-1469-5. Disponível em: <<http://doi.acm.org/10.1145/2407336.2407340>>.
- SMITH, R. et al. Open dynamics engine. 2005.

SOUSA, P. H. A. Q. de. *Interação com um personagem fisicamente simulado utilizando visão computacional e realidade aumentada*. 2014. Universidade Federal do Ceará. Projeto de Pesquisa.

SUTHERLAND, I. E. Sketch pad a man-machine graphical communication system. In: ACM. *Proceedings of the SHARE design automation workshop*. [S.l.], 1964. p. 6–329.

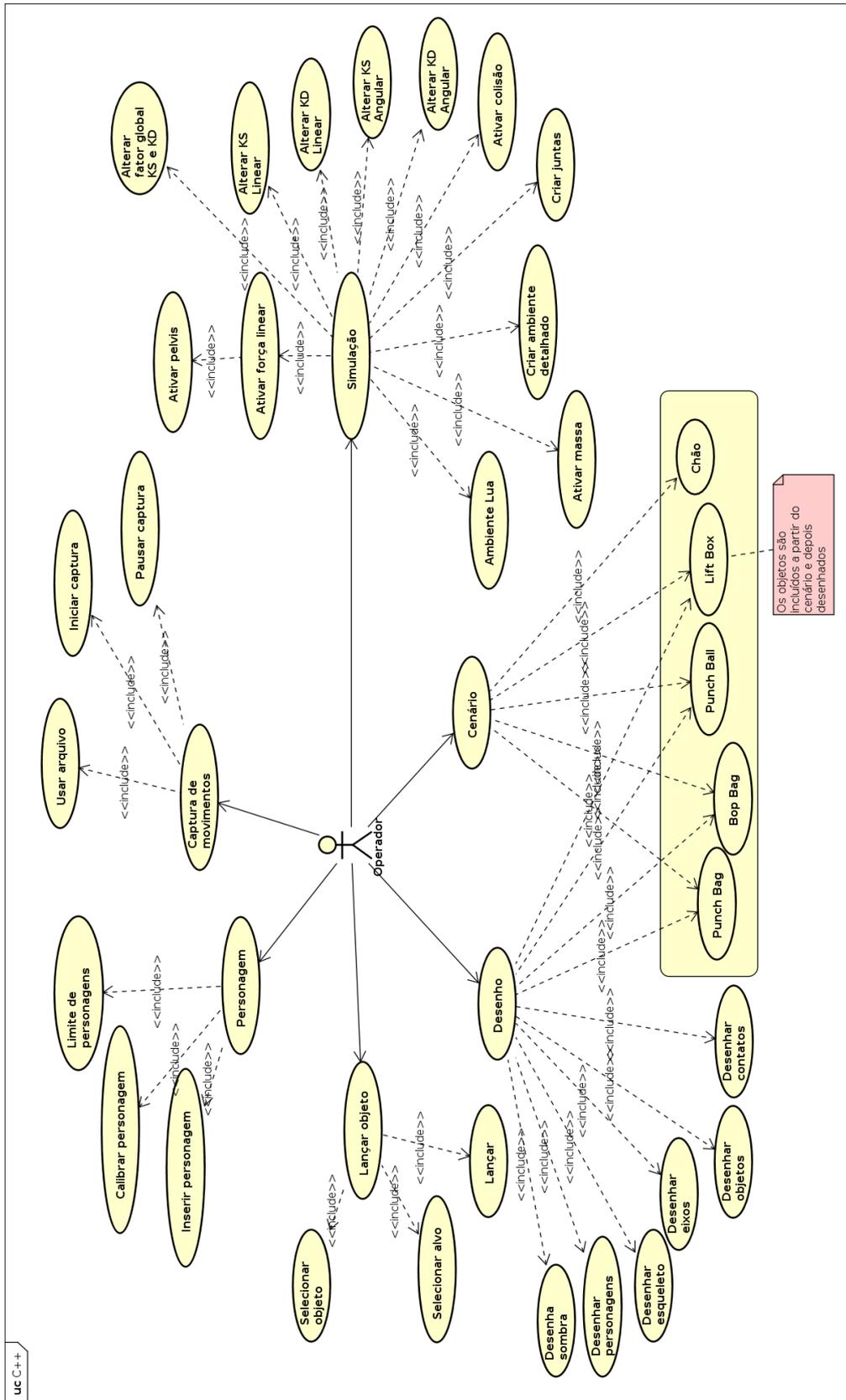
ZUHER, F.; ROMERO, R. Recognition of human motions for imitation and control of a humanoid robot. In: *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*. [S.l.: s.n.], 2012. p. 190–195.

ANEXO A - DIAGRAMA DE ATIVIDADES



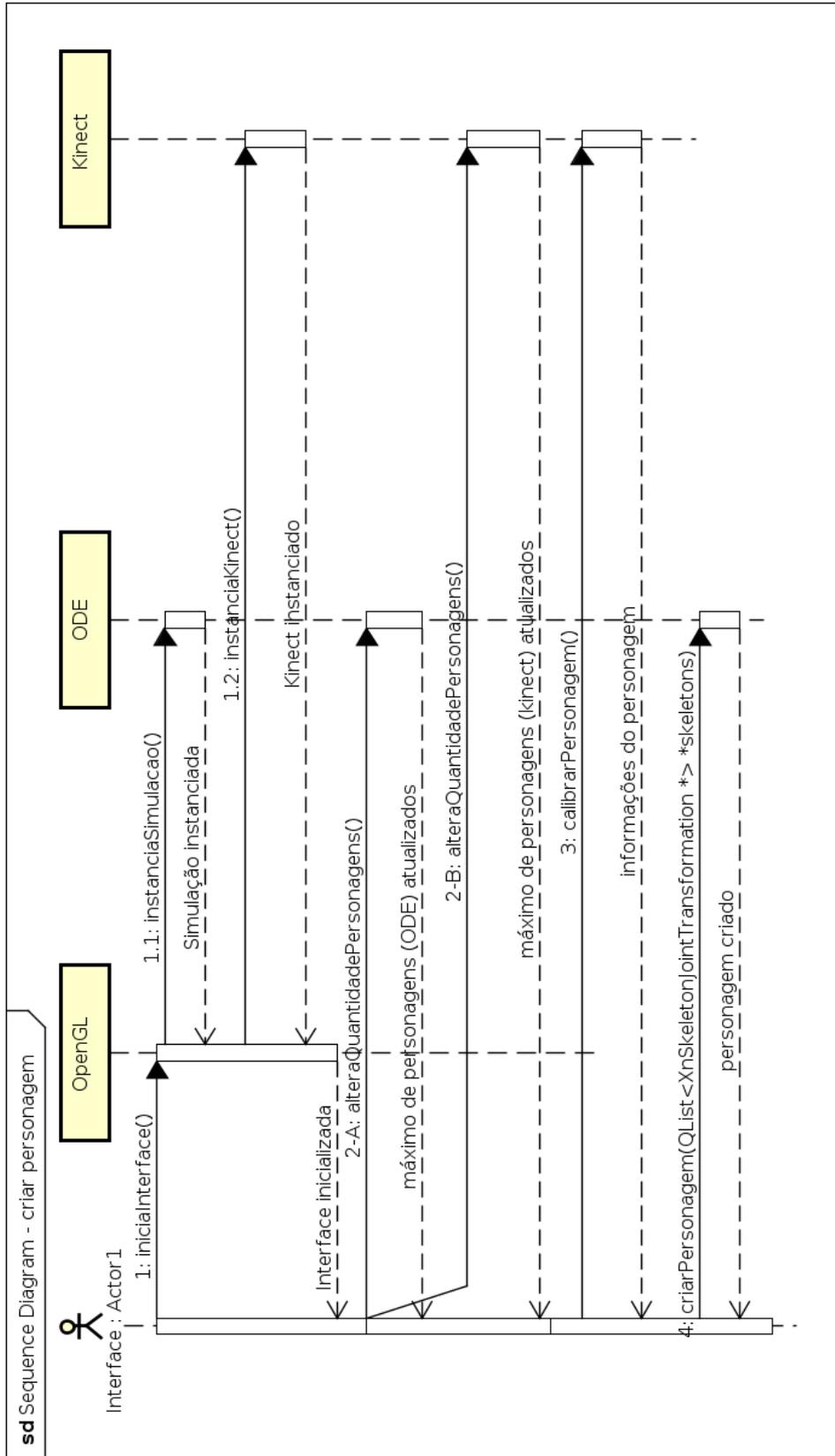
Fonte: Elaborada pelo autor.

ANEXO B - DIAGRAMA DE CASO DE USO



Fonte: Elaborada pelo autor.

ANEXO C - DIAGRAMA DE SEQUÊNCIA PARA CRIAR UM PERSONAGEM



Fonte: Elaborada pelo autor.