

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANTONIO JOSÉ GRANDSON BUSSON

***CONTROLE DE PERSONAGENS SIMULADOS FISICAMENTE  
USANDO SINAIS DO WIIMOTE: jogo master and slave***

São Luís  
2012

**ANTONIO JOSÉ GRANDSON BUSSON**

***CONTROLE DE PERSONAGENS SIMULADOS FISICAMENTE  
USANDO SINAIS DO WIIMOTE: jogo master and slave***

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, para obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Anselmo Cardoso de Paiva.

São Luís  
2012

Busson, Antonio José Grandson.

Controle de personagens simulados fisicamente usando sinais do wiimote: jogo master and slave / Antonio José Grandson Busson .- 2012.

42f.

Impresso por computador (Fotocópia).

Orientador: Anselmo Cardoso de Paiva.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2012.

1. Interface – Computação 2. Wiimote 3. Realidade aumentada 4. Simulação física 5. Acelerômetro I. Título.

CDU 004.451.84

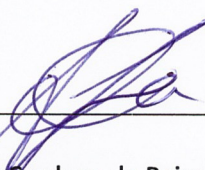
ANTONIO JOSÉ GRANDSON BUSSON

**CONTROLE DE PERSONAGENS SIMULADOS FISICAMENTE  
USANDO SINAIS DO WIIMOTE: *jogo master and slave***

Monografia apresentada ao Curso de Ciência da  
Computação da Universidade Federal do Maranhão,  
para obtenção do grau de Bacharel em Ciência da  
Computação.

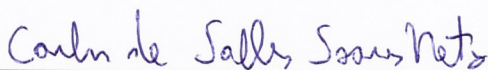
Aprovada em 04/10/2012.

Banca Examinadora



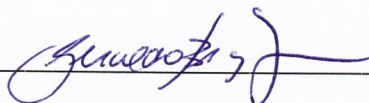
---

**Prof. Anselmo Cardoso de Paiva** (Orientador)  
Doutor em Ciência da Computação  
Universidade Federal do Maranhão



---

**Prof. Carlos de Salles Soares Neto**  
Doutor em Ciência da Computação  
Universidade Federal do Maranhão



---

**Prof. Geraldo Braz Junior**  
Mestre em Engenharia Elétrica  
Universidade Federal do Maranhão

*Aos meus pais e irmão.*

*Aos amigos, pelo apoio e companheirismo.*

## **RESUMO**

As novas experiências interativas em tempo real com o uso de controladores especiais, como o Wiimote, permitem inúmeras possibilidades, principalmente no espaço virtual. O uso da realidade aumentada potencializa a capacidade de visualização, pois apresenta este espaço virtual ao usuário de forma mais realística e amigável. Neste trabalho, apresenta-se o desenvolvimento de uma nova interface a partir de uma metodologia computacional que reconheça movimentos referentes as leituras do acelerômetro ( Wiimote), para reproduzi-los sob a forma de um avatar virtual em um jogo com ambiente de Realidade Aumentada.

Palavras-chave: Wiimote. Realidade aumentada. Simulação física. Acelerômetro.

## **ABSTRACT**

The new interactive experiences in real time with the use of special controllers, like the Wiimote, enable numerous possibilities, specially in virtual space. The use of augmented reality enhance visualization skills because present this virtual space in a more user friendly and realistic way. In this work, we develop a new interface based on a computational methodology that recognizes movements associated to the readings from the accelerometer (Wiimote) to represent them in the form of an virtual avatar in a game with Augmented Reality environment.

Keywords: Wiimote, Augmented reality. Simulation physics. Accelerometer.

# **AGRADECIMENTOS**

A minha família, pelo encorajamento e apoio.

Ao professor Anselmo Cardoso de Paiva pela orientação, e pela paciência, sem a qual este trabalho não se realizaria.

Ao professor Alexandre César Muniz de Oliveira pelos anos de tutoria no Programa de Educação Tutorial ( PET ).

Aos professores Carlos Salles e Geraldo Braz pelo apoio e por aceitar o convite para compor a banca examinadora deste trabalho de monografia.

Aos demais professores do Departamento de Informática pelos seus ensinamentos que contribuíram para nosso enriquecimento pessoal e profissional.

Aos amigos e colegas de graduação pelos anos de companheirismo, especialmente a Ruy Oliveira, André Felipe, Tiago Ribeiro, Raphael Gomes, Paulo Jansen e Júlio Filho.



*“Ninguém baterá tão forte quanto a vida. Porém, não se trata de quão forte pode bater, se trata de quão forte pode ser atingido e continuar seguindo em frente. É assim que a vitória é conquistada.”*

*Rocky Balboa*

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>2. FUNDAMENTAÇÃO TEORICA E TECNOLOGIAS UTILIZADAS .....</b>	<b>14</b>
<b>2.1 <i>Wiimote</i> .....</b>	<b>14</b>
<b>2.2 <i>Acelerômetro</i> .....</b>	<b>15</b>
<b>2.3 <i>Filtro de Kalman</i> .....</b>	<b>16</b>
<b>2.4 <i>Realidade Aumentada</i> .....</b>	<b>18</b>
<b>3 METODOLOGIA PARA SIMULAÇÃO FÍSICA DE UM PERSONAGEM VIRTUAL.....</b>	<b>20</b>
<b>3.1 <i>Aquisição</i>.....</b>	<b>20</b>
<b>3.2 <i>Filtragem</i> .....</b>	<b>22</b>
<b>3.3 <i>Classificação de movimentos</i> .....</b>	<b>24</b>
<b>3.3.1 <i>Posição dos membros superiores</i> .....</b>	<b>24</b>
<b>3.3.2 <i>Análise da ação do usuário com relação aos membros inferiores</i>.....</b>	<b>26</b>
<b>3.4 <i>Visualização</i> .....</b>	<b>27</b>
<b>4 JOGO DESENVOLVIDO .....</b>	<b>29</b>
<b>4.1 <i>Funcionalidades da aplicação</i> .....</b>	<b>29</b>
<b>4.2 <i>Entidades modeladas</i>.....</b>	<b>33</b>
<b>4.3 <i>Detalhamento das principais funcionalidades</i> .....</b>	<b>33</b>
<b>4.3 <i>Interação</i>.....</b>	<b>34</b>
<b>4.4 <i>Interface</i> .....</b>	<b>35</b>
<b>4.5 <i>Mecânica do jogo</i>.....</b>	<b>35</b>
<b>4.6 <i>Implantação</i> .....</b>	<b>36</b>

**5 CONCLUSÃO..... 37**

**REFERÊNCIAS ..... 38**

## LISTA DE FIGURAS

2.1 Acelerômetro do <i>Wiimote</i> .....	15
2.2 Uso de interação por controle virtual .....	18
2.3 Diagrama do sistema de visão de vídeo baseado em monitor .....	19
3.1 Módulos da aplicação .....	20
3.2 LEDs do <i>Wiimote</i> .....	21
3.3 Caixa de informação do <i>Wiimote</i> .....	23
3.4 Sinais da aceleração original .....	24
3.5 Sinais da aceleração após o <i>Filtro de Kalman</i> .....	24
3.6 Aceleração do <i>Wiimote</i> no eixo X segundo sua orientação .....	25
3.7 <i>Wiimote</i> acoplado aos braços .....	25
3.8 <i>Wiimote</i> acoplado as pernas do usuário .....	26
3.9 Ondas da aceleração dos <i>Wiimotes</i> dos membros inferiores.....	26
3.10 Personagem virtual com destaque aos membros coloridos .....	27
3.11 Plataforma de apoio .....	27
4.1 Diagrama de caso de uso da aplicação .....	29
4.2 Diagrama de classes .....	33
4.3 Diagrama de sequência – Movimentar .....	34
4.4 Diagrama de sequência .....	34
4.5 Tela principal do jogo .....	35
4.6 Modelo de implantação .....	36

## **LISTA DE TABELAS**

3.1 Codificação dos LEDs.....	22
4.1 Descrição de caso de uso – Conectar Wiimote .....	30
4.2 Descrição de caso de uso – Pausar jogo .....	30
4.3 Descrição de caso de uso – Começar Jogo .....	30
4.4 Descrição de caso de uso – Mover braço direito .....	31
4.5 Descrição de caso de uso – Mover braço esquerdo .....	31
4.6 Descrição de caso de uso – Pular .....	31
4.7 Descrição de caso de uso – Andar .....	32
4.8 Descrição de caso de uso – Chutar .....	32
4.9 Relação de botões do joystick com a movimentação do personagem.....	35

# 1 INTRODUÇÃO

A experiência de jogar vídeo game permaneceu inalterada durante muitos anos, até o lançamento do *Wii mote* (NINTENDO, 2006). Um dispositivo que possui um acelerômetro de três eixos, um giroscópio e uma interface de comunicação via *Bluetooth*. Este tipo de controlador permite que tanto desenvolvedores independentes como comerciais possam criar aplicações inovadoras. Hoje as experiências interativas em tempo real com o uso desse tipo de controlador permitem inúmeras possibilidades, principalmente no espaço virtual, pois um avatar virtual tem a vantagem de oferecer potencial a reações naturais. Personagens Simulados Fisicamente são mais difíceis de controlar em uma interface com joystick tradicional (SHIRATORI; HODGINS, 2008). Além disso, esta tecnologia também pode ser usada para a captura de movimentos.

Ao longo dos tempos, desde o primeiro curta-metragem todo feito com desenho animado em 1908, *Fantasmagoire*, de Emile Cohl, a prática para a animação de personagens vem sendo aprimorada graças ao grande interesse das indústrias cinematográficas e de jogos, na maioria das vezes o movimento fiel é indispensável para a aceitação pública, uma boa abordagem para solucionar este problema seria a simulação física dinâmica que garante movimentos bastante realistas, o desenvolvimento de uma interface de captura de movimento é algo muito caro, mas graças a fácil acessibilidade que o *Wii mote* oferece podemos criar uma aplicação similar a um preço bastante reduzido.

Também é de interesse das indústrias interativas a potencialização da capacidade de visualização e dinamismo das animações, a Realidade Aumentada é uma tecnologia que pode suprir estes requisitos. Com o uso do computador, a multimídia e a realidade virtual eliminaram algumas restrições, dando maior flexibilidade aos jogos, a revolução tecnológica propiciou a viabilização da Realidade Aumentada. A sobreposição de elementos virtuais em cenários reais, captados por câmeras. Mostrou-se promissor, sua integração neste setor está ganhando notoriedade devido ao seu diferencial de interação e as novas representações dos elementos virtuais.

Outros dispositivos também já foram utilizados para captura de movimentos, até mesmo um dispositivo comum como o mouse. Em (LASZLO; PANNE; FIUME, 2000)

foi desenvolvido um sistema interativo para controlar fisicamente um personagem através do mapeamento de movimentos do mouse. Em (NEFF; ALBRECHT; SEIDEL, 2007) utilizando mapas de correlação, permitiram ao usuário controlar múltiplas articulações de um robô simultaneamente, através do movimento do mouse.

Tablets também já foram utilizados para controlar personagens animados. Em (THORNE; BURKE; PANNE, 2007) foi usado este dispositivo para criar um personagem e animá-lo desenhando traços pré-definidos. O sistema tinha primitivas de movimentos pré-definidas que foram modificadas por propriedades como a velocidade. Em (OSHITA, 2004) foi construído uma interface baseada em uma caneta para o controle de animação, usaram o cursor da caneta para especificar a trajetória, a pressão para o salto e a inclinação da caneta para a altitude do personagem.

Sistemas interessantes também foram desenvolvidos utilizando dispositivos únicos. Johnson (JOHNSON et al., 1999) introduziu um novo conceito de interface, que usava um brinquedo de pelúcia embutido com um acelerômetro e um giroscópio, que reconhecia gestos a partir dos dados desses sensores. Em (YIN; PAI, 2003) desenvolveram o sistema FootSee que media a pressão entre o pé e o chão, e é usado para reconstruir movimentos com base em movimentos pré-existentes no bando de dados. Destacamos especialmente o trabalho (SHIRATORI; HODGINS, 2008), que propôs uma nova interface para o controle de personagens fisicamente simulados utilizando o *Wii mote*.

Neste trabalho, vamos desenvolver uma metodologia computacional que reconheça movimentos referentes às leituras do acelerômetro (*Wii mote*), para reproduzί-los sob a forma de um avatar virtual em um ambiente de realidade aumentada. A interface proposta consiste em acoplar um *Wii mote* em cada membro superior e inferior, para permitir que o usuário possa controlar um personagem dinâmico que pode correr, saltar, levantar cada perna e mover os braços. Este tipo de interface tem um maior desempenho se comparada ao joystick clássico, pois exige que o usuário imite padrões de locomoção que vemos na vida diária.

De maneira mais específica pretendemos:

- Capturar os sinais do acelerômetro;
- Utilizar um filtro de suavização para remover ruídos das leituras captadas;
- Reconhecer e classificar movimentos;

- Desenvolver um jogo com um ambiente de Realidade Aumentada;

A apresentação detalhada dos aspectos abordados neste trabalho esta distribuída do seguinte modo:

No Capítulo 2, Fundamentação Teórica e Tecnologias Utilizadas, segue informações importantes para o contexto e entendimento do trabalho.

No Capítulo 3, Metodologia para simulação física de um personagem virtual, será explicada a metodologia utilizada como ponto de partida para o desenvolvimento desse trabalho.

No Capítulo 4, Jogo desenvolvido, mostra-se o jogo desenvolvido, junto a sua modelagem.

No Capítulo 5, Conclusão, apresenta-se os resultados alcançados.



## 2 FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS

Nessa seção, abordaremos os conceitos e tecnologias necessárias para entender o funcionamento da metodologia computacional proposta.

### 2.1 *Wii mote*

O *Wii mote*, é o controle principal do Console Wii (NINTENDO, 2006). Possui dois sensores: uma câmera que captura a luz infravermelha emitida por dois pontos na *Sensor Bar*, e acelerômetros, que permitem obter a aceleração do dispositivo nos três eixos. O *Wii mote* processa a sua entrada e envia ao computador por meio de ondas de radio a partir de um chip *Bluetooth*. Também usa um sensor infravermelho para detectar a luz que vem do *Sensor Bar*, esta luz é como um sinal de retorno que auxilia o *Wii mote* a encontrar-se no espaço.

O controle também se comunica com o usuário através de um pequeno auto falante que reproduz sons e um pequeno motor que pode vibrar o dispositivo. Quatro pequenos diodos emissores de luz também podem ascender, um pedaço de plástico translucido leva essa luz até a parte frontal do controle para que o usuário possa vê-lo.

O *Wii Motion Plus* (NINTENDO, 2009) é um acessório adicional do *Wii mote*. Este periférico ao ser plugado na parte inferior do controle aumenta a precisão de movimentos de um para um, assim garantindo uma reprodução fiel dos movimentos do jogador ao invés de responder a comandos definidos anteriormente. O *Wii Motion Plus* possui uma passagem de conector para extensão externa, permitindo que outros periféricos sejam usados simultaneamente.

O que torna o *Wii mote* especial é o seu acelerômetro. O acelerômetro permite que você interaja com o videogame balançando ou girando o controle. As informações referente ao funcionamento do Acelerômetro seguem no próximo tópico.

## 2.2 Acelerômetro

O acelerômetro é um dispositivo capaz de medir a alteração da velocidade ao longo de um período de tempo.

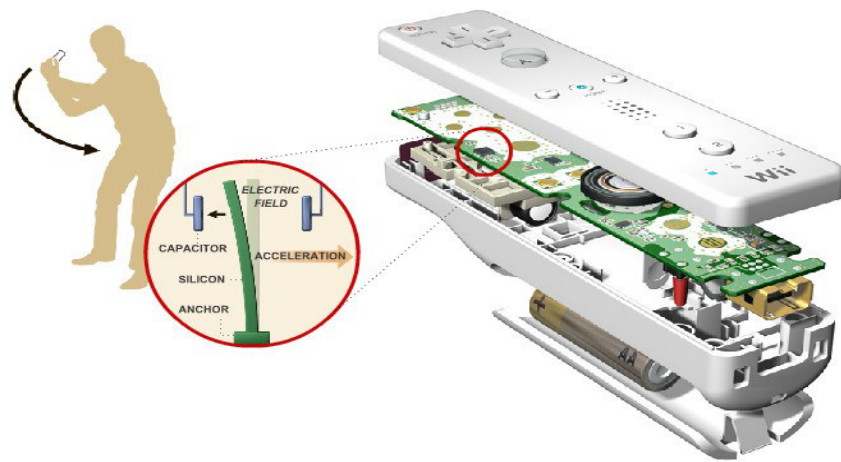


Figura 2.1: Acelerômetro do *Wii mote* (O'CONNELL; CAMPBELL, 2006)

Há muitos tipos diferentes de acelerômetros, mas todos servem para um mesmo fim primário, medir a aceleração. O *Wii mote* (Figura 2.1) possui um acelerômetro capacitivo, este utiliza um mecanismo de detecção conhecido como um sensor capacitivo de aceleração, para medir tanto a aceleração estática como as forças de acelerações dinâmicas. Com consideração a todos os acelerômetros, há normalmente uma espécie de circuito elétrico ligado ao mecanismo de detecção, a fim de representar a saída, ou tensão, das suas medições.

A força de aceleração estática é gerada por uma força gravitacional. Ao medir a aceleração estática com um acelerômetro capacitivo, as medições irão retratar o ângulo de inclinação a que o dispositivo está preso ou suspenso em relação à força gravitacional da Terra. A capacidade comum que o *Wii mote* possui é que a orientação muda em relação ao ângulo em que o dispositivo está sendo direcionado.

A força de aceleração dinâmica é gerada pelas vibrações do movimento. A medição da vibração permite a um acelerômetro capacitivo determinar em que direção o dispositivo está se movendo e com qual velocidade.

Os sensores capacitivos de aceleração contribuem para a captação de toda a energia de aceleração estática e dinâmica. A construção básica deste mecanismo é feita por uma microestrutura móvel ou um oscilador emparelhado com uma ou mais microestruturas em estado sólidas ou estacionárias que armazenam a capacitância, proporcionando definição

para a denominação “acelerômetro capacitivo.” A capacitância, ou a energia gerada por estes componentes, carregam um circuito elétrico sobre a qual o sensor está ligado.

O circuito elétrico é concebido para medir a saída ou a tensão gerada pelo mecanismo de detecção. Através de uma série de circuitos de detecção, que medem as tensões de pico, as tensões máximas são medidas de forma incremental. As medições são processadas por um amplificador que fornece uma soma de todas as medições, indicada como a saída final. A saída final seria a leitura mostrada ao usuário que é recolhido os dados a partir do dispositivo. Entretanto estes sinais estão sujeitos a ruídos, para solucionar este problema propõe-se utilizar o *Filtro de Kalman* para suavizar os dados.

## 2.3 *Filtro de Kalman*

O *Filtro de Kalman* é um método matemático proposto por Rudolf Kalman em (KALMAN, 1960). Seu propósito é utilizar medições de grandezas realizadas ao longo do tempo contaminadas com ruído e outras incertezas, e gerar resultados que tendam a se aproximar dos valores reais das grandezas medidas e valores associados.

O *Filtro de Kalman* é baseado em sistemas dinâmicos lineares discretizados em períodos de tempo. A cada passo de tempo, um operador linear é aplicado ao estado para gerar o próximo estado, com adição de ruído e, opcionalmente, alguma informação sobre as entradas de controle, caso sejam conhecidas. Depois, outro operador linear com mais adição de ruído gera as observações do estado real.

De modo a utilizar o *Filtro de Kalman* para estimar o estado completo de um processo dado apenas uma sequência de observações ruidosas, é necessário especificar as seguintes matrizes:  $F$ , o modelo de transição de estados;  $H$ , o modelo de observação;  $Q$ , a covariância do ruído do processo;  $R$ , a covariância do ruído da observação; e  $B$ , o modelo das entradas de controle. Para cada passo de tempo  $k$ , temos que a estimativa do vetor de estados pode ser dada por:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_k \quad (2.1)$$

, onde  $x_k$  é um vetor  $n$ -dimensional dos componentes de estado;  $x_{k-1}$  é o vetor de estados no tempo  $k - 1$ . O vector  $u_k$  é a entrada. O  $w_k$  é uma variável aleatória associada a eventos aleatórios ou forças que afetam diretamente o estado atual do sistema.

Em geral, fazem-se medições  $z_k$  que podem ou não serem medidas diretamente da variável de estado  $x_k$ . É possível resumir essa situação ao dizer que o vetor m-dimensional de medições  $z_k$  é dado por:

$$z_k = H_k x_k + v_k \quad (2.2)$$

, onde  $H_k$  é uma matriz MxN e  $v_k$  é o erro de medição.

Tudo o que resta agora é ligar essas expressões nas formas generalizadas das equações de atualização. A ideia básica é a mesma. Primeiro, é necessário calcular a *priori* uma estimativa  $x_k^-$  do Estado.

$$x_k^- = F x_{k-1} + B u_{k-1} + w_k \quad (2.3)$$

Utilizando  $P_k^-$  para denotar a covariância de erro, a estimativa a *priori* para esta covariância no tempo k é obtida, a partir do valor no tempo  $k - 1$ , por:

$$P_k^- = F P_{k-1} F^T + Q_{k-1} \quad (2.4)$$

Esta equação forma a base da parte de previsão do estimador, e define "o que se espera" com base no que já foi visto. A partir daqui, será obtido (sem derivação) o ganho de Kalman ou o fator de mistura, que diz como as informações novas pesam contra o que já se sabe:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (2.5)$$

Para um exemplo unidimensional em que se mede uma variável de posição diretamente,  $H_k$  é apenas uma matriz 1x1 contendo apenas um 1. Assim, se o erro de medição é  $\sigma_{k+1}^2$ , então  $R_k$  é também uma matriz de 1x1 que também contém 1. Da mesma forma,  $P_k$  é apenas a variância  $\sigma_k^2$ . Dessa forma a equação pode ser escrita como:

$$K_k = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_{k+1}^2} \quad (2.6)$$

O ganho permite calcular dos valores atualizados para  $x_k$  e  $P_k$  quando uma nova medição está disponível, isto é expresso nas equações (2.7) e (2.8):

$$x_k = x_k^- + K_k (z_k^- - H_k x_k^-) \quad (2.7)$$

$$P_k = (I - K_k H_k) P_k^- \quad (2.8)$$

## 2.4 Realidade Aumentada

A Realidade Aumentada pode ser definida como a sobreposição de objetos virtuais sob a imagem do ambiente físico, em tempo real, e com o apoio de um dispositivo tecnológico. A Realidade Aumentada é sensível aos sentidos humanos (AZUMA et al., 2001) e proporciona ao usuário uma interação segura, sem necessidade de treinamento, uma vez que ele pode trazer para seu ambiente real objetos virtuais, aumentando a visão que se tem do mundo real. Isto é obtido graças a técnicas de visão computacional e de realidade virtual.

Para que os objetos virtuais sejam visualizados e manipulados deve-se utilizar um programa com capacidade de visão do ambiente real e de posicionamento dos objetos virtuais, além de acionar dispositivos tecnológicos apropriados para Realidade Aumentada (ZORZAL, 2009).



Figura 2.2: Uso de interação por controle virtual (ZORZAL, 2009)

Os ambientes de Realidade Aumentada amplificam as capacidades de avaliação das informações tridimensionais, na medida em que flexibilizam a atuação no espaço tridimensional e permitem o uso de múltiplas interações, possibilitando maior riqueza de detalhes, melhores técnicas de interação e um melhor desempenho computacional (KINNER; SISCOUTTO, 2007).

Os sistemas de Realidade Aumentada podem ser classificados de acordo com seu esquema de exibição. Neste trabalho usaremos o sistema de visão por vídeo, baseado em monitor (Figura 2.3), utiliza uma webcam para capturar a imagem do ambiente físico. Após a captura os objetos gerados pelo computador são sobrepostos sobre a imagem e apresentados no monitor ao usuário.

Entre as maiores dificuldades que ocorrem no desenvolvimento de aplicações de Realidade Aumentada está a execução em tempo real uma série de atividades, envolvendo:

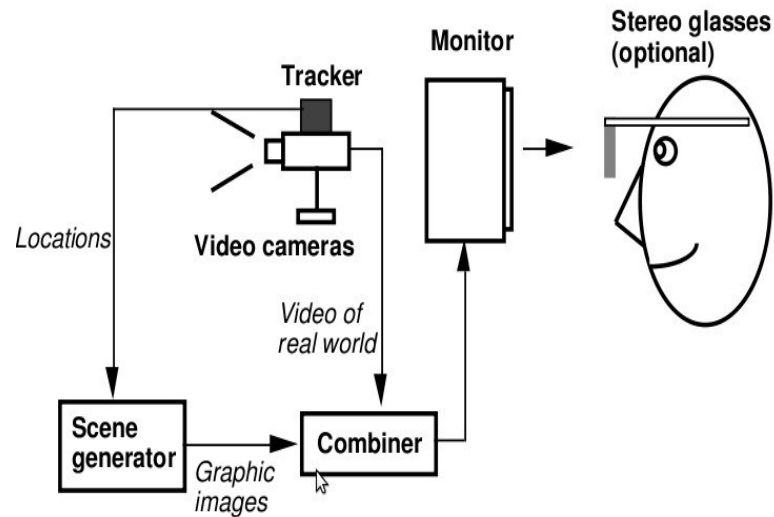


Figura 2.3: Diagrama do sistema de visão de vídeo baseado em monitor (AZUMA, 1997)

rastreamento de objetos reais, calibração e posicionamento de objetos virtuais. Os algoritmos de Realidade Aumentada rastream e calculam rapidamente a posição dos marcadores em relação a câmera, permitindo a iteração junto a objetos virtuais associados.

Neste capítulo abordamos sobre os dispositivos, algoritmos e bibliotecas usadas no projeto. No Capítulo 3, a seguir será explicada cada etapa desta proposta de trabalho.

# 3 METODOLOGIA PARA SIMULAÇÃO FÍSICA DE UM PERSONAGEM VIRTUAL

Nesse capítulo, será descrita a metodologia computacional da aplicação. Será explicado como as técnicas e recursos já conhecidos foram empregados. A Figura 3.1 mostra de forma simples as etapas que compõe a metodologia proposta.

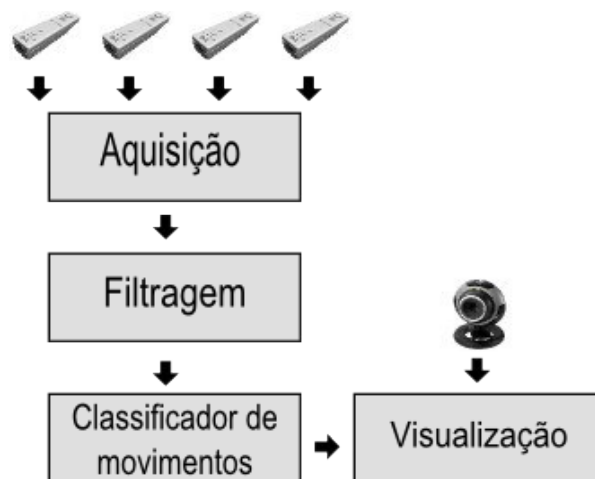


Figura 3.1: Módulos da aplicação

O primeiro módulo é a aquisição, esta tem a função de extrair as leituras de aceleração dos *Wiimotes* conectados; O segundo módulo é a filtragem, este objetiva suavizar os dados extraídos utilizando o *Filtro de Kalman*; O terceiro módulo é a classificação de movimentos, este analisa os dados e os classifica em diferentes tipos de ações; E, o ultimo módulo é a visualização, que desenha os elementos virtuais.

## 3.1 Aquisição

A aquisição dos dados da aceleração dos *Wiimotes* é uma etapa essencial para o funcionamento da aplicação como um todo. Primeiramente, porque todas as demais etapas dependem da aceleração extraída. Além disso, a quantidade de *Wimotes* que

devem estar conectados ao computador via *Bluetooth*, deve ser de pelo menos quatro para cada jogador. A aplicação suporta até oito *WiiMotes* conectados simultaneamente.

A leitura da aceleração do *WiiMote* por si só não é suficiente para obter melhores resultados no processamento do seu sinal. A presença de ruídos é menor quando utilizado com o acessório *Wii Motion Plus* que, ao ser plugado na parte inferior do controle aumenta a precisão de movimentos, garantindo assim uma reprodução fiel dos movimentos do usuário.

A extração da aceleração é feita através de funções da biblioteca *WiiMoteLib*. Para evitar diferença de performance, em máquinas diferentes, há um temporizador que limita as consultas da aceleração a uma taxa de trinta vezes por segundo.



Figura 3.2: LEDs do *WiiMote*

Após a inicialização da aplicação, a rotina de gerenciamento de *WiiMotes* procura por dispositivos que estão ociosos, para cada dispositivo encontrado é atribuída uma função única referente ao membro do usuário. Para auxiliar o usuário no reconhecimento de qual *WiiMote* é para seu membro, foi feita uma codificação nos seus LEDs (Figura 3.2). Na Tabela 3.1 segue a codificação dos LEDs segundo o membro do corpo que pertence para cada jogador, na primeira coluna '0' simboliza o LED desligado e '1' o LED ligado.

Se houver *WiiMotes* suficiente pelo menos para um usuário (quatro no mínimo) a aplicação seguirá realizando consultas à aceleração periodicamente, caso contrário a aplicação terminará. Por fim, as leituras serão usadas nas demais etapas da metodologia computacional. Os passos dessa etapa são resumidos no algoritmo (3.1).



LEDs	Membro/Usuário
0000	Desconectado
0001	Braço direito/Usuário 1
0010	Braço esquerdo/Usuário 1
0011	Perna direita/Usuário 1
0100	Perna esquerda/Usuário 1
0101	Braço direito/Usuário 2
0110	Braço esquerdo/Usuário 2
0111	Perna direita/Usuário 2
1000	Perna esquerda/Usuário 2

Tabela 3.1: Codificação dos LEDs

Listing 3.1: Algoritmo de Aquisição

---

```

1  Obter Wiimotes conectados
3  Se a Quantidade de Wiimotes conectados for menor que quatro
    Terminar aplicacao
5
   Para cada Wiimote na colecao faca
7     Codificar os LEDs do Wiimote
   Enquanto a quantidade de Wiimotes for maior que quatro
9     Para cada Wiimote na colecao faca
        Obter accelera o do Wiimote

```

---

Primeiro é chamado um processo que encontra todos os *Wiimotes* conectados ao computador e os adiciona em uma coleção, se o número de dispositivos conectados é inferior a quatro, a aplicação termina. Após a obtenção de cada *Wiimote*, estes recebem sua codificação de LEDs. Então a aplicação entra em um laço que consulta constantemente a aceleração de cada dispositivo.

As informações de cada *Wiimote* são mostradas ao usuário como na Figura 3.3. No topo do quadro é mostrado o usuário associado aquele dispositivo. Os três eixos são coloridos para mostrar qual a linha do gráfico é referente à aceleração. Também são mostrados o estado do *Wiimote* (conectado ou desconectado), os LEDs, nível da bateria e a qual membro pertence.

## 3.2 Filtragem

Uma vez adquiridos os dados da aceleração, observamos que os mesmos são adquiridos com ruídos. Os ruídos presentes nos dados extraídos atrapalham a análise da movi-

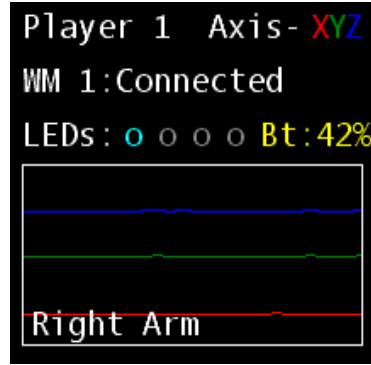


Figura 3.3: Caixa de informação do *Wiimote*

mentação, por isso neste passo iremos utilizar o *Filtro de Kalman* para removê-los. Será implementado o modelo proposto em (RASCO, 2007).

O vetor de estados,  $X$ ; e a matriz modelo de transição de estados,  $F$ ; estão expressos em (3.1). A entrada será a leitura da aceleração medida.

$$x = \begin{bmatrix} r \\ v \\ a_{bias} \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.1)$$

Os valores específicos usados para o vetor  $w$  (vetor de ruídos) depende da precisão do acelerômetro e da rapidez que se deseja que o filtro responda ao estimar o seu viés. O vetor  $B$ , é o modelo das entradas de controle, estão representados em (3.2).

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad u = a_{measured}, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (3.2)$$

O valor do ruído de precisão será determinado a partir da precisão da posição, e irá assumir o valor de 5 cm. Para o modelo de observação obtemos:

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad v = [v_1] = [v_{position}] \quad (3.3)$$

Uma boa maneira de iniciar a estimativa do estado é definir todos os estados como zero. Uma boa razão para definir um elemento de vetor de estado para um valor diferente de zero seria se o valor inicial já fosse conhecido a partir de uma medição preliminar. Por exemplo, se uma medida de posição está disponível, então essa poderia ser um bom valor a ser definido como a posição inicial. Uma forma base para definir a matriz de covariância,

$P(t)$ , é definir os elementos de folga diagonais como zero e os elementos diagonais como um valor suficientemente grande para aceitar novas medições. Porém não muito grande, caso contrário, o tempo para o filtro descobrir boas estimativas fica excessivo.

Após vários experimentos, os parâmetros de ruído com melhores resultados foram os seguintes:  $v1 = 0.005m$ ,  $w1 = 0.1m/s^2$ ,  $w2 = 0.1m/s^3$ .

Em uma execução de teste do *Filtro de Kalman*, em comparação com as leituras originais, como mostram as Figuras 3.4 e 3.5, o modelo implementado alcançou um bom resultado eliminando todos os ruídos do *Wii mote* em estado de repouso.

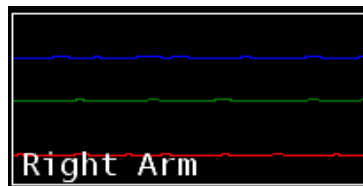


Figura 3.4: Sinais da aceleração original



Figura 3.5: Sinais da aceleração após o *Filtro de Kalman*

## 3.3 Classificação de movimentos

Com a suavização do sinal do acelerômetro a probabilidade de que ocorra uma classificação falsa é menor. A classificação de movimento se divide em duas partes: a primeira consiste em obter a orientação do *Wii mote* para descobrir a posição dos membros superiores, e a segunda que determinará a ação do usuário com relação aos membros inferiores, a partir da análise das amplitudes dos sinais da aceleração dos *Wii motes*.

### 3.3.0.1 Posição dos membros superiores

Mesmo que um dispositivo não possua um giroscópio interno, é possível descobrir a sua orientação analisando a aceleração ondular do eixo X do acelerômetro. Por exemplo,

quando o *Wii mote* está na posição vertical (Figura 3.6) a aceleração no eixo X marca -1.0, que por sua vez progride até 0 quando está na posição horizontal e até 1.0 quando está de ponta-a-cabeça. Desta maneira a partir das equações abaixo podemos descobrir a orientação angular do corpo.

Para aceleração no eixo X maior ou igual a zero:

$$Angle = X_{acel}(90)$$

Para aceleração no eixo X menor que zero:

$$Angle = X_{acel}(-90)$$

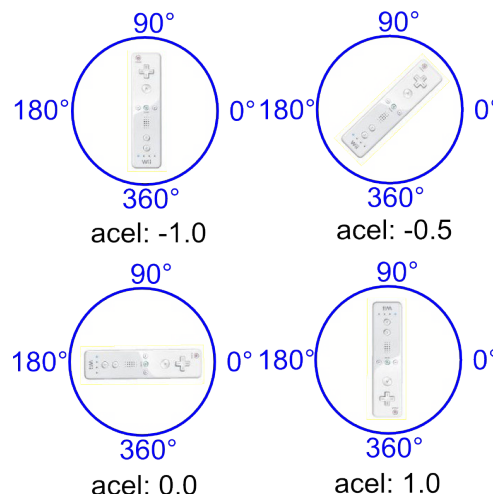


Figura 3.6: Aceleração do *Wii mote* no eixo X segundo sua orientação



Figura 3.7: *Wii mote* acoplado aos braços

Ao acoplar o *Wii mote* em um dos braços como na Figura 3.7 o usuário poderá controlar o ângulo de abertura desejado para que o personagem virtual imite com fidelidade o movimento.

### 3.3.0.2 Análise da ação do usuário com relação aos membros inferiores

Diferente dos membros superiores, onde cada braço tem sua própria orientação e trabalham individualmente, os membros inferiores trabalham de forma conjunta para classificar o movimento. Na Figura 3.8 podem ser vistos os *Wiiotes* acoplados as pernas do usuário.



Figura 3.8: *Wiiote* acoplado as pernas do usuário

Para determinar a ação do usuário com relação aos membros inferiores é feita uma análise das amplitudes ondulares da aceleração de ambas as pernas para criar um padrão de movimento.

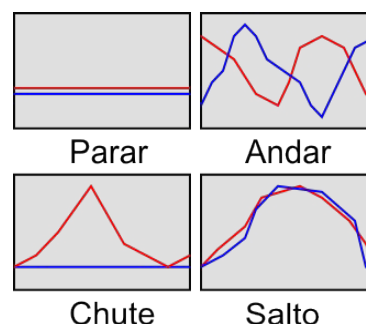


Figura 3.9: Ondas da aceleração dos *Wiiotes* dos membros inferiores. Vermelho para perna esquerda e Azul para perna direita

Para realizar a análise, o tempo deve ser discretizado em intervalos constantes de um segundo. Para cada intervalo será feita uma análise para classificar a ação do personagem virtual. Como pode ser visto na Figura 3.9 dois sinais constantes resultam na ação “Parar”; Sinais inversos resultam na ação “Andar”; Um Sinal constante e outro elevado resultam na ação “Chute” e ambos elevados resultam na ação “Salto”.

## 3.4 Visualização

Nesta etapa iremos desenhar os personagens e outros elementos virtuais no ambiente de Realidade Aumentada.

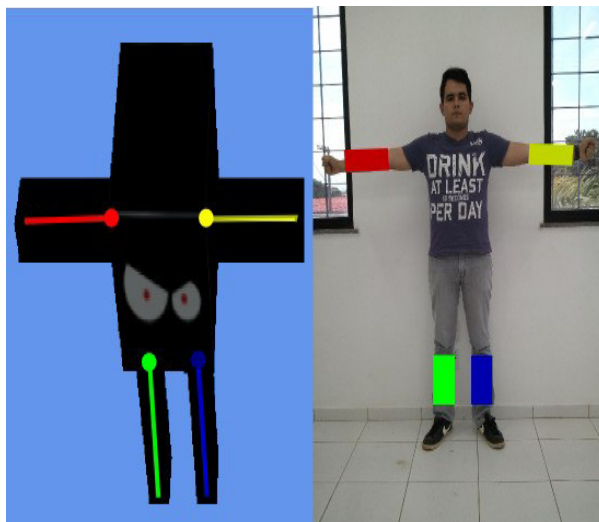


Figura 3.10: Personagem virtual com destaque aos membros coloridos que correspondem aos de mesma cor dos membros do usuário

O personagem virtual é simples, possui 1 par de pernas e braços que respondem à movimentação do usuário. Na Figura 3.10 podem ser vista a representação virtual dos membros do personagem, coloridos com a mesma cor na imagem do usuário. Quando o módulo de classificação de movimentos envia a posição dos membros do personagem, os membros de representação virtual são rotacionados em relação a uma junta. Cada junta possui apenas um grau de liberdade, pois a rotação relativa ocorre apenas em um único eixo.

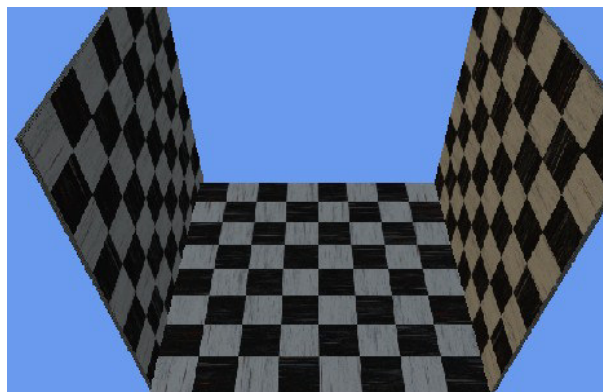


Figura 3.11: Plataforma de apoio

O personagem virtual permanecerá sobre uma plataforma de apoio que pode ser visto

---

na Figura 3.11.

As caixas de informações dos *Wimotes* conectados serão desenhados nos cantos, quatro de cada lado. Caixas de Instruções do jogo também serão desenhadas na parte inferior.

## 4 JOGO DESENVOLVIDO

Nesta seção será descrito todos os elementos, regras e dinâmicas da aplicação desenvolvida através dos diagramas da linguagem de modelagem UML.

### 4.1 Funcionalidades da aplicação

Os requisitos apresentam as funcionalidades do sistema de maneira mais detalhada. Para representar estes requisitos utilizaremos a descrição de casos de uso. A descrição dos casos de uso desta aplicação podem ser vistas na Figura 4.1 e nas Tabelas 4.1 a 4.8.

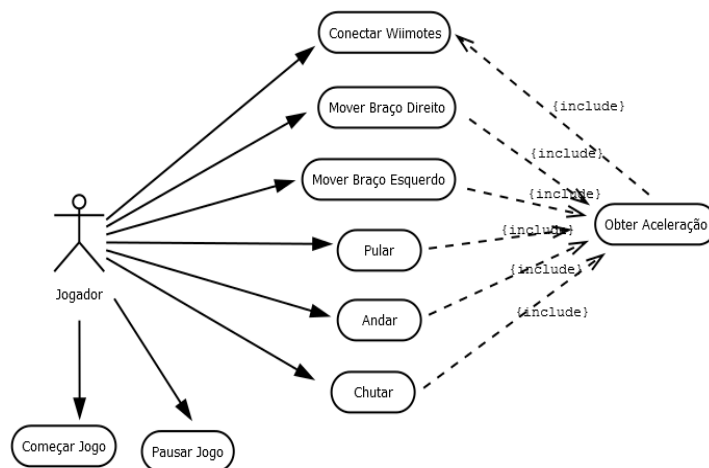


Figura 4.1: Diagrama de caso de uso da aplicação



Nome caso de Uso	Conectar Wiimote
Ator Principal	Jogador
Resumo	Jogador conecta os wiimotes a aplicação
Pré-Condição	Wiimotes conectados ao computador via BlueTooth
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Solicita Conexão do Wiimote	-
-	2 - Adiciona Wiimote a lista de dispositivos conectados

Tabela 4.1: Descrição de caso de uso - Conectar Wiimote

Nome caso de Uso	Pausar Jogo
Ator Principal	Jogador
Resumo	Jogador Pausa o jogo em execução
Pré-Condição	Jogo iniciado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Pausa o Jogo	-
-	2 - Para a atualização e congela o estado da memória

Tabela 4.2: Descrição de caso de uso - Pausar Jogo

Nome caso de Uso	Começar Jogo
Ator Principal	Jogador
Resumo	Jogador inicia o jogo ou retoma o jogo pausado
Pré-Condição	Jogo aberto ou pausado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Iniciar Jogo	-
-	2 - Carrega gráficos, informações e inicia.

Tabela 4.3: Descrição de caso de uso - Começar Jogo

Nome caso de Uso	Mover Braço Direito
Ator Principal	Jogador
Resumo	Jogador move o braço direito para a obtenção do grau de abertura
Pré-Condição	Wiimote conectado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Movimenta braço direito	-
-	2 - Obtém a aceleração do Wiimote referente ao braço direito.
-	3 - Retorna ângulo de abertura do braço direito.

Tabela 4.4: Descrição de caso de uso - Mover Braço Direito

Nome caso de Uso	Mover Braço Esquerdo
Ator Principal	Jogador
Resumo	Jogador move o braço esquerdo para a obtenção do grau de abertura
Pré-Condição	Wiimote conectado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Movimenta braço direito	-
-	2 - Obtém a aceleração do Wiimote referente ao braço esquerdo.
-	3 - Retorna ângulo de abertura do braço esquerdo.

Tabela 4.5: Descrição de caso de uso - Mover Braço Esquerdo

Nome caso de Uso	Pular
Ator Principal	Jogador
Resumo	Jogador realiza movimento de pulo
Pré-Condição	Wiimote conectado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Pula	-
-	2 - Obtém a aceleração do Wiimote referente as pernas.
-	3 - Retorna a pose da ação pular.

Tabela 4.6: Descrição de caso de uso - Pular

Nome caso de Uso	Andar
Ator Principal	Jogador
Resumo	Jogador realiza movimento de andar
Pré-Condição	Wiimote conectado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Pula	-
-	2 - Obtém a aceleração do Wiimote referente as pernas.
-	3 - Retorna a pose da ação andar.

Tabela 4.7: Descrição de caso de uso - Andar

Nome caso de Uso	Chutar
Ator Principal	Jogador
Resumo	Jogador realiza movimento de chute
Pré-Condição	Wiimote conectado
Pós-Condição	-
Ações do Ator	Ações da Aplicação
1 - Pula	-
-	2 - Obtém a aceleração do Wiimote referente as pernas.
-	3 - Retorna a pose da ação de chute.

Tabela 4.8: Descrição de caso de uso - Chutar

## 4.2 Entidades Modeladas

A partir dos requisitos descritos no tópico anterior é possível obter as entidades que comporão a aplicação. Com o diagrama de classe é possível descrever de forma organizada a implementação destas entidades (Figura 4.1).

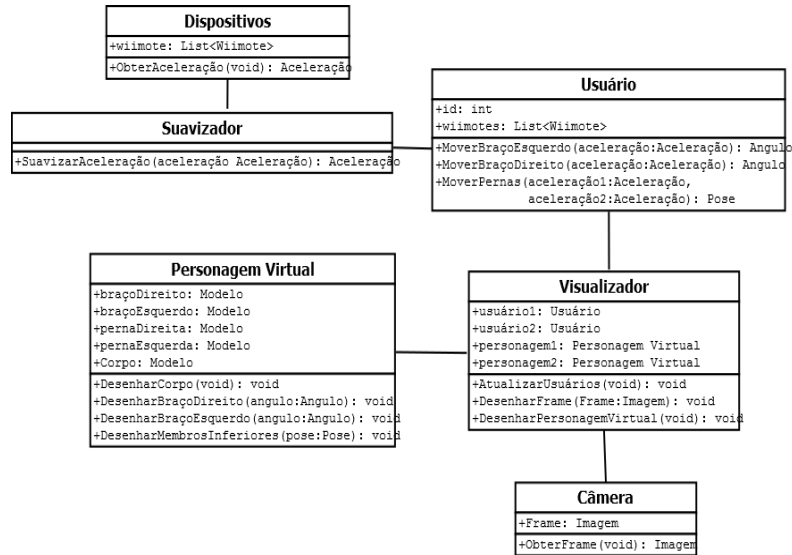


Figura 4.2: Diagrama de classes

As classes estão organizadas de acordo com os módulos apresentados no tópico do desenvolvimento. O diagrama está dividido em duas sessões. A primeira sessão é responsável pela obtenção e processamento da aceleração capturada. Esta entidade é composta das seguintes classes: Usuário, Suavizador e Dispositivo. A classe Usuário possui métodos de classificação de movimento e obtenção de ângulo, gerados a partir da aceleração obtida pela classe Dispositivos e filtrada pela classe Suavizador.

A segunda sessão é a responsável pela atualização e visualização dos elementos da aplicação. A principal componente desta entidade é a classe Visualizador, esta atualiza as leituras referentes a aceleração dos *Wiimotes*, desenha a imagem capturada da câmera e os personagens virtuais.

## 4.3 Detalhamento das principais funcionalidades

A partir das classes e suas funcionalidades é possível descrever uma sequência de passos ou troca de mensagens que devem ser realizadas para atender a determinada fun-

cionalidade. A seguir detalharemos algumas das principais funcionalidades observadas através de diagramas de sequência, são elas: Movimentar (Mover braços, Pular, Andar e Chutar) (Figura 4.3) e Conectar Wiimotes (Figura 3.4).

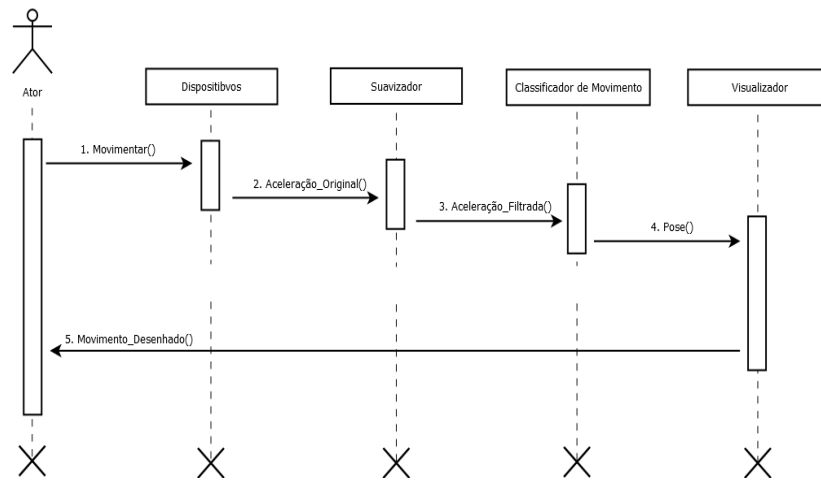


Figura 4.3: Diagrama de sequência - Movimentar (Mover braços, Pular, Andar e Chutar)

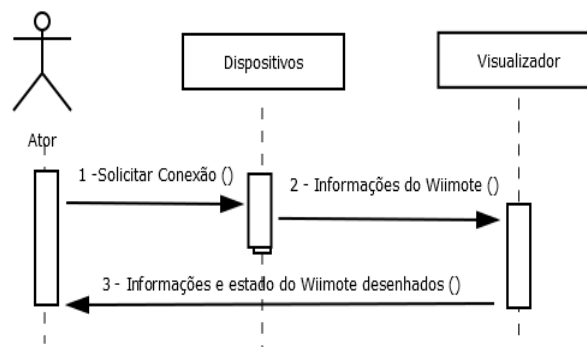


Figura 4.4: Diagrama de sequência -

## 4.4 Interação

Para jogar é necessário que cada jogador conecte pelo menos quatro *Wiimotes*, ou um Joystick ao computador. A interação com o jogo muda dependendo do tipo de controlador escolhido, o *Wiimote* utiliza uma interação por simulação física, já o Joystick utiliza interação com botões. Na Tabela 4.9 pode ser vista a relação do botões para os respectivos movimento na interação por Joystick.

Botão	Tipo de movimento
Alavanca Esquerda	Mover braço esquerdo
Alavanca Direita	Mover braço direito
A	Pular
B	Andar
Y	Chutar com a perna esquerda
X	Chutar com a perna direita

Tabela 4.9: Relação de botões do joystick com a movimentação do personagem

## 4.5 Interface

O jogo possui duas telas, a primeira é a tela do menu e a segunda é a tela principal do jogo (Figura 4.5). Na tela principal do jogo há duas colunas, uma em cada lado da tela, em cada coluna há quatro caixas que mostram as informações (aceleração, bateria e LEDs) para cada *Wiimote* conectado. Na parte central inferior da tela também há caixas com informações referente a partida, como: tempo para fim do turno, estado do jogador e contagem de pontos.

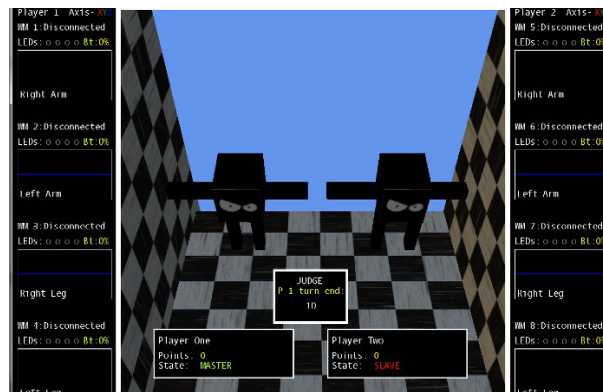


Figura 4.5: Tela principal do jogo

## 4.6 Mecânica do jogo

Neste tópico será descrito como o jogo funciona. No início do turno um jogador é escolhido para ser o mestre e o outro como escravo. O mestre terá dez segundos para realizar vários movimentos, ao fim deste tempo começa a vez do escravo, este deve imitar os movimentos feitos pelo mestre em dez segundos. O término do turno chega ao final da vez do escravo, o turno recomeça e os jogadores trocam de tarefas. Cada jogador recebe

uma pontuação de trinta pontos para cada movimento imitado com sucesso, e perde cinco pontos por cada movimento errado. O jogo é finalizado após quatro turnos. Vence o jogador com a pontuação mais alta.

## 4.7 Implantação

Como pode ser visto no modelo de implantação (Figura 4.6), para funcionar de forma correta a aplicação deve rodar sob o sistema operacional Windows, deve haver também uma conexão via Bluetooth, para que os Wiimotes se conectem ao jogo. Para o funcionamento do jogo é necessário os seguintes requisitos: 4 Wiimotes por jogador, Câmera Digital, .NET Framework 4.0 e XNA Framework 4.0.

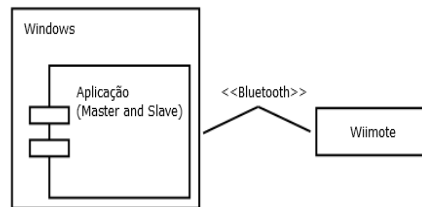


Figura 4.6: Modelo de implantação

## 5 CONCLUSÃO

Neste trabalho foi construído um jogo baseado em uma interface de simulação física, capaz de reconhecer vários tipos de movimentos, concebido em um ambiente de Realidade Aumentada. A partir da aplicação desenvolvida foi feito um estudo com parâmetros de comparação para analisar a eficiência da interface proposta. Embora a maioria dos jogos comerciais utilize uma interface com *joystick*, a interface de simulação física mostrou-se superior nesse tipo de aplicação por ser muito mais intuitiva e por não exigir uma manipulação precisa. Foram analisados dois parâmetros, o primeiro foi o tempo de resposta, e o segundo a familiaridade e facilidade de uso.

O tempo de resposta da execução do movimento na interface de simulação física foi satisfatório, embora haja um pequeno atraso da ordem dos milissegundos, pois o programa precisa de um tempo para analisar e classificar o movimento dos membros inferiores. Quanto a familiaridade e facilidade de uso, mostrou-se bastante eficiente, a interface imita padrões de locomoção que vivemos na vida diária, o que a torna mais envolvente que outras interfaces.

Um ponto negativo sobre a dinâmica do jogo é que depois de algum tempo, dependendo na capacidade física do usuário, operar a interface se torna cansativo. De outro ponto de vista esta característica pode se tornar uma vantagem, como o *Wii Fit*, que visa o exercício.

A interface fisicamente simulada é mais envolvente e com o uso da realidade aumentada o usuário tem uma maior sensação de imersão. Apesar do efeito de atraso na resposta da movimentação, o utilizador intuitivamente o compreende, e se adapta a aplicação.

Futuramente pretende-se aprimorar o trabalho, pesquisando e aplicando tecnologias e técnicas mais precisas de obtenção e análise de movimento, e por fim, aplicar a tecnologia desenvolvida em outros sistemas, como: manipulação de objetos, navegação, treinamento, apoio em terapias físicas e etc.



## Referências Bibliográficas

AZUMA, R. et al. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, December 2001.

AZUMA, R. T. A survey of augmented reality. *In Presence: Teleoperators and Virtual Environments*, August 1997.

JOHNSON, M. P. et al. Sympathetic interfaces: Using a plush toy to direct synthetic characters. *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, 1999.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, 1960.

KINNER, C.; SISCOUTTO, R. *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*. [S.l.]: Editora SBC – Sociedade Brasileira de Computação, 2007.

LASZLO, J.; PANNE, M. van de; FIUME, E. Interactive control for physically-based animation. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2000*, 2000.

NEFF, M.; ALBRECHT, I.; SEIDEL, H.-P. Layered performance animation with correlation maps. *Computer Graphics Forum*, October 2007.

NINTENDO. Wii. 2006. [Http://www.wii.com/](http://www.wii.com/).

NINTENDO. Wii motion plus. 2009. [Http://www.nintendo.com/wii/](http://www.nintendo.com/wii/).

O'CONNELL, F.; CAMPBELL, M. Wii's means motion. December 2006. [Http://www.nytimes.com](http://www.nytimes.com).

OSHITA, M. A pen-based interactive control of a human figure. *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, 2004.

RASCO, B. Where's the wiimote? using kalman filtering to extract accelerometer data. June 2007. [Http://www.gamasutra.com](http://www.gamasutra.com).

SHIRATORI, T.; HODGINS, J. K. Accelerometer-based user interfaces for the control of a physically simulated character. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2008*, December 2008.

THORNE, M.; BURKE, D.; PANNE, M. van de. Motion doodles: an interface for sketching character motion. *ACM SIGGRAPH 2007 courses*, 2007.

YIN, K.; PAI, D. K. Footsee: an interactive animation system. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2003.

ZORZAL, E. R. Estratégia para o desenvolvimento de aplicações adaptativas de visualização de informações com realidade aumentada. Novembro 2009.