

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**REABILITAÇÃO DE PACIENTES COM DESORDENS  
DE EQUILÍBRIO UTILIZANDO KINECT**

**TRABALHO FINAL DE GRADUAÇÃO**

**Rubens Andreas Sautter**

**Santa Maria, RS, Brasil**

**2014**

# **REABILITAÇÃO DE PACIENTES COM DESORDENS DE EQUILÍBRIO UTILIZANDO KINECT**

**Rubens Andreas Sautter**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de  
**Bacharel em Ciência da Computação.**

**Orientador: Prof. Dr. Marcos Cordeiro d'Ornellas**

**Trabalho de Graduação N. 384**

**Santa Maria, RS, Brasil**

**2014**

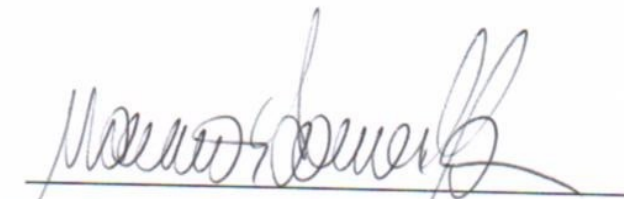
**REABILITAÇÃO DE PACIENTES COM DESORDENS DE  
EQUILÍBRIO UTILIZANDO KINECT**

elaborado por

**Rubens Andreas Sautter**

como requisito parcial para obtenção do grau de  
**Bacharel em Ciência da Computação**

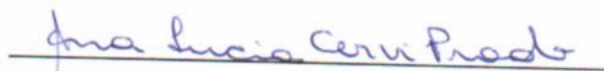
**COMISSÃO EXAMINADORA:**



**Prof. Marcos Cordeiro D'Ornellas (UFSM) (Presidente/Orientador)**



**Profa. Giliane Bernardi (UFSM)**



**Prof. Ana Lúcia Cervi Prado (UFSM)**

Santa Maria, 02 de dezembro de 2014.

## **AGRADECIMENTOS**

Dedico este trabalho de graduação a minha família pelo apoio, aos meus pais Frank e Cláudia por sempre estarem ao meu lado e me incentivarem a alcançar novas metas, a minha irmã Cecilia por sempre se preocupar comigo.

Também gostaria de agradecer aos grupos de pesquisa do LaCA e da Saúde Aplicada, por me aceitarem e auxiliarem a fazer este trabalho. Em especial ao professor Jefferson Potiguara Moraes, que cedeu seu tempo de folga para me auxiliar neste trabalho e introduzindo a uma área de pesquisa que até então eu desconhecia.

Aos meus colegas de aula, pelo companherismo, tanto nos passatempos, como nos trabalhos realizados durante todo o curso.

## RESUMO

Trabalho de Graduação  
Curso de Ciência da Computação  
Universidade Federal de Santa Maria

### **Desenvolvimento de um aplicativo para o auxílio a reabilitação de pacientes com desordens de equilíbrio utilizando Kinect**

AUTOR: RUBENS ANDREAS SAUTTER

ORIENTADOR: MARCOS CORDEIRO D'ORNELLAS

Data e Local da Defesa: Santa Maria, Dezembro de 2014.

A reabilitação de pacientes que sofreram lesões no joelho durante uma prática desportiva é e muitas vezes demorada, podendo afetar diversas habilidades motoras e de equilíbrio. Como forma complementar de reabilitação destes pacientes, foi proposto neste trabalho o estudo e implementação de um jogo utilizando o dispositivo *Kinect*, avaliando sua usabilidade e eficiência. O *Kinect* é um dispositivo de captura de imagens permite mapear os movimentos do paciente para um modelo em três dimensões e assim utilizar este modelo em um jogo. Este dispositivo possui compatibilidade com diversas *engines* gráficas, dentre elas a Unity foi utilizada para o desenvolvimento deste jogo devido a sua compatibilidade com diversos tipos de modelos em 3D. Além disto, a Unity possui métodos para a conexão com servidores de bancos de dados, que foram utilizados para enviar informações sobre a forma de execução dos exercícios realizados pelo paciente para um servidor. Isto permite que posteriormente possam ser avaliados a execução dos exercícios de reabilitação, com maior acuidade, auxiliando o fisioterapeuta a determinar o tipo de movimento que o paciente deve realizar.

**Palavras-chave:** Reabilitação, Kinect, Equilíbrio, Unity

## **ABSTRACT**

Many activities like walk, run or jump can damage the knees, and so it is an recurrent problem in the population. The treatment for this injury varies according to its severity, that may be treated according to it (Silva, 2007). One common step in many rehabilitation process is the cinesioterapia, where the patient is treated by repetitive movements. Because of this many technologies were developed to make these exercises more fun. But these technologies are mostly paid, making it difficult to achieve it. So we have made in this work a game, in a partnership between HUSM and UFSM, to help treat patients.

In this work we developed a game that aims to assist physiotherapists to rehabilitate athletes who have suffered some type of knee injury. To make this game we have studied the knee injuries, as its traditional treatment and then we developed it using Kinect. To evaluate the system requirements, we have performed a usability test with specialization physical therapy students.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura Musculoesquelética.....	13
Figura 2 – Área ideal de funcionamento do kinect.....	19
Figura 3 – Estrutura hardware do Kinect.....	19
Figura 4 – Windows SDK skeleton.....	23
Figura 5 – Interface do jogo de equilíbrio dinâmico.....	25
Figura 6 – Interface do jogo de equilíbrio estático.....	26
Figura 7 - Encriptação AES.....	35

## LISTA DE DIAGRAMAS

Diagrama 1 – Casos de uso do menu principal e dos jogos.....	28
Diagrama 2 – Classes do jogo de equilíbrio dinâmico.....	29
Diagrama 3 - Sequência para ativação de um plano.....	30
Diagrama 4 - Classes do jogo de equilíbrio estático.....	31
Diagrama 5 – Modelagem da base de dados.....	34



## **TABELA DE ABREVIATURAS E SIGLAS**

HUSM – Hospital Universitário de Santa Maria

IR – Infrared

UML – Unified Modeling Language

SQL – Structured Query Language

RGB – Red, Green, Blue

WAV – Waveform Audio File Format

AIFF –Audio Interchange File Format

GUI – Graphical User Interface

AES - Advanced Encryption Standard

XOR – Exclusive or

GF – Galois Field

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 Justificativa.....	12
1.2 Objetivos.....	13
1.2.1 Objetivos secundários.....	13
<b>2 REVISÃO DA LITERATURA.....</b>	<b>14</b>
2.1 Estrutura Musculoesquelética.....	14
2.2 Propriocepção.....	14
2.3 Ligamento Cruzado.....	15
2.4 Jogos Sérios.....	16
<b>3 MATERIAIS E MÉTODOS.....</b>	<b>18</b>
3.1 Regras do Jogo.....	18
3.1.1 Jogo de equilíbrio Estático.....	18
3.1.2 Jogo de equilíbrio Dinâmico.....	19
3.2 Especificações do Kinect.....	19
3.3 Engine Unity.....	21
3.3.1 Organização da Unity.....	21
3.3.2 Física e detecção de colisão na Unity.....	21
3.3.3 Sons e Efeitos Musicais.....	22
3.3.4 GUI.....	23
3.4 Biblioteca Kinect for Windows SDK 2.0.....	23
3.4.1 Alterações na biblioteca.....	24
3.5 Interface do sistema.....	25
3.6 Modelagem do sistema.....	27
3.6.1 Modelagem do menu principal.....	28
3.6.1 Modelagem dos jogos.....	28
3.7 Biofeedback.....	31
3.8 Conexão com base de dados.....	33
3.8.1 Modelagem da base de dados.....	33
3.8.2 Segurança.....	34
3.8.3 Exemplos de consulta.....	36

<b>4 RESULTADOS E DISCUSSÃO.....</b>	<b>37</b>
4.1 Testes do sistema.....	37
4.2 Discussão.....	37
<b>5 CONCLUSÃO.....</b>	<b>38</b>

# 1 INTRODUÇÃO

Muitas atividades diárias como caminhar, correr ou saltar podem causar danos aos joelhos, sendo assim um problema recorrente na saúde da população. O tratamento deste tipo de lesão varia de acordo com sua gravidade, podendo ser tratado de diversas formas (SILVA, 2007). Uma das etapas comum para a maioria dos processos de reabilitação é a cinesioterapia, onde o paciente é tratado através de movimentos repetitivos de acordo com sua capacidade de movimentação e assim aos poucos reestabelecer o movimento natural. Para tornar o tratamento lúdico diversas tecnologias já foram desenvolvidas, tais como os controles remotos e dispositivos de captura de movimento. Porém o desenvolvimento de jogos que utilizam estas tecnologias é específico para cada tipo de tratamento. Para facilitar o acesso a este tipo de tecnologia no tratamento de pacientes neste trabalho foi desenvolvid dois jogos, através da parceria entre o HUSM e o Curso de Ciência da Computação da UFSM.

O uso de jogos para o auxílio em tratamentos é frequente em diversas áreas da saúde, pois os jogos podem ser uma ferramenta muito útil para melhorar o estilo de vida além de motivar e prover informações sobre a atividade física (FERGUSON, 2012). Neste trabalho o jogo desenvolvido tem por objetivo auxiliar os fisioterapeutas a reabilitar pacientes que sofreram algum tipo de lesão no joelho, pois muitas vezes este processo é demorado. Para isto foram estudados os tipos de lesões, como estas lesões são tratadas tradicionalmente e então foi desenvolvido um jogo utilizando o dispositivo *Kinect* com base nestas formas de tratamento.

## 1.1 Justificativa

No Brasil, segundo o Censo de 2010, cerca de 4% da população sofre de algum tipo de deficiência motora, representando na época 8.832.249 habitantes e destes cerca de 41% tem uma grande deficiência motora. Estas deficiências físicas muitas vezes podem ser tratadas através de exercícios fisioterapêuticos. Porém, estes exercícios físicos muitas vezes não são atrativos, levando a apenas 31% dos pacientes aplicar corretamente os exercícios (SHAUGHNESSY, 2006). Abordagens lúdicas, como jogos de computadores, podem auxiliar este tipo de recuperação tornando esta abordagem mais eficaz e atrativo aos pacientes.

## 1.2 Objetivos

Implementar um sistema para o auxílio a reabilitação de pacientes. Para tanto, foi proposto o uso do dispositivo de captura de movimento Kinect para o desenvolvimento de dois jogos que trabalham os movimentos de reabilitação de períodos diferentes de reabilitação.

### 1.2.1 Objetivos secundários

- Desenvolver uma interface natural, tanto para o paciente quanto para o fisioterapeuta.
- Contribuir para o desenvolvimento da biblioteca Windows SDK
- Criar de um sistema de *biofeedback* preciso, para informar propriedades do movimento que está sendo realizado, tais como angulação do joelho e centro de massa.
- Integrar ao sistema uma base de dados, de forma segura para armazenar informações sobre os movimentos realizados e as propriedades configuradas no jogo.

## 2 REVISÃO DA LITERATURA

Nesta seção estão descritos os principais tipos de lesão, os tratamentos tradicionais e alternativas auxiliares para o tratamento destes tipos de lesão.

### 2.1 Estrutura Musculoesquelética

A prática desportiva é saudável, porém em sua prática é comum ocorrer lesões (PINHEIRO, 2009). Para tratar estas lesões é importante entender a estrutura Musculoesquelética do corpo humano. Esta estrutura é composta por diferentes tecidos, conforme ilustrado na figura 1, onde o músculo é composto por um conjunto de fibras que em conjunto a vasos capilares e nervos formam o endomísio. Enquanto o perimísio é composto por tecido conjuntivo, fornecendo uma propriedade elástica a estrutura.

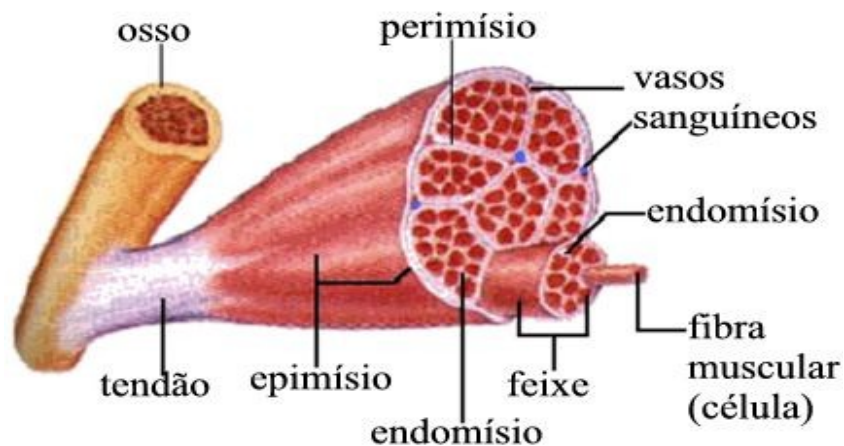


Figura 1-1 – Estrutura Musculoesquelética

Segundo Prentice (2011), quando uma força externa é aplicada ao músculo esta sofre uma deformação. Ao remover esta força externa o músculo tende a retornar ao estado inicial devido a sua propriedade elástica, porém se esta deformação passar de um ponto de cedência parte da deformação permanece, produzindo assim uma lesão.

### 2.2 Propriocepção

Na literatura há diversas definições de propriocepção, Hewett (2002) define a propriocepção como o resultado de aferências provenientes de articulações, músculos e

proprioceptores de músculo profundo associados, onde estes seriam projetados ao sistema nervoso central, produzindo assim o reflexo e o controle motor. Atualmente a propriocepção é geralmente referida como uma modalidade sensorial, que envolve a sensação de posição das articulações, sensação de movimento das articulações e detecção de força aplicada as articulações. Esta modalidade sensorial é proveniente de proprioceptores localizados nos músculos, ligamentos e tendões que produzem um sinal que é repassado ao sistema nervoso central (GUIMARÃES,2003).

A propriocepção é essencial para o controle motor, pois os receptores que estão presentes na estrutura musculoesquelética avaliam informações intracorpóreas e extracorpóreas. Entre as informações intracorpóreas destaca-se a posição do centro de gravidade, pois este é essencial para o equilíbrio, enquanto informações do ambiente externo como as irregularidades de um relevo são classificadas como informações extracorpóreas.

### **2.3 Ligamento Cruzado**

O ligamento cruzado é uma estrutura musculoesquelética que faz parte de diversos ligamentos, nos joelhos esta estrutura é ligada ao osso fêmur e a tibia. Esta estrutura é responsável por estabilizar as juntas. Entre os diversos tipos de lesões na estrutura musculoesquelética, o rompimento do ligamento cruzado possui alta incidência na população, com cerca de 0.03% ao ano (BOLLEN,1995). O ligamento cruzado é responsável por grande parte dos movimentos esportivos, e o seu rompimento pode causar instabilidade funcional, destacando-se a propriocepção (GUIMARÃES, 2003).

Existem diversos estudos sobre o tratamento de pacientes após a reabilitação, além de testes da progressão da reabilitação. Adams (2012) propôs em um estudo uma sequência de atividades e acordo com o período após a operação em sua grande maioria, conforme a compilação expressa na tabela 1. Nas etapas intermediária e tardia, não é estipulado o tempo pós-operatório, sendo necessário uma avaliação subjetiva do fisioterapeuta. O objetivo desta sequência de atividade é inicialmente melhorar a força muscular e a propriocepção dos joelhos, enquanto as últimas etapas procuram introduzir ou reintroduzir o paciente a um esporte.

Tabela 1 – Sequência de reabilitação

<b>Período</b>	<b>Fase</b>	<b>Atividade</b>
1 semana após a cirurgia	Pós operação	Movimentar de 0° a 90° o joelho, acrescentar pesos com o tempo. Aplicar estímulo neuromuscular.
2 semana após a cirurgia	Inicial	Apoiar as costas na parede e descer utilizando a força das pernas
nd	Intermediária	Equilibrar-se com uma perna, realizando algum tipo de exercício lúdico
nd	Tardia	Aplicar estímulo neuromuscular.
8 semanas após a cirurgia	Transicional	Corrida

nd – Período não estipulado

## 2.4 Jogos Sérios

Existem diversas definições de jogos sérios na literatura. Segundo Susi (2007), os jogos sérios são jogos outros com que possuem outros propósitos além do entretenimento, como a educação ou simulação. Enquanto . Neste projeto foi estudado e desenvolvido um jogo sério como forma complementar de tratamento para a fisioterapia. Em muito destes jogos, especialmente aqueles que são voltados a fisioterapia, utilizam-se dispositivos de captura de movimento para avaliar o desempenho do paciente.

Dentre os diversos dispositivos auxiliares podemos citar o controle remoto dos consoles, o *Wii Controller* ou *Kinect*. Cada um destes pode ser utilizado para atender um objetivo na reabilitação, por exemplo o controle remoto dos consoles pode auxiliar na reabilitação da motricidade da mão conforme descrito por Szturm (2008), enquanto o *Kinect* pode auxiliar na reabilitação dos movimentos de membros superiores como descrito por Cargnin (2014).

O estudo realizado por Szturm (2008) demonstra como a implementação de um jogo sério pode auxiliar na reabilitação. Este estudo tinha por objetivo reabilitar a motricidade da mão, para isto foi adicionado sensores a objetos como rodas, bolas e potes para assim desenvolver um jogo para os pacientes com este problema. No estudo de Szturm (2008) foi avaliado três casos, um paciente com lesão na medula espinhal, um paciente que sofreu um



acidente vascular cerebral isquêmico e um paciente com hemiplegia. Como resultado deste estudo foi observado a melhoria significativa da motricidade da mão.

O estudo realizado por Cargnin et al. (2014) tinha por objetivo determinar quais parâmetros são importantes para desenvolver um jogo de reabilitação de membros superiores em pacientes com hemiplegia utilizando o dispositivo de captura de movimentos kinect. Muitos parâmetros utilizados neste estudo também podem ser utilizados em outros tipos de jogos de reabilitação. Neste estudo um dos parâmetros que podem ser destacados é o sistema de biofeedback para a desenvolvimento de jogos de reabilitação em geral, o sistema de biofeedback fornece informações através de sinais luminosos e auditivos que permitem ao paciente ajustar corretamente seus hábitos e reações(CARGNIN,2014).

Existem diversos estudos relacionados ao desenvolvimento de jogos sérios para a reabilitação, porém poucos são relacionados a reabilitação dos joelhos. Grande parte dos trabalhos relacionados a reabilitação dos joelhos tratam do sistema de *biofeedback*, tais como o cálculo do centro de massa e sua distribuição pelo corpo, como descrito por González et al. (2014) e Rosário et al. (2013). No artigo de Gonzalez et al. (2014) foi estudado o método SESC (*Statically Equivalent Serial Chain*) para a estimação do centro de massa, neste método o centro de massa depende de fatores como a orientação de todas as ligações do corpo além disto esta técnica estima as posições dos centros de massa locais. Para estimar o centro de massa local com precisão, nesta técnica é construído um sistema linear. Existem diversas implementações, que tentam otimizar esta técnica para tentar tornar viável a estimação do centro de massa em um tempo razoável de processamento.

O estudo de Rosário et al.(2013) busca estimar o centro de massa utilizando uma técnica similar ao SESC, porém ao invés de resolver um sistema linear para encontrar os centros de massa relativos, foi utilizado a média das posições dos centros de massa locais ponderados pelo pesos deste segmento do corpo humano. Esta técnica é muito mais simples de ser implementada além de possuir complexidade menor que a técnica SESC, porém esta técnica tem precisão menor que a técnica SESC.

### 3 MATERIAIS E MÉTODOS

Neste capítulo está descrito como as tecnologias auxiliares utilizadas para o tratamento dos pacientes que sofreram algum tipo de lesão. Além disto também é descrito as capacidades e limitações destas tecnologias.

#### 3.1 Regras do Jogo

Neste projeto foi desenvolvido um sistema para o auxílio na reabilitação da propriocepção dos joelhos. Para isto foi desenvolvido dois jogos de reabilitação, um destes tem por objetivo reabilitar a propriocepção da estabilidade estática e outro tem por objetivo reabilitar a propriocepção da estabilidade dinâmica, pois segundo Guimarães (2003), a propriocepção é composta por estes dois tipos de sensação que são utilizados para gerar estabilidade.

Para o desenvolvimento dos jogos, os movimentos do paciente foram mapeados para um modelo em 3D e este modelo em 3D foi utilizado para construir a mecânica dos dois jogos. A interação do paciente com os jogos foi desenvolvida utilizando mecanismos como som e a alteração dos objetos em 3D presentes nas cenas do jogo.

##### 3.1.1 Jogo de equilíbrio Estático

Para o primeiro jogo buscou se trabalhar a memória do paciente além da reabilitação da propriocepção e força muscular. Neste jogo há um conjunto de botões, que ao serem ativados mudam de cor e produzem um som. O objetivo do paciente é ativar os botões em uma determinada sequência, se o paciente acertar a sequência uma nova ativação é adicionada ao final da sequência caso contrário a sequência é reiniciada. Para ativar um botão o paciente deve se movimentar de maneira a deslocar a mão do personagem virtual até esta encostar no botão, porém o movimento somente é validado se o paciente estiver com apenas um dos pés longe do chão.

O objetivo deste jogo é realizar movimentos que geralmente são aplicados durante a fase intermediária de reabilitação, conforme descrito na seção 2.3. Adams (2012) sugere um jogo similar onde o paciente deve pegar cones que estão no chão, enquanto apoiados em

apenas um dos pés, neste jogo a dificuldade é ajustada através da altura dos cones e a distância dos mesmos. De forma similar a configuração do jogo de pegar os cones a configuração do jogo de equilíbrio estático, permite alterar o nível de dificuldade através das propriedades de altura de cada um dos botões e a distância destes em relação ao paciente.

### 3.1.2 Jogo de equilíbrio Dinâmico

A reabilitação através de movimentos rápidos é denominada pliometria, esta técnica de reabilitação pode melhorar em cerca de 1.7% a velocidade em 20 dias, sendo uma técnica que não causa atrofia muscular significativa (HERRERO, 2002). No jogo desenvolvido utilizou-se a pliometria através do movimento de deslocamento do paciente. Neste jogo o objetivo do paciente é alcançar todos os círculos que aparecerão na tela com uma determinada cor, em uma região pré-definida pelo fisioterapeuta. Quando o paciente conseguir alcançar um destes círculos sua pontuação será incrementada e a cada intervalo de tempo predefinido nas configurações do jogo, as posições das circunferências são alteradas para uma nova posição.

Diferentes modalidades deste jogo foram desenvolvidas para avaliar a movimentação do paciente, e permitir tratar de acordo com o nível de recuperação do paciente. Em uma das modalidades do jogo o paciente deve se deslocar em direção a cada um dos círculos ativos, para assim pontuar. Em uma segunda modalidade do jogo de equilíbrio dinâmico o paciente precisa se deslocar em todos os círculos ativos em uma sequência e assim retornar ao círculo central e assim pontuar e reiniciar a sequência.

## 3.2 Especificações do Kinect

Segundo a Microsoft (2014) o *Kinect* é um dispositivo de captura de imagens da profundidade do ambiente a sua frente, possuindo um ângulo de visão horizontal de 57° e ângulo vertical de 43°, este dispositivo consegue captar com precisão imagem de objetos entre 2.0 m e 3.5 m. Funcionando de forma ideal na área definida pela figura 2. Segundo Khoshelham (2011), o erro deste dispositivo varia quadraticamente a cada 4 cm, e necessita da distância mínima de 7 cm do alvo e pode funcionar até 5 m do alvo desejado. Além disto o *Kinect* é capaz de gerar 30 imagens por segundos em sua atual versão.

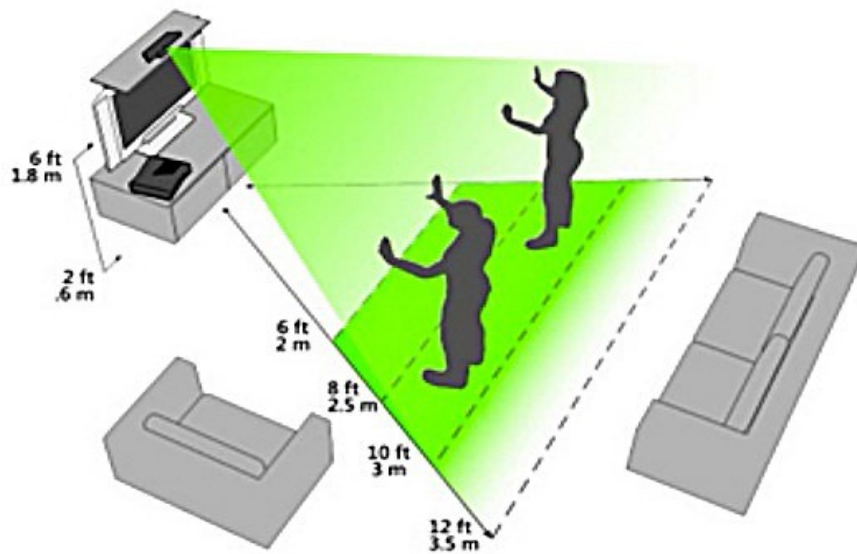


Figura 2 – Área ideal de funcionamento do kinect

A imagem capturada com profundidade permite o mapeamento de humanos para um modelo em 3D. Isto é possível devido ao conjunto das duas câmeras que este possui, conforme a figura 3.

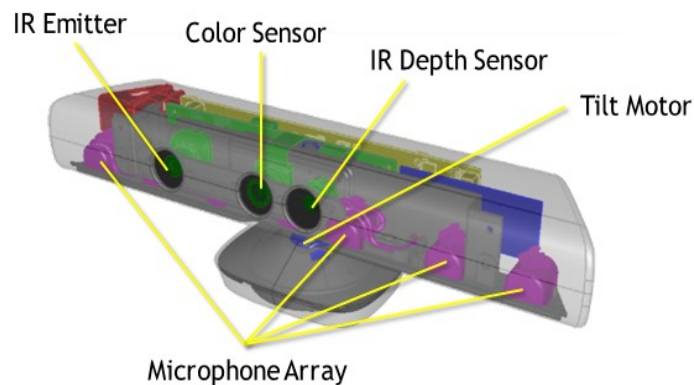


Figura 3 – Estrutura hardware do Kinect

Dentre as estruturas do *kinect* pode-se observar:

- IR Emitter: Emissor de raios infravermelho de 830nm não visíveis ao olho humano
- IR Depth Sensor: Receptor de raios infravermelho de 830nm
- Color Sensor: Sensor de cores visíveis ao olho humano (RGB), de resolução 640x480

### 3.3 Engine Unity

Motores gráficos, também conhecidos como *engine*, são plataformas de desenvolvimento de jogos, que são desenvolvidas para facilitar o desenvolvimento de jogos. Estas plataformas permitem a importação de diversos tipos de padrões de modelos em 3D ou em 2D, além de sons e texturas. Neste trabalho foi utilizado a Unity, que é um motor gráfico desenvolvido pela Unity Technologies em que utiliza o paradigma orientado a *Script*. Este motor gráfico permite a importação de modelos em diversos formatos, além de possuir uma biblioteca externa para o acesso ao *Kinect*.

No paradigma orientado a *Script*, o jogo é modelado em um conjunto de objetos de jogo, denominados pela Unity como *GameObject*, e um conjunto de classes características destes objetos que é denominada *Script*. Os *Scripts* são segmentos de código, que podem seguir diferentes tipos de paradigmas dependendo da linguagem escolhida. Dentre as linguagens de programação disponíveis na Unity temos: C#, Javascript e Boo Script.

#### 3.3.1 Organização da Unity

A Unity é uma plataforma de desenvolvimento de jogos, que organiza os jogos em desenvolvimento através de cenas, onde cada cena é composta de objetos, como o ambiente e os obstáculos (UNITY, 2014). Todos os objetos da cena podem ser do tipo *GameObject* e podem possuir diversos *Scripts*, contanto que cada um destes *Scripts* derive da classe *MonoBehaviour*. A extensão desta classe permite que o sistema faça chamadas para os métodos *Start*, *Update* e *Awake* que são úteis para o desenvolvimento dos jogos.

Nesta plataforma, para visualizar uma cena é necessário pelo menos uma câmera que deve ser posicionada e rotacionada de forma a focalizar o que se deseja visualizar. Esta câmera pode mostrar a imagem produzida diretamente no monitor, ou salva-la em uma textura, para posteriormente esta ser utilizada.

#### 3.3.2 Física e detecção de colisão na Unity

A Unity possui uma biblioteca de física, que permite simular o comportamento real dos objetos. Para tanto, a *engine* armazena informações como a massa, velocidade, aceleração e arrasto de cada objeto e além da gravidade do ambiente. Para que um modelo possua estas

propriedades físicas, deve-se adicionar a classe *RigidBody*, que também permite a adicionar forças ou alterar a velocidade do modelo em 3D. Neste sistema foram ajustadas estas variáveis de forma a permitir uma movimentação natural do paciente no ambiente virtual.

A detecção de colisões na Unity pode ser implementada através de dois tipos de métodos, a propriedade que determina qual o tipos destes métodos vão ser invocados é dado pela variável booleana *IsTrigger*, que está presente nas classes do tipo *Collider*. Estes métodos que são invocados durante a colisão podem ser sobrescritos, para assim .Se um objeto estiver com a variável *IsTrigger* com o valor verdadeiro, este não sofrerá ações físicas da colisão e ao colidir com outro objeto poderá ativar os métodos *OnTriggerEnter*, *OnTriggerExit* e *OnTriggerStay*. Caso o valor da variável *IsTrigger* seja falso, o objeto sofrerá uma força proveniente da colisão e poderá invocar os métodos *OnCollisionEnter*, *OnCollisionStay* e *OnCollisionExit*.

Quanto a estrutura geométrica, a detecção de colisões no ambiente da Unity é realizada utilizando a interface *Collider*, esta interface é implementada por diversas classes da *engine*, como a *MeshCollider*, *BoxCollider*, *SphereCollider* e a *TerrainCollider*. Cada classe possui uma propriedade diferente, relacionada a complexidade do modelo e a custo de processamento, a *SphereCollider* é uma classe de menor custo computacional, que representa apenas esferas, enquanto a classe *MeshCollider* permite representar qualquer modelo em 3D porém possui um custo computacional elevado (Unity, 2014).

### 3.3.3 Sons e Efeitos Musicais

Os efeitos musicais da Unity podem ser adicionados ao projeto através de diversas formas. A Unity suporta os formatos de arquivo de áudio WAV, AIFF, OGG e MP3, onde somente os dois últimos são formatos de dados comprimidos. Para utilizar estes arquivos importados esta *engine* possui as classes *AudioClip*, *AudioListener* e *AudioSource*. A classe *AudioSource* armazena o arquivo de música ou efeito sonoro, enquanto a classe *AudioClip* é responsável por controlar quando os *AudioSource* é ativado e como este é disperso no ambiente 3D. A classe *AudioListener* informa a posição do ouvinte no cenário e reproduz nos alto-falantes o som proveniente de todos os objetos que possuem um *AudioClip* com o som ativo.

### 3.3.4 GUI

Existem diversas formas de implementar a interface com o usuário, como padrão existe uma biblioteca de que possui uma classe principal denominada GUI. Esta classe implementa métodos estáticos de desenho de botões, caixas de texto, texturas, entre outros meios de comunicação com o usuário. Estes métodos registram um objeto que é desenhado na tela a cada atualização do método `OnGUI()`, que deve ser implementado pela classe que chamará estes métodos estáticos. Em geral estes métodos recebem dois ou três parâmetros, o primeiro indica o posicionamento da estrutura que se deseja desenhar e é do tipo `Rect()`, enquanto o segundo parâmetro é o conteúdo desta estrutura, podendo ser do tipo `string`, `texture` ou `GUIContent`. O último parâmetro é opcional, este indica o estilo do desenho sendo do tipo `GUIStyle`.

### 3.4 Biblioteca Kinect for Windows SDK 2.0

A *Kinect for Windows SDK 2.0* é uma biblioteca de acesso ao *Kinect* na linguagem `C#`, que possui uma versão para a *Unity*. Esta biblioteca permite o acesso a informações como posição do *skeleton*, a imagem em profundidade e a imagem colorida. Um *skeleton* é um conjunto de *bone* que por sua vez é estrutura composta por um ponto e direção de uma determinada parte do modelo em 3D. Por padrão um objeto humanóide em 3D é segmentado em *bones* conforme a figura 4.

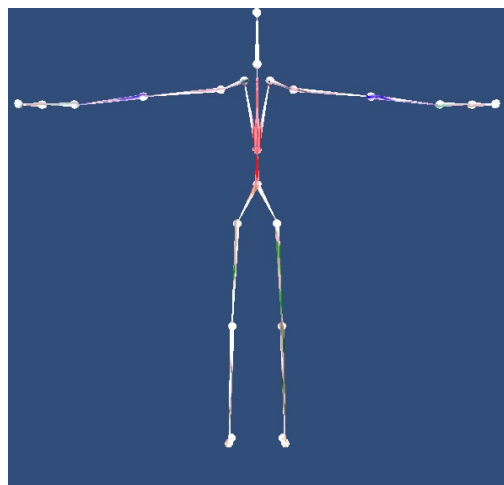


Figura 4 – Windows SDK skeleton

A biblioteca da Windows SDK possui três tipos de *scripts* de interface, que estão organizados nas pastas *KinectModelControllers*, *KinectImgControllers* e *KinectWrapper*. Os *scripts* da pasta *KinectModelControllers* são responsáveis por mapear os pontos de controle do modelo do *kinect* para um objeto 3D, enquanto os *scripts* da pasta *KinectImgControllers* são responsáveis por adicionar a textura das imagens capturadas pelo *kinect* a um modelo em 3D. da pasta *KinectWrapper* são responsáveis por conectar a biblioteca pré-compilada e armazenar as informações desta. Os *scripts* padrão implementados na Unity para ter esta compatibilidade são acessados através de uma *dll*. Neste arquivo *dll* é possível obter informações sobre a posição absoluta de cada *bone* e ângulo entre cada *bone*.

### 3.4.1 Alterações na biblioteca

Devido a implementação da biblioteca pré-compilada, foram realizadas algumas alterações na interface da biblioteca Windows SDK. Pois pode ser carregada apenas uma vez por jogo, caso contrário poderá ocorrer um erro. Por padrão ao inicializar a cena todos os objetos são construídos e ao finalizar são destruídos, porém as bibliotecas do tipo *dll* são mantidas em memória até a finalização do jogo. Se um objeto estiver com *script* da biblioteca e necessitar trocar de cena, para evitar o erro de carregar duas vezes esta biblioteca o objeto não pode ser destruído até a finalização. Para isto, foi utilizado o padrão de projeto Singleton modificado para um sistema orientado a *scripts* conforme o código 1.

```
public GameObject knPrefab{
    public static GameObject kinectObject = null;
    void Start () {
        kinectObject=GameObject.Find("kinect");
        if(kinectObject==null){
            Debug.Log ("Building Kinect Game Object");
            kinectObject = (GameObject)Instantiate (knPrefab);
            kinectObject.name = "kinect";
            GameObject.DontDestroyOnLoad (kinectObject);
        }
    };
};
```

Código 1 – Correção da biblioteca padrão do kinect



Nesta correção, é buscado um objeto padronizado para ser o *wrapper* da Unity, caso não encontre, um novo *wrapper* é instanciado e alterado para não ser destruído ao final da cena. Para que as modificações possam ser efetivas, deve-se colocar este segmento de código em todos os scripts pretcentes ao *KinectModelControllers* e ao *KinectImgController*, além de ser produzido um *prefab* dos *wrappers* da biblioteca.

A biblioteca possui duas classes na *KinectModelControllers*, uma destas realiza apenas transformações de rotações sobre o modelo que for conectada, enquanto outra realiza translações. O principal problema de trabalhar com a classe que realiza as translações é a escala, se esta operação for realizada todo o modelo sofrerá o efeito da escala. Como solução para este problema selecionou-se um ponto para determinar qual a posição relativa do personagem, e assim aplicar uma velocidade na direção deste ponto.

### 3.5 Interface do sistema

A interface foi implementada neste jogo para auxiliar o fisioterapeuta a avaliar a movimentação do paciente além de permitir a configuraração do nível de dificuldade do jogo de acordo com cada paciente. Além disto durante o jogo foram necessárias duas visualizações do jogador, uma lateral e outra frontal para avaliar a postura do paciente. A interface de configuração do jogo é mostrado na figura 6, onde à esquerda é possível configurado algumas propriedades do jogo através dos *sliders*, enquanto a direita há um conjunto de botões que permitem ver as regras do jogo, iniciar o jogo ou retornar ao menu principal. Também foi adicionado ao lado direito, uma visualização da câmera do *Kinect* para que o paciente possa se posicionar em um local adequado para o sensor.

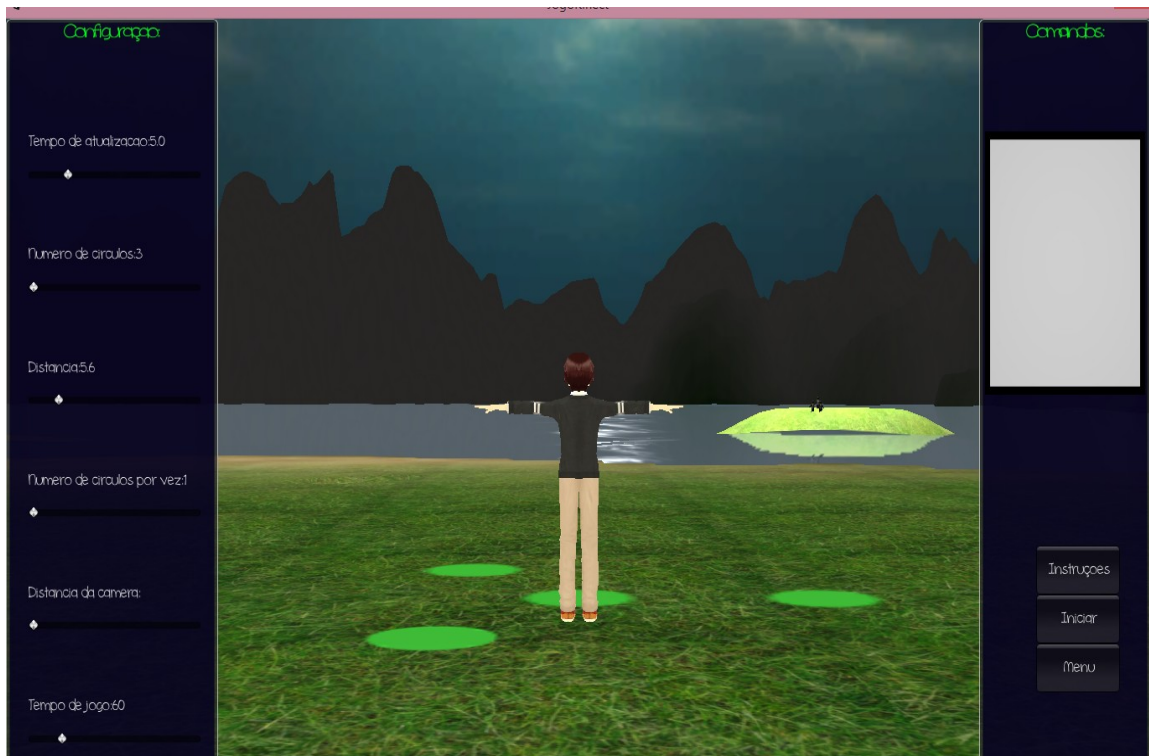


Figura 5 – Interface do jogo de equilíbrio dinâmico

Durante a configuração do jogo de equilíbrio dinâmico é mostrado todos os círculos ao redor do personagem do paciente. Estes círculos aparecerão durante o jogo de acordo com um temporizador. O jogo foi desenvolvido para que o fisioterapeuta possa alterar o nível de dificuldade para cada paciente, para tanto este pode alterar as seguintes propriedades no jogo de equilíbrio dinâmico:

- Tempo de atualização
- Número de círculos
- Distância dos círculos
- Número de círculos que aparecerão por vez
- Distância da câmera
- Tempo de jogo

A interface do jogo de equilíbrio estático foi desenvolvida de forma muito similar a forma do jogo de equilíbrio dinâmico, como pode ser visto na figura 7. O objetivo de desenvolver uma interface semelhante é facilitar a familiarização do fisioterapeuta que irá utilizá-la, pois uma vez que este aprenda a utilizar a interface de um jogo, não terá dificuldade

de interagir com a interface de outro jogo. Porém as configurações que estão localizadas do lado esquerdo da tela de configuração do jogo foram alteradas, pois os jogos possuem regras e mecânicas diferentes. Para o jogo de equilíbrio estático foi implementado além das configurações de visualização da câmera as configurações comumente utilizadas para este tipo de reabilitação, como descrito por Adams (2012), sendo estas a distância dos alvos e a altura dos alvos.



Figura 6 – Interface do jogo de equilíbrio estático

### 3.6 Modelagem do sistema

Para facilitar a modelagem do sistema, neste projeto foi utilizado a linguagem UML, pois segundo Castela (2001), esta linguagem facilita o reúso de código além de facilitar o compartilhamento de ideias. A linguagem UML utiliza o paradigma gráfico que geralmente é empregado em modelagem de sistemas orientados a objetos, devido a isto, para o desenvolvimento deste jogo foi utilizado a linguagem C# que utiliza o paradigma orientado a objetos.

### 3.6.1 Modelagem do menu principal

O menu de inicialização do sistema foi modelado de acordo com o diagrama 1, onde os casos de uso do sistema foram especificados. Este diagrama tem por objetivo descrever como o sistema pode se comunicar com o fisioterapeuta e o paciente. Dentre as principais funções que o fisioterapeuta pode realizar no menu principal temos Selecionar o jogo e Cadastrar paciente e ajustar posicionamento. No menu de inicialização o fisioterapeuta antes de iniciar o jogo, o fisioterapeuta deve cadastrar o paciente, informando os dados nome, idade e sexo. Estas informações serão armazenadas somente ao finalizar o jogo.

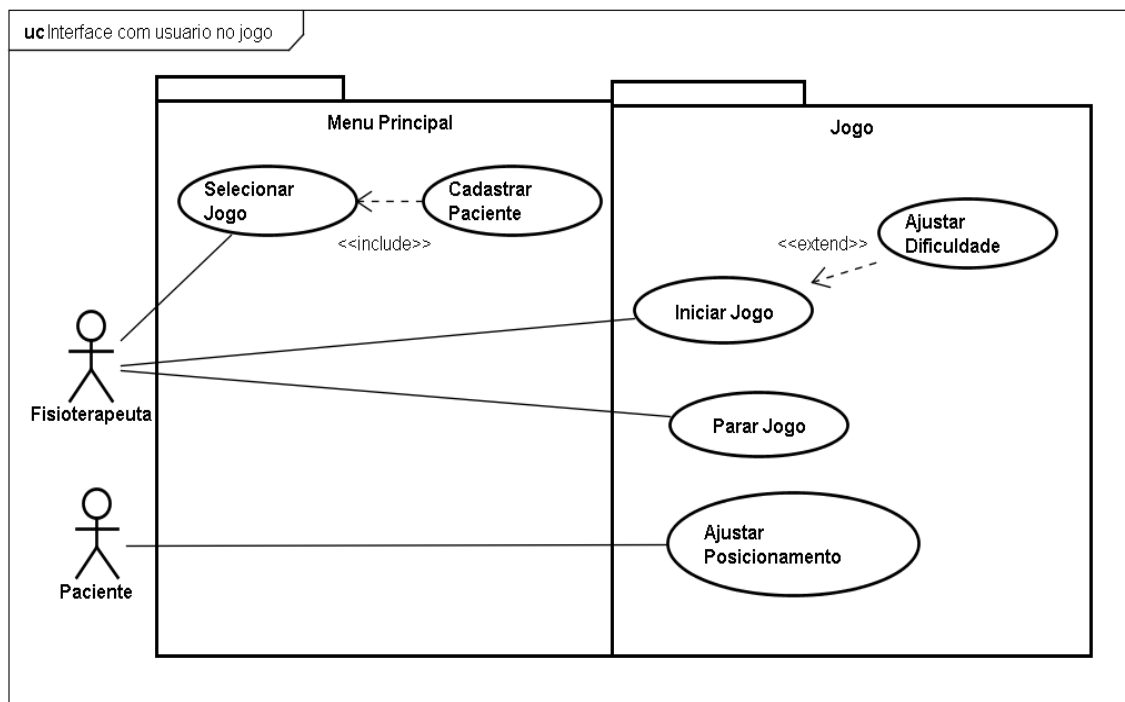


Diagrama 1 – Casos de uso do menu principal e dos jogos

### 3.6.1 Modelagem dos jogos

Para a inicialização dos dois jogos o diagrama de casos de uso possui a mesma estrutura, a diferença está na especificação de cada caso de uso. Em ambos os jogos o fisioterapeuta pode inicializar e finalizar o jogo, onde a opção ajustar o nível de dificuldade pode ser realizada de acordo com o paciente ou utilizar ao nível de dificuldade padrão, isto pode ser feito através do caso de uso Ajustar dificuldade.

A implementação dos dois jogos foi realizada de forma a aumentar o reúso de código. Isto pode ser visto nos diagramas 2 e 4 onde é especificado quais foram as classes utilizadas para implementar o jogo de equilíbrio dinâmico e o jogo de equilíbrio estático. Nestes diagramas é possível observar que as classes *ObjectFactory* e *Jogador* foram utilizadas para a implementação dos dois jogos.

Para a implementação do jogo de equilíbrio dinâmico, especificado pelo diagrama 2 está definido a classe *StateController*, esta classe é responsável por determinar em qual estado está o sistema. Enquanto a classe *MainSceneGUI* deve controlar a transição de estados e fornecer uma interface para o paciente e o fisioterapeuta. A classe *MovementGameController* gerencia todos os mecanismos do jogo, como gestos, movimentações, pontuações e tempo. Para auxiliar esta classe, foi implementado classes de controle de *Jogador* e uma classe de controle de objetos denominada *ObjectFactory*. A classe *Jogador* realiza todas as transformações de posição, para evidenciar a movimentação do paciente. Além disto esta classe calcula e fornece informações sobre o ângulo dos joelhos, ângulo dos braços e a posição do centro de massa, conforme descrito na seção 3.7.

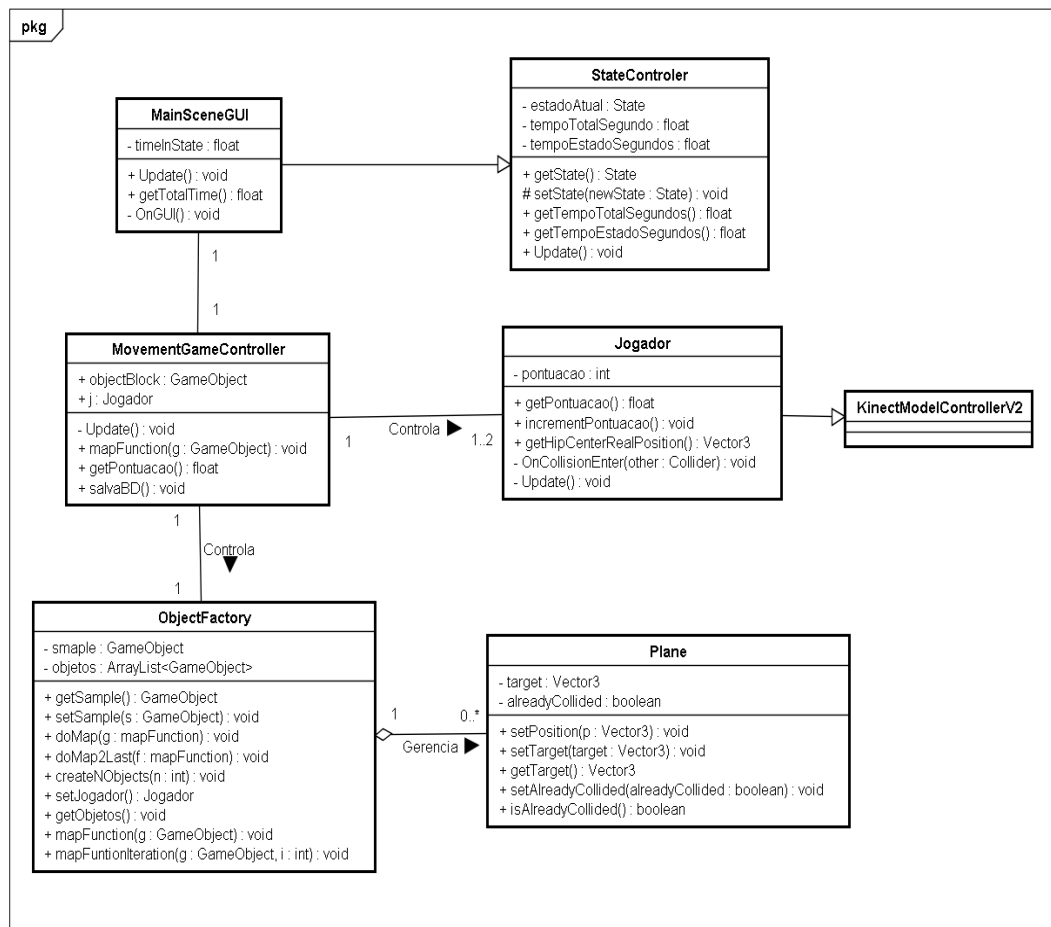


Diagrama 2 – Classes do jogo de equilíbrio dinâmico

Para a modelagem do sistema também foi construído um diagrama de sequência da operação de colisão do jogo de equilíbrio dinâmico, pois esta operação é essencial para a mecânica do jogo. O diagrama 3 é um diagrama de sequência da operação colisão, neste diagrama as ações do jogo são iniciadas a partir da chamada do método OnCollisionEnter. A chamada do método OnCollisionEnter é realizada automaticamente pela Unity através do objeto GameObject, quando o personagem colide com algum objeto que possua um Collider, como descrito na seção 3.3.2. Para discernir o objeto que colidiu com o personagem, foi utilizado a operação GetComponent e seu resultado é repassado para o controlador do jogo, pois a função desta instância é controlar o jogo, aplicando as regras. Após receber o sinal de colisão da instância do tipo Jogador, o controlador deve verificar qual é o tipo e estado da circunferência que este Jogador colidiu. Caso a circunferência que o Jogador colidiu seja a circunferência central, toda a pontuação acumulada do jogador deve ser contabilizada, e um sinal sonoro e de luz deve ser acionado. Caso a circunferência que o usuário colidiu não seja a circunferência central e esta esteja no estado não colidiu, a pontuação temporária do usuário deve ser incrementada.

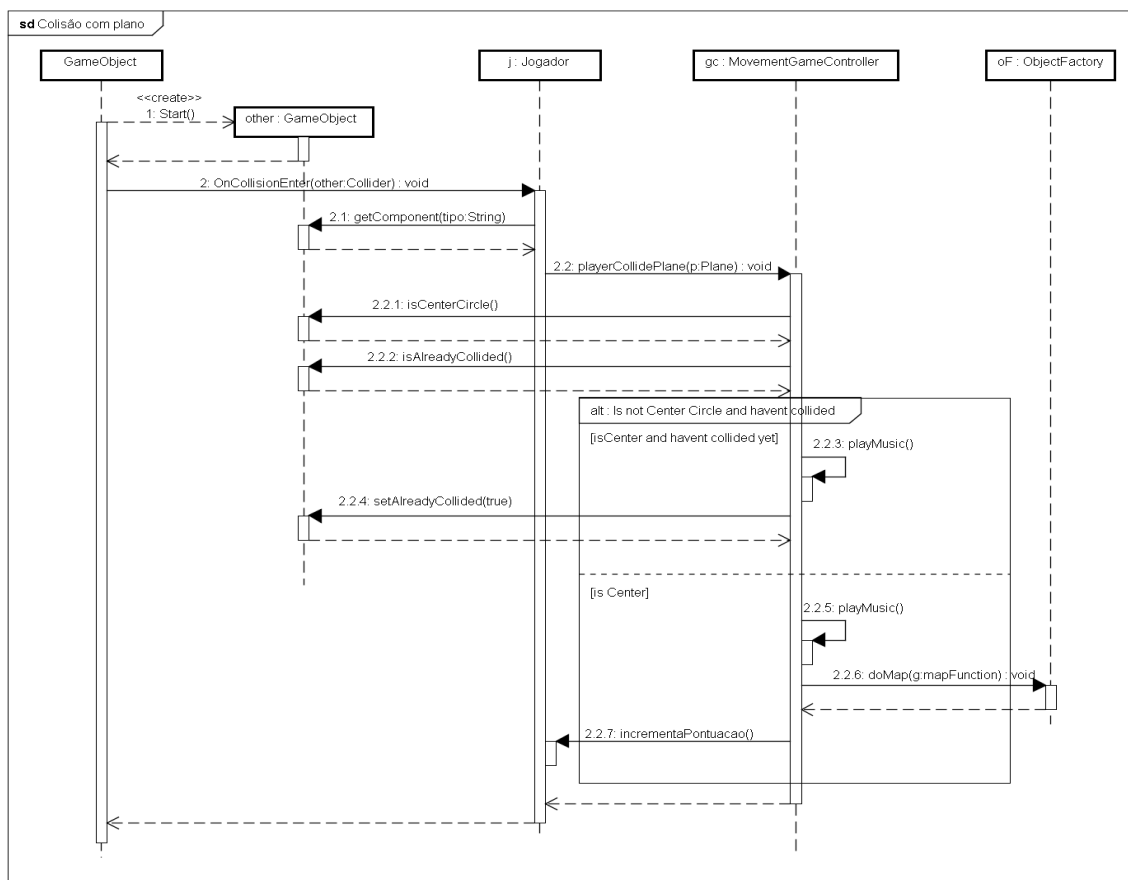


Diagrama 3 -Sequência para ativação de um plano

A modelagem do jogo equilíbrio estático foi realizado de forma similar ao jogo de equilíbrio dinâmico, permitindo o reúso das classes *ObjectFactory* e *Jogador*. A classe de controle denominada *MusicGameController* do diagrama 2 é análoga a classe *MovementGameController*, uma vez que ambas possuem a função de controlar e aplicar as regras de jogo. Além da classe *MusicGameController*, as foram implementadas as classes *MusicGameGUI*, *MusicStateController* e *BotaoNota*. Assim como a classe *MainSceneGUI*, a classe *MusicGameGUI* produz uma GUI e trata as transições dos estados relacionados a interface com o usuário. Enquanto isto a classe *MusicStateController* armazena informações sobre o estado atual do jogo através de um enumerador, onde este pode assumir os valores *esperandoSequencia*, *tocandoSequencia* e *configurando*. A classe *BotaoNota* deve ser adicionada a cada objeto em jogo que representa um botao, sendo responsável por reagir alterando a cor do objeto que está ligado e produzindo um som equivalente ao seu identificador.

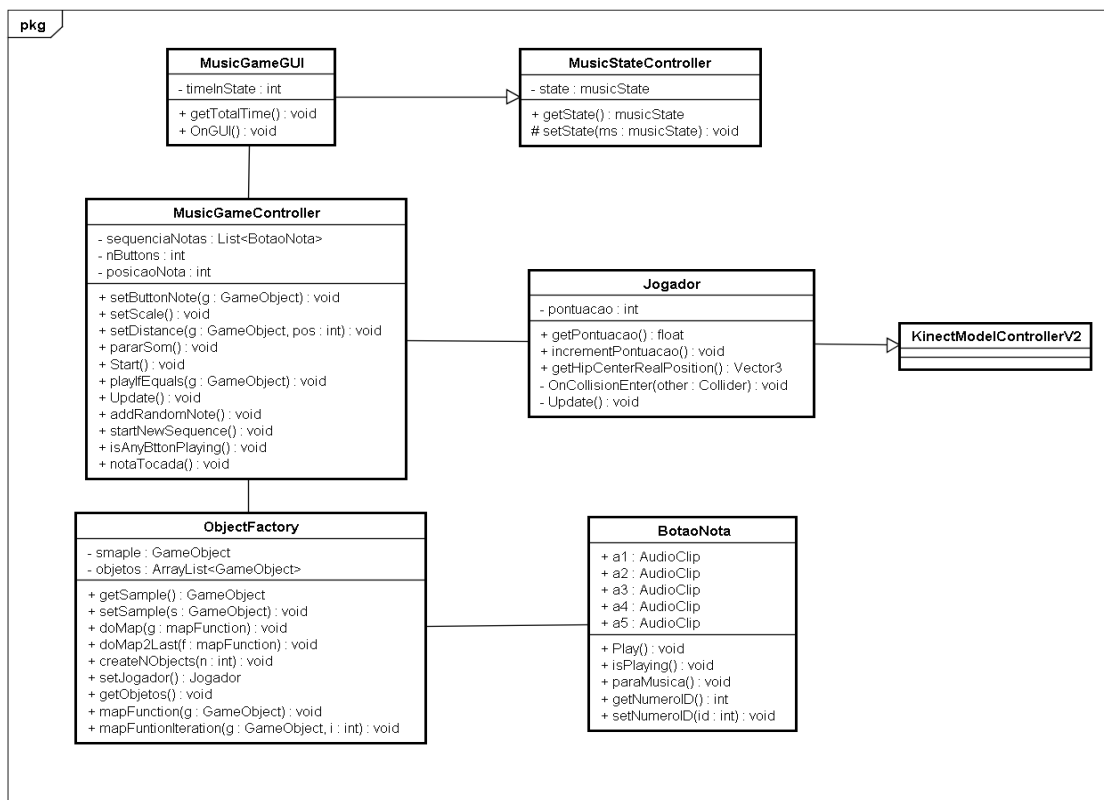


Diagrama 4 -Classes do jogo de equilíbrio estático

### 3.7 Biofeedback

*Biofeedback* é o processo de captura de informações fisiológicas, que pode ser utilizado para avaliar algum tratamento. Neste projeto, além da implementação dos jogos de

reabilitação foi implementado um sistema de *biofeedback* para o auxílio ao fisioterapeuta. Para tanto são coletadas informações de angulação do joelho, angulação dos braços e a posição do centro de massa. Estas informações são mostradas durante o jogo e ao finalizar o jogo algumas destas são armazenadas em uma base de dados conforme descrito na seção 3.8.

O ângulo do joelho geralmente são calculados utilizando um instrumento denominado goniômetro, este instrumento produz dados que são importantes para a avaliação do desenvolvimento do paciente, como descrito na seção 2.1 onde os objetivos de cada etapa são avaliados através dos ângulos que o paciente consegue realizar com os joelhos. Porém este instrumento não é capaz de avaliar durante a movimentação os valores obtidos pelo paciente e devido a isto neste projeto foi implementado este cálculo. Para o calcular estes ângulos utilizou-se dois vetores dados pela subtração da posição do pé pela posição do joelho e a subtração da posição da cintura pela posição do joelho. Sendo assim, o cálculo do ângulo entre dois vetores pode ser obtido através do arco cosseno produto escalar entre estes vetores.

O cálculo do centro de massa pode ser realizado por meio de diversos métodos, de acordo com a disponibilidade de informação. Estudos como ,utilizaram o *kinect* em conjunto do wii board para calcular o centro de massa produzindo resultados precisos, porém o uso desta ferramenta limita a mobilidade do paciente e devido a isto neste projeto foi calculado centro de massa utilizando apenas o *kinect*. Para o cálculo do centro de massa é necessário o peso percentual médio da população, que neste trabalho é dado pela tabela 2, proveniente do estudo de De Leva(1996). Com estas informações pode-se calcular o centro de massa através da soma de todas as posições mapeadas ponderada pelos seus respectivos valores da tabela.

Tabela 2 – Distribuição média de massa no corpo humano

<b>Posição</b>	<b>Massa (%)</b>
Cabeça	8.26
Tórax	20.1
Abdômen	13.06
Pélvis	13.66
Braço	3.25
Antebraço	1.87
Perna	16.68
Pé	1.43



### 3.8 Conexão com base de dados

Para que o sistema possa produzir informações aos fisioterapeutas possa demonstrar a progressão da reabilitação, o sistema precisa de uma forma de armazenamento de dados em memória estável. Para tanto, neste trabalho foi implementado um sistema de conexão com um servidor Mysql através de um script PHP na parte do servidor e na parte do cliente através das classes *WWW* e *WWWForm* da Unity. A classe *WWW* é responsável por estabelecer uma conexão com um site, enquanto a classe *WWWForm* da Unity utiliza o método *POST* da linguagem PHP para enviar informações, podendo ser textuais ou binário.

Para facilitar o uso das classes de comunicação de rede da Unity, foi implementado a classe *DatabaseConnection*. A classe *DatabaseConnection* possui os métodos *setSessionKey* e *sendDB*. A classe foi implementada utilizando o padrão de projeto *Singleton*, pois procurou-se evitar o uso de mais de uma conexão com a base de dados ao mesmo tempo durante o jogo.

O método *setSessionKey* é responsável por cadastrar as informações sobre a seção de um determinado jogo, recebendo como parâmetros o nome do paciente e o jogo. Ao ser realizado a chamada do método *setSessionKey*, o objeto do tipo *DatabaseConnection* cadastra uma nova entrada na tabela *Sessao*. Para enviar os dados para o servidor é carregado uma página no servidor que recebe os parâmetros de entrada do método, porém encriptados. Nesta tabela há um campo de data e hora, este campo do é gerado pelo script em php na hora em que o servidor recebe a requisição.

Para salvar uma propriedade do jogo foi implementado o método *sendDB*, este método foi sobrecarregado recebendo como parâmetros uma cadeia de caracteres que identifica a propriedade e os valores desta propriedade. A chave estrangeira da tabela *Sessao*, que é necessária para armazenar um novo valor nas tabelas *varVector*, *varDouble* e *varString* são enviados também para o servidor, pois este valor é armazenado internamente no objeto *DatabaseConnection*.

#### 3.8.1 Modelagem da base de dados

A modelagem da base de dados foi realizada através do software *Mysql Workbrench 6.1*. Esta ferramenta permite representar uma base de dados através de tabelas, conforme

demonstrado no diagrama 3, onde o armazenamento dos dados foi realizado utilizando 4 tabelas.

O objetivo desta modelagem era fornecer um meio genérico de armazenamento das informações sobre diversos jogos de reabilitação. Para isto foram criadas quatro tabelas, a seção é responsável por armazenar informações sobre o paciente jogo, enquanto as demais tabelas armazenam um conjunto de chave e valor de um tipo específico.

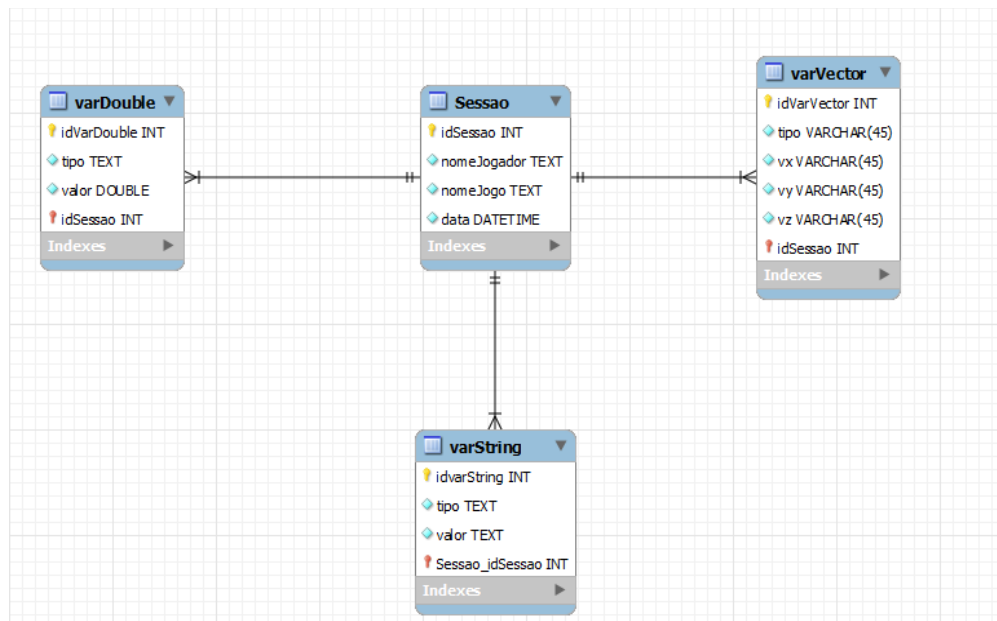


Diagrama 5 – Modelagem da base de dados

### 3.8.2 Segurança

A segurança da base de dados é de extrema importância, pois uma falha na autenticidade dos dados pode causar falhas na avaliação do desenvolvimento do paciente. Além disto, as informações das consultas dos pacientes com os fisioterapeutas em um tratamento devem ser sigilosas, conforme a resolução nº 424 do Conselho de Fisioterapia e de Terapia Ocupacional.

Para garantir a segurança, neste projeto foram implementados dois níveis de segurança. O primeiro nível de segurança trata a comunicação entre o jogo e o servidor PHP, onde foi utilizado o algoritmo AES, pois caso um pacote seja interceptado na rede este dificilmente será traduzido. No segundo nível de segurança foi tratado os ataques por injeção de código SQL, através de um código de verificação de subsequências de termos em SQL e a criação de um usuário com permissões etritas para a escrita na base de dados.

O AES é um algoritmo de criptografia de chave simétrica desenvolvido por Daemen (1998). Para desenvolver este algoritmo, Daemen (1998) propõe o uso de um espaço finito  $GF(2^8)$ , onde os bits são representados por um polinômio, onde a posição de cada bit representa um dos coeficientes do polinômio.

Este algoritmo é composto por um conjunto ciclico de operação que são aplicados a conjuntos de blocos de 16 bits de dados, representando uma matriz de 4x4 bits. As operações básicas de cada ciclo do algoritmo são SubBytes, ShiftRows, MixColumns e AddRoundKey. A figura 7 representa esta sequencia operações utilizadas para a encriptação e descriptação de dados.

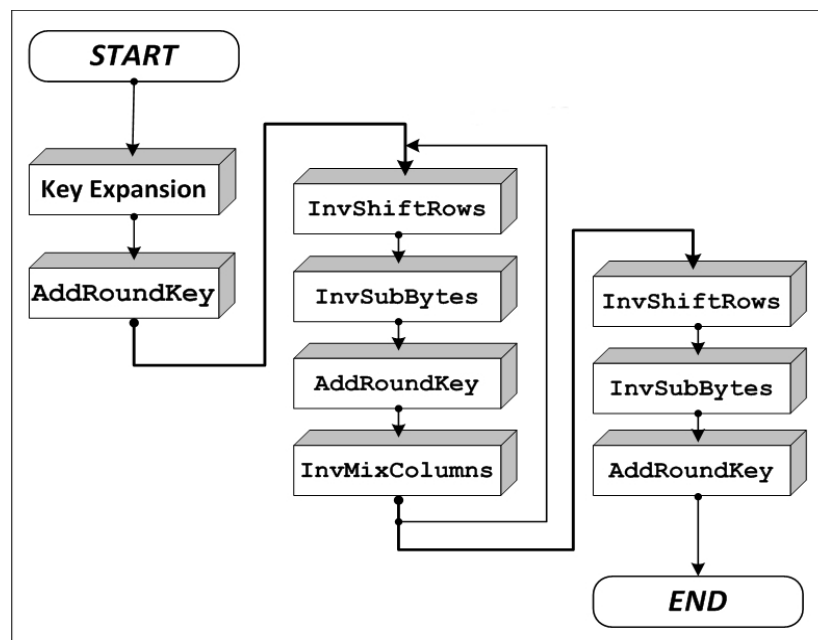


Figura 7- Encriptação AES

A operação de SubBytes é uma operação onde os dados são substituídos de acordo com uma caixa de substituição, denominada S-Box. Esta substituição pode ser revertida através de outra caixa de substituição, pois esta é uma transformação linear. A segunda operação de um ciclo da criptografia AES é a ShiftRows, onde em cada linha da matriz sofre um deslocamento de acordo com seu índice. Após a operação ShiftRows é aplicado a operação MixColumns, onde cada coluna da matriz de dados é multiplicada, por um polinômio no espaço  $GF(2^8)$ , o objetivo desta etapa é dispersar os dados.

O processo AddRoundKey é a última operação de cada ciclo, esta operação tem por objetivo cifrar os dados utilizando a chave. Para tanto, os dados são combinados através da operação XOR, com uma chave que é derivada da chave utilizada na entrada.

O resultado do processo de encriptação é uma sequência de bytes, que para ser enviada para o servidor deve ser convertida para uma sequência de caracteres. Em testes realizados com os dados foi observado que o padrão UTF8 e o padrão ASCII que utilizam 32 bits para cada caracter muitas vezes falha ao tentar representar os dados encriptados, pois o número de bits para indexação dos caracteres era insuficiente para representar todas as combinações possíveis de blocos de 32 bits. Como solução para tal problema, a sequência de bits foi separada em blocos de 32 bits e então convertida utilizando a base 64 bits.

### 3.8.3 Exemplos de consulta

A base de dados possui um padrão de consultas que será utilizado para desenvolver através de um *framework* um sistema de visualização dos dados. Este sistema de consulta precisa garantir a segurança, pois estas informações devem ser acessadas apenas pelo profissional da área de saúde, sendo necessário um sistema de autenticação de usuário. Além disto é necessário que este sistema evite ataques de injeção de código em SQL, para garantir a integridade dos dados armazenados. Para tanto há dois padrões de consultas, como exemplificado nos códigos 2 e 3, onde na primeira figura é buscado o conjunto de tuplas data e valor do jogador Rubens quando jogava o jogo “Telecinese”, enquanto na segunda figura é demonstrado uma consulta de desempenho da mesma característica, porém com todos os jogadores.

```
SELECT data, valor
FROM Sessao s INNER JOIN vardouble vd
WHERE s.nomeJogador='Rubens'
      AND s.nomeJogo LIKE 'Telecinese'
      AND vd.tipo LIKE 'Pontuacao'
```

Código 2 – Consulta de desempenho de um determinado jogador em um jogo

```
SELECT data, valor
FROM Sessao s INNER JOIN vardouble vd
WHERE s.nomeJogo LIKE 'Telecinese'
      AND vd.tipo LIKE 'Pontuacao'
```

Código 3 - Consulta de desempenho de todos os pacientes em um jogo

## **4 RESULTADOS E DISCUSSÃO**

Nesta seção estão descritos os testes realizados para avaliar a eficácia do sistema. Os testes descritos nesta seção foram realizados apenas com alunos da UFSM.

### **4.1 Testes do sistema**

A avaliação do sistema foi realizada através de um teste de usabilidade, para tanto foram selecionados 18 alunos de fisioterapia que atuam no Hospital Universitário de Santa Maria. Foi explicado a estes alunos o objetivo e o funcionamento do sistema, então foram aplicados dois testes, um para cada jogo. Inicialmente os alunos deveriam identificar as informações fisiológicas, como o ângulo dos joelhos e o centro de massa. Após identificarem as informações fisiológicas os alunos foram submetidos a uma rodada de cada jogo. As informações relacionadas a pontuação do jogo e ao sistema de biofeedback foram armazenadas em uma base de dados. Ao final dos testes os alunos foram convidados a responder um questionário sobre o sistema.

As questões de usabilidade do sistema aplicadas neste trabalho foram realizadas com base no questionário de Brockmyer (2009), que é utilizado para avaliar jogos. Também foram adicionados questões específicas sobre o tipo de movimento e a capacidade de identificação do sistema de biofeedback. Estas questões, que estão descritas na tabela 3, foram elaboradas através de afirmações que o aluno de fisioterapia deveria responder o grau de concordância. Em cada uma das questões foi utilizado a escala Likert pois esta escala possui distribuição homogênea e geralmente produz respostas consistentes (Croasmun, 2011). Cada resposta da escala Likert possui um valor numérico correspondente que varia de 1 a 5, onde 1 representa o maior grau de discordância com a afirmação e 5 representa o maior grau de concordância.

Tabela 3 – Questionário sobre a usabilidade dos jogos

Afirmação	Jogo de equilíbrio estático		Jogo de equilíbrio dinâmico	
	Média	Desvio Padrão	Média	Desvio Padrão
1.Os movimentos sugeridos no jogo são pertinentes para este tipo de reabilitação	4.17	0.60	4.33	0.47
2.Consigo identificar com facilidade as informações sobre o jogo e o paciente	3.89	0.74	4.28	0.56
3.A interface do jogo é intuitiva	3.94	1.03	4.33	0.47
4.Aprendi bastante com este jogo	3.72	0.73	3.94	0.70
5.Eu estava concentrado(a) no jogo	4.22	0.63	4.28	0.73
6.Não acho o jogo divertido	1.72	0.80	1.83	1.07
7.Me senti nervoso(a) ao jogar	1.94	0.78	1.89	0.87
8.Aprendi bastante com este jogo	3.78	0.79	3.78	0.71
9.Estou satisfeito(a) com minha performance neste jogo	2.83	0.76	3.50	1.01
10.Acho que esta atividade não é boa	1.78	0.97	1.56	0.50
11.Não me interessei por esta atividade	1.61	0.59	1.56	0.50
12.Me senti pressionado(a)	2.06	0.97	2.28	1.15
13.Os efeitos sonoros auxiliaram durante o jogo	4.44	0.60	4.39	0.59

## 4.2 Resultados e discussão

As respostas foram avaliadas e organizadas na tabela 3 onde está descrito a média e desvio padrão de cada uma das respostas. Além da média e desvio padrão, foi realizado o teste de correlação de Spearman entre as respostas obtidas, onde estes foram filtrados de acordo com a probabilidade produzindo os dados da tabela 4. Na tabela 3, é possível observar que os dois jogos são geralmente classificados como pertinentes ao problema proposto, pois a média observada é superior a 4 e possui um desvio padrão inferior a 0,7. Além disto a distribuição das respostas da questão 13 demonstra que tanto para jogo dinâmico, quanto para o jogo estático os alunos de fisioterapia concordam que os efeitos sonoros auxiliaram durante o jogo.

Tabela 4 – Teste Spearman de correlação entre as respostas das questões

	Jogo de equilíbrio dinâmico	p	Jogo de equilíbrio estático	p
Questão 10 & Questão 13	0,7	0,001	-0,95	0,000001
Questão 6 & Questão 10	0,74	0,0004	0,74	0,0004
Questão 6 & Questão 13	nc	nc	-0,73	0,0006
Questão 11 & Questão 6	0,73	0,0005	0,73	0,0003
Questão 12 & Questão 3	nc	nc	-0,73	0,0006

nc – Não computado devido a filtragem

A análise da questão 13 é importante, pois provavelmente a integração dos efeitos sonoros torna a atividade de reabilitação mais divertida se os efeitos sonoros forem condizentes com o jogo. Esta hipótese é confirmada pelo teste Spearman de correlação entre as questões 6 e 13, onde é observado a correlação de -0,73 entre as duas questões e um erro aproximado de  $6 \times 10^{-4}$ .

## 5 CONCLUSÃO

Neste trabalho foi implementado e analisado dois jogos para auxílio a reabilitação de pacientes com desordens de equilíbrio causado pela diminuição da propriocepção dos joelhos. Foi realizado um levantamento de requisitos e assim implementado um protótipo, entre os principais requisitos do sistema, como a jogabilidade, a pertinência dos movimentos, a confiabilidade do sistema de biofeedback e a segurança destas informações. Para o desenvolvimento deste sistema foi utilizado a engine gráfica Unity em conjunto a biblioteca Windows SDK 2.0 onde foi proposto a realização de alterações, para permitir que o personagem em 3D possa se transladar em sobre o espaço.

O sistema proposto atingiu a todos os requisitos, isto foi provado através de um teste de jogabilidade e usabilidade. Como resultado também foi observado que o sistema implementado é útil para reabilitação dos pacientes, pois segundo as respostas obtidas pelos alunos de fisioterapia os movimentos do jogo são pertinentes a reabilitação da propriocepção dos joelhos além de serem divertidos para os pacientes.

Para trabalhos futuros é sugerido uma análise mais detalhada do sistema e melhoria do mesmo de acordo com os resultados desta análise. Nesta análise será necessário avaliar a usabilidade e a jogabilidade com os pacientes que sofreram de perda parcial da propriocepção dos joelhos, procurando analisar também os resultados do sistema de biofeedback que são produzidos. Esta análise pode ser útil para melhorar o sistema, além de poder ser utilizados para a previsão do tempo de reabilitação e progresso dos pacientes.



## REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, D. **Current Concepts for Anterior Cruciate Ligament Reconstruction: A Criterion-Based Rehabilitation Progression.** *J Orthop Sports Phys Ther .*, Vol. 42, p. 601-614, 2012.

BOLLEN, S. **Ligament injuries of the knee-limping forward?** *British journal of sports medicine.* Vol. 32, p. 82-84, 1998.

BRASIL. Conselho de Fisioterapia e de Terapia Ocupacional. Resolução n. 424, de 01 de Agosto de 2013. Estabelece o Código de Ética e Deontologia da Fisioterapia

BROCKMYER, J. H. et. al. **The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing.** *Journal of Experimental Social Psychology.* Vol. 45, p. 624-634, 2009.

CARGNIN, D. et al. **A thoroughly approach to upper limb rehabilitation using serious games for intensive group physical therapy or individual biofeedback training.** Porto Alegre, 2014.

CASTELA, N.; et al. **Business Process Modeling with UML.** *ICEIS*, 2001.

CROASMUN, J. T.; OSTROM, L. **Using Likert-Type Scales in the Social Sciences.** *Journal of Adult Education*, Vol. 40, 2011.

DE LEVA, P. **Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters.** *Journal of biomechanics*, Vol. 29, p. 1223-1230, 1996.

FERGUSON, B. **The emergence of games for health.** *GAMES FOR HEALTH: Research, Development, and Clinical Applications*, Vol 1, p. 1-2, 2012.

GONZÁLEZ, A. et al. **Whole Body Center of Mass Estimation with Portable Sensors: Using the Statically Equivalent Serial Chain and a Kinect.** *Sensors*, Vol. 14, 2014.

GUIMARÃES, R. B.; et al. **Análise da propriocepção e sua relação com o desempenho funcional de indivíduos com deficiência do ligamento cruzado anterior.** *Revista Brasileira de Fisioterapia*. Vol. 7, p. 253-259, 2003.

HERRERO, J. A.; et al. **Análisis del entrenamiento pliométrico como trabajo de transferencia de la electroestimulación neuromuscular.** *Biomecánica*, p. 88-93, 2002.

HEWETT, T.E; PATERNO, M.V; MYER G.D. **Strategies for enhancing proprioception and neuromuscular control of the knee.** *Clin Orthop Rei Res*, p. 76-94, 2002.

IBGE-Instituto Brasileiro de Geografia e Estatística. **Censo demográfico:** resultados preliminares. São Paulo. Rio de Janeiro,2014.

KHOSHELHAM, K. **Accuracy analysis of kinect depth data.** *ISPRS workshop laser scanning*. Vol. 38, No. 5., 2011.

MICROSOFT. **Kinect for Windows Sensor Components and Specifications.** Disponível em: <<http://msdn.microsoft.com/en-us/library/jj131033.aspx>>. Acesso em: 14 Out. 2014

MORENO-GER, P.; et al. **UsabilityTestingforSeriousGames:MakingInformedDesign DecisionswithUserData.** In: *Advances in Human-Computer Interaction*, 2012.

PRENTICE, W. E. **Fisioterapia na prática esportiva:** Uma abordagem baseada em competências. 14 ed. Porto Alegre: AMGH Editora Ltda, 2012, 879 p.

ROSARIO, V. M; et al. **VERTICAL JUMP ANALYSIS USING THE MICROSOFT KINECT SENSOR.** In: *22nd International Congress of Mechanical Engineering*, 2013, Ribeirão Preto, Disponível em: <<http://www.abcm.org.br/pt/wp-content/anais/cobem/2013/PDF/487.pdf>>. Acesso em: 25 nov. 2014

SHAUGHNESSY, M.; RESNICK, B.; MACKO, R. **Testing a model of post-stroke exercise behavior.** *Rehabilitation Nursing*, v. 31, p.15-21,2006.

SILVA, A. L. P.; IMOTO, D. M.; CROCI, A. T. **Estudo comparativo entre a aplicação de crioterapia, cinesioterapia e ondas curtas no tratamento da osteoartrite de joelho.** *Acta ortopédica brasileira*, p. 204-209, 2007.

SILVESTRE, M.V; LIMA, W.C.. **Importância do treinamento proprioceptivo na reabilitação de entorse de tornozelo.** *Fisioter Mov.*, Vol. 16, p. 27-34, 2003

SUSI, T.; JOHANNESSON ,M.; BACKLUND, P. **Serious games: An overview.** 2007.

SZTURM, T.; et al. **Task-specific rehabilitation of finger-hand function using interactive computer gaming.** *Archives of physical medicine and rehabilitation*, p.2213-2217, 2008.

Unity. **Unity: Documentation.** Disponível em:  
<<http://docs.unity3d.com/Manual/index.html>>. Acesso em: 14 Out. 2014.