

UNIVERSIDADE FEDERAL DE ALFENAS

PAULO AFONSO MOLINA PEINADO

**UTILIZAÇÃO DE REALIDADE VIRTUAL E SENSORES DE
MOVIMENTO NO ENSINO DA FORÇA MAGNÉTICA SOBRE
UMA CARGA**

Alfenas/MG
2017

PAULO AFONSO MOLINA PEINADO

**UTILIZAÇÃO DE REALIDADE VIRTUAL E SENSORES DE
MOVIMENTO NO ENSINO DA FORÇA MAGNÉTICA SOBRE
UMA CARGA**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Alfnas como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Paulo Alexandre Bressan.

Alfnas/MG
2017

PAULO AFONSO MOLINA PEINADO

**UTILIZAÇÃO DE REALIDADE VIRTUAL E SENSORES DE
MOVIMENTO NO ENSINO DA FORÇA MAGNÉTICA SOBRE
UMA CARGA**

A Banca examinadora abaixo assinada aprova o Trabalho de Conclusão de Curso como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Aprovada em:

Prof. Dr. Paulo Alexandre Bressan
Instituição: UNIFAL-MG

Assinatura:

Prof. Dr. Flávio Barbieri Gonzaga
Instituição: UNIFAL-MG

Assinatura:

Prof. Dr. Frederico Augusto Toti
Instituição: UNIFAL-MG

Assinatura:

RESUMO

As dificuldades que muitos alunos apresentam na compreensão dos fenômenos físicos são conhecidas, gerando a necessidade de diversificar métodos para combater a reconhecida dificuldade na aprendizagem, que é particularmente nítida nas ciências exatas, conduzindo ao uso crescente e diversificado do computador no ensino da Física. O computador oferece atualmente várias possibilidades para ajudar a resolver problemas de ensino das ciências em geral, e da Física em particular. Pesquisas tentam comprovar as potencialidades dos recursos da informática nos processos de ensino e aprendizagem. O presente estudo objetivou desenvolver um ambiente virtual interativo utilizando conceitos de Realidade Virtual, associado à captura do movimento do corpo efetuado com auxílio do sensor Kinect®, para auxiliar professores durante aulas de magnetismo. A intenção é apresentar uma forma atrativa e interativa de ensinar e aprender Física, através de fases que desafiam o aluno, baseado na facilidade em que grande parte dos estudantes tem em interagir com novas tecnologias.

Palavras-Chave: Realidade Virtual, Sensor de Movimento, Ensino de Física, Magnetismo.

ABSTRACT

The difficulties that many students present in understanding physical phenomena are known, generating the need to diversify methods to combat the recognized difficulty in learning, which is particularly clear in the exact sciences, leading to the increasing and diversified use of the computer in physics teaching. The computer currently offers several possibilities to help solve teaching problems of the sciences in general, and physics in particular. Research tries to prove the potential of computing resources in teaching and learning processes. The present study aimed to develop an interactive virtual environment using concepts of Virtual Reality, associated to the capture of body movement performed using the Kinect® sensor, to assist teachers during magnetism classes. The intention is to present an attractive and interactive way of teaching and learning physics, through phases that challenge the student, based on the ease in which a great part of the students have in interacting with new technologies.

Keywords: Virtual Reality, Motion Sensors, Teaching Physics, Magnetism.

LISTA DE FIGURAS

Figura 1	- Movimento da esfera no campo.....	13
Figura 2	- Fases da captura do sensor de movimento, a partir do posicionamento frontal ao dispositivo (A), projeção e captura do infravermelho (B), e interpretação do mapa de profundidade com identificação do corpo humano(C).....	19
Figura 3	- Representação dos pontos de reconhecimento do corpo humano pelo Kinect®, imagem baseada na documentação Microsoft® Research (2011).....	20
Figura 4	- Funcionamento do ARToolkit por Souza e Kirner (2011).....	23
Figura 5	- Tela inicial do software.....	24
Figura 6	- Ambiente de desenvolvimento da Unity com suas principais janelas de trabalho: <i>Scene</i> (A), <i>Hierarchy</i> (B), <i>Project</i> (C), <i>Inspector</i> (D) e <i>Game</i> (E)	25
Figura 7	- Representações da esfera.....	28
Figura 8	- Ambiente da fase 1.....	28
Figura 9	- Ambiente da fase 2.....	31
Figura 10	- Definição das velocidades.....	32
Figura 11	- Ambiente da fase 3.....	35
Figura 12	- Ambiente da fase 4.....	37

LISTA DE CÓDIGOS

Código 1	- Checagem de colisão da esfera com os fios.....	29
Código 2	- Cálculo da diferença de posições para definição da velocidade..	33
Código 3	- Código responsável pelo movimento da esfera pelo campo.....	34
Código 4	- Campo magnético aleatório.....	36
Código 5	- Código responsável pela troca de esferas.....	38

LISTA DE TABELAS

Tabela 1	- Assunto apresentado em cada fase.....	14
----------	---	----

LISTA DE ABREVIATURAS E SIGLAS

- 2D** - Bidimensional ou duas dimensões
- 3D** - Tridimensional ou três dimensões
- JIT** - *Just in time*
- SDK** - *Software Development Kit*
- NUI** - *Natural User Interface*
- PC** - *Personal Computer*

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVO	11
1.1.1	Gerais	11
1.1.2	Específicos	11
2	REVISÃO BIBLIOGRÁFICA	12
2.1	MAGNETISMO	12
2.2	REALIDADE VIRTUAL	14
2.3	CONCEITOS DA ÁREA DE DESENVOLVIMENTO DE JOGOS	15
2.3.1	Coordenadas locais e globais	15
2.3.2	Vetores, <i>quaternions</i> e rotações	15
2.3.3	Renderização e Câmeras	16
2.3.4	Materiais e representação de formas	16
2.3.5	Corpos rígidos e detecção de colisões	17
2.4	UNITY – <i>GAME ENGINE</i>	17
2.5	KINECT	18
2.6	APLICAÇÕES QUE UTILIZAM O KINECT NO ENSINO	21
2.6.1	Cinefut	21
2.6.2	I Love Edros	22
2.6.3	Fusion4D	22
2.7	ENSINO DE ELETROMAGNETISMO UTILIZANDO RECURSOS DE REALIDADE VIRTUAL AUMENTADA	22
2.8	JOGOS COMERCIAIS NO ENSINO DE FÍSICA	23
3	PROJETO	24
3.1	VISÃO GERAL	24
3.2	AMBIENTE DE CRIAÇÃO E MODELAGEM GRÁFICA	25
3.3	CONEXÃO DO KINECT COM A APLICAÇÃO	26
3.4	DESENVOLVIMENTO DAS FASES	27
3.4.1	Fase Um	27
3.4.2	Fase Dois	30
3.4.3	Fase Três	35
3.4.4	Fase Quatro	36
4	CONCLUSÕES	40
4.1	CONSIDERAÇÕES FINAIS	40
4.2	TRABALHOS FUTUROS	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

O desenvolvimento das novas tecnologias de comunicação e informação nos últimos anos tem afetado vários setores da sociedade, diminuindo esforços e tempo nas atividades diárias, encurtando as distâncias e, principalmente, ampliando as possibilidades de acesso à informação. Scaramello (2002) afirma que no setor educacional, a difusão do uso dessas novas tecnologias tem permitido agilizar os serviços administrativos e elaborar materiais didáticos. No entanto, há uma grande expectativa quanto às possibilidades de anexar esses recursos no processo de ensino e aprendizagem, o que irá produzir grande impacto na forma de aprendizado.

De acordo com Oliveira et al. (2007) o ensino de conceitos físicos é usualmente criticado ou pouco interessante para os alunos devido ao alto grau de abstração exigido. Para diminuir essas barreiras, a interdisciplinaridade, interação e simulações são conceitos importantes para reverter esse quadro. Também defende que, por meio da Realidade Virtual, é possível que se tenha uma solução para esse problema, por se mostrar como forma alternativa e principalmente mais atrativa para simulação de acontecimentos reais e abstratos. Kirner e Tori (2004) afirmam que a Realidade Virtual pode ser considerada uma ferramenta para visualizar, manipular, explorar, interagir e modificar, por meio do computador, dados complexos de uma forma natural, muito semelhante ao que se faria no caso da ação sobre o dado real.

Uma forma de aumentar a interação com o ambiente virtual é a utilização do sensor de movimento Kinect® desenvolvido pela Microsoft®, que é capaz de capturar os movimentos realizados pelo jogador e reproduzi-los na tela.

Ao focar no magnetismo e no eletromagnetismo, observa-se que estes assuntos são repletos de simbologias para tentar representar propriedades, as quais muitas vezes demandam representações tridimensionais. O correto entendimento destas propriedades demanda do professor a difícil tarefa de alcançar esse entendimento apenas com a utilização de quadros ou livros.

Neste contexto, a proposta deste trabalho visa criar um ambiente virtual interativo que utiliza a Realidade Virtual e o sensor de movimento Kinect no intuito de auxiliar professores de Física no ensino de magnetismo. Com isso, espera-se fornecer ao professor uma forma icônica de ensino deixando a aula mais interativa e atrativa, além de fazer diferença na motivação e aprendizado dos alunos.

Este trabalho está organizado da seguinte maneira: No Capítulo 2, é abordado todo o referencial bibliográfico necessário para o entendimento do trabalho e uma descrição dos trabalhos publicados que defendem o uso de Realidade Virtual e sensores de movimento no ensino. O Capítulo 3 apresenta os detalhes da concepção do *software* e o resultado obtido. No último capítulo apresenta as conclusões obtidas na realização deste trabalho.

1.1 OBJETIVOS

1.1.1 Gerais

Este trabalho tem por objetivo o desenvolvimento de um ambiente virtual, que visa tornar o ensino de magnetismo mais interativo, através da captura de movimento do corpo efetuado pelo sensor Kinect, e mais atrativo, estimulando os alunos através de desafios e tarefas propostos pela mecânica do jogo, tornando o *software* uma ferramenta auxiliar durante as aulas de Física.

1.1.2 Específicos

A fim de se alcançar o objetivo geral deste trabalho, alguns objetivos específicos devem ser realizados:

- Descrição dos conceitos básicos necessários para a concepção de um ambiente virtual.
- Levantamento dos *softwares* educativos que utilizam realidade virtual;
- Análise dos atuais usos destes *softwares* no ambiente de ensino.
- Demonstração de como o ambiente virtual foi concebido e sua importância;

2 REVISÃO BIBLIOGRÁFICA

2.1 MAGNETISMO

O *software* desenvolvido neste trabalho tem o intuito de auxiliar professores durante aulas de magnetismo, principalmente no ensino de eletricidade e da Força Magnética. O *software* segue uma sequência de quatro fases, onde cada fase introduz um conceito diferente, aproveitando o que foi explicado pelas fases anteriores.

Um corpo é considerado neutro, se seu número de prótons e elétrons é o mesmo. Se o corpo perder elétrons, consequente ficando com mais cargas positivas do que negativas, considera-se que ele está eletrizado positivamente. Em contrapartida, ele estará eletrizado negativamente se ganhar elétrons, ficando com mais cargas negativas do que positivas. Esse conceito de carga elétrica é o primeiro abordado pelo *software* e se faz necessário durante todos os desafios apresentados ao usuário pela aplicação.

Forças de repulsão e atração atuam sobre partículas que possuem carga elétrica na presença de um campo elétrico. Quando estas se movimentam em um campo magnético, é a Força Magnética que começa a agir. Essa força é perpendicular à direção do deslocamento e também perpendicular à direção do campo magnético no qual ela está inserida, sua intensidade depende da direção, do sentido e do módulo da velocidade da carga, além é claro, da intensidade da carga (Ribeiro, 2008).

Considerando uma situação onde a carga elétrica está na presença de um campo magnético, surge então a atuação da Força Magnética, que é a soma da força devido às interações da carga com o campo elétrico e com o campo magnético. Assim, obtém-se a expressão:

$$\vec{F} = q \vec{v} \times \vec{B}$$

onde F é a intensidade da força, q é o valor da carga, v é a velocidade da partícula e B é o módulo do campo magnético e.

Quando uma partícula com carga elétrica entra com velocidade em campo magnético perpendicular à velocidade, cria-se uma força perpendicular à velocidade e ao campo. Esta força por sua vez provocará deslocamento na direção da partícula carregada que iniciará um movimento curvilíneo (Ribeiro, 2008). Neste movimento, a Força Magnética passa a atuar como uma força centrípeta. E o raio do movimento passa a ser dado pela fórmula.

$$R = \frac{m \cdot \vec{v}}{q \cdot \vec{B}}$$

onde R é o raio do giro provocado pela força, m é a massa da partícula, v é o módulo da velocidade, q é a carga e B é o módulo do campo magnético.

Através da equação descrita anteriormente, nota-se que o raio de curvatura é proporcional à velocidade e massa da esfera. A fórmula do raio do giro provocado pela força de Lorentz é outro ponto chave para a realização das ações durante a execução da aplicação. A Figura 1 mostra um exemplo da trajetória da partícula no campo.

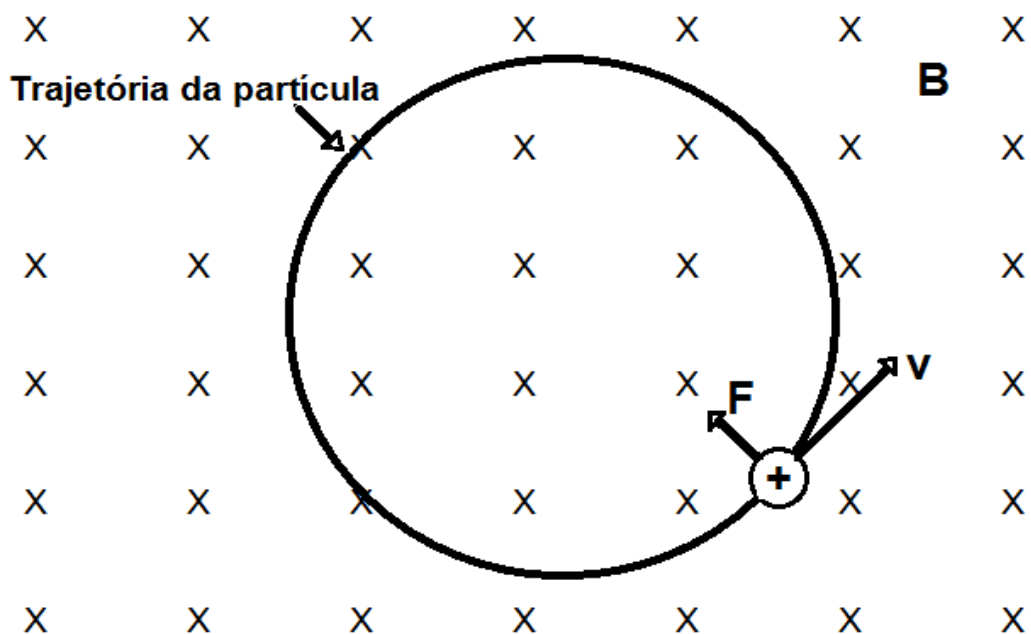


Figura 1 – Movimento da esfera no campo.

O comportamento de uma partícula no campo magnético disparada com diferentes velocidades é visto na segunda fase da aplicação. A terceira fase apresenta esse mesmo contexto tendo como diferencial a variação da direção do campo magnético. Por fim, na quarta fase o usuário deve se atentar à variação da massa e da velocidade para conseguir o raio de curvatura desejado. A Tabela 1 mostra melhor o assunto de cada fase.

Fases	Assunto
Fase Um	Conceito de carga elétrica
Fase Dois	Movimento de partículas no campo com diferentes velocidades
Fase Três	Movimento de partículas em campos com sentidos diferentes
Fase Quadro	Movimento de partículas com diferentes massas no campo magnético constante

Tabela 1 – Assunto apresentado em cada fase.

2.2 REALIDADE VIRTUAL

Segundo Pimentel (1995), o termo Realidade Virtual refere-se a uma experiência imersiva e interativa baseada em imagens gráficas 3D geradas em tempo real por computador. Machover (1994) afirma que a qualidade dessa experiência em Realidade Virtual é crucial, pois deve estimular ao máximo e de forma criativa e produtiva o usuário - a “realidade” precisa reagir de forma coerente aos movimentos do participante, tornando a experiência consistente. De acordo com Braga (2001), para que isso aconteça de uma forma eficiente, é necessária a utilização de alguns dispositivos de interface. Estes dispositivos são capazes de tornar o ambiente dinâmico e interativo, seguindo os movimentos executados pelo usuário. Estes dispositivos são mouse, teclado, luvas, monitor, sensor de movimento, entre outros.

Braga (2001) afirma que um sistema de Realidade Virtual combina estudos e recursos envolvendo *hardware*, *software*, interface de usuário, fatores humanos e aplicações, além de exigir o domínio sobre dispositivos não convencionais em computadores de alta tecnologia, computação gráfica e 3D para sua elaboração. Braga (2001) conclui que o usuário poderá desenvolver um comportamento natural e intuitivo estando imerso no ambiente criado virtualmente, buscando agir de forma natural e receber resposta para suas ações através da interação.

Com a utilização de Realidade Virtual na educação pode-se descobrir e explorar o conhecimento sobre lugares inimagináveis ou inalcançáveis. São essas possibilidades que demonstram o grande potencial da Realidade Virtual, não só através de aulas ou objetos físicos, mas também da manipulação virtual do alvo a ser explorado, analisado e estudado.

2.3 CONCEITOS DA ÁREA DE DESENVOLVIMENTO DE JOGOS

2.3.1 Coordenadas Locais e Globais

Em um sistema de coordenadas tridimensionais, a posição de um ponto pode ser representada utilizando-se três variáveis: x , y e z . Considerando o ponto como um objeto no espaço, a posição global deste objeto é representada em relação à origem do plano. Às vezes é necessário representar a posição de um objeto a outro, neste caso utiliza-se as coordenadas locais do objeto de referência, considerando que todo objeto tem seu próprio ponto de origem.

2.3.2 Vetores, *quaternions* e rotações

Vetores são segmentos de reta usados para representar direção, intensidade e sentido no plano. Através do uso de vetores é possível representar além da posição de um objeto no espaço, a sua direção.

Quaternions são utilizados para realizar a rotação de objetos no espaço tridimensional. Para representar a direção que um objeto aponta no espaço não basta apenas representar sua posição, é necessário atribuir um ângulo a ela. Quando um objeto é criado em um jogo, é definida a sua posição e um ângulo inicial a ele. Se for necessária alguma rotação nesse objeto, esta será realizada com base em sua posição e seu ângulo atual, ou seja, no *quaternion* do objeto.

2.3.3 Renderização e Câmeras

Renderização é a ação de criar imagens 2D dada uma descrição geométrica de um mundo tridimensional e uma câmera virtual que especifica a perspectiva que o mundo está sendo visto.

De acordo com Goldstone (2009), câmeras em jogos servem para representar o comportamento dos olhos em um sistema em três dimensões. Através delas são determinados quais objetos devem aparecer na tela e como eles devem ser mostrados. Elas podem ser posicionadas em qualquer lugar no espaço, possuem uma direção e um campo de visão. Esse campo de visão pode ser interpretado como uma pirâmide invertida com origem no ponto de origem da câmera e com direção determinada pela direção da câmera, assim, os pontos mostrados na tela são os que estão dentro dessa pirâmide. Ao limitar-se a distância do campo de visão, especifica-se uma distância máxima que um objeto deve estar do ponto de origem da câmera para que ele seja mostrado na tela, também é possível especificar uma distância mínima que os objetos deverão estar da câmera para que eles sejam mostrados na tela.

2.3.4 Materiais e representação de formas

Materiais servem para definir a aparência de um modelo após renderizados. Materiais são cores ou texturas, que em conjunto com um *script* (código-fonte) de

renderização, conhecido como *shader*, determina a aparência de um objeto quando visualizado na tela.

Para representar formas em três dimensões são utilizados polígonos, no desenvolvimento de jogos geralmente são utilizados triângulos, ou faces, que são representados por três pontos ligados por vértices. A combinação de polígonos permite a construção de formas mais complexas, denominadas *Meshes*, e são através desses dados que são feitos os cálculos de detecção de colisões entre um objeto e outro no espaço (Goldstone, 2009).

2.3.5 Corpos rígidos e detecção de colisões

Segundo Goldstone (2009), corpos rígidos são objetos que são afetados pelas leis da física durante o jogo, ou seja, os motores de física realizam cálculos para tornar os movimentos desses objetos mais realistas. Corpos rígidos podem possuir uma ou mais propriedades, sendo elas: massa, atrito, gravidade e velocidade.

A detecção de colisões entre objetos durante o jogo também é de responsabilidade do motor físico do jogo. Ao adicionar o elemento *collider* (colisor) à um determinado objeto, este passa a agir de um modo diferente ao colidir com outro objeto que possui um corpo rígido, geralmente alterando a velocidade e direção de seu movimento, ou agindo de acordo com um *script* pré-determinado (Goldstone, 2009).

2.4 UNITY – GAME ENGINE

Unity é uma ferramenta de criação de jogos 3D para Mac e PC. Os motores (*game engine*) são partes essenciais por trás de cada jogo digital. Desde a arte até a matemática que decide cada quadro na tela, o "motor" toma as decisões. Começando com a renderização - o método de exibição de gráficos na tela, e interagindo um método de controle e um conjunto de regras para se seguir durante o

jogo - o motor é o que um desenvolvedor constrói para servir de base ao jogo. Os modernos motores de jogos 3D são uma junção de códigos meticulosamente escritos, e como tal, uma vez utilizados para o seu propósito pretendido (que é a produção de um jogo que são feitos para), estes motores são muitas vezes vendidos, modificados e reutilizados. Devido ao nível de complexidade e custo de tais motores de jogos comerciais, a indústria de desenvolvimento de jogos é uma área de interesse difícil para novos talentos em potencial, sem estudar extensivamente linguagens de programação como C++ ou C #.

A estrutura e os comandos dos motores de jogos comerciais exigem milhares de funções e linhas de código. Este código foi entregue no Unity com a ajuda da compilação *just-in-time* (conhecida como JIT), usando a biblioteca de código aberto C# Mono. Ao usar a compilação JIT, os mecanismos como o Unity podem aproveitar a compilação de alta velocidade, em que o código que você gravará para Unity será compilado em Mono imediatamente antes de ser executado. Isso é crucial para jogos que devem executar código em momentos específicos durante o tempo de execução. Além da biblioteca Mono, a Unity também tira proveito de outras bibliotecas de *software* em sua funcionalidade, como o motor de física PhysX da Nvidia®, OpenAL para áudio além de OpenGL e DirectX para renderização 3D. Todas estas bibliotecas são integradas na aplicação, logo não existe preocupação em aprender a utilizá-las individualmente. Então, simplesmente sente-se e desfrute deles trabalhando para você sem problemas dentro da Unity (Goldstone, 2009).

A escolha do Unity tornou-se essencial devido à facilidade de comunicação entre os *softwares* de criação de objetos 3D e o sensor de movimentos Kinect®, além de contar com uma comunidade ativa e rica em documentação e seções entre as áreas de desenvolvimento.

2.5 KINECT

O Kinect é um sensor de movimentos desenvolvido pela Microsoft para o Xbox 360 e Xbox One. O Kinect criou uma nova tecnologia capaz de permitir aos jogadores interagir com os jogos eletrônicos sem a necessidade de ter em mãos um

controle. Com um quesito inovador, ele foi um elemento chave para a idealização desse projeto e muitos outros.

O Kinect possui duas câmeras: uma para a captura de cor, que permite o fácil reconhecimento do jogador; outra de profundidade, para a captura dos raios infravermelhos no ambiente, além de uma coleção de microfones. A Figura 2 ilustra a captura da presença usuário.

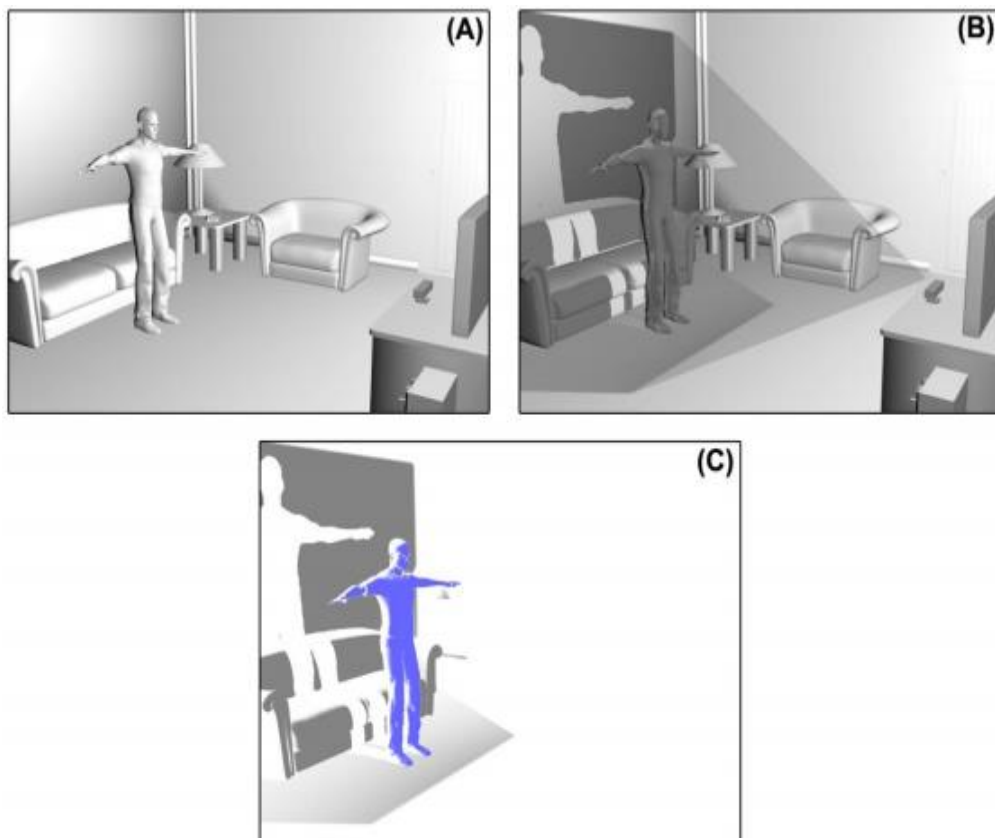


Figura 2 - Fases da captura do sensor de movimento, a partir do posicionamento frontal ao dispositivo (A), projeção e captura do infravermelho (B), e interpretação do mapa de profundidade com identificação do corpo humano(C).

Fonte: Santos et al. (2012).

Após o lançamento do Kinect e o interesse de entusiastas tecnológicos, a Microsoft® lançou o Kinect SDK (*Software Development Kit* ou kit de desenvolvimento de *software*), conjunto de ferramentas usadas para auxiliar na criação de *softwares* ligados ao Kinect.

Através da utilização do Kinect SDK, pesquisadores de diversas áreas podem adaptar o dispositivo de forma a passar e receber informações dele mais facilmente. Essas ferramentas permitem explorar as seguintes funcionalidades:

- Rastreamento do movimento do esqueleto de até duas pessoas que estão no campo de visão do Kinect;
- Acesso às informações da câmera de profundidade tridimensional (XYZ) que indica a distância dos objetos e acesso a câmera de cores;
- Captura de alto processamento de áudio que cancela ruídos e ecos possibilitando a identificação e origem do som.

A captura do esqueleto NUI (*Natural User Interface* ou Interface natural do usuário) provém de informações de localização e orientação de pontos que representam o esqueleto humano como mostra a Figura 3.

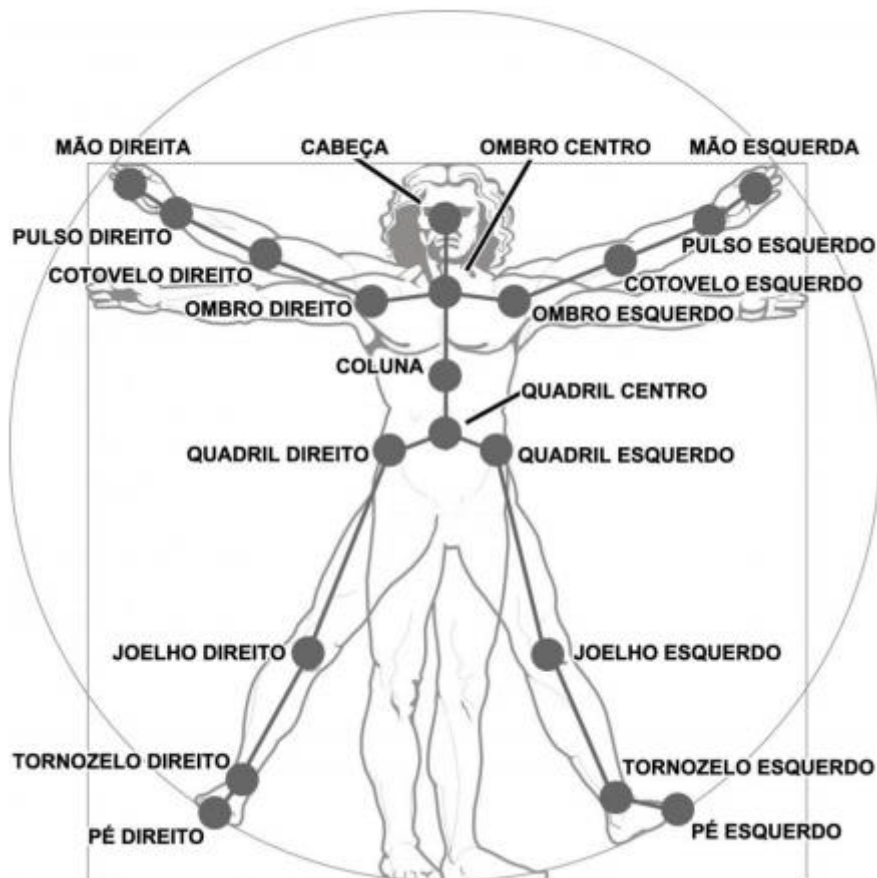


Figura 3 - Representação dos pontos de reconhecimento do corpo humano pelo Kinect.
Fonte: Microsoft Research (2011).

Hsu (2011) divulgou um artigo apresentando o potencial do Kinect como ferramenta de ensino em sala de aula. Hsu afirma no seu trabalho que o Kinect pode melhorar as práticas pedagógicas, além de ter grande potencial para aumentar a interatividade dentro de sala de aula, pois o Kinect possui a interação gestual como algo além da interação audiovisual.

Hsu (2011) ainda defende em seu trabalho que os *softwares* para Kinect precisam incorporar a concepção pedagógica, ou seja, precisa-se criar *softwares* voltados para o ensino. Entretanto para que o Kinect seja usado de maneira eficiente é necessário que o professor passe primeiro por um treino para se adaptar a tecnologia. Este trabalho é uma contribuição para o professor de Física utilizar essa tecnologia na sala de aula de Física.

2.6 APLICAÇÕES QUE UTILIZAM O KINECT NO ENSINO

2.6.1 CineFut

No trabalho de Reis (2014), foi utilizado o sensor de movimento Kinect para se ensinar princípios básicos da cinemática vetorial. O jogo gerado por ele, chamado de CineFut, apresenta basicamente o movimento oblíquo de uma bola de futebol sendo chutada por um jogador.

Reis (2014) criou para seu jogo um sistema de fases que aos poucos insere o conteúdo de física a ser abordado. As fases também aumentam o nível de dificuldade para se concluir o jogo. Tornando o conteúdo mais atraente. Por fim, concluiu que é válido o uso da tecnologia (Sensor de movimento) para o ensino de física. E afirma que a utilização desta tecnologia proporcionou maior interatividade entre professor e alunos com a tecnologia e com o conteúdo em si.

2.6.2 | Love Edros

O *software* desenvolvido por Pereira et al. (2013), tem o objetivo de inserir novas tecnologias ao ensino da geometria. No *software* são trabalhados conceitos de dualidade e simetria de poliedros, incluindo o sensor de movimento Kinect para realizar a interação do usuário com as fases criadas, onde cada fase tenta transmitir ao usuário determinado conceito relacionado à geometria.

2.6.3 Fusion4D

Com o intuito de aumentar a imersão dos alunos de medicina e de áreas relacionadas, o *software* Fusion4D, desenvolvido por Matsumura e Sonnino (2011), utiliza as tecnologias de geração de imagens tridimensionais junto ao sensor Kinect para proporcionar uma maior sensação de imersão. Um diferencial deste *software* é a utilização, não somente de comandos gestuais, mas também de comando de voz para a interação entre usuário e *software*, permitindo ao usuário manipular modelos 3D de órgãos do corpo humano.

2.7 ENSINO DE ELETROMAGNETISMO USANDO RECURSOS DE REALIDADE VIRTUAL AUMENTADA

Com o objetivo de tornar o ensino de magnetismo mais lúcido aos alunos, Souza e Kirner (2011) utilizando a tecnologia de realidade virtual aumentada construíram um simulador da experiência do físico dinamarquês Oersted. Para descrever o funcionamento da realidade virtual aumentada de seu simulador eles utilizam a Figura 4.

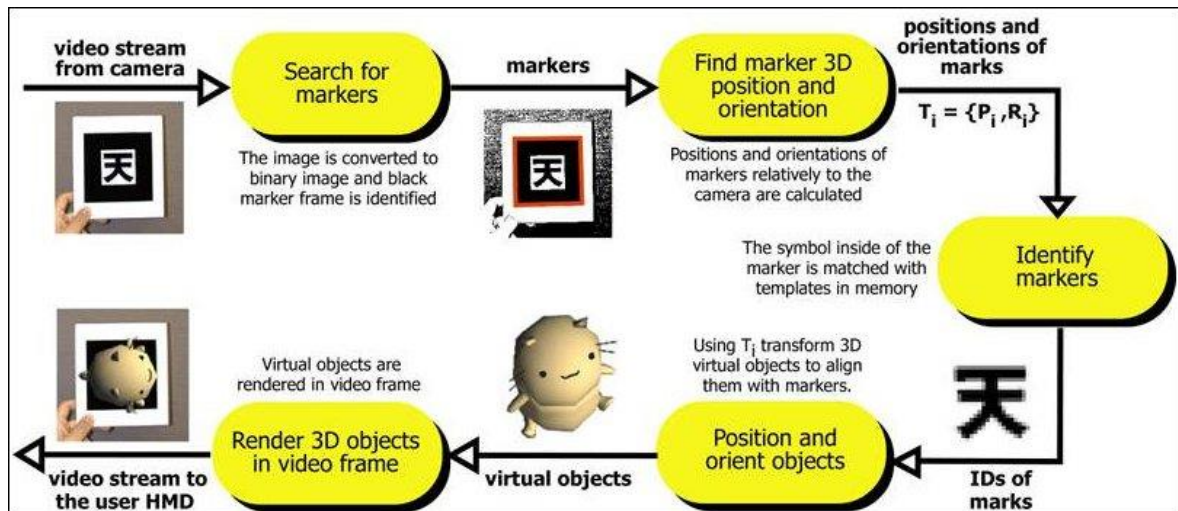


Figura 4 - Funcionamento do ARToolkit
Fonte: Souza e Kirner (2011).

O mínimo necessário para a utilização do simulador é de um computador, uma webcam, um *software* compatível com o ARToolkit e algumas plaquinhas de papel conhecidas como marcadores.

No simulador, o computador, via webcam, reconhece o marcador e insere na tela vários componentes do minilaboratório virtual com o intuito de executar a experiência. Com os componentes posicionados, outro marcador movimenta os objetos virtuais executando assim “manualmente” a experiência virtual.

2.8 JOGOS COMERCIAIS NO ENSINO DE FÍSICA

No trabalho de Sun (2015) foram analisados os efeitos dos jogos *Cut the Rope* e *Angry Birds Space* no ensino de conceitos de física. Após dividir 83 alunos em dois grupos, verificou-se que o jogo *Cut the Rope* melhorou o aprendizado dos conceitos em relação ao ensino tradicional. O mesmo não ocorreu com a utilização do jogo *Angry Birds Space*. Analisando os testes concluiu-se que os jogos comerciais não oferecem um avanço significativo para o aprendizado devido a sua falta de instruções com relação aos conceitos físicos. Assim é necessário adicionar tutoriais e ferramentas que explicam os princípios básicos da Física para não prejudicar a transferência de experiências perceptivas.

3 PROJETO

3.1 Visão Geral

O presente trabalho propõe o desenvolvimento de uma ferramenta que auxilie os professores no ensino de magnetismo, estimulando os alunos de forma divertida, de fácil manuseio e compreensão.

O *software* chamado de Magnet3D utiliza o conceito de Realidade Virtual para criar um ambiente virtual 3D, e o sensor de movimento para aumentar a imersão e interação com este ambiente.

O *software* é dividido em quatro fases, onde cada uma apresenta um conceito sobre magnetismo e objetivos a serem atingidos pelo aluno para que este possa prosseguir. O aluno, utilizando-se do sensor de movimento, deve interagir com essas fases e realizar seus objetivos utilizando uma esfera. Além disso, o usuário pode visualizar a imagem capturada pelo Kinect no canto inferior direito da tela de execução do *software* como mostra a Figura 5.

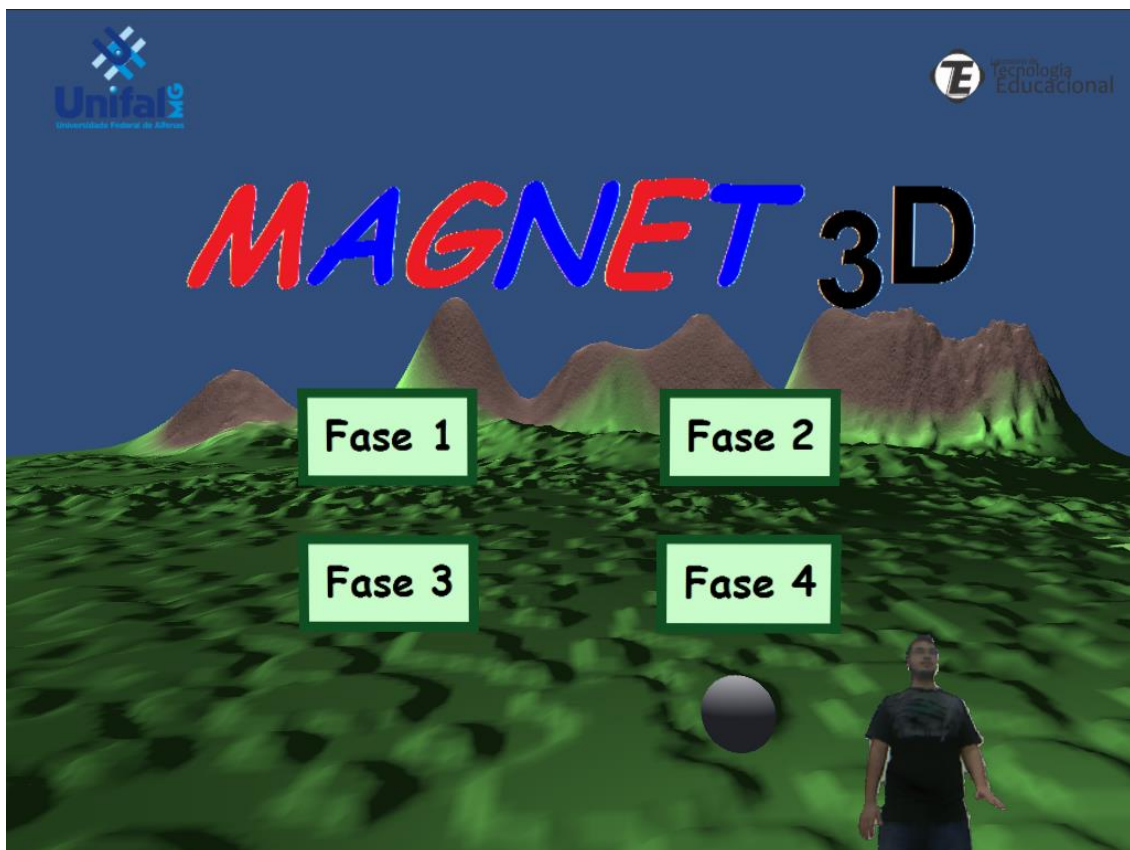


Figura 5 - Tela inicial do software.

Não há demanda de equipamentos caros e robustos para usar a aplicação em sala de aula, o mínimo necessário a ser utilizado é um computador, um monitor ou projetor e o sensor de movimento Kinect.

3.2 AMBIENTE DE CRIAÇÃO E MODELAGEM GRÁFICA

Na criação do ambiente virtual do *software* são necessários recursos gráficos e multimídias, esses elementos podem ser: texturas de imagens, superfícies, sons e modelos 3D. A Figura 6 mostra o ambiente de desenvolvimento da Unity, plataforma utilizada na criação do Magnet3D.

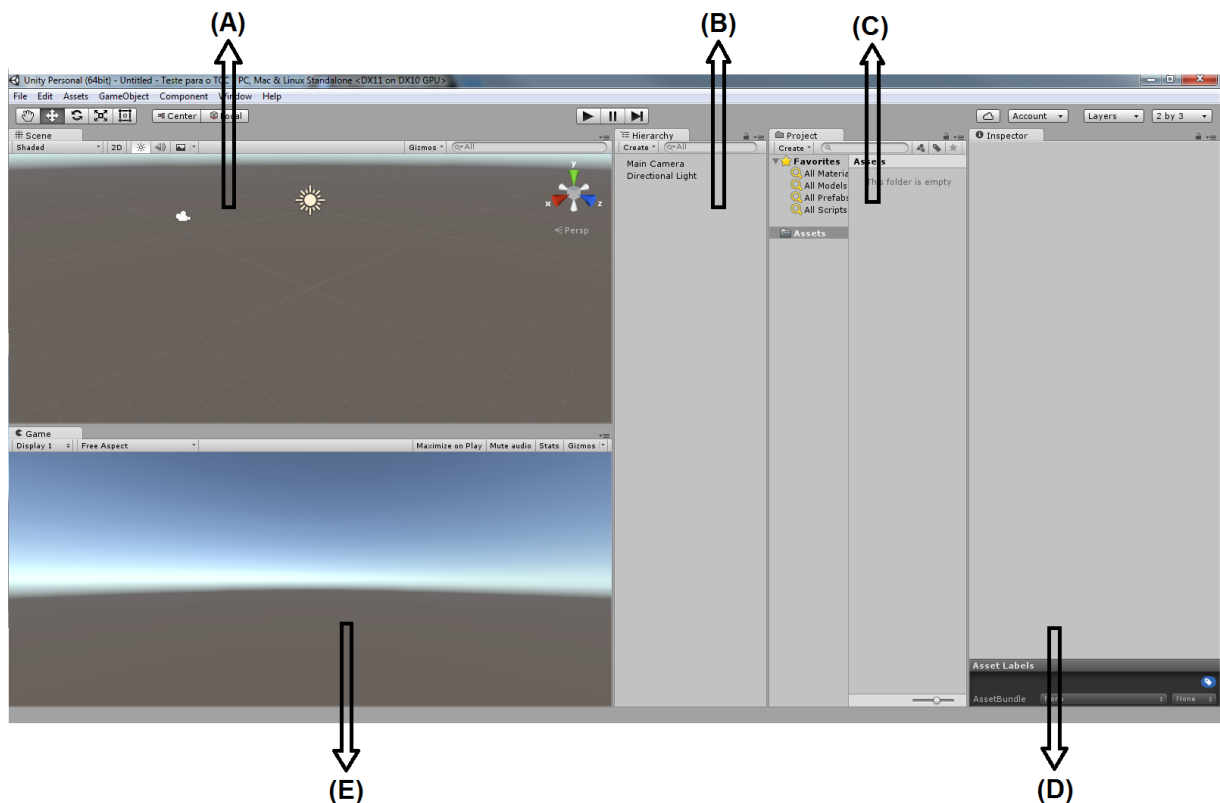


Figura 6 - Ambiente de desenvolvimento da Unity com suas principais janelas de trabalho: Scene (A), Hierarchy (B), Project (C), Inspector (D) e Game (E);

Na Figura 6 vemos as cinco principais janelas utilizadas na criação e desenvolvimento de jogos ou ferramentas através da Unity. A janela *Scene* (cena) é onde molda-se o ambiente virtual inserindo elementos visuais ao usuário, esses

elementos vão desde objetos tridimensionais modelados e animados até imagens e propriedades de iluminação, além disso um projeto criado na Unity pode ter mais de uma cena. Na janela *Hierarchy* (hierarquia) vemos os nomes de cada objeto utilizado na cena do ambiente virtual bem como a relação entre estes. Cenas diferentes vão expor diferentes objetos nesta janela. A janela *Project* (projeto) contém os elementos utilizados na aplicação, podendo estes ser: modelos 3D, texturas de imagens e *scripts* (código-fonte). A janela *Inspector* (inspetor) é uma das mais importantes, pois nela podemos “inspecionar” as características de cada elemento e sua posição na cena, essas características variam de acordo com o objeto selecionado. Nela também podemos anexar um *script* a um objeto, definindo seu comportamento. Por fim a janela *Game* (jogo) onde podemos ver a execução das cenas criadas no projeto, observando o comportamento de seus elementos de acordo com os *scripts* utilizados.

Ao longo de cada fase o usuário se encontra dentro de diferentes *containers*, estes possuem objetos variados em seu interior, tais como: sensores, bancadas e fios desencapados.

3.3 CONEXÃO DO KINECT COM A APLICAÇÃO

O sensor de movimento Kinect é parte fundamental deste projeto, pois ele captura os movimentos corporais e os transformam em elementos de entrada para o computador, assim o sistema atribui os movimentos de membros superiores e inferiores e de todo o corpo em comandos que operam os quesitos do sistema. Mas para que o sistema possa fazer isso é necessário o Kinect SDK, *software* que dá acesso e controle sobre as funcionalidades do sensor e às propriedades das imagens capturadas. Com a utilização do Kinect SDK, o *software* Magnet3D tem acesso aos comandos necessários para manipular essas informações através de linhas de código-fonte. Esses códigos são obtidos através do “Kinect *with* MS-SDK”, protótipo gratuito disponível na própria Unity.

O Kinect *with* MS-SDK é um protótipo de aplicação onde o usuário controla um personagem virtual através de seus movimentos, mas para o Magnet3D são

necessários apenas dois *scripts* indispensáveis encontrados neste protótipo, são eles: KinectManager e KinectOverlayer.

A classe KinectManager é responsável por gerenciar a captura das imagens, indicando se o Kinect está calibrado ou se algum usuário está posicionado à sua frente. Se um usuário já posicionado à frente do Kinect, essa classe começa o mapeamento do esqueleto de conforme mostrado anteriormente. Após isso entra em ação a classe KinectOverlayer responsável por reproduzir na tela as imagens trabalhadas pela classe KinectManager, essas imagens podem ser exibidas da forma que o desenvolvedor desejar.

3.4 DESENVOLVIMENTO DAS FASES

O software Magnet3D foi desenvolvido no formato de um jogo que possui quatro fases. Cada fase apresenta objetivos a serem realizados pelos alunos, ensinando um conceito de magnetismo, ao longo de cada fase esse conceito é aprimorado e a dificuldade dos objetivos aumenta. Como dito anteriormente, as fases são ambientadas em diferentes *containers* e o usuário deve interagir com o ambiente utilizando uma esfera controlada pelo movimento de seu braço direito. Todas as fases exibem seus objetivos na forma de mensagens na tela e começam com a esfera neutra.

3.4.1 Fase Um

Esta fase tem como objetivo apresentar as primeiras movimentações do jogo e o conceito de eletricidade. A esfera controlada pelo usuário possui três estados, podendo encontrar-se neutra, carregada positivamente e carregada negativamente. Essas formas são representadas de acordo com a Figura 7.

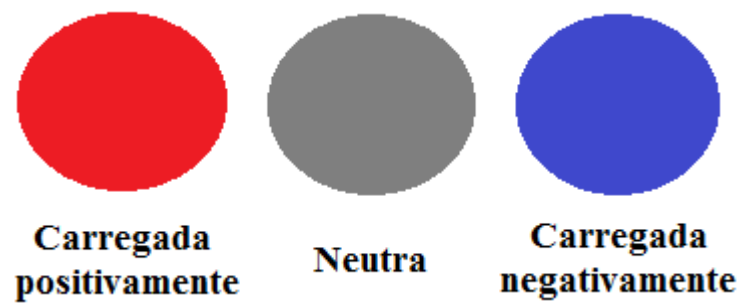


Figura 7 - Representações da esfera.

Em linha de código, a esfera neutra é representada pelo número inteiro 0 (cor cinza), quando a esfera está positiva ela é representada pelo valor 1 (vermelha) e quando está negativa recebe o valor -1 (azul). Para carregar a esfera positivamente, o usuário deve encostá-la em um fio desencapado posicionado à sua esquerda, e para carregá-la negativamente deverá fazer o mesmo com o fio posicionado à sua direita. A Figura 8 mostra o ambiente da fase.



Figura 8 - Ambiente da fase Um mostrando a esfera neutra (1), o fio positivo (2) e o fio negativo (3).

O Código 1 mostra as condições necessárias para a troca de carga da esfera. A princípio é verificado qual fio desencapado foi tocado pela esfera, como ambos

sempre adicionam ou retiram a mesma quantidade de elétrons, se ela estiver positiva e tocar no fio posicionado à direita ela irá se tornar neutra. Após a mudança de carga é chamada a função para mudar a cor da esfera de acordo com o novo valor.

```

1.  //Colisão Fio Esquerdo
2.      if (other.gameObject.tag == "fiosEsq") {
3.          if (cargaAtual != 1) {
4.              if (cargaAtual == -1) {
5.                  cargaAnterior = cargaAtual;
6.                  cargaAtual = 0;
7.              } else {
8.                  cargaAnterior = cargaAtual;
9.                  cargaAtual = 1;
10.             }
11.         trocarCorEsfera ();
12.         Fase1.GetComponent<FuncionamentoFase01>
13.             ().checarObjetivos ();
14.     }
15. }
16. //Colisão Fio Direito
17. if (other.gameObject.tag == "fiosDir") {
18.     if (cargaAtual != 1) {
19.         if (cargaAtual == -1) {
20.             cargaAnterior = cargaAtual;
21.             cargaAtual = 0;
22.         } else {
23.             cargaAnterior = cargaAtual;
24.             cargaAtual = 1;
25.         }
26.     trocarCorEsfera ();
27.     Fase1.GetComponent<FuncionamentoFase01>
28.         ().checarObjetivos ();
29. }
30. }

```

Código 1 – Checagem de colisão da esfera com os fios.

No decorrer da fase Um o usuário deve seguir os objetivos para avançar. Esses objetivos são simples mensagens que pedem para o usuário deixar a esfera neutra, positiva ou negativa. A sequência de objetivos da fase Um é a seguinte:

1. Deixar a esfera negativa;
2. Deixar a esfera neutra;
3. Deixar a esfera positiva;
4. Deixar a esfera negativa;

Durante a realização dos três primeiros objetivos o aluno é avisado por mensagens simbolizando um erro caso queira deixar a esfera negativa quando o objetivo é deixá-la positiva. No último objetivo o aluno fica livre para realizá-lo de acordo com o que aprendeu nos anteriores.

3.4.2 Fase Dois

Todas as quatro fases do jogo utilizam a mesma forma para troca de cargas da esfera, porém, a fase Dois acrescenta uma forma de arremessar a esfera atravessando um campo magnético contínuo e uniforme com o objetivo de acertar três sensores. A Figura 9 mostra o sentido do campo magnético e o posicionamento dos sensores.



Figura 9 - Ambiente da fase Dois.

Para acertar os sensores, o usuário deve pensar na equação do raio de curvatura, descrita anteriormente, considerando que a massa da esfera é constante e o ângulo de entrada no campo é sempre 90° . O *software* possui três velocidades predefinidas, que são selecionadas através do arremesso.

O usuário, controlando a esfera com o braço direito, deve selecionar o botão Arremessar, e utilizar seu braço esquerdo para arremessar a esfera. Para o arremesso, são utilizados três pontos capturados pelo Kinect em tempo real, esses pontos são as posições do pulso, cotovelo e ombro esquerdos do usuário, cada um é representado por um vetor tridimensional. Apenas os valores de y dos vetores são utilizados para a definição da velocidade, verificadas da seguinte forma:

- Velocidade 1 - Mais lenta. Se o y da mão do usuário for menor que o y de seu ombro e maior que o y de seu cotovelo, ou seja, sua mão estar abaixo do ombro e acima do cotovelo;

- Velocidade 2 - Normal ou intermediária. Se o y do ombro do usuário for maior que o y de seu cotovelo e menor que o y de sua mão, e o y de seu cotovelo menor que o y de seu ombro, ou seja, seu ombro estar abaixo de sua mão e acima do cotovelo;
- Velocidade 3 - Mais rápida. Se o y da mão e do cotovelo do usuário forem maiores que o y de seu ombro, ou seja, são mão e cotovelo estarem acima de seu ombro;

O arremesso é computado assim que a mão do usuário estiver abaixo de seu cotovelo. A Figura 10 deixa mais clara a definição das velocidades.

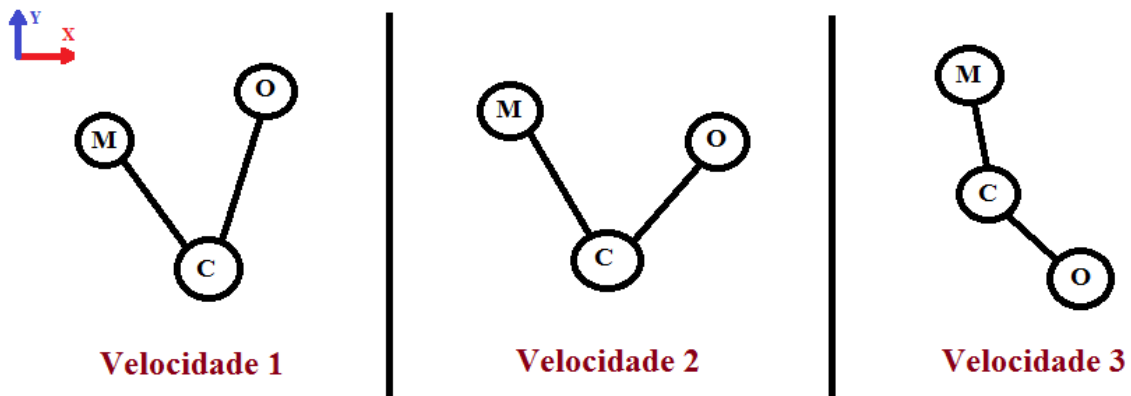


Figura 10 - Definição das velocidades.

O Código 2 realiza o cálculo da diferença de posições e definição da velocidade e executado novamente a cada *frame* até que o arremesso seja computado.

```

1. if (!preparou) {
2.     if (leftHandPosition.y > leftElbowPosition.y) {
3.         preparou = true;
4.         if (leftElbowPosition.y < (leftShoulderPosition.y -0.1f)) {
5.             velocidade = 1;
6.         } else {
7.             if (leftElbowPosition.y > (leftShoulderPosition.y +0.1f))
8.                 {
9.                     velocidade = 3;
10.                } else {
11.                    velocidade = 2;
12.                }
13.            }
14.        }
15.    } else {
16.        if (leftHandPosition.y > leftElbowPosition.y) {
17.            if (leftElbowPosition.y > (leftShoulderPosition.y +0.1f)) {
18.                velocidade = 3;
19.            } else {
20.                if (leftHandPosition.y > (leftShoulderPosition.y +0.1f))
21.                    {
22.                        velocidade = 2;
23.                    } else {
24.                        Velocidade = 1;
25.                    }
26.                }
27.            } else {
28.                lancou = true;
29.            }
30.        }

```

Código 2 – Cálculo da diferença de posições para definição da velocidade.

Após o arremesso ser computado é iniciada a movimentação da esfera pelo campo magnético. Para isso foi necessário a inserção de um objeto na cena, que não é renderizado ao longo da execução, simbolizando o centro de curvatura do movimento da esfera, ou seja, a esfera irá rotacionar em volta dele simulando o

movimento no campo magnético. O Código abaixo é responsável por indicar a posição deste objeto de acordo com a velocidade e a carga da esfera. O método *Update* é chamado a cada frame para atualizar a posição da esfera durante o movimento, após o arremesso é chamada a função *ajusteMovimento*, que é responsável por ajustar a posição do objeto, chamado de *cube*, de acordo com os atributos da esfera.

```

1. void Update () {
2.     if (start) {
3.         if (carga=0) {
4.             Transform.Translate (new Vector3 (-velocidade, 0, 0)
5.                                     *10* Time.deltaTime);
6.         } else {
7.             if (carga == -1) {
8.                 Transform.RotateAround (cube.transform.position,
9.                                         Vector.up, Velocidade *25* Time.deltaTime);
10.            } else {
11.                Transform.RotateAround (cube.transform.position,
12.                                        Vector.down, Velocidade *25* Time.deltaTime);
13.            }
14.        }
15.    }
16. }

```

Código 3 – Código responsável pelo movimento da esfera pelo campo.

Os *scripts* de colisão irão detectar caso a esfera acerte algum sensor e sua luz vermelha se tornará verde. Se o usuário errar os sensores, as variáveis receberão seus valores iniciais e ele poderá arremessar novamente.

Ao final desta fase o aluno deve entender que quando uma esfera carregada atravessa um campo magnético sobre ela surge uma força. Esta força tem sentido e direção diferenciados de acordo com a carga elétrica. Ainda é possível mostrar para o aluno que o corpo neutro não possui interação com o campo sendo assim não surge uma força magnética nele.

3.4.3 Fase Três

A fase Três aprimora o que foi visto na fase anterior, acrescentando um o botão para alterar o sentido do campo e aumentando o número de sensores a serem atingidos pela esfera. Dessa forma o usuário deve agora prestar atenção, não só na velocidade e carga da esfera, mas também com o sentido do campo magnético. A Figura 11 mostra essas mudanças.



Figura 11 - Ambiente da fase Três.

Esta fase possui quatro campos magnéticos com sentidos diferentes, são eles: para cima, para baixo, para esquerda e para direita. Para gerar o campo de forma aleatória foi utilizado o seguinte código.

```

1. public void aleatorizarCampo() {
2.     if (sensores [3] && sensores [4]) {
3.         campoSelecionado = Random.Range (0,2);
4.     } else {
5.         if (sensores [0] && sensores [1]) {
6.             campoSelecionado = Random.Range (2,4);
7.         } else {
8.             campoSelecionado = Random.Range (0,4);
9.         }
10.    }
11.    if (campoAnterior == campoSelecionado) {
12.        aleatorizarCampo ();
13.    }
14.    campoAnterior = campoSelecionado;
15.    ativarCampo (campoSelecionado);
16. }

```

Código 4 – Campo magnético aleatório.

Através do código nota-se a preocupação em se gerar campos com todos os sensores atingíveis através dele já acionados, tornando a chance de gerar um campo “útil” aleatória, podendo gerar impaciência por parte do usuário, tornando a experiência frustrante.

A ideia desta fase é mostrar que a força magnética é perpendicular ao campo e a velocidade simultaneamente. Ao mudar a direção e o sentido do campo magnético em giros de 90° auxilia o aluno a perceber claramente que quando a força surge, tem 90° com o campo magnético.

3.4.4 Fase Quatro

A quarta fase, assim como a terceira, é baseada na fase Dois, adicionado agora uma variação na massa da esfera. Nesta fase o usuário deve acertar cinco sensores posicionados de uma forma diferentes dos encontrados na fase Três, além

de alguns deles só serão acionados por esferas de massa diferente. A Figura 13 detalha o ambiente da quarta fase.



Figura 12 - Ambiente da fase Quatro.

Como se pode ver na Figura 12, agora tem-se três esferas com massas diferentes, fazendo com o usuário precise de mais velocidade para aumentar o raio de curvatura da esfera mais pesada e menos velocidade para que a esfera mais leve atinja o alvo desejado.

Para concluir esta fase o usuário deve acertar todos os sensores com as esferas específicas, sendo elas:

- Esfera leve - sensores vermelhos;
- Esfera normal - sensor azul;
- Esfera pesada - sensores amarelos.

O código abaixo é responsável por simular a troca entre as esferas de acordo com a escolhida pelo usuário, este pode mudar de esfera a qualquer momento entre os arremessos.

```

1. public void atualizarEsfera (string name) {
2.     switch (name) {
3.         case "esferaP":
4.             This.transform.localScale =
5.                 esferaPequena.transform.localScale;
6.             cursorArremesso.transform.localScale =
7.                 esferaPequena.transform.localScale;
8.             cursorArremesso.GetComponent<CursorArremesso04>
9.                 ().numEsfera = 0;
10.            esferaPequena.SetActive (false);
11.         break;
12.         case "esferaM":
13.             This.transform.localScale =
14.                 esferaNormal.transform.localScale;
15.             cursorArremesso.transform.localScale =
16.                 esferaNormal.transform.localScale;
17.             cursorArremesso.GetComponent<CursorArremesso04>
18.                 ().numEsfera = 1;
19.            esferaNormal.SetActive (false);
20.         break;
21.         case "esferaG":
22.             This.transform.localScale =
23.                 esferaGrande.transform.localScale;
24.             cursorArremesso.transform.localScale =
25.                 esferaGrande.transform.localScale;
26.             cursorArremesso.GetComponent<CursorArremesso04>
27.                 ().numEsfera = 2;
28.            esferaGrande.SetActive (false);
29.         break;
30.         default:
31.         break;
32.     }
33. }

```

Código 5 – Troca de esferas.

A troca entre esferas simulada pelo código anterior é apenas a mudança do tamanho da esfera, fixada à sua mão direita, pelo tamanho da esfera selecionada.

Durante seu percurso na quarta fase o aluno deve entender o conceito de inércia no campo magnético trabalhando a relação da massa com movimento dentro do campo magnético.

De acordo com a modelagem e implementação do *software* Magnet3D, que foi idealizado de forma exclusiva para o ensino de magnetismo, este deve servir na realização de aulas de magnetismo a fim de se atingir a aprovação de professores e alunos.

É necessário esclarecer que o jogo não exclui a presença do professor. A ideia é melhorar o processo de ensino e aprendizado em sala de aula, onde o professor deve explicar o conteúdo e demonstrar o que está sendo ensinado utilizando a aplicação. Assim facilitando o trabalho do professor e a compreensão do aluno.

4 CONCLUSÕES

4.1 CONSIDERAÇÕES FINAIS

Atualmente a tecnologia tem se tornado um potencial recurso para auxiliar professores a transmitir o conhecimento, principalmente por despertar curiosidade e prender a atenção dos alunos que, muitas vezes, se encontram entediados depois de passarem algum tempo focados apenas nos livros e no professor.

Sendo assim o presente estudo desenvolveu um ambiente virtual que permite a integração do aluno com o sensor de movimento Kinect, exclusivamente elaborado para o ensino de Física, levando o usuário a experimentar uma realidade diferente daquela presenciada diariamente na sala de aula.

É importante destacar a relevância dessa ferramenta como instrumento de ensino, pois o *software* Magnet3D é capaz de fornecer ao professor uma forma inovadora de ensino que muitas vezes não é possível vivenciar, respeitando o processo de desenvolvimento intelectual do aluno. Resta saber se a utilização do jogo em sala de aula terá impacto na motivação e aprendizado dos alunos.

4.2 TRABALHOS FUTUROS

A fim de aperfeiçoar as diretrizes do trabalho proposto e ampliar o alcance do *software* dentre outras áreas da Física, propõe-se realizar testes para validação e verificação de melhorias sugeridas pelos alunos e professores através de questionários técnicos.

Planeja-se também, acrescentar um modo de jogo onde a imagem do usuário, captada pelo Kinect, seja exibida em tamanho maior no centro da tela, em um ambiente novo onde ele possa interagir com os objetos encontrados “descobrimo” suas propriedades físicas, aumentando o nível de imersão.

REFERÊNCIAS

SCARAMELLO, J. M. **Atlas digitais escolares: proposta de avaliação e estudos de caso.** I Simpósio Ibero-Americano de Cartografia para Criança. Rio de Janeiro, 2002.

OLIVEIRA, R. P.; WERNECK, M. M.; MACIEL, F. L. **Fotofone: Experimento para Transmissão de Dados.** VII Encontro Nacional de Licenciatura em Física– Novembro. Rio de Janeiro, 2007.

KIRNER, C.; TORI, R. **Introdução à realidade virtual, realidade misturada e hiper-realidade.** Realidade Virtual: Conceitos, Tecnologia e Tendências. São Paulo, 2004.

RIBEIRO, J. E. A. **Sobre a Força de Lorentz, os Conceitos de Campo e a “Essência” do Eletromagnetismo Clássico.** São Paulo, 2008.

MACHOVER, C.; TICE, S.E. **Virtual reality.** IEEE Computer Graphics & Applications. [S.l.:s.n.], 1994, p. 15-16.

PIMENTEL, K.; TEIXEIRA, K. **Virtual reality through the new looking glass.** 2ed. New York, 1995.

BRAGA, M. **Realidade virtual e educação.** Revista de biologia e ciências da terra. Paraíba, 2001, p. 1-13.

JUNKER, G. **Pro Ogre 3d Programming: Leverage the power of Modern Real-Time Hardware-Accelerated 3d Graphics with the Best-in-Class 3d Graphics Library.** New York: Springer-Verlag New York, 2006, p. 288.

GOLDSTONE, W. **Unity game development essentials.** Mirmingham: Packt Publishing, 2009.

HSU, H. J. **The potential of Kinect in education.** International Journal of Information and Education Technology. Singapore: IACSIT Press, 2011, p. 334-338.

REIS, T. H. D. **Ensinando Conceitos de Física com Sensores de Movimento.** Simpósio Brasileiro de Entretenimento Digital. Porto Alegre, 2014.

PEREIRA, D. C., PINHEIRO, J. V. L., VAZ, C. E. V. **I LOVE EDROS – Um Game Educacional para o Ensino Da Geometria.** [S.l.:s.n.], 2013.

MATSUMURA, K.; SONNINO, R. **FUSION4D – Interface Natural e Imersiva Para Manipulação De Objetos 3D.** São Paulo, 2011.

SOUZA, R. C.; KIRNER, C. **Ensino e Aprendizagem de Eletromagnetismo usando Recursos de Realidade Aumentada.** RENOTE, v. 9, 2011.

SANTOS, J. V. S.; CARVALHO, L. C.; BRESSAN, P. A. **Physioplay: Um exergame para reabilitação física aplicando a interatividade do Kinect® como biofeedback visual.** IX Workshop em Realidade Virtual e Aumentada. Blumenau, 2012.

Microsoft® Research, Kinect® for Windows® SDK Beta - **Programming Guide,** [Documentação integrada no kit de desenvolvimento Kinect SDK]. Julho, 2011.

SUN, C.; YE, S.; WANG, Y. **Effects of commercial video games on cognitive elaboration of physical concepts.** Computers & Education, v. 88, 2015, p. 169-181.