

Desenvolvimento de Aplicação Móvel com Realidade Virtual para Práticas de Atividades Esportivas

Daniel Rocha Beck¹, Reiner Franthesco Perozzo¹

¹Curso de Ciência da Computação – Centro Universitário Franciscano Santa Maria – RS – Brasil

danielrochabeck@yahoo.com.br, reiner.perozzo@gmail.com

Abstract. *The use of technology in sports is increasingly present, in the search for better performance, monitoring of training information or simply to make some practices more attractive. In this way, this work presents the proposal of a system in virtual reality to offer a greater immersion and to make the practice of indoor exercises more pleasant using a cadence and velocity sensor to move the user through a virtual environment. The sport to be explored, as a case study, will be cycling, so the user can pedals and moves virtually, have information about his performance on the spectacle screen, through a smartphone and the use of sensors installed in the bicycle.*

Resumo. *A utilização da tecnologia no meio esportivo está cada vez mais presente, seja na busca pelo melhor desempenho, pela monitoração de informações de treino ou, simplesmente, para tornar algumas práticas mais atrativas. Dessa forma, este trabalho apresenta a proposta de um sistema em realidade virtual que oferece uma imersão maior e tornar mais agradável a prática de exercícios indoor utilizando um sensor de cadência e velocidade para movimentar o usuário em meio virtual. O esporte a ser explorado, como estudo de caso, é o ciclismo, em que o usuário pode, conforme pedala e se desloca virtualmente por um cenário, ter as informações de seu desempenho na tela do óculos, por meio de um smartphone e da utilização de sensores instalados na bicicleta.*

1. Introdução

A busca pela maestria em esportes, sempre foi algo almejado pela humanidade. Esportes tem sido amplamente praticados por inúmeras pessoas ao redor do mundo para diferentes finalidades, seja lazer, saúde ou competitividade.

A crescente tecnologia em conjunto com esportes, que concebe artefatos dos mais variados tipos para ajudar no treinamento de atletas, tais como: amortecedores para tênis, monitoramento cardiovascular, dispositivos de medição de rendimento, entre muitos outros, estão cada vez mais interconectados [Iglesias M. 2009].

A computação móvel é uma das áreas que mais tem crescido nos últimos anos. Um exemplo são os *smartphones*, considerados comuns atualmente, munidos de uma série de sensores que ajudam um atleta a se monitorar. Outro exemplo são os microcontroladores, os quais também são capazes de extrair informações de atletas para a busca contínua de ascensão no desempenho de esportes. Nesse caso, pode existir uma ligação direta entre *smartphones* e microcontroladores de uso geral [Figueiredo C. M. S., Nakamura E., 2003].

Somando-se a isso, a realidade virtual vem se destacando no auxílio para a prática de atividades esportivas *indoor*, tornando o exercício mais imersivo, real e atrativo [VirZOOM, 2017].

Dentro desse contexto, o presente trabalho propõe o desenvolvimento de um sistema que utiliza realidade virtual em atividades esportivas *indoor*.

1.1. Objetivo Geral

Construir um aplicativo móvel, com realidade virtual, para a atividade esportiva de ciclismo *indoor*.

1.2. Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Construir a aplicação para dispositivos móveis;
- Conter realidade virtual;
- Capturar a velocidade e a cadência da bicicleta por um sensor;
- Criar um ambiente 3D dentro da aplicação móvel;
- Locomover o usuário, virtualmente, de acordo com a velocidade adquirida pelo sensor;
- Mostrar as informações de cadência e velocidade na tela do *smartphone*.

2. Referencial Teórico

Nesta seção serão apresentados os conceitos e as tecnologias que englobam a presente proposta, bem como um levantamento de trabalhos relacionados.

2.1. Computação Gráfica

A Computação Gráfica é o conjunto de procedimentos e técnicas que tem como objetivo a reprodução e a manipulação de imagens digitais [Velho L., Gomes J. M. 2000]. Entre as técnicas, existe, por exemplo, o *Ray Tracing* [Karasinski E. 2008], que visa reproduzir uma imagem da forma mais real possível. Também é possível lidar com variáveis de rotação, o quaternion, determinadas por números complexos [Unity Documentation, 2017]. É uma parte da informática presente no cotidiano da vida das pessoas, seja assistindo um filme, editando uma foto ou até mesmo projetando um casa em um *software* de desenho arquitetônico, ou algo fora do comum como fazer a simulação de uma turbina de avião em ambiente virtual.

Hoje em dia, a computação gráfica é responsável, principalmente, por imagens e cenas tridimensionais em jogos, filmes, simulações, entre outros. É possível trabalhar com ambientes virtuais de duas ou três dimensões. Para facilitar trabalhos com Computação Gráfica, podem ser utilizados motores de jogos (*game engines*), que são *integrated development environment* (IDEs) e que unem bibliotecas gráficas, físicas e pacotes de funcionalidades, oferecendo recursos e ferramentas úteis ao desenvolvedor [Santos A. K. 2017]. O motor de jogos selecionado para este projeto é o Unity.

2.2. Unity

O Unity é um motor de jogos composto por uma IDE (em português: ambiente de desenvolvimento integrado) que busca facilitar a criação de ambientes virtuais, cujo foco é a criação de jogos, simuladores e animações digitais. Ele dispõe de inúmeras ferramentas, dentre elas a interface gráfica no Unity, fazendo parte do processo contornar a programação em código e se fazendo muito necessária na parte dos *scripts*.

Dentre as bibliotecas disponíveis no Unity como a biblioteca de Inteligência Artificial, de áudio, é importante destacar a de física e gráfica, a de física pois permite simular a física do mundo real definindo, por exemplo, o peso de objetos e a força da gravidade, e a gráfica, que se utiliza do Direct X (uma interface de programação de aplicativos) viabilizando o processamento de gráficos tridimensionais [Unity Documentation, 2017]. Uma característica importante do Unity é o suporte às múltiplas plataformas. O desenvolvedor pode exportar seus projetos para plataformas, como: Playstation 4, Xbox One, Nintendo Switch, sistemas operacionais Windows e Mac e plataformas móveis Android e iOS [Unity, 2017].

Atualmente, o Unity se encontra na versão 5.6, sendo que uma das novidades é o suporte nativo ao Google VR - que é a plataforma da Google para realidade virtual [Google VR, 2017].

2.3. Realidade Virtual

A realidade virtual é uma tecnologia com a capacidade de confundir os sentidos de um usuário através de imagens reproduzidas por meio de um aparelho similar a um óculos. Cada olho do usuário é exposto a uma imagem com perspectivas diferentes do mesmo ambiente virtual. Essas perspectivas simulam a distância entre os olhos ao exibirem a mesma visão, porém em ângulos diferentes [Coutinho D., 2015]. Existem dois tipos de óculos de realidade virtual: os que possuem sistema e processamento dedicado e os que fazem uso de um *smartphone*.

No óculos dedicado existem diferentes sistemas, sendo possível destacar um modelo, o HTC Vive, que, conforme o site oficial da empresa, [HTC, 2017] e [SteamVR, 2017] é um conjunto de dispositivos composto por dois sensores de parede, dois controles, o óculos de realidade virtual e o *link box*, que é o dispositivo que conecta tudo isso ao computador. O Vive é usado para uma imersão em cenários virtuais com possibilidade de movimentação.

Os dois sensores de movimento de parede são posicionados em dois vértices superiores da sala, um oposto ao outro, e são conectados no *link box* por meio de cabos. Assim como o óculos, que se conecta por cabos também e contém sensores, dentre eles uma câmera frontal. Já os controles se conectam ao link box via *wireless*, possuem bateria interna e sensores próprios também.

O Vive, além de possibilitar ver e interagir com o ambiente virtual, permite ao usuário se mover pela sala fisicamente, se deslocando também no ambiente virtual. Os controles são visíveis dentro da realidade virtual, exatamente na mesma posição onde eles estão na realidade, funcionando como as mãos do usuário. Caso o usuário se aproxime de algo na realidade, ele tem um aviso de que está se aproximando de algo físico. Equipado com a câmera frontal do óculos, não é necessário tirar o óculos para enxergar a realidade. Na Figura 1, é ilustrada uma visão geral do usuário utilizando o Vive e interagindo com os ambientes (virtual e real).



Figura 1: Mostra os dispositivos do Vive posicionados [SteamVR 2017].

Já nos modelos que fazem uso do *smartphone*, este é colocado em um suporte no óculos em frente ao visor, com a tela do aparelho posicionada de frente para o usuário. Assim, a tela é dividida pelo visor, metade para cada olho. Os sensores do *smartphone*, o giroscópio e o acelerômetro, são responsáveis pela leitura da movimentação da cabeça do usuário, para que o *software* possa atualizar a visão exibida no ambiente virtual conforme os movimentos reais do usuário [Prada R. 2009] [Silveira D. P. 2016].

Os óculos de realidade virtual que fazem uso de *smartphones* são mais acessíveis, com menor custo, por utilizarem o processamento do próprio *smartphone* para reproduzir o ambiente virtual. Além disso, os *smartphones* já contam com uma série de sensores integrados de fábrica e que podem ser utilizados para múltiplas finalidades, inclusive para aplicações de realidade virtual e práticas esportivas.

2.4. Sensores e comunicação

Sensores são dispositivos capazes de detectar e converterem grandezas físicas ou químicas proveniente do ambiente em grandezas elétricas. Assim o resultado dessa conversão pode ser enviado e processado por um outro dispositivo [Silveira C. B. 2016]. Sensores são usados para inúmeros propósitos tais como os sensores que são encontrados em *smartphones* ou sensor de velocidade e cadência presentes em alguns modelos de bicicletas.

O sensor de velocidade e cadência para bicicletas é um dispositivo em duas partes: uma delas, a principal, deve ser acoplado na bicicleta, próximo da roda traseira. A outra parte, no aro da bicicleta. Assim quando a roda gira, essa parte acoplada a roda passa perto do sensor continuamente, resultando na geração de informações como velocidade, cadência, queima de calorias, distância e tempo. Existem diversos tipos de comunicação de dados utilizadas por esses sensores, como ant+ e *wireless*, sendo que uma das mais comuns é Bluetooth [Goodrich R. 2013].

Dentre os sensores presentes em *smartphones* atuais, salienta-se o giroscópio e o acelerômetro, fazendo parte da estrutura da maioria desses dispositivos e são os principais sensores na questão relacionada com captura de movimento. O acelerômetro é um instrumento usado para medir a aceleração aplicada a ele, cujas medidas são dadas por meio de molas e corrente elétrica, sensíveis às variações dessa corrente elétrica [Prada R. 2009].

Já o giroscópio monitora a direção para onde o dispositivo está apontando. Não é sensível a sua localização ou posição. Em síntese um giroscópio é composto por um rotor, suportado por um pêndulo, feito por dois círculos articulados, com juntas do arquétipo Cardã. "O eixo em rotação tem um efeito de memória que guarda a direção fixa em relação ao círculo máximo, dispensando coordenadas geográficas" [Silveira D. P. 2016].

2.5. Trabalhos Correlatos

Nesta subseção, são descritos trabalhos com atributos similares ou que de alguma forma estão associados com a proposta do presente trabalho.

2.5.1. Zwift:

O Zwift é um *software* comercial para computadores que faz a simulação da prática de ciclismo indoor, permitindo aos usuários a estarem conectados juntos em um ambiente virtual via internet [Zwift 2017].

De acordo com [Zwift ,2017], ele funciona com o auxílio de um rolo na roda traseira, um sensor de cadência e de velocidade, e um dispositivo (*universal serial bus*) USB receptor para comunicação do sensor com a aplicação, funcionando no padrão ANT+ ou Bluetooth Smart.

A aplicação define a velocidade do personagem fazendo um cálculo usando peso, potência, resistência do ar, inclinação do terreno (*in game*) e a velocidade da roda traseira da bicicleta onde o sensor está detectando os dados. Assim se o personagem no jogo estiver subindo um terreno inclinado o usuário terá que pedalar mais, caso esteja em uma descida o usuário pode pedalar menos. A Figura 2 (a) ilustra uma visão geral do sistema do Zwift.

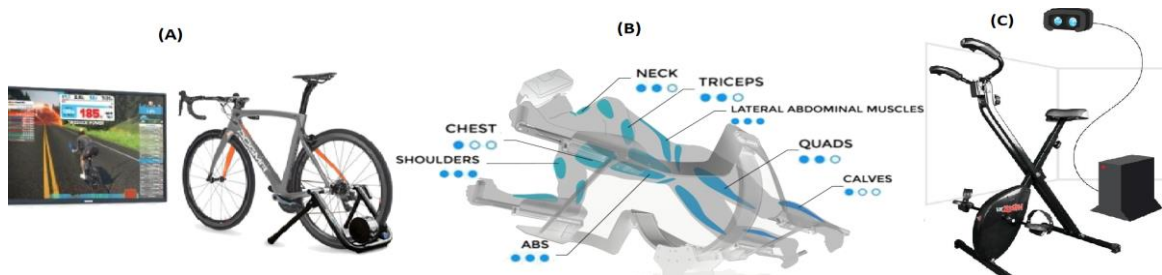


Figura 2: Estrutura do Zwift [Zwift 2017], do Icaros [Icaros 2017] e VirZOOM bike [VirZOOM, 2017]

2.5.2. Icaros:

Icaros é um aparelho de simulação que o usuário pode posicionar-se sobre ele, permitindo um movimento de inclinação para frente, para trás e para as laterais. Aliado ao software mobile em realidade virtual e um controle, o sistema simula que o usuário está voando [Icaros 2017].

Ao posicionar a estrutura do aparelho no chão, fixa-se na posição de uma das mãos do usuário o controle do Icaros, que atua como um sensor que mensura todos os movimentos feitos no aparelho. O *smartphone* é encaixado no óculos de realidade virtual, e então pode-se ter a sensação de voar em um ambiente virtual. O controle e o *smartphone* onde está sendo executado o *software* são conectados via *wireless*.

O Icaros possui ajuste para o descanso das pernas e para o descanso dos braços e oferece alguns benefícios ao corpo, exercitando os ombros, os músculos do abdome e laterais abdominais, ao tríceps e aos músculos do pescoço, além de melhorar os reflexos, capacidade de concentração e equilíbrio. Figura 2 (b) ilustra o sistema Icaros.

2.5.3. VirZOOM:

VirZOOM é um conjunto de jogos em realidade virtual para computador ou para Playstation 4. Dentre os jogos é possível encontrar alguns em que o usuário controla um blindado, um helicóptero, um cavalo e outros, possibilitando ser jogado *online* com outros usuários [VirZOOM 2017].

Ainda é disponibilizada a virZOOMbike, algo similar a uma bicicleta de academia, em que o usuário se move no jogo conforme suas pedaladas. A virZOOMbike se conecta, via Bluetooth, ao dispositivo que vai estar executando os jogos.

Os jogos do VirZOOM podem ser executados e jogados tanto com óculos de realidade virtual quanto em um monitor, também podem ser jogados com *joysticks* ao invés de utilizar a bicicleta, apesar do *software* ter sido desenvolvido com foco na VirZOOM Bike. Desenvolvido no Unity. A Figura 2 (c) mostra a visão geral do VirZOOM.

2.6. Considerações sobre os trabalhos relacionados:

Os trabalhos abordados ofereceram, uma visão de como funciona um aplicativo em realidade virtual e sua estrutura para construir o trabalho proposto. Pode-se observar o material necessário para ser feito este trabalho com o Zwift, o qual se aproxima muito com a estrutura deste trabalho. Porém o foco do Zwift é computadores de mesa e *notebooks*, diferente do trabalho proposto em que o foco é para tecnologia de *smartphones* acoplados em óculos de realidade virtual, tornando o exercício mais imersivo.

No caso do VirZOOM, ele utiliza um gerenciador de jogos e é necessária uma bicicleta específica para uso do sistema. Da mesma forma como o Icaros, que precisa de aparatos específicos para se debruçar no aparelho, o que torna sua distribuição muito custosa monetariamente. Diante disso, a proposta deste trabalho poderia se beneficiar por meio da utilização de uma bicicleta e de um *smartphone* já utilizados pelo usuário.

3. Metodologia

As metodologias tornam o processo de criação de *softwares* mais rápida, aumentam a qualidade do resultado e padronizam o procedimento, focando em sempre aprimorar perenemente. Para isso elas comumente incentivam a comunicação da equipe para eficiência na elaboração do sistema. Também se elege datas para findar processos da criação, o que aumenta a produtividade [Berbarado K. 2015]. Sendo assim, a metodologia escolhida para o desenvolvimento deste trabalho foi a *Feature Driven Development* (FDD), por ser uma metodologia pouco burocrática, coadjuvando com projetos em que existe um ou poucos colaboradores.

FDD (em português: Desenvolvimento Voltado à Funcionalidades) é uma metodologia ágil para desenvolvimento de sistemas que facilita os processos de organização, maximiza o rendimento em questão de trabalho, reduz o tempo de todo o processo, sendo pouco burocrática [Goyal S. 2008] e deixando projetos em que existem um ou poucos colaboradores, mais eficazes. Além disso, é permitido reprojeter funcionalidades. É exigido o fornecimento de datas definidas para entregas de funcionalidades. A metodologia é composta de cinco partes, descritas nos tópicos seguintes e ilustrada na Figura 3.

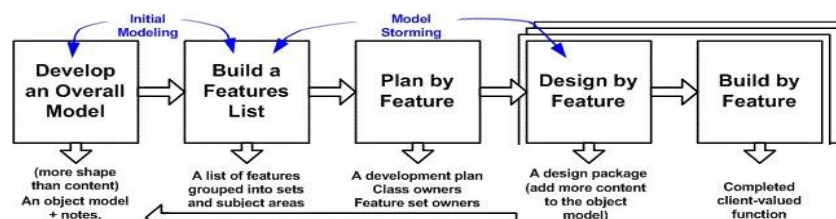


Figura 3: Estrutura do FDD. Imagem adaptada de [AgileModeling 2017].

Primeira fase (concepção e planejamento):

(I): Desenvolvimento do modelo global: etapa que ocorre o estudo do objetivo a ser atingido e a criação de um diagrama de domínio ou de classe que demonstre a estrutura do *software*.

(II): Construção da lista de funcionalidades: momento para se refinar o entendimento do escopo e onde são determinadas as funcionalidades do *software* decompondo o diagrama de domínio da etapa passada, repercutindo em uma lista de funcionalidades e suas hierarquias.

(III): Planejamento por funcionalidade: fase em que as funcionalidades pré-determinadas são escaladas em ordem de importância e dificuldade a serem feitas, determinando, assim, *deadlines* para serem entregues as funcionalidades.

Segunda fase (construção):

(I): Detalhamento por funcionalidade: é o detalhamento das funcionalidades, em que são selecionadas as funcionalidades a serem feitas. Nessa fase deve ser criado o diagramas de sequência para cada uma delas.

(II): Construção por funcionalidade: essa fase é, de fato, a implementação das funcionalidades, orientando-se pelo material resultante das fases anteriores.

Maiores detalhes poderão ser obtidos em [AgileModeling ,2017] e [Goyal S. ,2008]. Os passos realizados em cada uma das fases são apresentados na sessão 4.1 e 4.2.

4. Proposta

Este trabalho propõe um sistema de realidade virtual para ser utilizado em práticas de exercícios *indoor*. Através do uso de um óculos de realidade virtual para *smartphones* e um aplicativo que foi desenvolvido, a proposta visa imergir o usuário em um ambiente virtual tridimensional fornecendo, durante a prática do exercício *indoor*, informações de desempenho ao usuário. Essas informações serão cadência e velocidade do usuário no exercício, conforme a atividade desempenhada. O esporte escolhido para ser explorado neste trabalho é o ciclismo, pelo fato do ambiente *indoor* ser melhor controlado e ser mais efetiva a captura de informações, em comparação com outros esportes.

Uma visão geral da proposta é apresentada na Figura 4.

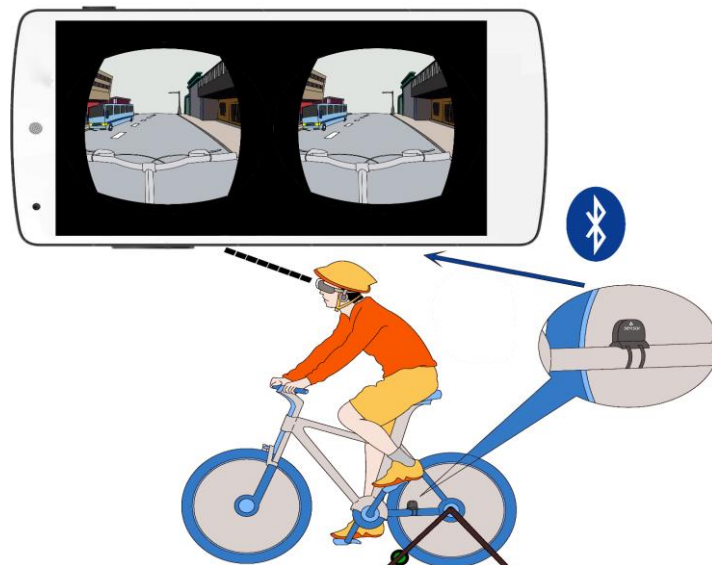


Figura 4: Visão geral do trabalho.

Conforme ilustrado na Figura 4, a proposta está centrada na realidade virtual apresentada ao esportista no momento de suas atividades físicas. Os dados de velocidade e cadência que, no exemplo de ciclismo, são obtidos por meio de um sensor e enviados, por Bluetooth, ao *smartphone* do esportista que está acoplado a um óculos de realidade virtual. O usuário, que está utilizando esse óculos, visualiza as informações na interface do simulador, sendo que os dados serão exibidos nas laterais da tela de visualização do usuário.

O uso da tecnologia de realidade virtual, junto à prática de exercícios *indoor*, agrega valor à atividade física de diversas formas. Ela poderia tornar a prática da atividade mais atraente e imersiva, ao possibilitar que diferentes cenários possam ser explorados no ambiente virtual. Também seria possível personalizar diferentes condições em cada cenário, com o objetivo de oferecer uma variação maior de experiências ao usuário. Além da realidade virtual, o sistema também coleta e exibe informações relativas à jornada diária de cada atleta.

Considerando esse cenário, a Figura 5 ilustra todo o fluxo de construção do projeto, desde a parte de implementação da aplicação de realidade virtual até o momento da execução/utilização dessa aplicação pelo atleta.



Figura 5: Fluxo do projeto

Assim, a presente proposta segue, nas próximas subseções, a estrutura da metodologia apresentada na seção 3, sendo composta pelo desenvolvimento do modelo global e pela construção da lista de funcionalidades com o seu respectivo planejamento.

4.1.Desenvolvimento do modelo global

Nesta subseção é apresentado um diagrama de domínio da aplicação de realidade virtual, contemplando a estrutura do sistema e identificar as classes conceituais e seus relacionamentos entre si e com o cenário. Assim, a aplicação móvel a ser desenvolvida conta com as seguintes classes conceituais (Figura 6):

- (i) BluetoothLEHardwareInterface: Essa classe tem todos os métodos para tornar possível a conexão entre o *smartphone* e o sensor de cadência e velocidade.

- (ii) **ConexãoBluetooth**: Por sua vez, é a classe que é empregada para manipular dados e os utilizar os métodos da classe “BluetoothLEHardwareInterface”.
- (iii) **DadosBluetooth**: Essa classe é responsável pela atualização na tela do usuário e cálculos dos valores vindos da classe “ConexãoBluetooth”.
- (iv) **MoverJogador**: Classe responsável por atribuir a velocidade à bicicleta tridimensional movendo-se no vetor Z, além de mover a bicicleta para o início do percurso caso ela ultrapasse uma distância pré-determinada dentro do cenário.
- (v) **AmbienteUnity**: Esta é a cena, onde estão todos os objetos tridimensionais do cenário, a câmera do jogador e a bicicleta.

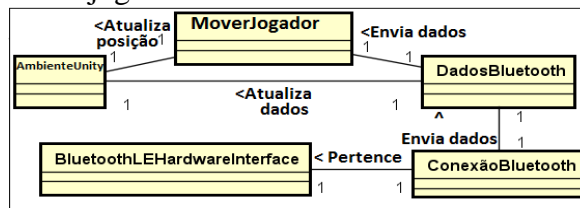


Figura 6: Diagrama de domínio da aplicação móvel.

4.2. Construção da lista de funcionalidades

Nesta subseção é apresentada a lista de funcionalidades da proposta, separadas em requisitos funcionais e não funcionais.

Requisitos Funcionais:

- Fazer usuário se deslocar conforme os dados do sensor Bluetooth;
- Permitir o usuário olhar para qualquer lado no ambiente;
- Capturar posição do óculos pelos sensores;
- Calcular dados de posição e velocidade e exibir na tela;
- Capturar dados do sensor Bluetooth;
- Atualizar as informações na tela, periodicamente;

Requisito Não Funcional:

- Criar ambiente tridimensional.

4.3. Planejamento por funcionalidade

Seguindo a construção na metodologia FDD, essa subseção mostra as funcionalidades e o tempo reservado para a construção de cada uma delas, assim foi planejado e construída as funcionalidades.

4.4. Projeto por Funcionalidade

Esta subseção expõe as fases de projeto de *software* e de *hardware* deste trabalho.

4.4.1. Projeto de *hardware*

Para a construção do projeto, foi utilizado um *smartphone* Motorola, o Moto Maxx e um sensor de cadência e velocidade de baixo consumo de energia, chamado DMTR30 (Figura 7) do fabricante Echowell [Echowell, 2017], o qual transmite tais informações, via Bluetooth.



Figura 7: Imagem do sensor de cadência e velocidade.

Assim, esse sensor pode ser acoplado a uma bicicleta (sob um rolo de treinamento) e o *smartphone* inserido em um óculos de realidade virtual para esse propósito. Entretanto, neste projeto optou-se por utilizar uma bicicleta de *spinning* (Figura 8), pelo fato da mesma estar à disposição do projeto e ser comumente encontrada em academias e algumas residências.



Figura 8: Visão geral da instalação.

Conforme ilustrado na Figura 8, o sensor de cadência é instalado no braço direito do pedivela e o sensor de velocidade instalado próximo a roda inercial da bicicleta.

Dentro desse contexto, o sensor de cadência é responsável por mensurar a quantidade de rotações por minuto (RPM) que o pedivela está girando. Em outras palavras, é a medida de RPM em que o ciclista se encontra. No que tange o sensor de velocidade, ele obtém o valor de velocidade em que a roda inercial está girando. Em síntese, a velocidade é obtida pelo cálculo entre o tempo de giro entre cada volta que a roda realiza.

4.4.3 Projeto de *Software*

Para esse projeto, foi desenvolvido um aplicativo no Unity, na linguagem c# que projeta o meio virtual, e capta as informações do sensor de velocidade e cadência para que sejam mensuradas no ambiente tridimensional, o diagrama de classes da Figura 9 ilustra a estrutura da aplicação. Essa comunicação é estabelecida com a ajuda de um *asset*, *Bluetooth LE for iOS, tvOS and Android*, da empresa Shatalmic [Shatalmic, 2017] e uma API da Google, para que seja possível tornar a visada do jogador imersiva [Google VR, 2017].

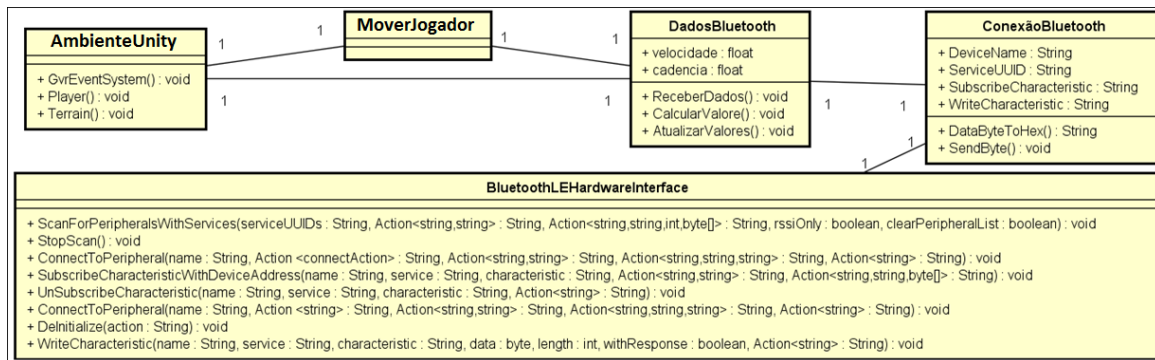


Figura 9: Diagrama de classe da aplicação.

Para esse Na Figura 9, podem ser visualizados todos os métodos utilizados em cada classe, refinando o diagrama de domínio exibido na Figura 6.

4.5. Construir por Funcionalidade

Nesta subseção é apresentada a implementação do projeto e como os recursos são utilizados.

4.5.1. Visada do óculos jogador

Para que a visão do jogador seja imersiva, foi usada uma API da Google que possibilita importar, para o projeto no Unity, uma câmera especial composta por dois focos de visão, simulando o olho direito e o olho esquerdo do jogador.

Junto dessa câmera, também contém os algoritmos responsáveis pela leitura do acelerômetro e do giroscópio do *smartphone*, simulando a visada do jogador conforme o lado que usuário olha.

Assim, foi substituída a câmera padrão do Unity por essa nova câmera, a qual pode ser identificada na Figura 9, na classe “AmbienteUnity”. Nesse caso, a câmera está dentro de “Player” e o “GvrEventSystem” é responsável pela captura dos dados do giroscópio e acelerômetro do *smartphone*.

4.5.2. Conexão e aquisição dos dados do sensor Bluetooth

Para a realização da conectividade entre o sensor de cadência e velocidade e o *smartphone* é necessária uma alteração no arquivo de manifesto do Android, contido dentro do projeto, o qual contém as permissões do sistema operacional Android e as necessidades da aplicação. Tal alteração pode ser visualizada na Figura 10, sendo que a mesma é responsável por configurar o aplicativo para quando seja inicializado, ele requisite uma permissão para utilização do Bluetooth do *smartphone* para que o usuário possa permitir.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figura 10: Linha adicionada no manifesto do Android.

A importação do *asset* ao projeto, permite, dentre outras funcionalidades, realizar ações como procurar por dispositivos Bluetooth, conectar-se a periféricos e inscrever-se em serviços de dispositivos encontrados. Então o que o método faz e como foi utilizado no projeto.

O método *ScanForPeripheralsWithServices* é responsável por encontrar os dispositivos Bluetooth ligados, dentro do raio de detecção do *smartphone*, que suportem o serviço especificado. No caso deste trabalho, o sensor com o serviço de detecção de cadência e velocidade.

Outro método do *asset* que merece destaque é o *ConnectToPeripheral*. Nesse caso, ele tem como tarefa conectar o *smartphone* ao dispositivo identificado e suas características também apontadas. No caso, esse método estabelece a conexão entre o dispositivo e o aplicativo.

Por fim, o método *SubscribeCharacteristicWithDeviceAddress* é responsável pela inscrição do aplicativo em serviço do dispositivo. Dessa forma, o serviço que retorna a informação de velocidade e cadência.

Então para ser descoberto qual serviço e qual característica do serviço seria utilizada, começou-se um trabalho de teste, um a um os serviços foram sendo testados, até ser encontrado o serviço e a característica que retornava os dados de cadência e velocidade em hexadecimal conforme mostra a Figura 11:

Nome:	Endereço:	Serviço	Característica
EC DMTR30 08AE	EE:3E:BE:FF:87:25		
Serviços:			
00001800-0000-1000-8000-00805f9b34fb		00002a00-0000-1000-8000-00805f9b34fb	
Generic Attribute:			
00001801-0000-1000-8000-00805f9b34fb		00002a05-0000-1000-8000-00805f9b34fb	
Cycling Speed and Cadence:			
00001816-0000-1000-8000-00805f9b34fb		00002a5b-0000-1000-8000-00805f9b34fb	
00001816-0000-1000-8000-00805f9b34fb		00002a5c-0000-1000-8000-00805f9b34fb	
00001816-0000-1000-8000-00805f9b34fb		00002a5d-0000-1000-8000-00805f9b34fb	
00001816-0000-1000-8000-00805f9b34fb		00002a55-0000-1000-8000-00805f9b34fb	
Battery Service:			
0000180f-0000-1000-8000-00805f9b34fb		00002a19-0000-1000-8000-00805f9b34fb	
Device Information:			
0000180a-0000-1000-8000-00805f9b34fb		00002a24-0000-1000-8000-00805f9b34fb	
0000180a-0000-1000-8000-00805f9b34fb		00002a26-0000-1000-8000-00805f9b34fb	
0000180a-0000-1000-8000-00805f9b34fb		00002a27-0000-1000-8000-00805f9b34fb	
0000180a-0000-1000-8000-00805f9b34fb		00002a28-0000-1000-8000-00805f9b34fb	
0000180a-0000-1000-8000-00805f9b34fb		00002a29-0000-1000-8000-00805f9b34fb	

Figura 11: Lista de serviços e características do sensor utilizado com o serviço certo destacado.

Ao obter esses dados eles foram separados e convertidos para inteiro, pois são recebidos concatenados em uma só informação e em hexadecimal.

4.5.3. Cálculo e atualização de valores e posições no cenário

Com a conexão estabelecida, o sensor começa a enviar uma mensagem, composta por onze números, sendo os primeiros sete números, responsáveis pela velocidade e os outros quatro responsáveis pela cadência. A partir disso, foi realizado um estudo de identificação da mensagem e uma dedução do significado de cada valor, obtendo-se a seguinte conclusão:

Para explicar melhor o funcionamento desses números, foi considerado o seguinte exemplo de mensagem recebida do sensor de cadência e velocidade:

“3 123 2 0 0 72 37 11 1 30 12”

O primeiro número (3) é identificado como o início da mensagem, sendo que ele não sofre alterações, ao contrário dos demais. O segundo número (123), conta quantas vezes foi alterada a velocidade capturada. O terceiro (2), quarto (0) e o quinto (0) números, começam em zero e são utilizados como contador para quantas vezes o segundo

número chega a duzentos e cinquenta e cinco, reiniciando toda a vez que alcança esse valor e incrementando um nesses números. O sexto (72) e o sétimo (12) são responsáveis pela velocidade. O oitavo (11) número, conta quantas vezes a cadência capturada foi alterada. O nono (1) número conta quantas vezes o oitavo número chegou a 255, assim ele zera e o nono número incrementa. E por fim o décimo (30) e o décimo primeiro (12) são responsáveis pela cadência em si.

Finalizada a identificação da mensagem, a sua utilização ao meio virtual foi dada por meio da classe chamada “DadosBluetooth”. Ela recebe a mensagem da classe “ConexaoBluetooth” contendo o contador de mudança de velocidade, a velocidade, o contador de mudança de cadência e a cadência. Assim, essa classe aplica a cadência recebida na tela do jogador, fazendo o com que seja exibida a cadência que está sendo emitida pelo sensor, como ilustra a Figura 12.

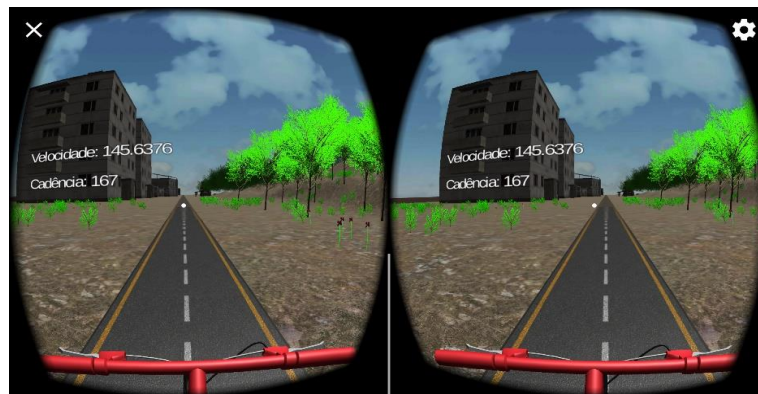


Figura 12: Aplicação com a velocidade e a cadência.

Em seguida, a classe “MoverJogador” recebe a velocidade capturada pelo sensor e é aplicado um cálculo para que o jogador tenha uma velocidade similar com a emitida pelo sensor. Porém, é realizada a ação/movimentação no meio virtual. Pode-se visualizar na Figura 13, o fragmento do código responsável pela transformação da velocidade recebida para adequá-la ao meio virtual.

```
transform.position += new Vector3 (0f, 0f, dadosBluetooth.velocidade * 0.005f);
```

Figura 13: Fragmento do código responsável pelo reposicionamento do jogador.

Essa classe “MoverJogador” também é responsável por reiniciar a posição do jogador, caso ele ultrapasse a posição oitocentos e cinquenta no eixo z, como pode ser observado na Figura 14.

```
if (transform.position.z > 850)  
{  
    transform.position = new Vector3(0f, 2f, -20f);  
}
```

Figura 14: Parte do código responsável pelo reposicionamento do jogador.

4.5.4. Construção do cenário

O cenário é construído dentro do Unity, não necessitando código para isso. Foi construído um plano quadrado, cortado ao meio por uma estrada saindo do meio de um de seus lados. Isso, por si só, já caracteriza um cenário para testes, mas foi adicionado textura no chão desse terreno, para imitar terra, e textura na estrada, para imitar asfalto. Após isso, foi adicionado alguns modelos tridimensionais de prédios, um modelo de bicicleta, plantas, muros e um céu com nuvens (Figura 15).



Figura 15: Cenário do aplicativo.

5. Validação

Para testar o sistema, o aparato completo, incluindo bicicleta de *spinning* e óculos de realidade virtual, foi montado e exposto em um ambiente público no campus do Centro Universitário Franciscano, para que qualquer pessoa interessada pudesse utilizá-lo. Professores e alunos testaram o sistema e ofereceram suas opiniões e *feedbacks*.

Aqueles que participaram comentaram sobre suas experiências utilizando o sistema, como que acharam uma aplicação inovadora ou/e muito interessante, que chama atenção, responderam se gostaram e se fariam uso de um sistema como este. Nesse caso, todas as respostas foram positivas.

6. Conclusão e Trabalhos Futuros

Com a implementação da proposta deste trabalho, a ideia é que a prática de exercícios físicos *indoor* torne-se mais interessante e possa ser melhor explorada com uso de algumas tecnologias. Este projeto, o qual visa oferecer uma maior imersão ao usuário de esportes *indoor* eliminando, também as dependências de um clima favorável e da condição de segurança das ruas para a prática da atividade.

Dentro desse contexto, foram analisados alguns trabalhos relacionados, entre os quais o Zwift e o VirZOOM mereceram destaque por tratarem especificamente de ciclismo. Com eles, é possível acompanhar a velocidade e a cadência que o usuário pedala, em que essas informações podem modificar diretamente o deslocamento do usuário no meio virtual, o que é uma das bases deste projeto.

Entretanto, uma das preocupações deste trabalho é a utilização de dispositivos em excesso e o quão agradável ou desagradável pode ser o uso de um óculos de realidade virtual para a prática esportiva. Após a utilização do sistema durante um certo período de tempo, a transpiração do corpo humano pode embaçar as lentes do óculos, gerando desconforto em usos prolongados desses dispositivos. Portanto, é preciso levar em consideração a necessidade de encontrar alternativas de óculos que ofereçam soluções a esses problemas e verificar se a aplicação desenvolvida realmente torna a prática do ciclismo *indoor* mais imersiva e interessante quando comparada com a sua não utilização.

Como sugestão para trabalhos futuros, está a parte de mensurar, com precisão, a velocidade pedalada em relação a roda e as informações retornadas do sensor de cadência e velocidade para replicar a distância percorrida no meio virtual com exatidão. Isso, para que a proposta seja ainda mais imersiva, implicando na reconstrução do cenário com tamanhos e medidas reais. Outro desafio, que pode ser um futuro aprimoramento, é o fato de quando o aplicativo inicia, a visada do jogador está centralizada para o lado em que o *smartphone* estiver posicionado, ou seja, não é direcionado para a roda dianteira da

bicicleta. Pode-se melhorar o trabalho instalando um novo dispositivo pra que aconteça a detecção do lado da roda dianteira ou, simplesmente, programar o menu disponível e preparado para a implementação entre o guidão e o banco da bicicleta. Assim, apenas com a visada do jogador em cima do botão de centralizar, seria o suficiente para centralizar a visão do jogador na roda dianteira, permitindo que essa centralização seja usada sempre que necessário. E como última sugestão para a continuidade do projeto, poderiam ser construídos mais cenários, melhor otimizados para a tecnologia móvel.

Em futuras exposições, a pesquisa qualitativa realizada com os participantes poderá ser complementada por uma pesquisa quantitativa, com questões avaliativas como, se sentiu estranhamentos utilizando o sistema, se doeu os olhos, se sentiu alguma tontura, se adquiriria um sistema como esse, entre outras perguntas que poderão resultar, com maior nível de detalhe, em dados sobre a usabilidade do sistema e sobre a interação dos participantes.

Referências Bibliográficas

- AgileModeling. Disponível: agilemodeling.com/essays/fdd.htm. Acesso maio/2017.
- Bernardo K. “O que são métodos ágeis?” Disponível: <https://www.culturaagil.com.br/oque-sao-metodos-ageis/>, 02/agosto/2015.
- Coutinho D. “O que é Realidade Virtual? Entenda melhor como funciona a tecnologia” Disponível: <http://www.techtudo.com.br/noticias/noticia/2015/09/o-que-e-realidadevirtual-entenda-melhor-como-funciona-a-tecnologia.html>, 29/setembro/2015
- Echowell Disponível: <http://www.echowell.com/>, Acesso Novembro/2017
- Figueiredo C. M. S., Nakamura E. “Computação Móvel: Novas Oportunidades e Novos Desafios” Disponível: researchgate.net/profile/Eduardo_Nakamura2/publication/268435975_computacao_movel_novas_oportunidades_e_novos_desafios_computacao_movel_novas_oportunidades_e_novos_desafios/links/55004c230cf28e4ac347f396.pdf, Junho/2003.
- Goodrich R. (2014) “Cadence Sensors: What Are They and How Do They Work?” Disponível: livescience.com/40064-cadence-sensors.html, 30/Setembro/2013.
- Google VR Disponível: <https://vr.google.com/>, Acesso Abril/2017.
- Goyal S. “Agile Techniques for Project Management and Software Engineering” Disponível: <http://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf>, Julho 2008.
- Icaros Disponível: <http://www.icaros.net/> Acesso abril/2017.
- Iglesias M. “A ciência e a tecnologia melhoram a prática esportiva?” Disponível: <http://universidadedofutebol.com.br/a-ciencia-e-a-tecnologia-melhoraram-a-praticaesportiva/>, 21/Setembro/2009.
- Karasiniski E. “O que é raytracing?” Disponível: <https://www.tecmundo.com.br/video-game-e-jogos/881-o-que-e-ray-tracing-.htm>, Novembro/2008.
- Moraes C. “Computação Gráfica Para Todos.” Disponível: <http://www.hardware.com.br/artigos/computacao-grafica-iniciantes/> 31/Outubro/2008.
- Prada R. “O que é um acelerômetro” Disponível: <https://www.tecmundo.com.br/curiosidade/2652-o-que-e-um-acelerometro-.htm> 27/Agosto/2009.
- Santos K. S. “Os IDE’s (Ambientes de Desenvolvimento Integrado) como ferramenta de trabalho em informática” Disponível: usr.inf.ufsm.br/~alexks/elc1020/artigo-elc1020-alexks.pdf Acesso: junho/2017.
- Santos F. VR Vive Disponível: blog.waz.com.br/2017/02/13/vr-vive-comofunciona/ 13/Fevereiro/2017.

Silveira C. B. “Sensor: Você Sabe o Que é Quais os Tipos?” Disponível:
www.citisystems.com.br/sensor-voce-sabe-que-quais-tipos/ 17/Novembro/2016.

Silveira D. P. “O que o giroscópio?” Disponível: <https://www.oficinadanet.com.br/post/17290-o-que-e-giroscopio-nos-smartphones> 14/Setembro/2016.

Shatamic Disponível: <http://www.shatamic.com/>, Acesso novembro/2017

SteamVR “Como funciona” <http://store.steampowered.com/steamvr> Acesso: abril/2017.

Unity Disponível: <https://unity3d.com/pt/> Acesso abril/2017

Unity Documentation “Quaternion” Disponível:
<https://docs.unity3d.com/ScriptReference/Quaternion.html> Acesso junho/2017

Velho L., Gomes J. M. “Computação Gráfica e Estilos Visuais” Disponível:
http://www.visgraf.impa.br/Data/RefBib/PS_PDF/eav00/ 19/Setembro/2000.

virZOOM Disponível: <https://virzoom.com/> Acesso abril/2017>.

Vive HTC (2017) Disponível: <https://www.vive.com/us/> Acesso abril/2017.

Zwift (2017) Disponível: <https://zwift.com/get-started#ready> Acesso abril/2017