

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

**DESENVOLVIMENTO DE UM *CASE* PARA
SIMULAÇÃO DE MOTOCICLETA**

TRABALHO DE GRADUAÇÃO

Ricardo Luiz Paul

Santa Maria, RS, Brasil

2014

DESENVOLVIMENTO DE UM *CASE* PARA SIMULAÇÃO DE MOTOCICLETA

Ricardo Luiz Paul

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de

Bacharel em Ciência da Computação

Orientador: Prof. Dr. Raul Ceretta Nunes

Co-orientador: Prof. Dr. Cesar Tadeu Pozzer

**385
Santa Maria, RS, Brasil**

2014

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**DESENVOLVIMENTO DE UM CASE PARA SIMULAÇÃO DE
MOTOCICLETA**


elaborado por
Ricardo Luiz Paul

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

COMISSÃO EXAMINADORA:


Raul Ceretta Nunes, Dr.
(Presidente/Orientador)


Giliane Bernardi, Prof^a. Dr^a. (UFSM)


Lisandra Manzoni Fontoura, Prof^a. Dr^a. (UFSM)

Santa Maria, 3 de Dezembro de 2014.

AGRADECIMENTOS

Agradeço a minha família por todo apoio e dedicação para proporcionar condições para a realização de todo o curso de Ciência da Computação na Universidade Federal de Santa Maria.

Agradeço os professores de todo o curso, e principalmente aos orientadores Raul Ceretta Nunes e Cesar Tadeu Pozzer pelo auxílio prestado durante o desenvolvimento deste trabalho.

Agradeço aos amigos e colegas do curso de Ciência da Computação, pela ajuda, compreensão e amizade durante os 4 anos de muitos estudos.

RESUMO

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

DESENVOLVIMENTO DE UM CASE PARA SIMULAÇÃO DE MOTOCICLETA

AUTOR: RICARDO LUIZ PAUL

ORIENTADOR: RAUL CERETTA NUNES

CO-ORIENTADOR: CESAR TADEU POZZER

Local da Defesa e Data: Santa Maria, 3 de Dezembro de 2014.

Simuladores de direção vêm ganhando espaço no treinamento de novos motoristas. Esta ferramenta proporciona um treinamento direcionado à situações específicas, capacitando novos condutores à circular no trânsito. Considerando o treinamento para pilotos de motocicletas, o uso de simuladores tem a vantagem de não expor o condutor a possíveis acidentes. Para simuladores serem válidos no treinamento de condutores é necessário que o simulador, como um todo, represente o mais próximo possível a realidade. A busca pela realidade é uma constante no âmbito de simuladores, pois a dificuldade de transformação do mundo real em um sistema de computador exige alto grau de processamento, mapeamento, integração e sincronização afim de obter uma ferramenta de simulação para direção veicular. Este trabalho visa desenvolver um simulador de motocicleta com ênfase na sincronização entre, eventos decorridos em um cenário de simulação de motocicleta, e, a realização de movimentos em um protótipo da Plataforma de Movimentos (*motion*). Por ser um sistema complexo, no desenvolvimento deste trabalho é necessária a utilização de uma *game engine* para construção do cenário, movimentação de objetos inseridos no cenário e comunicação com a Plataforma de Movimentos. Também faz-se necessário a utilização da plataforma Arduino para prototipação de uma Plataforma de Movimentos.

Palavras-chave: Simulador. Motion. Motocicleta.

ABSTRACT

Undergraduate Final Work
Undergraduate Program in Computer Science
Federal University of Santa Maria

DEVELOPMENT OF A CASE FOR MOTORCYCLE SIMULATION

AUTHOR: RICARDO LUIZ PAUL

ADVISOR: RAUL CERETTA NUNES

COADVISOR: CESAR TADEU POZZER

Defense Place and Date: Santa Maria, December 3rd, 2014.

Driving simulators have been gaining ground in the training of new drivers. This kind of tool provides a training directed to specific situations, enabling new drivers to guide in traffic. Considering training for motorcycle riders, the use of simulators has the advantage of not exposing the driver to potential accidents. Simulators should represent the real environment as close as possible to be considered as a valid tool for training. The search for reality is a constant within simulators, since the complexity of translating the real world into a computer system, requiring a high degree of processing, mapping, integration and synchronization in order to build an effective driving simulator. This work aims at the development of a motorcycle simulator with emphasis on synchronization between events occurred in a motorcycle simulation scenario, and also performing movements in a prototype of the Motion Platform. Due to the complexity, in this work we adopted a game engine to ease scenario construction, as well representation of moving entities and communication with the Motion Platform. It is also necessary to use the Arduino to prototype a Motion Platform.

Keywords: Simulation. Motion. Motorcycle.

LISTA DE FIGURAS

2.1	Exemplo de imersão e interação em um simulador de motocicleta (COS-SALTER, 2010, p. 224).	13
2.2	Plataforma de Movimentos (<i>Motion</i>) (http://www.moog.com/).	15
2.3	Ilustração dos Graus de Liberdade em uma aeronave (http://www.en.wikipedia.org/). 16	16
2.4	Área de Trabalho da Unity3D.	17
2.5	Placa Arduino Uno (http://www.arduino.cc/).	18
2.6	Potenciômetro (http://www.twenga.com/)	18
3.1	Ambiente Experimental do Simulador de Motocicleta.	20
3.2	Casos de Uso do Simulador de Motocicleta.	21
3.3	Diagrama de Sequência para o Caso de Uso Acelerar Motocicleta.	21
4.1	Visão do cenário criado.	24
4.2	Mapa do Cenário desenvolvido.	24
4.3	Ligação entre Objetos do Cenário e Variáveis do script de Movimentação. . .	25
4.4	Trecho de código da aplicação de forças para movimentação da motocicleta. .	26
4.5	Potenciômetro conectado a Protoboard e ligado ao Arduino Uno.	27
4.6	Representação do movimento de rotação Pitch na motocicleta introduzida no cenário criado.	28
4.7	Representação do movimento de rotação Roll em uma motocicleta (http://www.conti-online.com/).	29
4.8	Cálculo do Movimento Pitch da Motocicleta.	30
4.9	Referenciais para cálculo do movimento Roll na Motocicleta.	31
4.10	Base e Pistões da Plataforma de Movimentos.	32
4.11	Protoboard e Leds.	33
4.12	Protótipo da Plataforma de Movimentos.	34
5.1	Trajatória do Experimento 1.	35
5.2	Posição da motocicleta em um determinado instante do Experimento 1.	37
5.3	Simulação da Plataforma de movimentos para a posição da motocicleta na Figura 5.2.	38
5.4	Trajatória a ser percorrida no Experimento 2.	38
5.5	Protótipo para teste do movimento de inclinação Roll.	39
5.6	Instante que a motocicleta está efetuando uma curva.	40
5.7	Protótipo indicando o movimento dos pistões da Plataforma de Movimen- tos para o instante da motocicleta da Figura 5.6.	41

SUMÁRIO

1 INTRODUÇÃO	9
2 REVISÃO BIBLIOGRÁFICA	11
2.1 Simulação para treinamento de motoristas	11
2.2 Realidade Virtual e Cenários	13
2.2.1 Leis da Física	14
2.2.2 Plataforma de Movimentos (<i>Motion</i>).....	14
2.3 Graus de Liberdade de uma Plataforma de Movimentos	15
2.4 Engine	16
2.5 Arduino	17
3 PROPOSTA DE DESENVOLVIMENTO E SINCRONIZAÇÃO DE EVENTOS DE UM SIMULADOR DE MOTOCICLETA	19
3.1 Requisitos para Desenvolvimento do Simulador	19
3.2 Ambiente Experimental	19
3.3 Metodologia	21
4 DESENVOLVIMENTO	23
4.1 Cenário	23
4.2 Motocicleta	24
4.2.1 Movimentação da Motocicleta	25
4.2.2 Propriedades Físicas da Motocicleta	26
4.2.3 Aceleração da Motocicleta via Potenciômetro	27
4.3 Graus de Liberdade da Plataforma de Movimentos	27
4.4 Detecção de eventos para Motion	29
4.4.1 Ângulo de Inclinação do Movimento Pitch	29
4.4.2 Ângulo de Inclinação do Movimento Roll.....	30
4.4.3 Sensação de Acelerações e Desacelerações no <i>Motion</i>	30
4.5 Padrão de Comunicação entre Unity3D e Plataforma de Movimentos	31
4.6 Protótipo da Plataforma de Movimentos	32
5 TESTES E RESULTADOS	35
5.1 Experimento 1	35
5.1.1 Configuração do protótipo da Plataforma de Movimentos	36
5.1.2 Resultados do Experimento 1	36
5.2 Experimento 2	37
5.2.1 Configuração do protótipo da Plataforma de Movimentos	38
5.2.2 Resultados do Experimento 2	39
5.3 Experimento 3	40
6 CONCLUSÃO	43
REFERÊNCIAS	44

1 INTRODUÇÃO

A utilização de simuladores no treinamento de pessoas que buscam obter habilitação para dirigir tem sido vista como uma solução que reduz riscos, devido às diversas vantagens de se utilizar um ambiente simulado ao invés do treinamento habitual nas ruas. A simulação de direção para motoristas pode ser obtida através da utilização de ferramentas de hardware e software que simulam o controle de um veículo e o ambiente em que o veículo é controlado.

Em nota o Detran-RS posicionou-se em relação a Resolução 493/2014 do Conselho Nacional de Trânsito, tornando obrigatório o treinamento com a utilização de simuladores de direção para motoristas que buscam habilitação na categoria B. Apesar de os simuladores serem limitados na representação do mundo real, eles são considerados importantes na formação de condutores por proporcionarem opções de reações quando ocorrem condições adversas (chuva, neblina, tráfego intenso). Porém, a tendência é de melhora significativa na representação da realidade nos simuladores, o que demanda contínua investigação e experimentos nesta área.

No Japão, o treinamento de novos motoristas para motocicletas faz uso de simuladores desde setembro de 1996, quando uma lei introduziu esta ferramenta (Yamasaki, 1998). Com essa necessidade das escolas japonesas de treinamento de motoristas para motocicletas, diversos trabalhos foram desenvolvidos visando obter um simulador com alto grau de realismo. Para Yamasaki (1998), a necessidade dos simuladores fornecerem ao usuário um *feedback* físico com impactos, inclinação da motocicleta nas curvas e sensação de aceleração é crucial para a educação de motoristas. Para isso, a integração com uma Plataforma de Movimentos (*motion*) deve estar em perfeita sincronia com as demais funcionalidades.

Em se tratando de simuladores veiculares, a simulação realista da forma como o veículo se move e responde a ações do usuário é vista como um dos elementos chave para representação precisa de situações cotidianas, principalmente se considerado a capacitação para habilitação na Categoria A (motocicletas). Comandos inseridos pelo usuário devem estar em perfeita sincronia com: a) parte gráfica (componente visual) e b) hardware que procura dar ao usuário a sensação de estar em um veículo real. Orsolin e Rampelotto (2010) destacam em seu trabalho a importância da integração do software e hardware em um simulador veicular. Quanto melhor for a sincronia entre o software e o hardware, melhor a eficiência do treinamento.

Alcançar o sincronismo desejado é uma tarefa desafiadora e que exige conhecimentos de programação de jogos, computação gráfica, sistemas distribuídos, engenharia de software e

padrões de interface. Nesse sentido, este trabalho visa investigar técnicas de sincronismo para simuladores de motocicleta. Para isso faz-se uso de uma *Game Engine* responsável pelo processamento gráfico, de dados e troca de mensagens, além da sincronização da *Game Engine* com a Plataforma de Movimentos através da troca de dados. A sincronização de sons foi explorada em (OLIVEIRA, 2014).

Este trabalho está organizado da seguinte forma. No capítulo 2 é apresentada uma revisão da literatura com conceitos importantes utilizados, juntamente com trabalhos relacionados. No Capítulo 3 é descrito a proposta do ambiente experimental do simulador, metodologia utilizada no trabalho e requisitos para o desenvolvimento do trabalho. O Capítulo 4 descreve o desenvolvimento do trabalho, o capítulo 5 descreve como foram formulados e realizados os testes da ferramenta desenvolvida e por fim é apresentada a conclusão do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Jogos eletrônicos estão em constante evolução. Isto se deve as pesquisas e trabalhos constantes nesta área. A implementação e sincronização de vários elementos, como som, animação, interação com o jogador, Inteligência Artificial, entre outros, faz com que a busca pelo realismo seja uma constante no universo dos jogos.

Os jogos estão em continua evolução, tanto na área de computação gráfica, Inteligência Artificial, como na capacidade de imersão do usuário no jogo. Simuladores veiculares buscam estas características de jogos para proporcionar um treinamento controlado. As características desta ferramenta são muito parecidas com a de um jogo, pois o usuário interage com os controles disponíveis e a ferramenta trata dos comandos recebidos. É possível distinguir os simuladores em duas categorias: simuladores com a base fixa e simuladores com a base de movimento (HIMA et al., 2008). Nos simuladores de base fixa, o usuário tem as referências visuais e auditivas para controlar o seu personagem no cenário virtual. No simulador com base em movimento, uma nova propriedade é adicionada trazendo para o usuário a sensação de movimento físico através da base de movimentação (*motion*). Assim, o nível de imersão do usuário é destaque para o sucesso de um simulador com *motion* (HIMA et al., 2008).

2.1 Simulação para treinamento de motoristas

Segundo Sauve et al. (2007), simulação é uma representação simplificada, dinâmica e precisa da realidade definida como um sistema. Simuladores podem trabalhar muito próximo da realidade. Incidentes ou situações de perigo podem ser simulados, logo, pode-se avaliar se a reação tomada pelo usuário foi apropriada ou não para diferentes situações de perigo. Desta maneira, o usuário pode adquirir experiência, conhecimento e habilidades para tratar situações de perigo.

Para Sauve et al. (2007), simulação não necessariamente envolve conflitos ou competição entre os participantes, o que ocorre rotineiramente em jogos. Apesar de serem comparáveis, há também características que distinguem simuladores de jogos. Em simuladores o objetivo é o treinamento e a obtenção de habilidades para desenvolver uma determinada tarefa, já nos jogos, o objetivo é mais explícito, como por exemplo chegar em local, destruir uma certa quantidade de objetos, tudo isso, sem fins de obter habilidades úteis para o mundo real.

Para Hima et al. (2008), os projetos iniciais de simulação de direção para motoristas tanto de carros como de motocicletas, fizeram referência aos simuladores de treinamento de voos, eficientes e redutores de risco no treinamento de novos pilotos de aeronaves. O baixo custo e a segurança de realizar treinamento de pilotos de aeronaves em simuladores foi a alavanca para o crescimento desta ferramenta, que hoje se torna importante também no cenário de treinamento de condutores de carros e motocicletas.

Riener (2010) cita em seu trabalho as novas tendências do mercado de veículos como outro fator que requer treinamento simulado em computadores. Entre as tendências do mercado encontra-se o surgimento de um veículo autônomo encarregado de transportar passageiros com segurança até seus destinos. Uma das maneiras de realizar testes sem perigos é via simulação digital.

A atividade de simulação digital, seja ela na área veicular, traz aprendizado e exclui a probabilidade de danos materiais. Riener (2010) diz que a razão para utilizar simulação no domínio veicular é a sua aplicação em diversas áreas críticas, onde o treinamento, se feito no mundo real, trás riscos a integridade física de alunos e objetos materiais. Um exemplo é o treinamento de condutores em áreas de tráfego intenso, onde qualquer deslize pode resultar em acidentes. Levando em consideração este aspecto para aulas de treinamento de motoristas de motocicletas, um aluno sem experiência pode adquirir habilidades e confiança para conduzir uma motocicleta real.

No estudo de Pereira (2009), são citados exemplos positivos quanto a utilização de simuladores como ferramenta para aprimorar a capacidade de seus usuários. Lewis Hamilton, piloto de corrida da categoria Fórmula I, venceu o grande prêmio do Canadá de 2007 correndo pela primeira vez neste circuito (Spurgeon, B., 2007). O treinamento de Lewis Hamilton para a corrida foi realizado através de um simulador, onde aspectos da pista, colisões com zebras, vibrações do carro, eram simuladas, fazendo com que o piloto percebesse alto grau de imersão e realismo.

Nilsson (1993) realça que os benefícios da simulação para treinamento de motoristas são diversos, como a possibilidade de realizar testes específicos, treinamento direcionado a maximizar a eficiência e segurança, além da coleta de dados, afim de obter benefícios via avaliação pós-treinamento. Casutt et al. (2014) faz um estudo sobre o uso de simulação para treinamento de motoristas com idade avançada. A perda de habilidades, percepção, reação a eventos na direção de um veículo pode se tornar a causa de acidentes. Com o uso de simulador para treina-

mento de motoristas, Casutt et al. (2014) pôde notar que o treinamento melhora as habilidades de um motorista tornando-o mais seguro no trânsito.

2.2 Realidade Virtual e Cenários

Tudo que é simulado em um jogo precisa estar inserido dentro de um cenário virtual, tornando possível a interação do jogador com o ambiente criado. Os cenários ganham importância nesta área, pois é onde o jogador tem uma visão do que está acontecendo, auxiliando assim nas decisões tomadas. Basicamente, não existe um jogo/simulação sem um cenário.

Braga (2001) define realidade virtual como uma experiência imersiva e interativa baseada em imagens 3D geradas em tempo real por computador. No contexto deste trabalho, a imersão do usuário com o sistema de realidade virtual deve ser comparada com uma tarefa real, como por exemplo girar a cabeça para poder visualizar o que está ao lado do usuário, tornando real a ilusão de estar dentro de um cenário virtual em 3 dimensões. No contexto deste trabalho, a interatividade ocorre quando uma ação física realizada pelo usuário gera movimentos correspondentes no cenário virtual ao qual se está interagindo através de uma máquina. Resumindo, ações reais devem ser sincronizadas com o que acontece no cenário virtual através de controles fornecidos pela aplicação. A Figura 2.1 é uma representação dos conceitos de imersividade e interação inseridos em um simulador de motocicleta.



Figura 2.1 – Exemplo de imersão e interação em um simulador de motocicleta (COSSALTER, 2010, p. 224).

O nível de realismo é um fator importante no contexto de realidade virtual e cenários.

Definir o mundo real como um sistema de computador é um desafio que envolve controle de forças, como a gravidade, controle de acelerações, colisões entre objetos, entre outros aspectos.

2.2.1 Leis da Física

O uso de Leis da Física é fundamental para a construção de cenários para simuladores que buscam o realismo na representação de forças, colisões, acelerações, no ambiente simulado. Por exemplo, quando uma aceleração é empregada em um objeto, a velocidade do mesmo deve aumentar gradualmente, ou quando o objeto colide com algum outro objeto do cenário, a *engine* de física não deve permitir que dois corpos permaneçam no mesmo espaço. Objetos em um mundo real se movimentam através de forças aplicadas. O simulador deve realizar esse controle de forças, em seu cenário, para tornar a simulação mais próxima da realidade.

No domínio de simuladores dois aspectos são utilizados para caracterizar a qualidade de um simulador: fidelidade e validade (RIENER, 2010). O segundo aspecto refere-se a como o simulador é capaz de simular o mundo real. Para que isso seja possível, deve-se ter um controle sobre os objetos do cenário para que todos estejam de acordo com as Leis da Física.

2.2.2 Plataforma de Movimentos (*Motion*)

Sempre na busca pela realidade, simuladores podem utilizar sistemas de movimento ou *motion* (Figura 2.2). Este hardware/ferramenta permite aplicar forças físicas ao usuário de modo a imitar a sensação física que esse sentiria se estivesse no cenário virtual, promovendo ao usuário sensação de imersão no simulador. Por exemplo, em um simulador de motocicleta quando o usuário passa em uma ondulação do cenário, a motocicleta deveria realizar um movimento real correspondente ao tamanho da ondulação com objetivo do usuário sentir, na simulação, fisicamente o que aconteceria no mundo real.

Segundo Yamasaki et al. (1998), este requisito de sensação de movimento real vem sendo desenvolvido desde o surgimento dos simuladores de motocicleta, crescendo após a lei de uso de simuladores para treinamento de motoristas de motocicletas no Japão. Em seu trabalho, Yamasaki et al. (1998) desenvolveu um simulador com *motion* através de motores que alavancavam o usuário pra frente em uma frenagem ou inclinava o usuário para trás em uma aceleração, trazendo a real sensação para dentro do simulador já naquela época.

De acordo com Groot et al. (2011), foi demonstrado que o uso de *motion* no treinamento de motoristas reduz a velocidade de condução e aceleração, diminuindo os riscos de

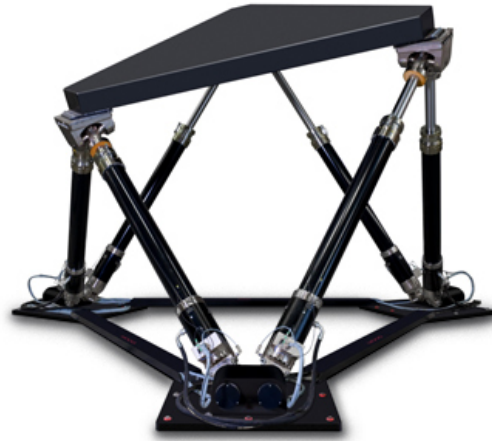


Figura 2.2 – Plataforma de Movimentos (*Motion*) (<http://www.moog.com/>).

incidentes. Além de aumentar a sensação de estar dirigindo um veículo real, *motion* ajuda na conscientização de motoristas da real periculosidade na condução de um veículo, melhorando suas habilidades na direção.

Para os simuladores com base de movimento, a funcionalidade de movimentar a base deve estar de acordo com o que ocorre no cenário de simulação (HIMA et al., 2008). O software deve verificar no cenário a ocorrência de movimentos que se tornarão movimentos físicos no *motion*. Ondulações no terreno ou colisão com objetos são exemplos que devem ser verificados para serem sincronizados com os movimentos físicos, afim de estar mais próximo da realidade possível.

2.3 Graus de Liberdade de uma Plataforma de Movimentos

Graus de Liberdade refere-se ao número de coordenadas no espaço que devem ser especificados para determinar a posição de um corpo (PENNESTRI; CAVACECE; VITA, 2005). Se o corpo é um ponto de massa, 3 graus de liberdade são suficientes para dar a posição exata. Por outro lado, o corpo pode ser um objeto como uma motocicleta, então são necessários também 3 ângulos que representam a posição do objeto em relação a cada coordenada(x, y e z).

Uma Plataforma de Movimentos pode movimentar sua base em diversos sentidos. Na Figura 2.3 são apresentados três movimentos, Roll, Pitch e Yaw, que são movimentos de rotação dos eixos coordenados. O ângulo de rotação dos eixos coordenados define esses 3 movimentos. Há também os movimentos de translação sob os eixos coordenados. A combinação de movimentos que uma plataforma pode executar, define a quantidade de graus de liberdade da plataforma.



Figura 2.3 – Ilustração dos Graus de Liberdade em uma aeronave (<http://www.en.wikipedia.org/>).

2.4 Engine

Segundo Clua (2005), uma *engine* é o principal elemento para o funcionamento de um jogo ou simulador. Fazendo a comparação com uma motocicleta, o motor é responsável por fazê-la andar. Logo, o piloto não está interessado em como ocorre o funcionamento do motor, o controle da entrada de combustível para combustão, entre outros fatores. Uma *engine* abstrai questões de baixo nível em programação para quem está desenvolvendo um simulador, auxiliando o programador a desenvolver um trabalho complexo de alto nível, em um período menor de tempo.

Uma *engine* pode ser constituída de diversas ferramentas que auxiliam nas diversas etapas da criação de um simulador. Segundo Clua (2005), estas ferramentas podem servir para auxiliar na renderização, tratamento de sons, simular as Leis da Física no cenário criado, e outros. A *game engine* Unity3D¹ é um ambiente de desenvolvimento projetado para desenvolvimento de jogos. É composta por funcionalidades para desenvolvimento com alto nível de realismo, podendo também ser integrada a diversas plataformas para entrada e saída de dados. Também devido ao seu vasto material encontrado facilmente na Web, ela foi a escolhida para desenvolver o trabalho.

Unity3D permite atribuir a um jogo diversos objetos, chamados Game Objects, podendo inserir em cada um destes objetos, qualquer lógica, característica ou propriedade que a Unity3D oferece. Tudo que se pode adicionar a um Game Object na Unity3D é chamado de componente (GOLDSTONE, 2009). A Unity3D possibilita trabalhar com a física de objetos adicionando

¹ <http://www.unity3d.com/>

componentes referentes a física, como o componente Rigidbody, ou através da chamada de funções em scripts. Script é uma dentre as ferramentas oferecidas pela Unity3D, onde pode-se manipular qualquer objeto do cenário através de lógica de programação na linguagem C Sharp, JavaScript ou Boo.

A Unity3D permite criar jogos em 2D e 3D, para várias plataformas como MAC, Windows, Android, IOS e PS3 (GOLDSTONE, 2009). A interface de desenvolvimento da Unity3D permite manipular e gerenciar todos os elementos presentes no jogo, como também realizar testes a partir da visão final do usuário. A área de trabalho da Unity3D é editável de acordo com a preferência do desenvolvedor. A Figura 2.4 representa a área de trabalho da Unity3D, onde se encontra as Tabs Hierarchy contendo os Game Objects da cena, Project contendo os Assets do projeto criado para desenvolvimento, Inspector contendo os componentes de um Game Object selecionado, Scene e Game.



Figura 2.4 – Área de Trabalho da Unity3D.

2.5 Arduino

Segundo Banzi (2014), Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software de fácil utilização. Arduino tem características que visam auxiliar o contato de um computador ao mundo físico. Também é utilizada para construir protótipos de simulação, tendo a vantagem de ser de baixo custo. Um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes que estejam conectados a ele externamente (MCROBERTS, 2011). O Hardware é composto

por uma placa micro-controladora, que tem suporte para entrada e saída de dados para sensores, leds, protoboards, potenciômetros, dentre outros dispositivos. A Figura 2.5 representa uma placa micro-controladora Arduino Uno utilizada neste trabalho.

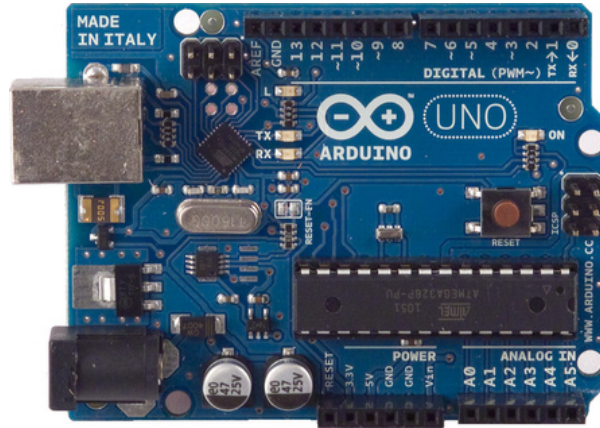


Figura 2.5 – Placa Arduino Uno (<http://www.arduino.cc>).

Dentre os dispositivos de interação do Arduino, o potenciômetro é um objeto que transfere dados para o Arduino conforme é acionado o pino de rolagem. A Figura 2.6 representa um potenciômetro utilizado no trabalho.



Figura 2.6 – Potenciômetro (<http://www.twenga.com/>)

3 PROPOSTA DE DESENVOLVIMENTO E SINCRONIZAÇÃO DE EVENTOS DE UM SIMULADOR DE MOTOCICLETA

A integração correta de ferramentas que auxiliam na interatividade e perceptividade do usuário com o cenário criado para simulação de motocicleta traz resultados positivos se comparados com simuladores de motocicleta que não dispõe dessas ferramentas (GROOT et al., 2011). Este trabalho visa buscar a integração correta de dados vindos do usuário através da interação do mesmo com controles fornecidos pelo simulador, e, a integração correta de uma Plataforma de Movimentos que visa auxiliar na percepção do usuário do que está acontecendo no cenário virtual.

3.1 Requisitos para Desenvolvimento do Simulador

- Delimitação de um cenário virtual para tráfego de veículos automotores.
- Inserção de uma motocicleta no cenário com propriedades físicas de movimentação similares as propriedades físicas reais de movimentação de uma motocicleta.
- Especificação de controles para o usuário interagir e guiar a motocicleta inserida no cenário.
- Definição dos Graus de Liberdade que serão simulados na Plataforma de Movimentos.
- Especificação do padrão de comunicação entre o cenário virtual e Plataforma de Movimentos.
- Detecção de eventos da Plataforma de Movimentos no cenário virtual.
- Envio de dados do movimento Pitch e Roll para a Plataforma de Movimentos.
- Prototipação de uma Plataforma de Movimentos.
- Recebimento e processamento de dados no protótipo da Plataforma de Movimentos.

3.2 Ambiente Experimental

O ambiente experimental do simulador de motocicleta é composto por uma *game engine*, controles para o usuário (potenciômetro para aceleração da motocicleta e teclado para

direcionar e frear a motocicleta), componente para visualização e o protótipo da Plataforma de Movimentos. Na Figura 3.1 está representado o ambiente experimental do simulador. O usuário inserido na Figura 3.1 interage com os controles e tem sua visão do cenário virtual através do componente visual (Monitor de computador).

Quando o usuário interage com os controles do simulador, dados para controle da motocicleta são recebidos na *game engine*. Na *game engine* são definidos o cenário de simulação, Scripts de movimentação de Game Objects e Scripts de comunicação. O componente visual é atualizado a todo momento. Este componente está ligado a *game engine* que é responsável pela produção de imagens e renderização das imagens no componente visual.

O protótipo da Plataforma de Movimentos recebe dados da *game engine* de acordo com eventos decorridos no cenário. Quando um evento é detectado, a *game engine* envia dados do evento para o protótipo da Plataforma de Movimentos. Estes dados serão processados na plataforma Arduino de forma que sejam gerados eventos adequados no protótipo construído com Protoboarde Leds para sinalização dos movimentos.

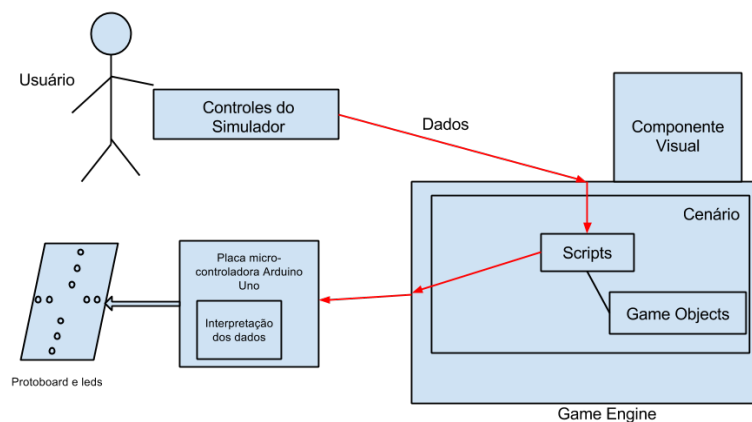


Figura 3.1 – Ambiente Experimental do Simulador de Motocicleta.

Os dados entre controles e *game engine* seguem um padrão, assim como dados entre *game engine* e o protótipo da Plataforma de Movimentos. Para o usuário foram definidos os Casos de Uso Acelerar Motocicleta, Direcionar Motocicleta e Frear Motocicleta, todos estes relacionados ao controle da motocicleta no cenário virtual. A Figura 3.2 contém os Casos de Uso descritos. A Figura 3.3 contém o Diagrama de Sequência para o Caso de Uso Acelerar Motocicleta.

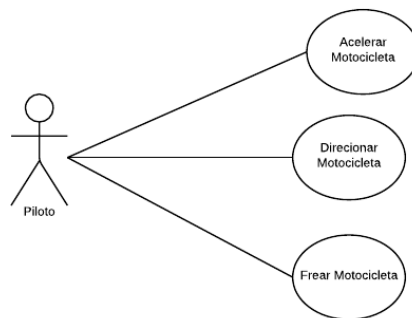


Figura 3.2 – Casos de Uso do Simulador de Motocicleta.

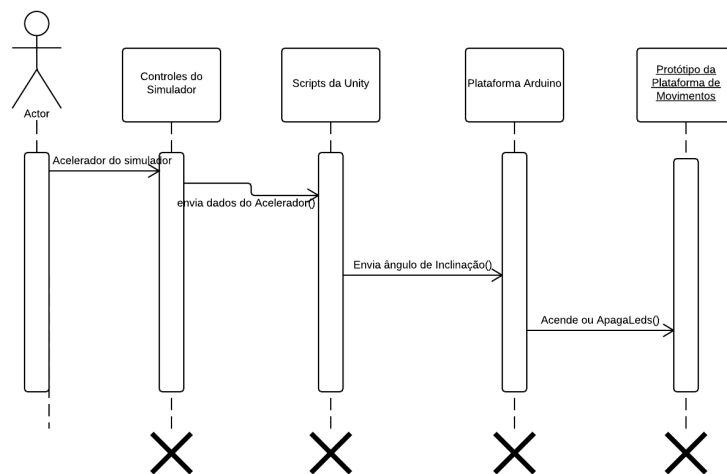


Figura 3.3 – Diagrama de Sequência para o Caso de Uso Acelerar Motocicleta.

3.3 Metodologia

O trabalho foi dividido em 4 passos principais, descritos a seguir.

1. O primeiro passo do trabalho é construir um cenário na Unity3D onde se possa realizar testes com uma motocicleta. A necessidade de aclives e declives no cenário fazem-se necessários para posteriormente realizar testes com a Plataforma de Movimentos. Há também a necessidade do cenário ser característico de uma cidade, com ruas, delimitando por onde o usuário pode circular.

2. O segundo passo é inserir uma motocicleta no cenário e simular movimentos reais. Nesta etapa as propriedades físicas da motocicleta são o principal foco, pois forças são adicionadas para simular a aceleração, frenagem, direcionamento em curvas e inclinação em curvas. Para a Plataforma de Movimentos, definir quantos graus de liberdade serão simulados é fundamental para a realização de testes e modelagem dos dados que farão parte da comunicação entre as plataformas (*motion* e *game engine*).
3. A visualização do que acontece no cenário virtual e a movimentação da plataforma que sustenta a motocicleta devem ser coerentes afim de tornar o sistema válido. A sincronização entre os eventos no cenário, mapeamento de dados para a camada de comunicação e recebimento dos comandos na plataforma para movimentação, devem atuar sem perdas ou atrasos de comunicação.
4. Para a integração da Plataforma de Movimentos ao simulador, optou-se por desenvolver um protótipo de Plataforma de Movimentos com o auxílio da plataforma Arduino, placa Protoboard e Leds. Assim, o protótipo a ser desenvolvido participará da comunicação necessária entre as plataformas (Unity3D e Plataforma de Movimentos) e simulará movimentos mecânicos do *motion* através do acionamento de uma sequência de Leds.

4 DESENVOLVIMENTO

Para o desenvolvimento do trabalho foram utilizadas duas plataformas, uma plataforma de desenvolvimento de jogos (Unity3D) e outra para prototipação da Plataforma de Movimentos (Arduino). Para programação, utilizou-se a linguagem JavaScript na Unity3D e a linguagem de programação C para desenvolvimento de rotinas no Arduino.

As subseções a seguir descrevem o trabalho realizado nas etapas do desenvolvimento do simulador de motocicleta, de acordo com a metodologia especificada no Capítulo 3.

4.1 Cenário

O cenário de simulação foi desenvolvido na plataforma Unity3D versão livre. Assets do Projeto criado são arrastados até a cena, tornando o Asset um Game Object da cena. Todos Game Objects são posicionados manualmente na cena. A partir de então, componentes físicos e de renderização são adicionados aos Game Objects criados. Também é inserida iluminação na cena. A iluminação utilizada é a Directional Light, que é constituída por um ponto que ilumina em uma direção definida. Também é adicionada a propriedade da cena chamada SkyBox Material, responsável por adicionar textura no céu.

Na construção do cenário também foi utilizada a versão trial da ferramenta SketchUp², da Google. Nesta ferramenta consegue-se obter modelos de objetos 3D para importação, criando um novo Asset no projeto da Unity3D. Assim não foi necessário realizar a modelagem de novos objetos.

O cenário criado consiste em ruas que delimitam por onde o usuário deve trafegar. Estas ruas foram posicionadas de forma que fossem formados aclives e declives. Também foram posicionados cilindros para simular quebra-molas, afim de realizar testes mais precisos de sincronização com a Plataforma de Movimentos. A Figura 4.1 apresenta uma visão do cenário com a motocicleta inserida. A Figura 4.2 representa o mapa do cenário desenvolvido que representa por onde o usuário pode trafegar com a motocicleta.

² <http://www.sketchup.com/>



Figura 4.1 – Visão do cenário criado.

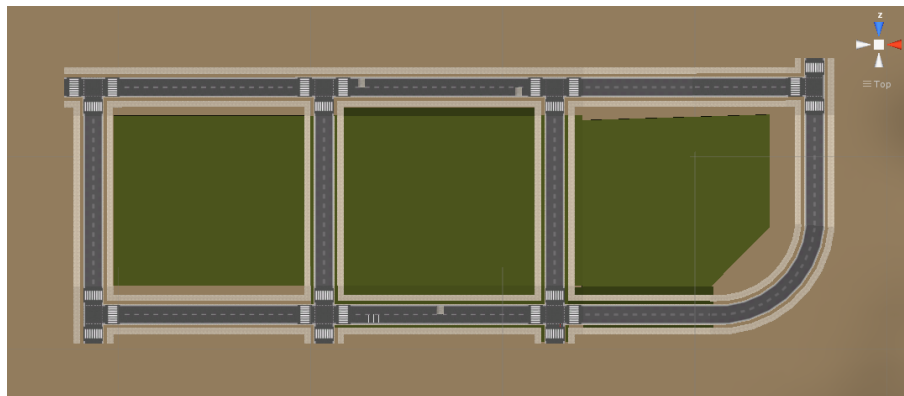


Figura 4.2 – Mapa do Cenário desenvolvido.

4.2 Motocicleta

Como a Unity3D não fornece modelos 3D de objetos variados, foi necessário o uso do SketchUp, ferramenta fornecida pela Google, utilizada no trabalho para obtenção de um modelo de motocicleta 3D. Após importação para a Unity3D, a motocicleta é representada por uma união de Game Objects. Este conjunto de objetos pertencentes a motocicleta é definido por quem construiu o modelo da motocicleta.

Na motocicleta utilizada, cada roda é um Game Object, e as demais partes da motocicleta são tratadas como um Game Object. Na inserção de componentes, é possível inserir um componente geral a todos os Game Objects que fazem parte de um único Game Object pai, ou também componentes a um Game Object específico.

4.2.1 Movimentação da Motocicleta

Um dos componentes que pode ser adicionado a um Game Object é o Script. Scripts tem a principal funcionalidade de adicionar animação aos objetos do cenário através de lógica de programação. Quando um Script é criado, duas funções são declaradas automaticamente para programação: a função Start e a função Update. A função Start é chamada na inicialização do objeto, quando o projeto é executado. Já a função Update é chamada a cada *frame*, no decorrer da execução do projeto.

As variáveis criadas no Script podem ser referenciadas a um Game Object do cenário. A Figura 4.3 representa a ligação entre as variáveis do Script de movimentação da motocicleta e objetos inseridos no cenário pertencentes a motocicleta a partir de um menu do Script na área de trabalho da Unity3D.

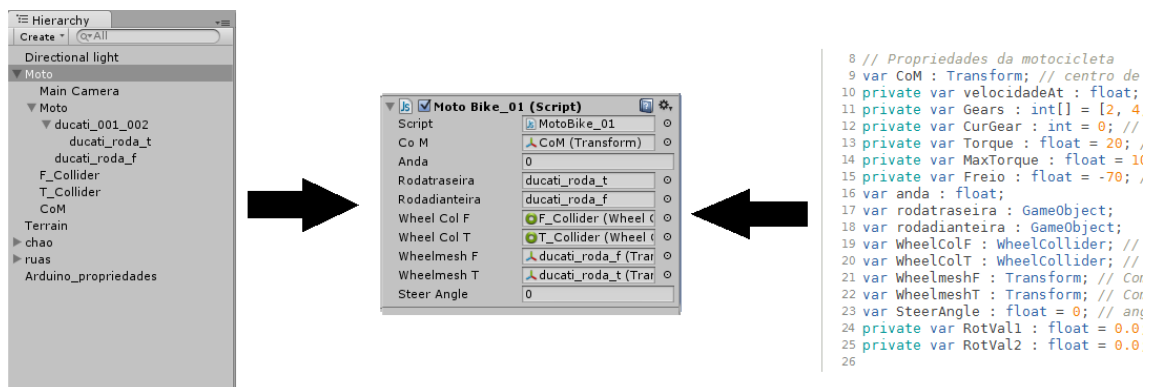


Figura 4.3 – Ligação entre Objetos do Cenário e Variáveis do script de Movimentação.

Um objeto que tem a propriedade Rigidbody deve ter sua movimentação realizada através da aplicação de forças (Unity, 2014). Para a motocicleta se movimentar, adicionamos uma força chamada motorTorque a roda traseira da motocicleta, simulando o que acontece em uma motocicleta no mundo real, onde quem recebe a força para andar é a roda traseira. Esta força pode ser positiva ou negativa, mas ela só é empregada na roda quando for positiva, para simular a motocicleta sendo deslocada para frente.

Para realizar a frenagem, outra força é aplicada a roda da motocicleta. Essa força é chamada de brakeTorque. A intensidade da força de aceleração ou freio pode ser ajustada através da multiplicação de uma variável de valor fixo pela variável que define a intensidade com que foi acionada a aceleração ou freio. O trecho de código da Figura 4.4 demonstra o uso destas forças aplicadas nas rodas da motocicleta. Na Seção 4.2.3 deste Capítulo é detalhado como o valor da variável potenciometro é obtido. A chamada de função "Input.GetAxis("Vertical")" representa

um valor decimal entre -1 e 1, representando a interação do usuário com o simulador através do teclado do computador.

```

152 //Aceleração da moto
153 WheelColT.motorTorque = Torque * potenciometro;//WheelColT.motorTorque = 10 * accelera;
154
155 //freio da moto
156 if (Input.GetAxis("Vertical") < 0){
157     WheelColT.brakeTorque = Freio * Input.GetAxis("Vertical");
158 }
159
160 }
161
162 //direção da moto
163 WheelColF.steerAngle = SteerAngle * Input.GetAxis ("Horizontal");
164

```

Figura 4.4 – Trecho de código da aplicação de forças para movimentação da motocicleta.

A direção da motocicleta é ajustada através de uma propriedade da roda dianteira, chamada *SteerAngle*. De acordo como o usuário solicita a mudança de direção pelo teclado do computador, a variável aumenta seu valor e o ângulo, em graus, é alterado.

4.2.2 Propriedades Físicas da Motocicleta

Interpretar as propriedades físicas do mundo real em um computador, requer um processamento pesado, e ainda assim, não é possível interpretar todas as propriedades físicas do mundo real. Muitas *game engines* contêm um motor de física responsável por realizar o mapeamento e processamento destas características dos jogos. A Unity3D faz uso da Nvidia's PhysX Engine, uma popular e eficiente *engine* comercial de física.

A base para criar movimentos realistas na Unity3D é o uso do componente *RigidBody* em *Game Objects*. A partir deste componente, qualquer *Game Object* pode conter quatro propriedades físicas principais: Velocidade, Massa, Gravidade, e Fricção. Como o objeto contém uma massa específica, representada em quilos, é necessário adicionar um centro de massa ao objeto. Este centro de massa possibilita que a motocicleta receba forças e tenha um ponto de equilíbrio. A posição do centro de massa fica abaixo da motocicleta, pois assim ela ganha estabilidade para manter-se na posição vertical.

Na realização de uma curva, a motocicleta é levemente inclinada para manter o equilíbrio e não deixar a força centrífuga jogá-la para a direção contrária da curva. Para estabilizar a motocicleta na posição vertical, após a realização de uma curva, é utilizada a função *addRelativeTorque*.

4.2.3 Aceleração da Motocicleta via Potenciômetro

Para aceleração da motocicleta é utilizado um potenciômetro o qual tem funcionamento semelhante ao manete de aceleração de uma motocicleta real. Nesta etapa do trabalho é utilizado um potenciômetro conectado a uma placa protoboard. A placa protoboard esta conectada a uma placa Arduino Uno, que por sua vez está conectada ao desktop onde o cenário de simulação é executado pela Unity3D.

Ao girar o pino do potenciômetro, um valor analógico é recebido na placa microcontroladora Arduino conectada a este potenciômetro. A placa contém uma rotina na linguagem de programação C que a todo instante lê o valor enviado pelo potenciômetro. Este valor é então acondicionado em uma variável e enviado para a Unity3D, onde está o cenário virtual.

Para envio do valor referente ao estado do potenciômetro, é criada uma conexão serial assíncrona via Serial Port que envia um valor inteiro entre 0 e 1023 para a porta 9000, a qual o Arduino está conectado para escrita de valores no *buffer*.

No Script de movimentação na Unity3D, é criada uma Serial Port na porta 9000 para leitura de valores escritos no *buffer* desta conexão. Logo que um valor é recebido, o mesmo é utilizado na aceleração da motocicleta. A Figura 4.5 representa o acelerador da motocicleta com o potenciômetro instalado na placa Protoboard, que está conectada a placa Arduino Uno.

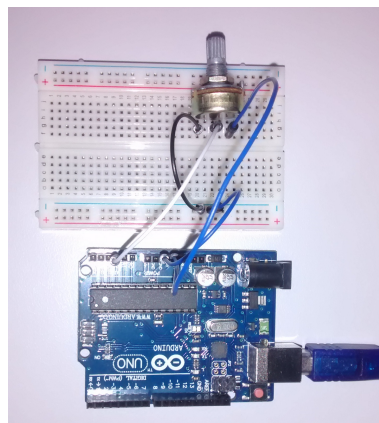


Figura 4.5 – Potenciômetro conectado a Protoboard e ligado ao Arduino Uno.

4.3 Graus de Liberdade da Plataforma de Movimentos

Para um simulador de motocicleta com *motion*, é importante simular frenagens, acelerações, inclinação em curvas e inclinação em aclives e declives. No estudo de Yamazaki (1998) foi

ressaltado que os graus de liberdade Pitch e Roll são responsáveis pelo comportamento principal da movimentação da motocicleta. Conclui-se então que dois graus de liberdade são suficientes para dar a sensação de estar em uma dessas situações. Neste trabalho usa-se dois graus de liberdade para a simulação.

Os movimentos usados são chamados de Pitch e Roll, ilustrados na Figura 2.3. Pitch é o movimento de rotação do eixo coordenado responsável por deslocar verticalmente a parte traseira ou dianteira de uma motocicleta, simulando estar em um aclave ou declive. A Figura 4.6, extraída do projeto criado na Unity3D, representa o movimento e o cálculo do ângulo relacionado.

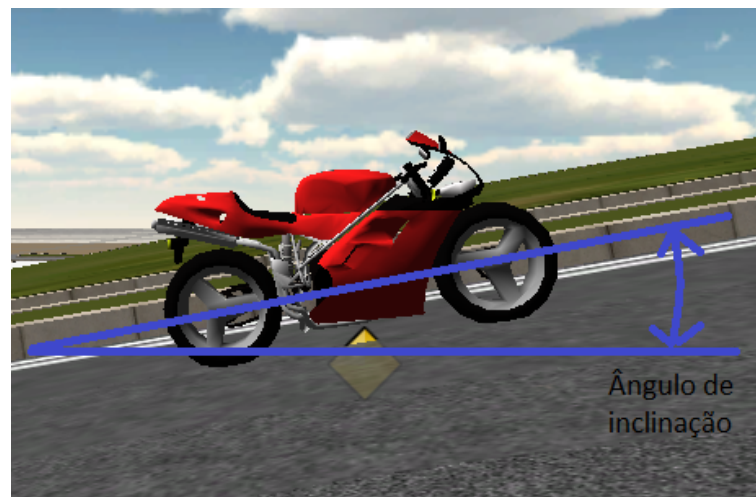


Figura 4.6 – Representação do movimento de rotação Pitch na motocicleta introduzida no cenário criado.

O movimento chamado Roll é a rotação de outro eixo coordenado, que simula a inclinação da motocicleta em uma curva. Supondo que a motocicleta esteja em uma reta com velocidade Q , e em um certo momento a mesma começa a realizar uma curva em seu trajeto. O ângulo responsável pela inclinação lateral da motocicleta passa de um ângulo reto, para um valor, em graus, diferente, variando a posição vertical da motocicleta para uma posição inclinada, dependente também da velocidade. A Figura 4.7 exemplifica o movimento Roll em uma motocicleta.

Uma frenagem pode ser simulada através de um leve deslocamento vertical da roda traseira da motocicleta para cima. Uma aceleração positiva faz com que o corpo do motociclista receba uma força que o empurre para trás. Realizando uma leve inclinação na roda dianteira, pode-se fazer a aceleração ser percebida pelo motociclista.



Figura 4.7 – Representação do movimento de rotação Roll em uma motocicleta (<http://www.conti-online.com>).

4.4 Detecção de eventos para Motion

Para que o *motion* represente eventos ocorridos no cenário virtual de simulação, é necessário que dados sejam calculados, afim de representarem eventos correspondentes aos movimentos que a Plataforma de Movimentos irá realizar. As Subseções 4.4.1 e 4.4.2 exemplificam como dados são capturados no Script de movimentação da motocicleta e qual é o significado destes dados para a Plataforma de Movimentos. A subseção 4.4.3 exemplifica como são simuladas frenagens e acelerações no protótipo criado.

4.4.1 Ângulo de Inclinação do Movimento Pitch

Para o cálculo do ângulo de inclinação do movimento Pitch, conforme visto na Figura 2.3, é necessário saber a diferença da distância entre as rodas, e a diferença da altura das rodas em relação ao solo. Assim, através da equação de cálculo do seno de um ângulo, é possível definir o ângulo de inclinação da motocicleta. A Equação 4.1 representa este cálculo, e a Figura 4.8 representa estes valores em uma motocicleta.

$$angulo = \arcsin\left(\frac{diferencadealturadasrodas}{distanciadasrodas}\right) \quad (4.1)$$

Através do cálculo do ângulo de inclinação, foi definido que ângulos negativos correspondem a inclinação da motocicleta quando ela está subindo um aclave. Já ângulos positivos são gerados quando a motocicleta está descendo um declive.



Figura 4.8 – Cálculo do Movimento Pitch da Motocicleta.

4.4.2 Ângulo de Inclinação do Movimento Roll

Para o cálculo do movimento de inclinação Roll da motocicleta foram utilizados dois referenciais inseridos na parte superior da motocicleta, sendo que um está à direita da motocicleta e o outro está à esquerda. Estes referenciais são Game Objects inseridos para auxiliar na detecção de possíveis eventos na Plataforma de Movimentos. Através da posição dos referenciais é possível definir o ângulo de inclinação da motocicleta quando ela realiza uma curva.

Para calcular o ângulo de inclinação Roll é utilizada a Equação 4.2, onde a diferença de altura dos referenciais é dividida pela distância dos referenciais, sendo que o quociente desta divisão é utilizado como parâmetro na função do cálculo do ArcoSeno, resultando assim, o ângulo da inclinação. A Figura 4.9 representa onde os referenciais estão inseridos e os valores utilizados na equação do cálculo do ângulo.

$$angulo = \arcsin\left(\frac{diferençadealturadosreferenciais}{distanciaentreferenciais}\right) \quad (4.2)$$

Para o padrão de comunicação entre as Plataformas, é definido que os ângulos negativos referentes ao movimento Roll são inclinações da motocicleta para o lado direito, já ângulos positivos para o movimento Roll são inclinações da motocicleta para a esquerda.

4.4.3 Sensação de Acelerações e Desacelerações no *Motion*

Para a simulação de acelerações e frenagens no *motion* é detectado quando a aceleração sofre alta variação, e transformada essa variação em um valor de ângulo para ser adicionado ao ângulo do movimento Pitch. Assim, a sensação de aceleração e frenagens se dará por uma leve

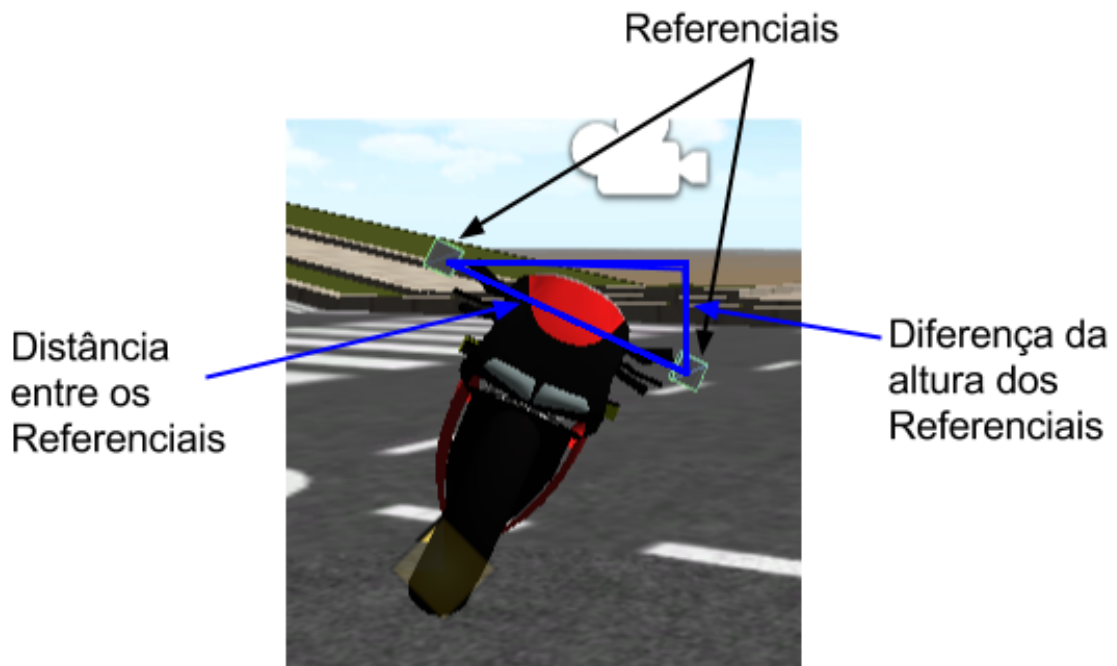


Figura 4.9 – Referenciais para cálculo do movimento Roll na Motocicleta.

inclinação da motocicleta com a variação do ângulo Pitch.

A variação da aceleração detectada, gera um ângulo negativo para no mesmo instante ser adicionado ao ângulo de inclinação Pitch. Para fins deste trabalho, é empregada uma função linear para reduzir ao longo do tempo o ângulo que gera a sensação de aceleração, aproximando-se ao que ocorre na realidade. A mesma técnica é utilizada para dar ao usuário sensação de uma força o empurrando para frente, quando uma frenagem ocorre. O que varia nessa situação é o valor do ângulo incrementado ao ângulo Pitch, tendo de ser um valor positivo.

4.5 Padrão de Comunicação entre Unity3D e Plataforma de Movimentos

Dados para comunicação entre duas plataformas independentes devem seguir um padrão para que os dados tenham o mesmo sentido no sistema criado. A comunicação entre o cenário virtual e a Plataforma de Movimentos é unidirecional, sendo que dados são enviados apenas pela Unity3D e são recebidos apenas pela Plataforma de Movimentos. Um canal de comunicação serial assíncrono é estabelecido entre as plataformas, utilizando a Serial Port 9696.

Os dados que são transmitidos estão no formato String e cada pacote contém 6 bytes. O primeiro byte indica o sinal do ângulo Pitch, os segundo e terceiro bytes são o valor do ângulo Pitch transmitido em questão. O quarto byte indica o sinal do ângulo Roll e o quinto e sexto

bytes indicam o valor do ângulo Roll. O ângulo Pitch tem variação entre -30 e 30, dependendo do aclave enfrentado. O ângulo do movimento Roll, varia de -45 a 45, dependendo do ângulo de inclinação lateral da motocicleta.

4.6 Protótipo da Plataforma de Movimentos

Devido ao alto custo para obter uma Plataforma de Movimentos, optou-se por construir um protótipo da mesma. A Plataforma de Movimentos tem seu funcionamento mecânico com base em 4 pistões dispostos sob uma base quadrada onde se encontra a motocicleta. Os pistões são posicionados próximo ao centro de cada um dos quatro lados da base. Os pistões posicionados em lados opostos da base são acionados em conjunto, sendo que se um pistão sobe, o outro pistão desce, e vice e versa, fazendo com que a região central da base permaneça na mesma altura e as extremidades se movimentem verticalmente, gerando os movimentos de inclinação Pitch e Roll da motocicleta fixada sobre a base. A Figura 4.10 representa a base e os pistões para acionamento.

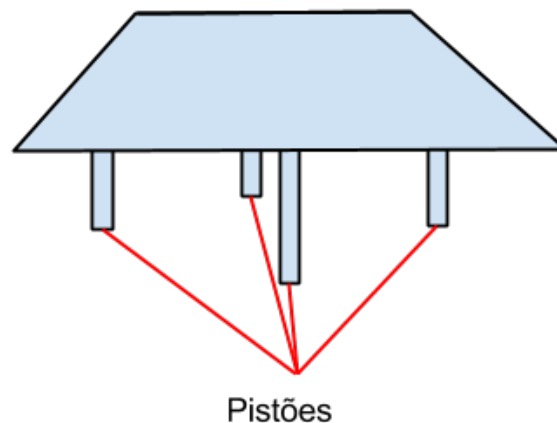


Figura 4.10 – Base e Pistões da Plataforma de Movimentos.

Para este trabalho utilizou-se uma placa micro-controladora Arduino Uno, placa Proto-board e Leds para construir um protótipo da Plataforma de Movimentos. O protótipo utiliza a linguagem de programação C, disponibilizada para desenvolvimento e programação no Arduino.

A placa micro-controladora é carregada com um código que recebe dados via conexão

serial. A conexão é inicializada na Serial Port 9696 e aguarda que dados sejam recebidos. Com o dado recebido, é inicializada a avaliação do ângulo recebido para ao fim enviar comandos até a placa Protoboard para acionamento dos Leds correspondentes ao ângulo de inclinação solicitado. Estes Leds simulam o acionamento dos pistões verticalmente ou quando ele está estável na posição inicial.

Na montagem e configuração dos Leds na Protoboard, é definido que um Led quando acionado demonstra que um pistão foi movimentado para cima ou para baixo do seu estado estável, conforme a configuração desejada e implementada no protótipo da Plataforma de Movimentos.

Supondo que tenha-se duas fileiras com três Leds na parte superior da Protoboard, e duas fileiras com três Leds na parte inferior da Protoboard, conforme disposto na Figura 4.11. A fileira de Leds na parte superior direita da Protoboard indica que o pistão foi movimentado para baixo e a fileira de Leds na parte superior esquerda da Protoboard significa que quando acesa, o estado do pistão está acima do estado inicial. A sequência de Leds posicionada serve para ver em qual intervalo determinado está o ângulo de inclinação solicitado.

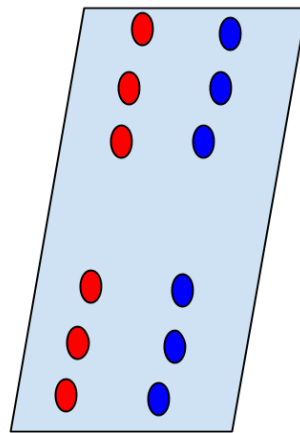


Figura 4.11 – Protoboard e Leds.

Partindo da suposição acima, e que os Leds representem o acionamento dos pistões para o ângulo de inclinação Pitch, quando todos os Leds estão apagados, significa que o ângulo de inclinação Pitch é nulo. Agora, se o primeiro Led está aceso na parte superior esquerda e o primeiro Led está aceso na parte inferior direita, significa que o ângulo de inclinação Pitch está em um intervalo negativo, onde a motocicleta se encontra em um aclive. Conforme os Leds seguintes a estes são acionados, significa que a inclinação do aclive é maior. Da mesma forma ocorre para as outras duas fileiras de Leds. Se o Led da parte superior direita estiver aceso e o

Led da parte inferior esquerda estiver aceso, significa que a motocicleta está com um ângulo de inclinação positivo, o que implica que ela está percorrendo um declive.

Para o ângulo de inclinação do movimento Roll recebido, utiliza-se da mesma técnica do ângulo Pitch de inclinação para prototipação dos pistões da Plataforma de Movimentos. A Figura 4.12 representa o protótipo da Plataforma de Movimentos com os Leds dispostos e conectados a placa Arduino Uno.

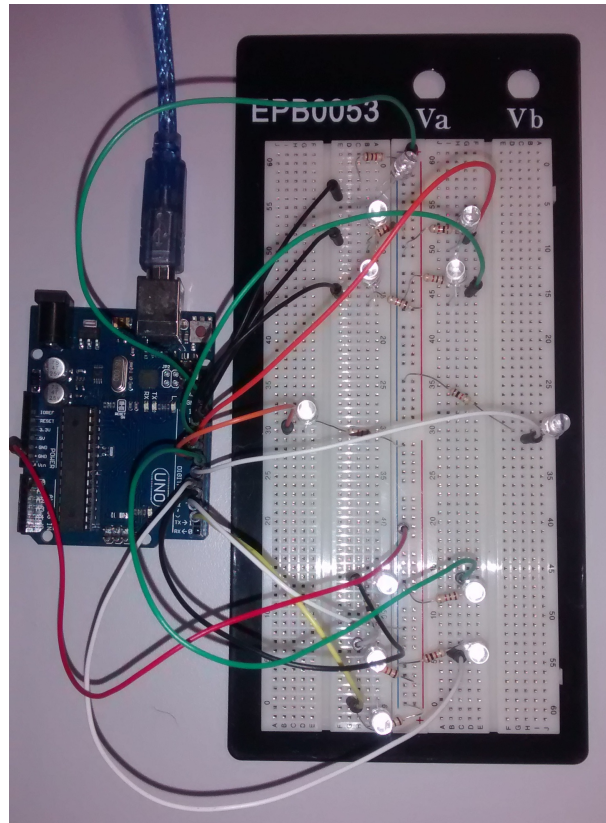


Figura 4.12 – Protótipo da Plataforma de Movimentos.

5 TESTES E RESULTADOS

Para o presente trabalho foram definidos cenários específicos para a realização dos testes afim de validar o sistema de simulação de motocicleta desenvolvido. É importante ressaltar que para cada teste é alterada a disposição dos Leds na plataforma de Movimentos, e consequentemente, alterada a configuração da rotina de recebimento de dados no protótipo da Plataforma de Movimentos. Esta alteração se dá pelo fato de o hardware Arduino Uno ter quantidade limitada de conexões para saída de dados. As subseções a seguir descrevem os cenários de teste, a configuração do protótipo da Plataforma de Movimentos, os resultados esperados e resultados obtidos.

5.1 Experimento 1

Para o movimento de rotação Pitch, faz-se necessário testes com a motocicleta passando por aclives e declives. O teste é composto por um trajeto onde a motocicleta percorrerá um aclive de 90 metros, um declive de 90 metros, dois quebra-molas, onde a variação do ângulo de inclinação ocorre rapidamente, e ainda 3 ondulações pequenas, para teste mais preciso da sincronização. Ao longo do trajeto é necessária a captura de dados para comparação. A Figura 5.1 representa o trajeto percorrido neste teste.

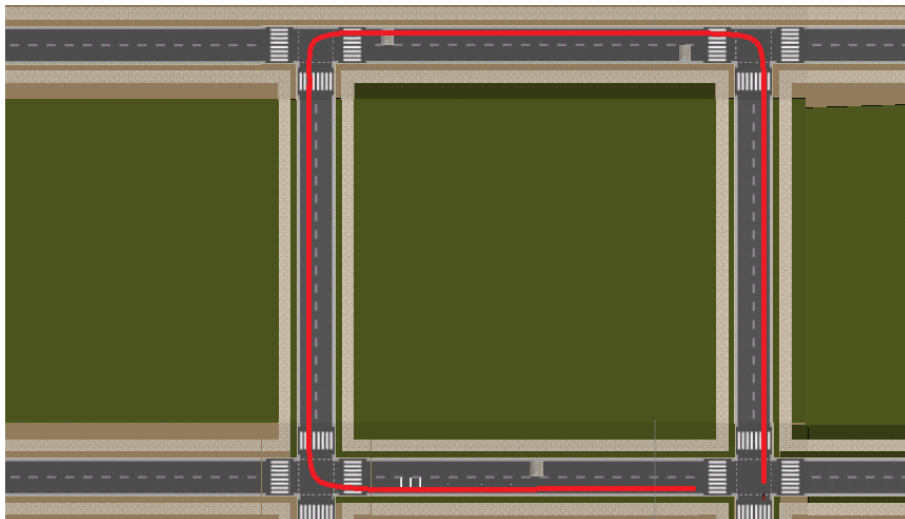


Figura 5.1 – Trajetória do Experimento 1.

Espera-se neste teste que os dados entre ângulo da inclinação da motocicleta no cenário virtual e ângulo de inclinação da motocicleta no protótipo da Plataforma de Movimentos sejam

correspondentes nos testes variados.

Para a visualização dos ângulos calculados para o movimento Pitch e Roll, é colocada uma função no Script de movimentação com a finalidade de imprimir os ângulos na tela que o usuário utiliza para visualização. Assim será possível extrair dados do teste para comparação.

5.1.1 Configuração do protótipo da Plataforma de Movimentos

Para este teste a disposição dos Leds concentra-se na parte superior e inferior da Protoboard. Duas fileiras na parte superior e duas fileiras na parte inferior da Protoboard fazem a simulação dos pistões para o movimento Pitch de inclinação. A Figura 4.12 representa a configuração de Leds disposta na Protoboard para este experimento.

A fileira de Leds da parte esquerda, tanto inferior como superior, infere que o pistão é acionado para cima, já a sequência de Leds na parte direita, infere que o pistão foi acionado para baixo. Dois Leds adicionais são colocados no centro à esquerda e no centro à direita, indicando que a motocicleta está inclinada para a direita ou para a esquerda, em relação ao movimento Roll.

Como no protótipo tem-se 3 Leds para sinalizar que o pistão foi acionado para cima, o primeiro Led da sequência acende quando o ângulo de inclinação é maior que 2 graus, o segundo Led acende quando o ângulo é maior que 7 graus e o terceiro Led acende quando é maior que 14 graus. Para as fileiras de Leds à direita inferior e superior da Protoboard, o primeiro Led é aceso quando o ângulo é maior que 2 graus e o segundo Led é aceso quando o ângulo é maior que 7 graus.

5.1.2 Resultados do Experimento 1

Para a trajetória traçada (Figura 5.1), foram extraídos dados do cenário virtual e do protótipo da Plataforma de Movimentos durante o percurso com a motocicleta. Estes dados foram coletados em momentos críticos, onde o ângulo varia muito rapidamente, para a troca de mensagens entre Unity3D e Arduino. A Tabela 5.1 contém uma amostra dos dados coletados, onde tem-se o ângulo de inclinação da motocicleta do cenário virtual, e a simulação da Plataforma de Movimentos através do acendimento dos Leds, conforme configuração especificada. Os números das linhas identificadas por Led superior esquerda, Leds superior direita, Leds inferior esquerda e Leds inferior direita representam a quantidade de Leds acesos nesta sequência.

As Figuras 5.2 e 5.3 representam o momento em que a motocicleta está em um quebra-

Tabela 5.1 – Ângulos capturados no cenário e quantidade de Leds acesos no protótipo.

Ângulo no Cenário	0	-2	-12	-17	-14	-3	-5	-11	-6	5	12	16	13	5
Leds superior esquerda	0	1	2	3	2	1	1	2	1	0	0	0	0	0
Leds superior direita	0	0	0	0	0	0	0	0	0	1	2	3	2	1
Leds inferior esquerda	0	0	0	0	0	0	0	0	0	1	2	2	2	1
Leds inferior direita	0	1	2	2	2	1	1	2	1	0	0	0	0	0

molhas e o respectivo movimento na Plataforma de Movimentos ilustrado através do protótipo. Neste instante, o cenário virtual apontava para um ângulo no valor de -11 graus.



Figura 5.2 – Posição da motocicleta em um determinado instante do Experimento 1.

5.2 Experimento 2

Para validar a sincronização de eventos detectados no cenário e eventos decorrentes na Plataforma de Movimentos relacionados ao movimento Roll, exemplificado na Figura 4.8, é realizada uma trajetória com a motocicleta de modo que ela passe por uma curva para a esquerda com velocidade relativamente baixa, uma curva para a direita com velocidade baixa, uma curva para a esquerda com velocidade alta e uma última curva para a direita com velocidade relativamente alta. A Figura 5.4 exemplifica a trajetória acima descrita.

Este teste tem a principal finalidade de detectar a diferença de inclinação da motocicleta de acordo com a velocidade, e também o movimento correspondente de cada inclinação na Plataforma de Movimentos.

Para este experimento espera-se que a motocicleta adquira um ângulo de inclinação pequeno quando efetuada a curva com velocidade relativamente baixa (entre 10 e 20 Km/H). Já para velocidades maiores (entre 40 e 60 Km/H), espera-se que o ângulo de inclinação seja maior quando a mesma curva é efetuada.

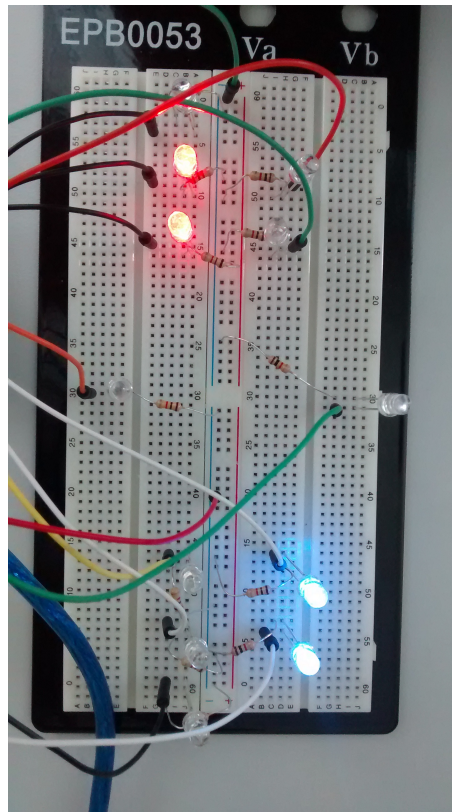


Figura 5.3 – Simulação da Plataforma de movimentos para a posição da motocicleta na Figura 5.2.

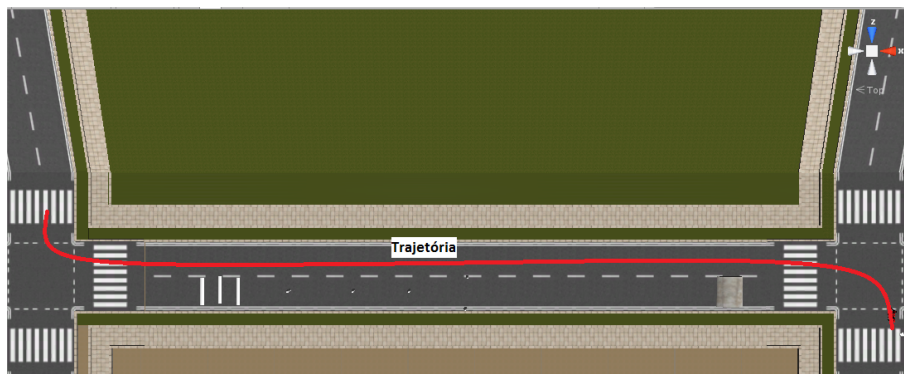


Figura 5.4 – Trajetória a ser percorrida no Experimento 2.

5.2.1 Configuração do protótipo da Plataforma de Movimentos

Para este teste os Leds do protótipo devem se concentrar nas duas laterais da Protoboard, onde os ângulos do Movimento Roll são simulados. Portanto, posiciona-se duas fileiras de Leds na lateral direita da Protoboard e duas fileiras de Leds na lateral esquerda da Protoboard. A fileira de Led inferior, tanto da esquerda como da direita, indica que o respectivo pistão da Plataforma de Movimentos é acionado para baixo. A fileira de Led superior, tanto da direita como da esquerda significa que o respectivo pistão foi acionado para cima. A Protoboard e os

Leds posicionados para este teste podem ser vistos na Figura 5.5.

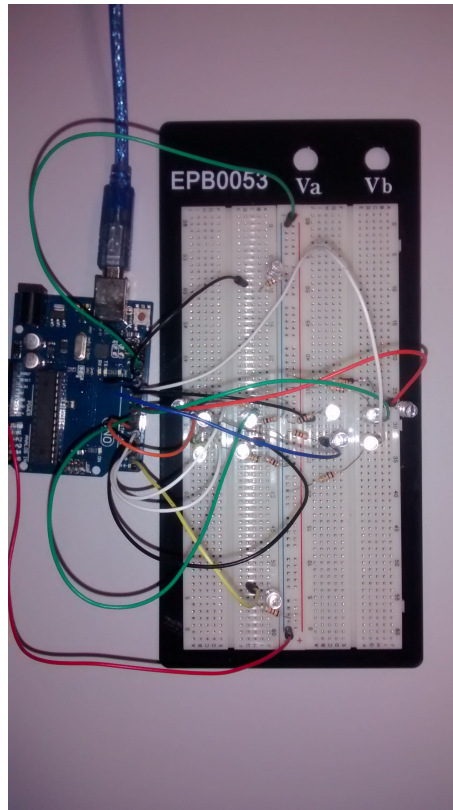


Figura 5.5 – Protótipo para teste do movimento de inclinação Roll.

No protótipo tem-se 3 Leds para sinalizar se o pistão da Plataforma de Movimentos foi acionado para cima, e 2 Leds para simular que o pistão foi acionado para baixo. Quando o primeiro Led da sequência superior acende significa que a motocicleta está com ângulo de inclinação Roll maior que 4 graus, já o segundo Led indica um ângulo maior que 15 graus, e o último Led indica um ângulo de inclinação maior que 30 graus. Para a sequência de Leds inferior à acima citada, quando o primeiro Led acende significa que a motocicleta está com ângulo de inclinação maior do que 4 graus, já o segundo Led aceso significa que a motocicleta está com mais de 15 graus de inclinação Roll.

5.2.2 Resultados do Experimento 2

A Tabela 5.2 refere-se aos dados capturados durante o trajeto com a motocicleta em velocidade baixa (10 a 20 km/h). Já na Tabela 5.3, a motocicleta estava com velocidade superior ao primeiro teste (20 a 40 km/h).

Na comparação dos dados, nota-se que para velocidades maiores a motocicleta tem maior inclinação, o que também é refletido no protótipo. A Figura 5.6 traz o instante em

Tabela 5.2 – Ângulos capturados no cenário e quantidade de Leds acesos no protótipo.

Ângulo no Cenário	0	-5	-14	-13	-8	0	6	13	14	5
Leds esquerda superior	0	1	1	1	1	0	0	0	0	0
Leds esquerda inferior	0	0	0	0	0	0	1	1	1	1
Leds direita superior	0	0	0	0	0	0	1	1	1	1
Leds direita inferior	0	1	1	1	1	0	0	0	0	0

Tabela 5.3 – Ângulos capturados no cenário e quantidade de Leds acesos no protótipo.

Ângulo no Cenário	0	-10	-20	-35	-23	-14	-7	0	16	32	36	24	11
Leds esquerda superior	0	1	2	3	2	1	1	0	0	0	0	0	0
Leds esquerda inferior	0	0	0	0	0	0	0	0	2	3	3	2	1
Leds direita superior	0	0	0	0	0	0	0	0	2	2	2	2	1
Leds direita inferior	0	1	2	2	2	1	1	0	0	0	0	0	0

que a motocicleta está efetuando uma curva, e a Figura 5.7 é o mesmo instante no protótipo da plataforma de movimentos.



Figura 5.6 – Instante que a motocicleta está efetuando uma curva.

5.3 Experimento 3

O Experimento 3 busca a validação da sensação de acelerações e frenagens da motocicleta. Acelerações com diferentes intensidades são empregadas na motocicleta que inicia estacionada, já na frenagem, é acionado o freio com diferentes intensidades. Para este teste espera-se que o usuário sinta uma força o empurrando para trás em uma aceleração positiva, para isso, o ângulo de inclinação Pitch varia seu valor para dar esta sensação. Na frenagem ocorre a ação semelhante ao ângulo de inclinação Pitch, mas com valor oposto, simulando que o usuário sinta uma força o empurrando para frente.

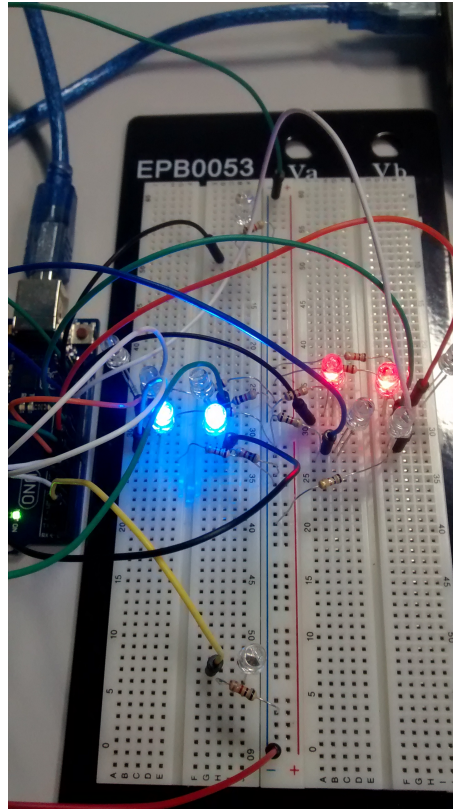


Figura 5.7 – Protótipo indicando o movimento dos pistões da Plataforma de Movimentos para o instante da motocicleta da Figura 5.6.

A disposição dos Leds na protoboard é semelhante a utilizada no Experimento 1. A mudança ocorre no intervalo de valores que um Led simula, tendo valores menores para este teste, já que o ângulo não sofre grande variação. A Tabela 5.4 representa uma amostra dos dados do valor relacionado à aceleração, do ângulo gerado, e dos Leds acesos em cada sequência, durante uma aceleração da motocicleta. Já a Tabela 5.5 representa uma amostra de dados do valor relacionado a frenagem, o ângulo gerado, e dos Leds acesos em cada sequência, durante uma frenagem.

Tabela 5.4 – Dados durante a aceleração da motocicleta.

Valor de Aceleração	0	100	200	250	250	250	250
Ângulo gerado	0	-2	-4	-4	-3	-2	-1
Leds superior esquerda	0	1	2	2	1	1	0
Leds superior direita	0	0	0	0	0	0	0
Leds inferior esquerda	0	0	0	0	0	0	0
Leds inferior direita	0	1	2	2	1	1	0

Analisando os resultados das Tabelas 5.4 e 5.5 pode-se notar que o protótipo realiza a simulação de frenagens e acelerações gerando pequenas inclinações na Plataforma de Movi-

Tabela 5.5 – Dados durante frenagem da motocicleta.

Valor de frenagem	0	0.3	1	1	1	1	1
Ângulo gerado	0	1	4	4	3	2	0
Leds superior esquerda	0	1	2	2	1	1	0
Leds superior direita	0	0	0	0	0	0	0
Leds inferior esquerda	0	0	0	0	0	0	0
Leds inferior direita	0	1	2	2	1	1	0

mentos.

6 CONCLUSÃO

Este trabalho teve como objetivo sincronizar um cenário virtual de simulação de motocicleta e um protótipo de uma Plataforma de Movimentos afim de fornecer um *feedback* dos acontecimentos decorridos no cenário para o usuário. A comunicação realizada por Serial Ports se mostrou eficiente e satisfaz os requisitos exigidos no trabalho, como persistência dos dados transmitidos, sem atrasos e falhas.

No desenvolvimento de um sistema real para movimentação da motocicleta na *game engine* Unity3D não foram desenvolvidos comportamentos realistas quando ocorrem frenagens em curvas ou a motocicleta sendo empinada quando uma aceleração muito forte é empregada. Esta dificuldade era esperada no desenvolvimento da simulação dos movimentos da motocicleta, por se tratar de um sistema complexo de Física para simulação em computador.

Apesar de o hardware do protótipo ser limitado na representação da Plataforma de Movimentos, o gerenciador do protótipo continha todas as informações para a simulação completa dos eventos detectados no cenário, sendo comprovado com a análise dos testes realizados. Adicionalmente, o hardware mostrou-se capacitado para trocar um maior número de informações, não influenciando o desempenho do sistema.

Os aspectos seguintes são sugestões de trabalhos futuros:

- Instrumentação de uma motocicleta para pilotagem.
- Inserção de outros Graus de Liberdade para o sistema.
- Utilização do Óculos Rift como componente visual do usuário.
- Inserção de Áudio 3D no cenário virtual e Surround no ambiente de simulação (OLIVEIRA, 2014).

REFERÊNCIAS

- BALBINOT, A. B.; TIMM, M. I.; ZARO, M. A. Aplicação de Jogos e Simuladores como Instrumentos para Educação e Segurança no Trânsito. **Novas Tecnologias na Educação**, [S.l.], v.7, n.1, 2009.
- BANZI, M. **Arduino**. Disponível em: <http://www.arduino.cc>, Acessado em Novembro de 2014.
- BRAGA, M. Realidade Virtual e Educação. **REVISTA DE BIOLOGIA E CIÊNCIAS DA TERRA**, [S.l.], v.1, n.1, 2001.
- CASUTT, G. et al. The drive-wise project: driving simulator training increases real driving performance in healthy older drivers. **Front Aging Neurosci**, [S.l.], v.6, 2014.
- CLUA, E. W. G.; BITTENCOURT, J. R. Desenvolvimento de Jogos 3D: concepção, design e programação. **Anais da XXIV Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação**, [S.l.], p.1313–1356, 2005.
- COSSALTER, V. et al. A motorcycle riding simulator for assessing the riding ability and for testing rider assistance systems. **DSC**, [S.l.], p.48–58, 2006.
- FERRAZZIN, D. et al. THE MORIS MOTORCYCLE SIMULATOR: an overview. **SAE Technical Paper**, [S.l.], 2001.
- FRANCO, G. Effects of AI on driving experience. **Master 's thesis. University of Skövde**, [S.l.], p.44, 2013.
- GOLDSTONE, W. **Unity Game Development Essentials**. BIRMINGHAM - UK: Packt Publishing, 2009.
- GROOT, S. de et al. Nonvestibular Motion Cueing in a Fixed-Base Driving Simulator: effects on driver braking and cornering performance. **Presence**, [S.l.], v.20, n.2, p.117–142, 2011.
- GRUMAN, E. **Detran/RS divulga nota sobre nova Resolução do Contran e simuladores de direção**. Disponível em: http://www.portal.rs.gov.br/conteudo/198444/detran/rs-divulga-nota-sobre-nova-resolucao-do-contran-e-simuladores-de-direcao/termosbusca=*, Acesso em Setembro de 2014.

HIMA, S. et al. Suitable Two Wheeled Vehicle Dynamics Synthesis for Interactive Motorcycle Simulator. **17th IFAC World Congress (IFAC'08)**, [S.l.], p.96–101, 2008.

LEBRAM, M.; ENGSTROM, H.; GUSTAVSSON, H. A driving simulator based on video game technology. **In Proceedings of SIGRAD**, [S.l.], p.39–43, 2006.

MCROBERTS, M. **Arduino Básico**. São Paulo - Brasil: Novatec Editora, 2011.

NILSSON, L. . Behavioural research in an advanced driving simulator-experiences of the VTI system. **Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting 37**, [S.l.], p.612–616, 1993.

OLIVEIRA, V. D. de. **Desenvolvimento de um Protótipo de Sistema de Audio para um Simulador de Motocicleta**. 2014.

PENNESTRI, E.; CAVACECE, M.; VITA, L. On the Computation of Degrees-of-Freedom: a didatic perspective. **International Design Engineering Technical Conferences and Computers and Information in Engineering Conference**, [S.l.], 2005.

PEREIRA, J. A. Estudo sobre a Influência dos Jogos de Corrida no Comportamento de Motoristas. **VIII Brazilian Symposium on Games and Digital Entertainment**, [S.l.], p.30–33, 2009.

RIENER, A. Assessment of Simulator Fidelity and Validity in Simulator and On-the-road Studies. **International Journal on Advances in Systems and Measurements**, [S.l.], v.3, n.3 e 4, p.110–124, 2010.

SAUVE, L. et al. Distinguishing between games and simulations: a systematic review. **Educational Technology e Society**, [S.l.], v.10, n.3, p.247–256, 2007.

Spurgeon, B. **Lewis Hamilton's great performance, and his great preparation**. Disponível em: <http://rendezvous.blogs.nytimes.com/2007/06/09/lewis-hamiltons-great-performance-and-his-great-preparation/>, Acesso em Setembro de 2014.

UNITY3D. **Unity**. Disponível em: <http://www.unity3d.com>, Acessado em Novembro de 2014.

YAMASAKI, G. et al. Development of motorcycle training simulator. **JSAE**, [S.l.], p.81–85, 1998.