6-10-2019

Virtual Reality Physics Scenarios

Patric Zhang

Thomas King

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 10, 2019

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Patric Zhang
Thomas King

ENTITLED

Virtual Reality Physics Scenarios

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Daniel W. Lewis
Thesis Advisor

Department Chair

Virtual Reality Physics Scenarios

by

Patric Zhang
Thomas King

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 10, 2019

Virtual Reality Physics Scenarios

Patric Zhang
Thomas King

Department of Computer Engineering
Santa Clara University
June 10, 2019

ABSTRACT

Many students taking physics early on in their education find that it may be difficult to associate the theory they learn in class with how physics works in real world scenarios. Through various experiments in class, students are able to see examples of physics phenomena, but those experiments are limited by equipment, and do not offer precise data. To combat this, we are creating a virtual reality application for students to use to help learn physics. This report details the requirements the system will meet, as well as the use cases and subsequent activity diagrams for all users. We have also included a conceptual model of our system, as well as an explanation for technologies used, and a test plan and development timeline.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Physics is a necessary subject that many students take early in their education. When learning about physics concepts, students need to see examples of how those concepts work. The two main methods of working with physics scenarios are exercises on paper and real life demonstrations. Exercises help students learn how to apply mathematics to physics problems, but students can have a hard time visualizing how those exercises would work in real life. To compensate for this, many teachers provide real life experiments or labs. The problem with those methods is that it is difficult to get precise data from experiments, and the kind of experiments that can be performed is limited by the equipment that is available. Our solution is to create a virtual reality (VR) application to simulate physics experiments.

When learning about kinematics, students are commonly asked to determine the trajectory of a thrown object. It teaches students how to use kinematics equations to figure out how to derive a solution from given data. For example, they might be asked to determine the trajectory of an object given the angle and force with which it was thrown. Working with numbers is helpful for learning about how physics works, but it can be hard to take those numbers and imagine what it would look like. Most students dont have a good feel for how fast 3.5 m/s is or how much force 10N is. If a teacher wanted to do a real life demonstration to solve that, their options are limited. They could throw a ball themselves, but then they couldn't get enough data to be useful. Another approach would be to set up an experiment with lab equipment, but these experiments are usually not very precise or limited in terms of what they can do. A common solution for kinematics problems is to move a cart along a track with sensors. The issue with that setup is that it can only record movement in one dimension. It doesn't record the force applied to the cart to get it moving, and the movement of the cart would be affected by variables like friction and air resistance that are outside of the model the students are trying to learn. A perfect solution would allow an experiment to be performed that is identical to the exercises the students are working with.

1.2 Solution

With virtual reality, we can create a learning experience that solves the issues of both methods. A student could watch the trajectory of a thrown ball while seeing how that trajectory follows the mathematics of physics equations. The application would be able to show the theory behind the physical actions. As many physics concepts are observable, virtual reality can bolster the learning of topics such as kinematics, forces, thermodynamics, magnetism, and light. Furthermore, students may be more motivated to learn through virtual reality, especially when learning about concepts that are traditionally more boring or hard to visualize.

Chapter 2

Requirements

The following requirements, divided into functional, non-functional, and design constraints, define what needs to be completed. The critical requirements are necessary, while the recommended requirements will be completed if given enough time.

2.1 Functional

- Users can interact with objects in a 3D space
- Software will portray realistic physics scenarios
- Data based on the scenario will be viewable
- Teachers can choose what scenario is experienced
- There will be some specific scenarios regarding...
- Users can give feedback on the application

2.2 Non-Functional

- The application should be easy enough for a middle school student to use
- The application should be able to be run by a standard smart phone
- The application should be easy to add new scenarios
- The application should be fast and responsive

2.3 Design Constraints

- Uses a cell phone run VR headset
- Runs on Android

Chapter 3

Use Cases

The system has two roles, students and teachers. Any user can access the whole functionality of the app, but students and teachers will generally be doing different roles.

3.1 Use Case Diagram

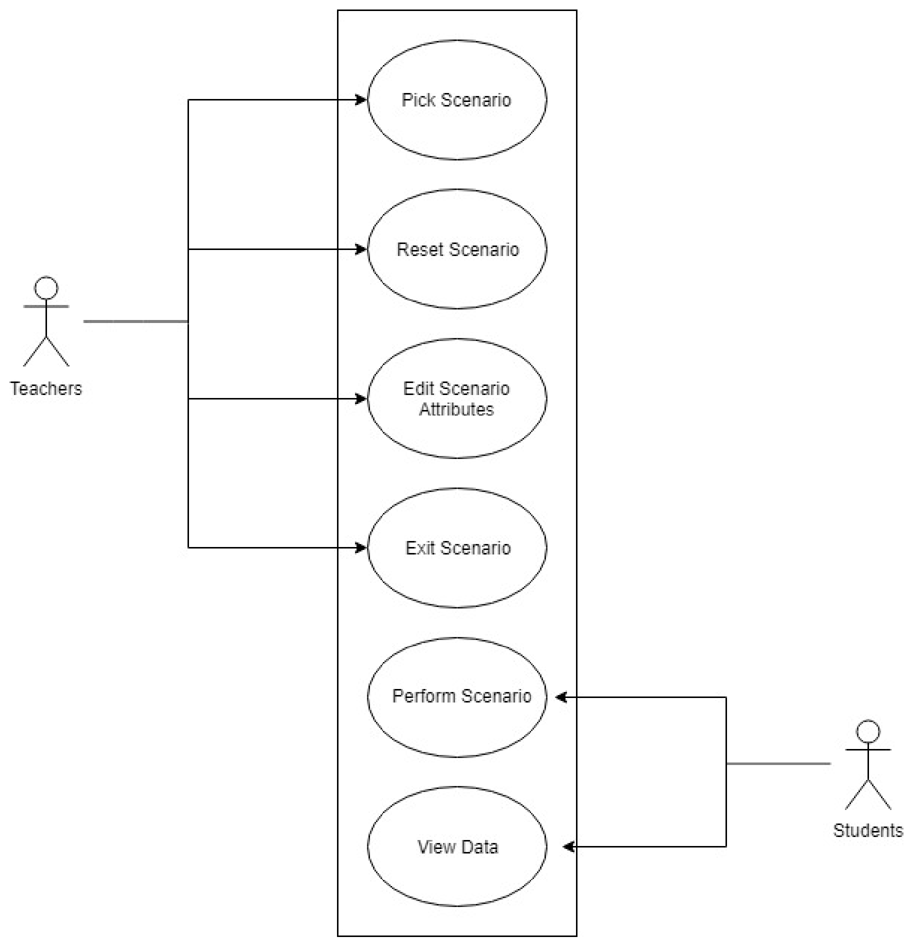

Figure 3.1: Use Case Diagram

3.2 Use Cases

- **Pick Scenario**

  Goal: View a list of all the scenarios and pick one to run

  Actors: Teachers

  Preconditions: Must not be an a scenario

  Postconditions: The scenario will begin

  Exceptions: None

- **Reset Scenario**

  Goal: Reset a scenario to its original state

  Actors: Teachers

  Preconditions: A scenario must be running

  Postconditions: Scenario reset to original state

  Exceptions: None

- **Edit Scenario Attributes**

  Goal: Edit variables in the scenarios to exhibit different behavior

  Actors: Teachers

  Preconditions: A scenario must be chosen

  Postconditions: Scenario will be changed to reflect variable changes

  Exceptions: Variables are invalid

- **Perform Scenario**

  Goal: View a scenario,

  Actors: Students

  Preconditions: A scenario must be chosen

  Postconditions: A scenario will be running

  Exceptions: None

- **View Data**

  Goal: View the data behind a scenario

  Actors: Students

  Preconditions: A scenario is running

Postconditions: Scenario data will be shown

Exceptions: None

- **Exit Scenario**

  Goal: Exit a running scenario

  Actors: Teachers

  Preconditions: Scenario is running

  Postconditions: Scenario is no longer running

  Exceptions: None

Chapter 4

Activity Diagrams

This chapter will outline the flow of activities for teachers and students.

4.1 Teacher

Figure 4.1 describes how a teacher uses the app. They would begin by picking the scenario they want to use for the day's lesson. Then they can set parameters to control certain aspects of the scenario. The teacher explains the scenario and then gives them the headsets. As students finish the scenario the teacher can give a headset to another student, until everyone has finished. To finish the lesson the teacher can talk with the students and close the app.

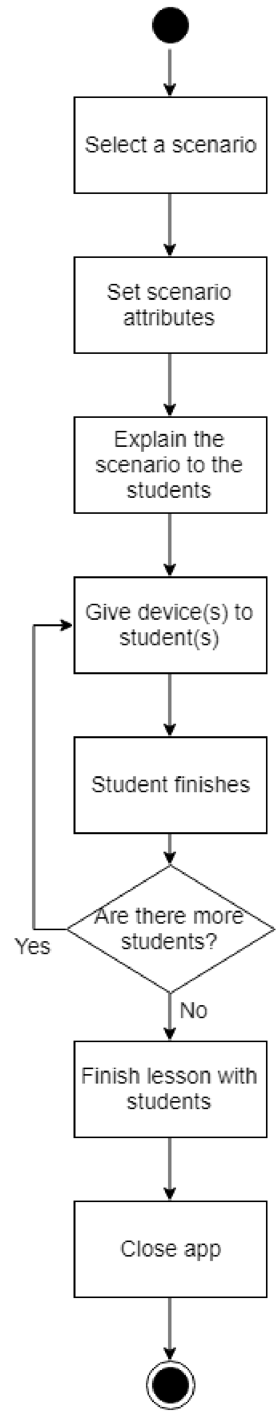

Figure 4.1: Teacher Activity Diagram

4.2 Student

Figure 4.2 describes how a student uses the app. First they listen to the teacher explain what the scenario is. Then they wait until a headset if free for them to start the scenario. They read the instruction screen in the app and then go through the scenario. Once everyone is done, they participate in whatever the teacher whats to do to wrap up the lesson.

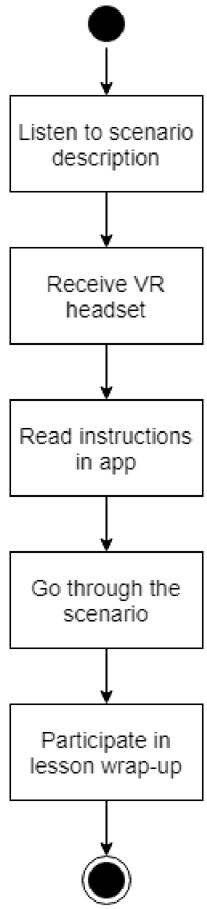

Figure 4.2: Student Activity Diagram

Chapter 5

Conceptual Model

Below you can see the conceptual model of our system, with mockups of the app screen, and sketches of possible virtual reality lessons. Figure 5.1 is a mockup of the home screen, where the teacher selects a scenario and sets its attributes. Figure 5.2 shows the interface used the select a scenario. Figure 5.3 is an example for a type of scenario, where a student throws a ball, and then can look at a 3D graph of the ball's position. Figure 5.4 describes a scenario where a student can move two magnetic objects around and see how that changes the field lines. In the actual application the field would be visible in three dimensions.

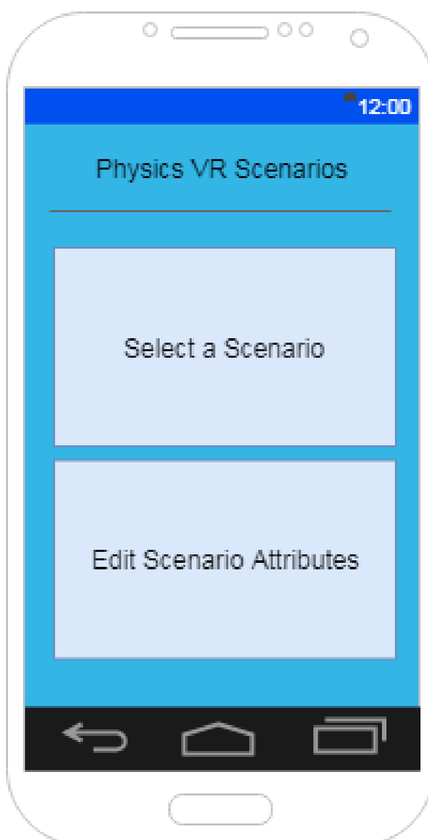

Figure 5.1: Mobile App Home Screen

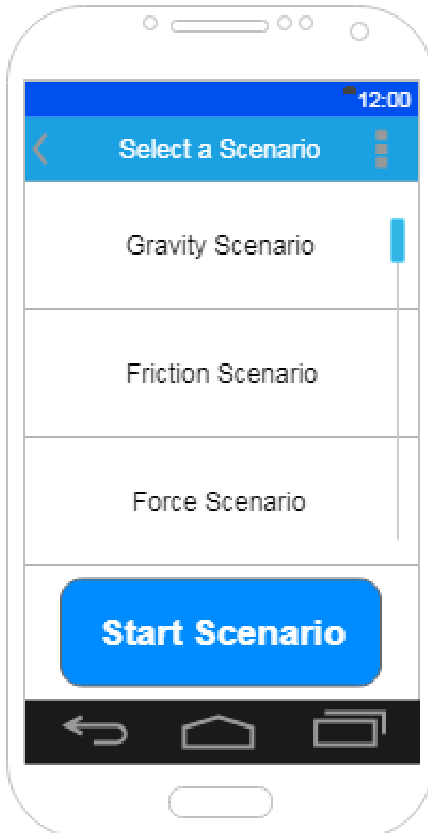

Figure 5.2: Mobile App Scenario Selection

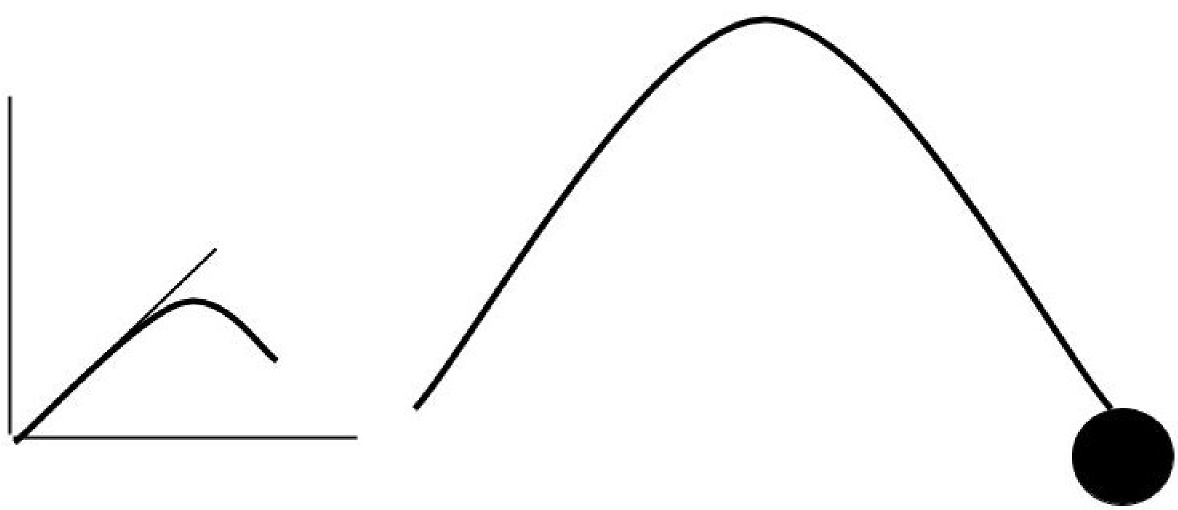

Figure 5.3: Thrown Ball Scenario Example

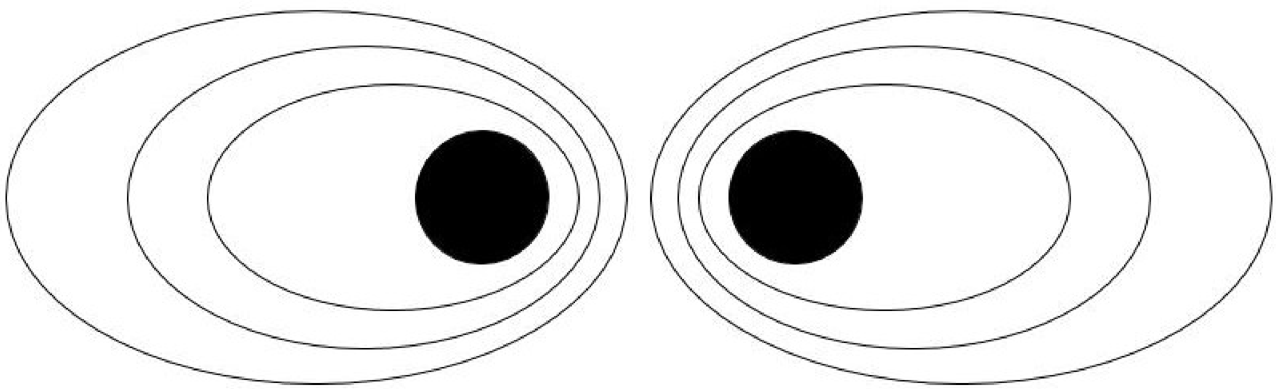

Figure 5.4: Magnetic Field Scenario Example

Chapter 6

Technologies Used

This chapter will outline the technologies we will use to implement the application.

6.1 Samsung Gear VR

The virtual reality headset we are going to use is Samsung Gear VR. It uses an android phone as the screen for the headset and uses the phones sensors to sense the movement of the users head. It also comes with a controller so users can interact with the virtual environment instead of just looking.

6.2 Unity

Unity is a 3D real time engine. We will be using to create our 3D environments and handle the physics simulation. Unity has integration with Samsung Gear through Oculus so it will also handle user input. Scripting in Unity is done with C#.

6.3 Blender

Blender is a 3D graphics toolset. We will be using it to create any 3D models we will need for the scenarios.

6.4 Adobe Photoshop

We will be using Photoshop, an image editing application, to create any 2D assets we need for the interface or textures.

Chapter 7

Architectural Diagram

This chapter describes the system architecture we will use to implement the applcication.

7.1 Model View Controller

Our application will use a Model View Controller architecture, as shown in Figure 7.1. The model contains the scenario information as chosen by the teacher. The box around the view and the controller represents the virtual reality device. The device's view shows the user part of the scenario based on what they are looking at, and the controller handles user input. When the user does certain actions the model is updated to reflect what changed in the environment.

7.2 Diagram

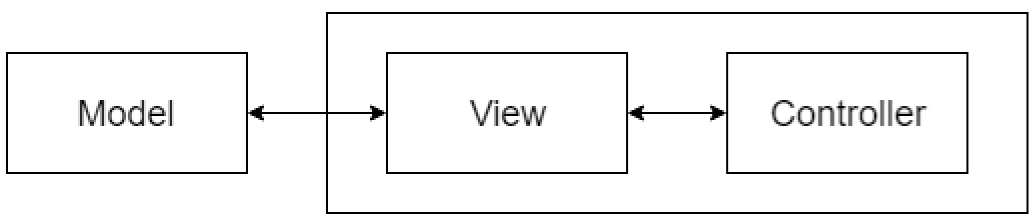

Figure 7.1: MVC Architectural Diagram

Chapter 8

Design Rationale

In this chapter we will design the rationale behind certain design decisions.

8.1 Virtual Reality

We chose to use a virtual environment for our application because we believe being able to see how things work in three dimensions will be helpful for students' learning. For the headset we decided on using a mobile platform instead of a desktop one to minimize the cost. The Samsung Gear VR headset works with Android phones, and unlike other mobile VR solutions it has a controller. The controller allows students to interact with the environment, which makes the application more engaging.

8.2 Development Tools

The main development tool we will be using is the Unity engine. We chose to use this because is has support for Android and Samsung Gear VR, and one of us has used it before. It uses C# as a scripting language, which is easy to learn since we have C/C++ experience. Unity also has a built in physics engine which will make it easy for us to implement scenarios. We are using Blender for 3D models because it is free but still powerful enough to create any kind of model we might need. For 2D images we are using Photoshop because it is a versatile application that we can use to create both interface designs and textures for models.

Chapter 9

Testing

The following sections will detail the different testing phases that our project went through.

9.1 Unit Testing

In order to make sure all of the components worked correctly, we unit tested extensively each time an update was made. This involved testing the headset itself, the Samsung Gear controller, as well as each of the scenarios and the menu screen.

9.2 Integration Testing

After all of the individual parts were finished, we needed to test the integration of our project, such as th functionality of the controller in various scenarios, the reset button for each of the scenarios, as well as the menu screen connecting all of the scenarios together.

9.3 User Testing

We plan on doing user testing in the future, where we will hopefully get middle school teachers and students to test out our software and give us feedback on how it could be improved. We have already seeked out advice from people from the physics department, but getting feedback from the target demographic would be very useful.

Chapter 10

Risk Analysis

Below is what we have selected as the greatest risks for the project.

Table 10.1: Risk Analysis Table

Risk	**Consequences**	**Probability**	**Severity**	**Impact**	**Mitigation**
Time	The project may not have all features finished on time.	0.5	6	3	Set deadlines and prioritize critical requirements.
Bugs	The project might behave differently than expected.	0.99	2	1.98	Perform code reviews and test extensively.
Learning new frameworks and languages	Work on parts of the project may be slower and less optimized.	0.8	2	1.6	Set aside time to familiarize ourselves with the new technology. Also seek out someone with experience.
Headset breaks	Will have to buy a new one.	0.05	8	0.4	Create mouse and keyboard control scheme so development can be done with only a computer.
Data Loss	May have to recover it or rewrite it.	0.01	10	0.1	Back up data, and use Github.

Chapter 11

Development Timeline

On the following timeline you can see the deadlines we have, and the timeframes we have set to meet those deadlines.

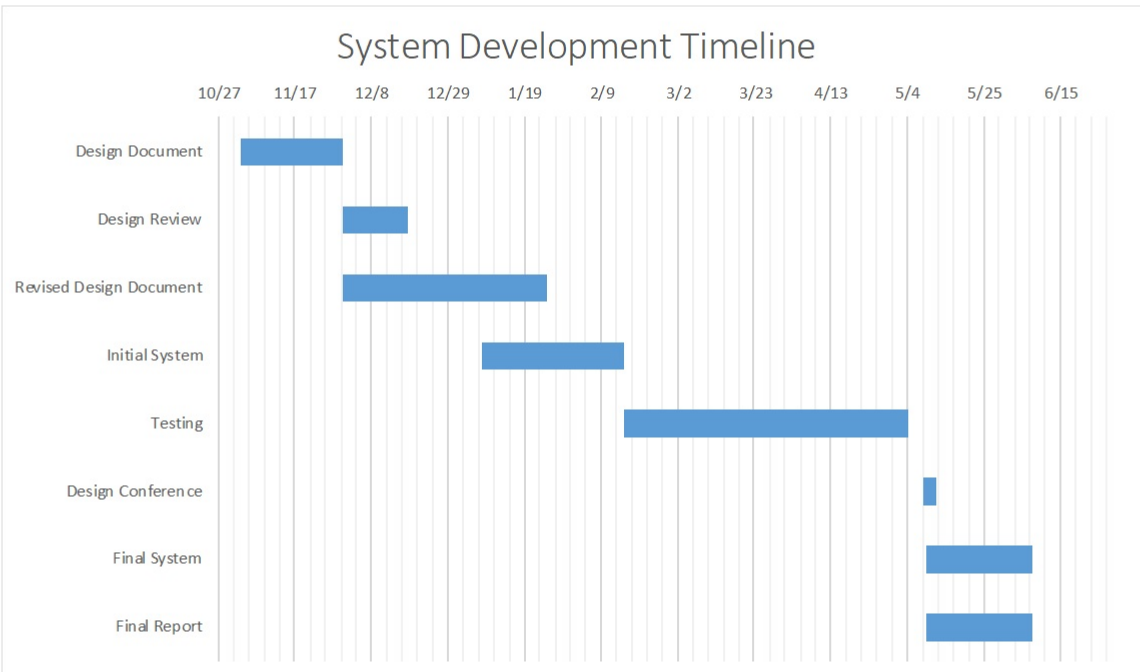

Figure 11.1: Development Timeline

Chapter 12

Societal Issues

All engineering projects deal with several societal issues that are important to consider during development. The following sections contain the societal issues that we deemed most relevant for this project.

12.1 Ethical

As our project is meant to be made as a purely educational tool, it does not deal with many ethical issues itself. However, we created our project so that all of the students using it would have equal opportunity to learn. While this does not apply to our project specifically, virtual reality that creates immersive environments for people to interact with each other have their own set of ethical issues.

12.2 Social

Social factors were a large influence on the design of our project. Many schools, especially public ones, do not have a large budget and are often accommodating students from low-income backgrounds. We chose to create this project using a mobile VR headset that is much more budget-friendly so that the school could afford more headsets so all the students could have an equal chance to learn. Furthermore, this project allows many students that would never be able to experience virtual reality to have their first experience with it.

12.3 Usability

When creating our project we wanted to make sure that the user interface was clear and easy to use for both teachers and students. While it can be a process to load the project onto the mobile device, the actual VR environment itself is very intuitive, with a menu button to choose between scenarios and reset buttons as well.

12.4 Economic

We chose the Samsung Gear VR headset over other headsets such as the Oculus Rift because of the cheaper price tag, allowing for schools to purchase more headsets for students to use. Despite this, virtual reality is still an up and coming technology that is rather expensive for schools to buy. In the future, if more budget-friendly headsets come out, we may port our project to that.

12.5 Health and Safety

Virtual reality in a classroom environment can transform the type of experiments students are allowed to partake in. Performing experiments in virtual reality do not pose any danger to the students at all. This allows students to be able to see more dangerous physics experiments up close without fearing any repercussions.

12.6 Lifelong Learning

This project was a very interesting dive into the up and coming world of virtual reality. For this project, we were able to use concepts we had learned from classes such as 3D modelling, but for the actual development of the code in Unity, it was all learning done on our own. It gave us practice looking through documentation to figure out what we need. Furthermore, we learned a ton about how a software engineering project is developed, with many factors constantly changing the design.

Chapter 13

Conclusion

Our project was a virtual reality application designed to help teach middle school physics students. It was developed in Unity, using the Samsung Gear VR headset. Doing this project, we learned a few important lessons. First, we learned that the design can change quickly and drastically. During both design creation, development, and even testing, we were forced to make changes to our design either to make it easier to develop or improve the product. We learned the virtual reality can be hard to work with, especially for testing. Whenever we wanted to test any changes, we had to go through the process of loading it onto the mobile device before being able to test. Lastly, we learned that sometimes it is easier to write your own solutions rather than relying on libraries. The Unity library for the VR controller was created for the Oculus Rift, and was very lacking for the Samsung Gear, so we coded our own controls for the controller.

We encountered a couple of obstacles while doing our project. We were inexperienced with Unity, and it took us a while to get started after figuring out how it worked. Also, we only had one headset, which made it so that only one of us could test changes at a time.

As for future work to be done, we would like to continue developing VR scenarios, for different physics concepts. We would like to improve the menu functionality, adding an option to change variables in the VR environment, such as temperature or gravity. Lastly, as mentioned in the testing section, we would like to conduct a user study with a middle school class to improve our application even more.

Chapter 14

Appendix A: Installation Guide

This appendix describes how to install the Android app and set up the Samsung Gear VR headset.

14.1 App Installation

1. Download the VR Physics Scenarios APK file.
2. Go into your phone's settings and enable installation from unknown sources.
3. Tap the APK file to open it.
4. Hit "Yes" on the prompt.
5. Wait for the app to be installed.

14.2 Samsung Gear VR Headset Setup

1. Insert the main strap through the large loops on the side on the headset.
2. Adjust the strap to a comfortable length and fasten the velcro.
3. Take the top head strap and insert the hook into the front bar on the top of the headset,
4. Attach the top head strap the main strap and pull the tab at the front to adjust its length.
5. Pull the front cover off.
6. Pull the device holder on the right side of the front face of the headset to the right.
7. Open the VR Physics Scenarios app on your phone.
8. Insert your phone into the USB-C port and push down into the headset. The device holder will snap into place.
9. Put the headset on and the app will load automatically.

Chapter 15

Appendix B: User Manual

This appendix describes how to use the app.

15.1 Controls

The app uses the Samsung Gear VR Headset to look around the environment and the Gear Controller to interact with it. Figure 15.1 describes the controller. The player can move around by touching the touchpad in the direction they want to move in. The controller contains motion sensors, so the play can move the controller around to point at objects in the environment. Pulling the trigger pushes menu buttons and picks up objects. If the position of the controller or the vision of the headset in virtual space is different from the real world the user can hold down the home key to recenter them.

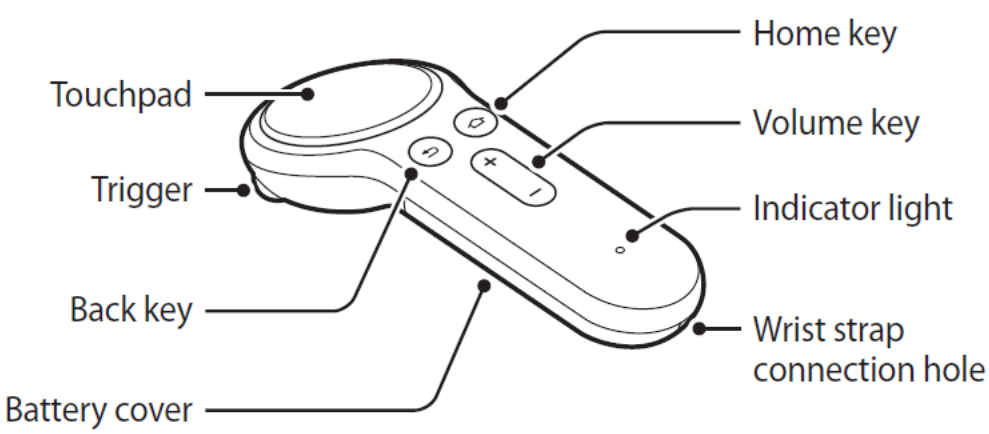

Figure 15.1: Samsung Gear Controller

15.2 Scenario Menu

When the user first loads into the app they see the scenario menu shown in Figure 15.2. There is a blue line drawn coming out from the controller to indicate where it is pointing. To select a scenario the user points at one of the scenario buttons and pulls the trigger on the controller.

Figure 15.2: Scenario Menu

15.3 Interacting with the Scenario

The Ball Throw scenario will be used as an example of how to interact with a scenario. Figure 15.3 shows the ball used in the scenario, with the user pointing the controller at the ball. The user can pull the trigger to pick up the ball, as shown in Figure 15.4. Once the ball is picked up the user can throw it by making a throwing motion and letting go of the trigger. When the ball is thrown a line is drawn showing its trajectory as shown in Figure 15.5. The line will remain until the ball is thrown again, at which point the old line will be erased and a new one will be drawn. The scenario also contains a graph where the balls position is plotted, similar to the trajectory line, as shown in Figure 15.6. There is also a data panel, shown in Figure 15.7 that has different measurements about the last ball throw. In this scenario the data panel contains an exercise. The user is given the initial height, horizontal velocity, and vertical velocity of the ball, and the user would have to calculate the distance the ball was thrown.

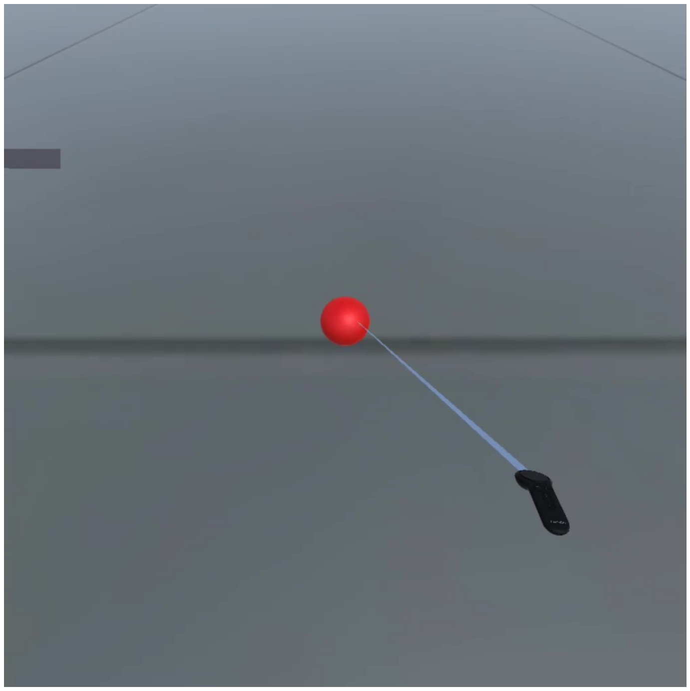

Figure 15.3: Ball Throw Scenario

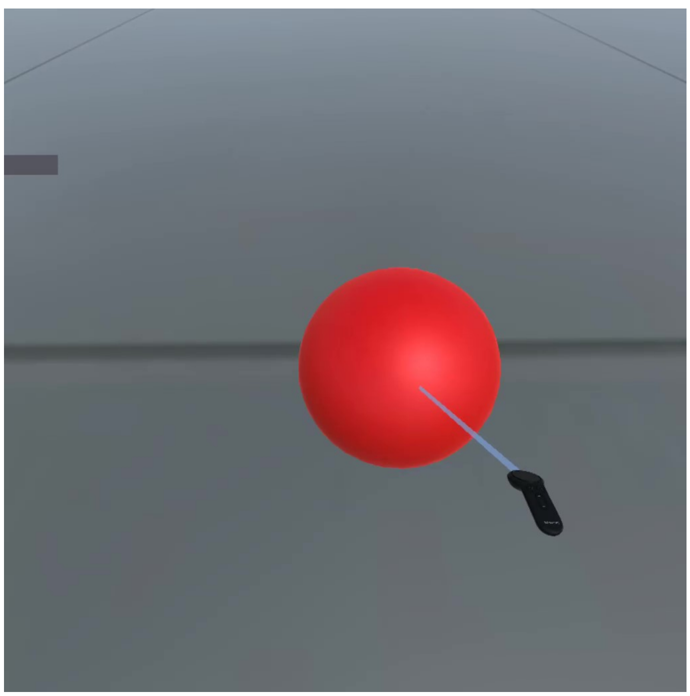

Figure 15.4: Picked up ball

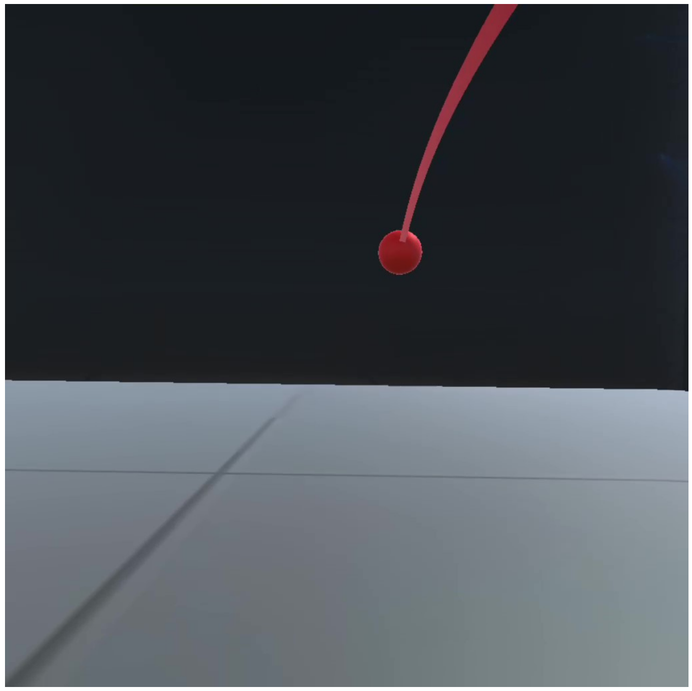

Figure 15.5: Thrown Ball

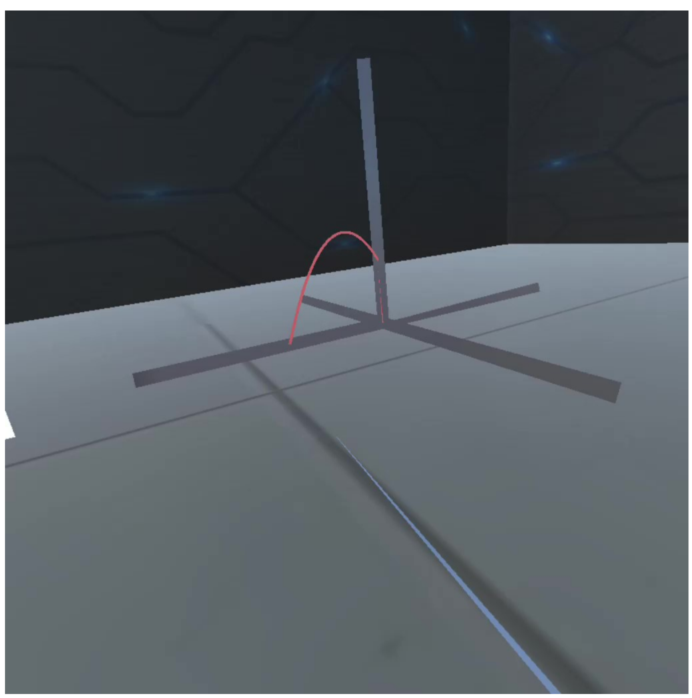

Figure 15.6: Graph

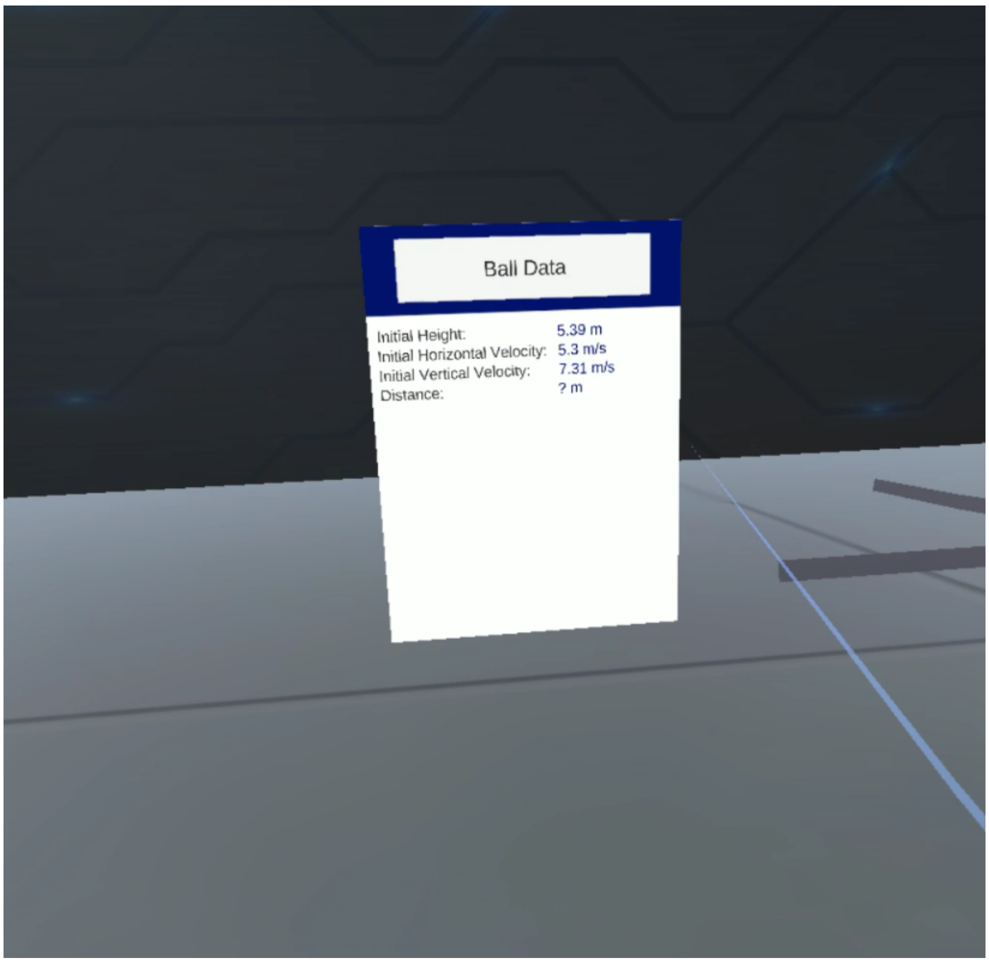

Figure 15.7: Data Panel

Bibliography

[1] Unity Technologies. *Unity User Manual.*

https://docs.unity3d.com/Manual/index.html

[2] Unity Technologies. *Unity Engine.*

https://unity.com/

[3] Samsung. *Samsung Gear VR.*

https://www.samsung.com/global/galaxy/gear-vr/