A Virtual Reality based Electrical Laboratory Simulation with enhanced visualizations for learning in a safe and controlled environment.

A Course Project Report Submitted
in Partial Fulfillment of the Requirements
for the Course of

CS560: Virtual and Augmented Reality Systems

by

GROUP 1
Aadi Aarya Chandra,
Akshat Mittal,
Anindya Vijayvargeeya,
Gunjan Dhanuka,
Pranjal Singh
under the guidance of

Dr. Samit Bhattacharya

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI,
GUWAHATI - 781039, ASSAM

Acknowledgements

We take this opportunity to express our regards and gratitude to our supervisor **Dr. Samit Bhattacharya** for his exceptional guidance and encouragement throughout this project. His extraordinary knowledge and unwavering enthusiasm kept the project enjoyable and their generosity helped us immensely. The completion of this project would not have been possible without his support.

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Project Motivation . . . 1
- 1.3 Organization of The Report . . . 2

2 Related Works **3**
- 2.1 Non-VR simulations . . . 3
  - 2.1.1 Circuit-lab[1] . . . 3
  - 2.1.2 Autodesk[2] . . . 3
  - 2.1.3 DC/AC Virtual Lab[3] . . . 3
  - 2.1.4 PartSim[4] . . . 3
  - 2.1.5 DoCircuits . . . 4
- 2.2 VR simulations . . . 4
  - 2.2.1 The VR Chemistry Lab[5] . . . 4
  - 2.2.2 Onlabs[6] . . . 4
  - 2.2.3 METaL[7] . . . 5

3 Our Work **6**
- 3.1 Problem formulation . . . 6
- 3.2 Specifications . . . 6
  - 3.2.1 Components . . . 6
  - 3.2.2 Features . . . 6
  - 3.2.3 Environment . . . 7
- 3.3 Implementation . . . 7
  - 3.3.1 3D Modelling . . . 8
  - 3.3.2 Animation and Scripting . . . 8
  - 3.3.3 SpiceSharp and Circuit Simulation . . . 9
  - 3.3.4 Electron Flow and Visualization . . . 10

4 Evaluation and Testing **11**
- 4.1 Testcases . . . 11

5 Future Work **13**

References **15**

List of Figures

1.1 Immersive Virtual Reality Flow Chart[8] 2

2.1 An experiment in The VR Chemistry Lab 4
2.2 A screenshot of Onlabs Virtual Reality biology laborator 5
2.3 Multimodal Experience Testbed and Laboratory 5

3.1 A in-application view of the virtual laboratory. 7
3.2 Code snippet that sets the bulb intensity according to current flow. 8
3.3 Screenshot of a user building a circuit in our application. 9
3.4 Screenshot of a running circuit in our application. 10

4.1 Informative charts in the virtual laboratory environment. 11
4.2 User can pick up objects from both of his hands, and attach them on the board. 12
4.3 Kirchoff's Current Law is verified in our software. *Kirchhoff's Current Law states that the total current entering a junction or a node equals the charge leaving the node as no charge is lost.* 12
4.4 The principle of Wheatstone Bridge works as expected in our software. *The Wheatstone bridge works on the principle of null deflection, i.e., the ratio of their resistances is equal, and no current flows through the circuit.* 12

Chapter 1

Introduction

1.1 Background

In the past few years, there has been a significant surge in the research and applications of Virtual Reality (VR) software. The term Virtual Reality was first proposed in 1989, describing computer simulation technology. However, at present, VR is used in a plethora of domains, including research, education, defense, entertainment, medicine, architecture, and training. VR can be defined as *an artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment.* VR can be of two types: non-immersive and immersive. The former refers to a computer-based environment that can simulate places in real or imagined worlds, while the latter gives the perception of being physically present in the non-physical world. A number of advanced devices like Meta Quest and Rift provide such immersive experiences by not only a surrounding display but also by tracking the movements, gestures, and gaze of the user in different ways. One of the main advantages of immersive VR is that the environment is a full scale replica of the real world counterpart and it relates to the human scale. Hence, the users get a feeling of interacting with the real environment. Using VR, we can provide a system for participants to enter and interact with a world that can be difficult to access due to costs or safety reasons.

1.2 Project Motivation

As engineering students, we have all had experience building and studying electrical circuits. It is a great way to complement the theoretical knowledge students gain in classes and through textbooks. However, getting access to such an electrical lab is not easy at all times due to a multitude of factors – limited labs, logistic problems in carrying the components, costs associated with various components, and risk of short circuits for new practitioners. Also, traveling to such labs is a pain point if a student quickly wants to try out an experiment at his current location. Keeping all these points in mind, we decided

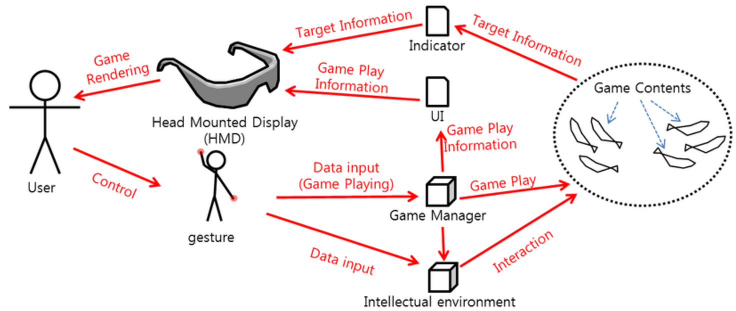

Fig. 1.1 Immersive Virtual Reality Flow Chart[8]

to develop a VR Electrical Laboratory Simulation System for learning and demonstration purposes in a safe and controlled environment. Our idea stems from the fact that it can significantly boost the accessibility of Electrical Labs, which are very crucial to develop an intuition and bolstering learning in students from high school up to undergraduates.

1.3 Organization of The Report

The rest of the report is organized as follows:

In Chapter 2, we discuss the prior work done in this field. We discuss about the non-VR electrical and electronic simulations software available to public and a brief discussion about each one of them. We also throw light on the VR lab simulations from other domains like Chemistry and Biology that gave us an inspiration to develop this project.

In Chapter 3, we first explain the system developed by us, its various features, components and scope.

In Chapter 4, we discuss upon the evaluation and feedback of the developed system.

In Chapter 5, we give a conclusion about our work.

Chapter 2

Related Works

2.1 Non-VR simulations

2.1.1 Circuit-lab[1]

Circuit-lab provides user-friendly tools leveraging the power of web-based distribution to make everything available everywhere, instantly. It allows users to iterate and collaborate: simulate, experiment, adjust, and share the results in just a few clicks, taking the friction out of design and analysis.

2.1.2 Autodesk[2]

Circuits.io is an online platform created by Autodesk for hardware hackers. It provides a browser-based application for designing, simulating electronic circuits, and creating PCB boards. The Autodesk circuits simulator can simulate Arduino-based projects to test designs and programs before creating them in real life. The simulator allows users to learn electronics using a virtual Arduino board and breadboard. It is free to use, but more features are available with premium accounts.

2.1.3 DC/AC Virtual Lab[3]

DC/AC Virtual Lab is an easy to learn, online simulator which is capable of building DC/AC circuits, users can build circuits with batteries, resistors, wires and other components. DC/AC Virtual Lab has pretty attractive graphics and components are real looking, with certain limitations in parts library, incapability of drawing circuits and some other reasons.

2.1.4 PartSim[4]

PartSim is a free and full-featured web-browser based SPICE circuit simulator. In addition to the online simulator,it contains a comprehensive circuit simulation website provided with tutorials, SPICE references, and a circuit repository.PartSim schematic editor runs as a native application in browser using javascript and doesn't require the use of any plugins.

PartSim uses a cloud-based version of ngspice, which is an open source simulator based on Berkely's Spice3f5 Spice engine.

2.1.5 DoCircuits

DoCircuits is a cloud based circuit simulation tool where users can build, run and analyze their circuit designs. It can be used by students to learn, create and simulate lab experiments online and share them. It can be used by practicing engineers to design analog and digital circuits and save their creations privately on the cloud. It is being used by students and teachers in 300+ colleges in over 150 countries.

2.2 VR simulations

2.2.1 The VR Chemistry Lab[5]

The VR Chemistry Lab is a virtual reality simulation of a school chemistry lab as well as a molecule modelling experience.Users can do an experiment in a chemistry lab or experience life-size atoms and molecules..It aims to be a supplementary tool to real, physical, hands-on science experiences that students usually have in school. In the case where students cannot have these traditional physical experiences, the VR Chemistry Lab can provide a comparable learning experience. It also can provide a unique experience where students can enter the experiment at the molecular level and interact with the molecules as the experiment is happening. It is designed to be a bridge between the real-life chemistry experience and a model of the abstract concepts that are usually difficult to visualize.

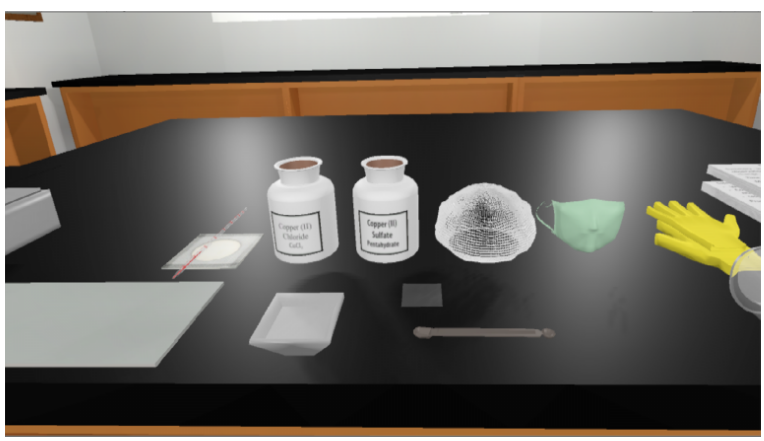

Fig. 2.1 An experiment in The VR Chemistry Lab

2.2.2 Onlabs[6]

Onlabs is a virtual reality biology lab developed by an interdisciplinary team in the School of Science and Technology at the Hellenic Open University, provided to the distance learning students, as a supplement to their printed learning material, so as to be trained on their basic biology lab experiments. students can practice on operating basic laboratory instruments before appearing in the physical biology lab, acquiring this way skills and confidence needed for performing successfully the real experiments.

Fig. 2.2 A screenshot of Onlabs Virtual Reality biology laborator

2.2.3 METaL[7]

The METaL (Multimodal Experience Testbed and Laboratory) lab is a human-scale immersive environment for development of natural interaction techniques in virtual reality. Key to the lab is an integrated haptic force-feedback device from Haption. This facility supports rapid development iteration and is comparable to virtual reality systems found in industry. Integration of haptic force-feedback with visual display, audio effects, and physical simulation permits the creation of multimodal experiences.

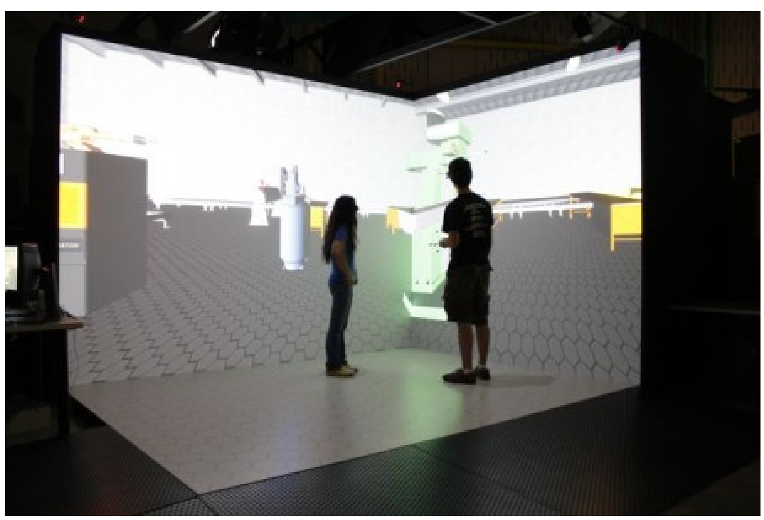

Fig. 2.3 Multimodal Experience Testbed and Laboratory

Chapter 3

Our Work

3.1 Problem formulation

The project aims to develop an immersive VR simulation of an Electrical Lab where the user can construct basic electric circuits using components like wire, switch, bulbs and resistors. We will be using accurate simulation using the SpiceSharp open-source circuit simulation library. The system should also be interactive and provide cues to the user using labels, tooltips, and animations wherever possible to enhance the learning experience. The system should support conducting standard experiments like Ohm's Law and Kirchoff's Laws and arbitrarily complex electric circuits depending on the user's requirements. To facilitate this, current, resistance, and voltage values should be displayed on each component.

3.2 Specifications

3.2.1 Components

- Breadboard, on which all circuit elements are placed.
- Battery.
- Switch.
- Bulbs with different resistances.
- Wire.

3.2.2 Features

- Unlimited supply of circuit components.
- Components snap to the grid with auditory feedback when dropped for easy circuit creation.
- Short circuit detection with visual feedback, indicating the exact components involved.
- Speed of electron flow indicate the amount of current flowing in the circuit.

- Interactive components - switches open/close when pinched.
- Label lever activates/deactivates current, resistance, and voltage drop labels on active circuits.
- Reset lever sends all components back to their dispensers for easy cleanup.
- Table height can be easily adjusted by grabbing the front bar for seated or standing play.
- Teleport locomotion with controller button as well as smooth locomotion and snap-turning with the thumbstick.

3.2.3 Environment

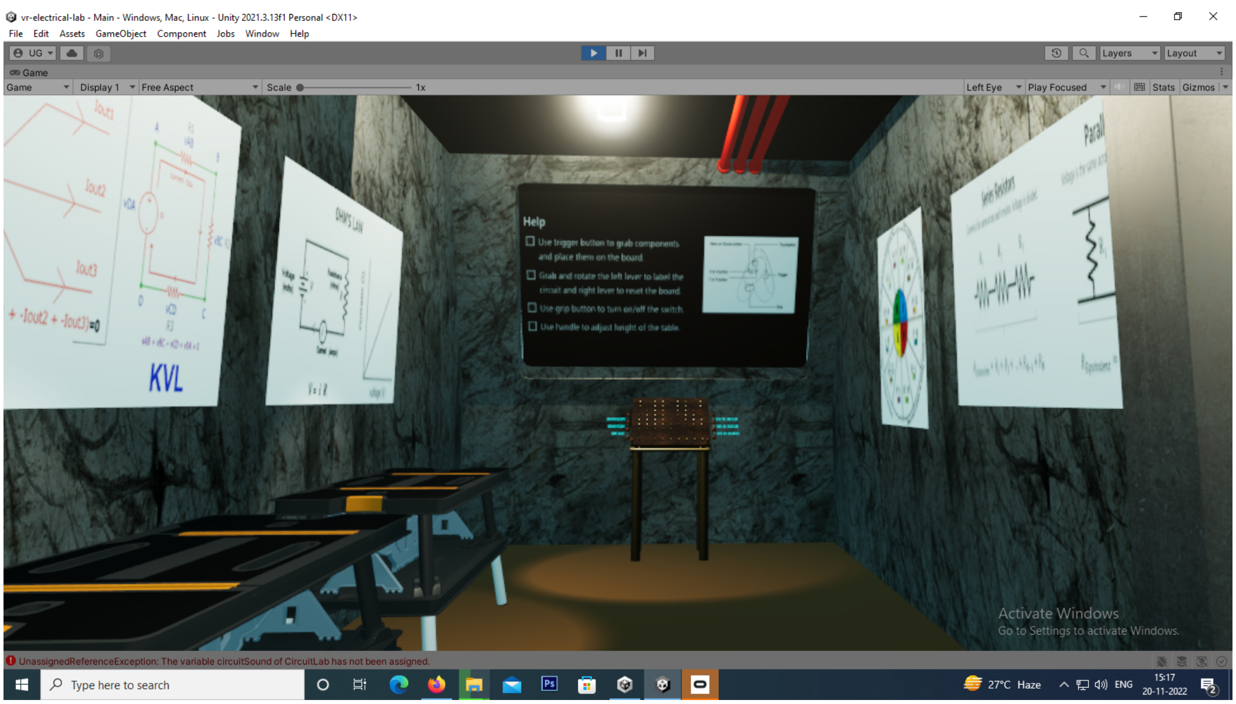

Fig. 3.1 A in-application view of the virtual laboratory.

We have created a virtual lab environment with electric circuit-related charts on the walls to create a good learning environment.

3.3 Implementation

First we created the various 3D models of lab environment, breadboard table and circuit components. Then we worked on scripts to implement basic functionality like adding components on breadboard, turning the switch on and off and its corresponding animation, etc. Each component has corresponding scripts attached for corresponding functioning and animations. The we added `CircuitLab` script that contains the core logic of circuit

simulation. Lastly to better visualize current flow, we added electron flow animation in wires.
Tech Stack used in this project:

- **Unity** for the overall project – including interaction, animation and scripting.
- **Blender** for making the various models.
- **SpiceSharp Library** to simulate the electrical circuit.

3.3.1 3D Modelling

For our project, we used an open-source 3D modeling software called **Blender** for making 3D assets like breadboard tables, resistors, switches, etc. For modelling, we used various tools provided by Blender, like proportional editing, modifiers, etc and used various modifiers like *Subdivision Surface Modifier*, *Mirror Modifier*, and *Array Modifiers* to model our assets. The *UV editor* in Blender was used for UV unwrapping the model. We found the image textures, normal, roughness and displacement maps in the unity asset store and used them to texture our model. Finally, we baked the textures and exported the assets to unity.

3.3.2 Animation and Scripting

Unlimited Components

To enable the user to use multiple instances of the same component, we implemented the "Unlimited Components" feature. This is done by using the `onTriggerExit` function, where whenever a component is grabbed, a clone gets spawned in the hands of the user.

Reset Lever

We initially faced a problem that whenever we had to build from scratch, we needed to exit the application. So to overcome that, we implemented the *Reset function* so that all the components come back to their original positions and the circuit is reset.

Bulbs Glow Intensity

To add to the realism of the simulation, we have tuned the intensity parameter in the bulbs to correspond to the current flowing through them using the formula:

```
// Calculate light intensity based on current
float maxCurrent = 0.01f;
float maxIntensity = 5.0f;
float minIntensity = 3.0f;
float pctCurrent = ((float)current > maxCurrent ? maxCurrent : (float)current) / maxCurrent;
intensity = (pctCurrent * (maxIntensity - minIntensity)) + minIntensity;
```

Fig. 3.2 Code snippet that sets the bulb intensity according to current flow.

Labels Lever

To enable the user to verify the various laws of electricity, we display the voltage, resistance, and current values on top of each component. To measure the current flowing, we attach an *invisible* 0V battery with each component so that it can act as an Ammeter. When the Label Lever is triggered, the labels get toggled.

3.3.3 SpiceSharp and Circuit Simulation

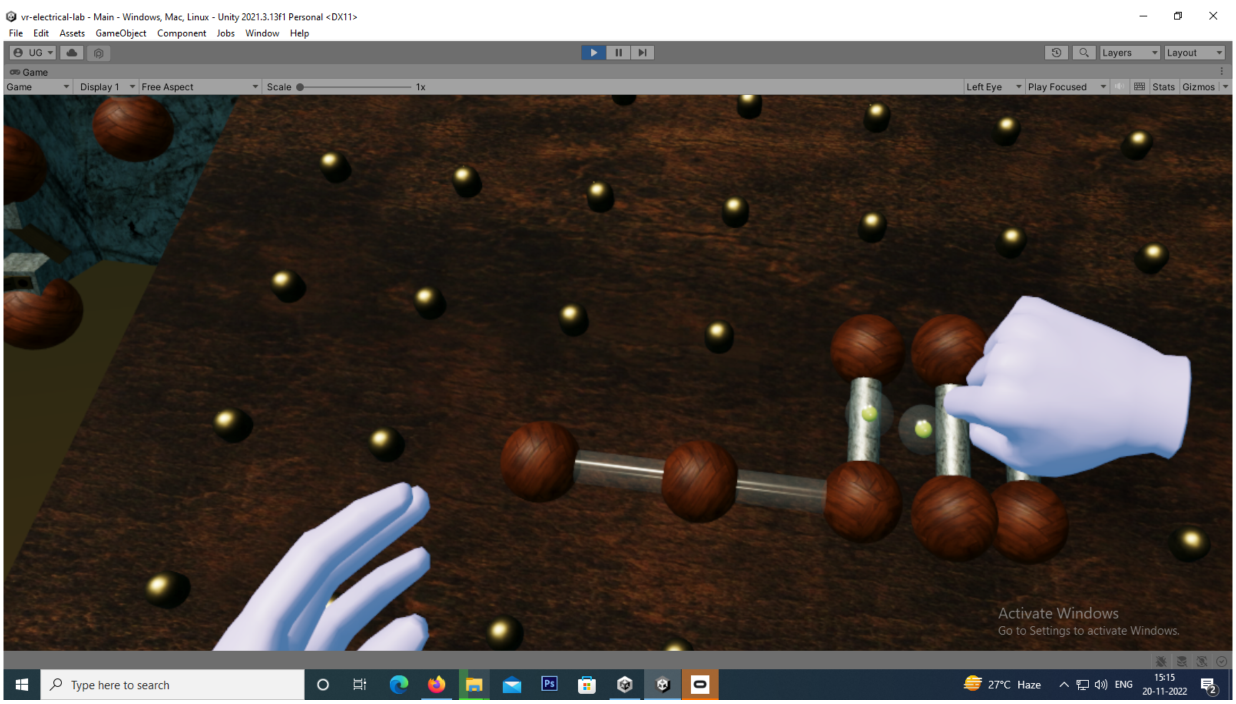

Fig. 3.3 Screenshot of a user building a circuit in our application.

1. Our implementation can support multiple circuits running on the breadboard simultaneously. To keep track of the circuits and prevent them from interfering with each other, we assign a Generation Number to each circuit.

2. To detect circuits, we use batteries as starting points and employ a Depth-First Search starting at the positive terminal of each battery on the board to find each circuit that eventually terminates at the negative terminal of the battery.

3. In the Depth-First Search, we dive into the individual circuits using `FindCircuit` and iterate component-wise. We check if there is a short circuit by tracking the number of resistors. If we encounter the negative terminal of the battery from where we started the search, we consider the circuit as complete. We activate all the components in this circuit and decide the current flow direction. If we encounter a previously added component again, there is a loop that doesn't include the battery from where we started. So we must bail out and finish the search. If the circuit is not yet complete, we recursively call the same function `FindCircuit` again to go to the next component.

4. The function `AddSpiceSharpEntity` adds a new component to the existing circuit building in the SpiceSharp space. We attach a 0 voltage source next to every component we add to it to act as an ammeter. If we add a new battery, we ground one terminal of the battery.

5. Once we have the components for a circuit ready, we go back to the `SimulateCircuit` function, we then create an operating point analysis for the circuit. We loop through the components and find the lowest voltage, to normalize the entire circuit to start at 0V. Then we update the voltage values corresponding to the new 0V component.

6. Then we run the simulation from the Operating Point and also play the necessary sounds to complement the simulation. To handle any possible simulation errors that might arise, we have used `try-catch` statements that will log out the error and deactivate all the components in the circuit.

3.3.4 Electron Flow and Visualization

To enhance the visualization and learning experience, we have implemented the movement of electrons in the wire, which also has a speed-current mapping, i.e., for greater current intensity, the electrons will move faster. In each wire, we have two child electron objects. When the wire is active (current is flowing through it), the `Update()` function is called. We set the `step` value equal to the speed $*$ Time.deltaTime, and for each Electron in the wire, we shift it in the direction of the current by a distance `step`. If the electron moves outside the boundary of the wire, then we wrap it around to the beginning to make it look like a continuous animation. The speed of the electron's movement is updated as:

$$\text{speed} = \text{normalCircuitSpeed} \times \frac{\text{current}}{\text{baseCurrent}}$$

There are different colored bulbs in our implementation that indicate different resistances visually of these bulbs.

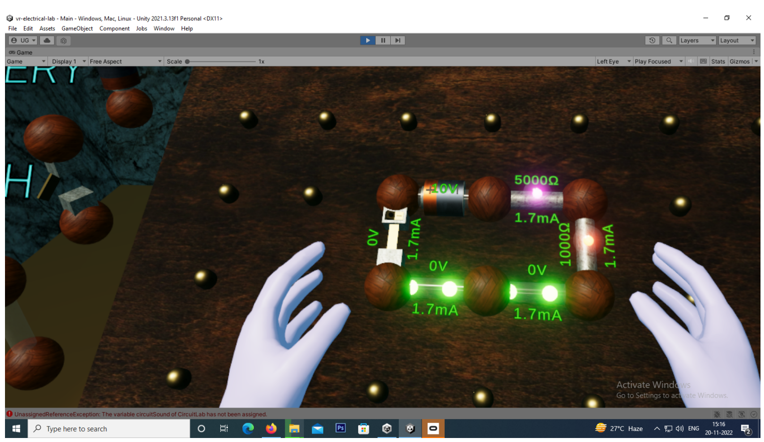

Fig. 3.4 Screenshot of a running circuit in our application.

Chapter 4

Evaluation and Testing

Software Testing is the evaluation of the software against requirements gathered from users and system specifications. We have tried our best to evaluate our software against the list of specifications and what a user may expect when using the application. We showcase our software as a “Virtual Electrical Lab for Learning in a Safe Environment”, and hence we need to ensure the learning component to it. We have added various charts to educate the user about the different concepts like *Ohm’s Law, Kirchoff’s Current Law, and Kirchoff’s Voltage Law*. We have also tested our application on various testcases that a user might expect from the application.

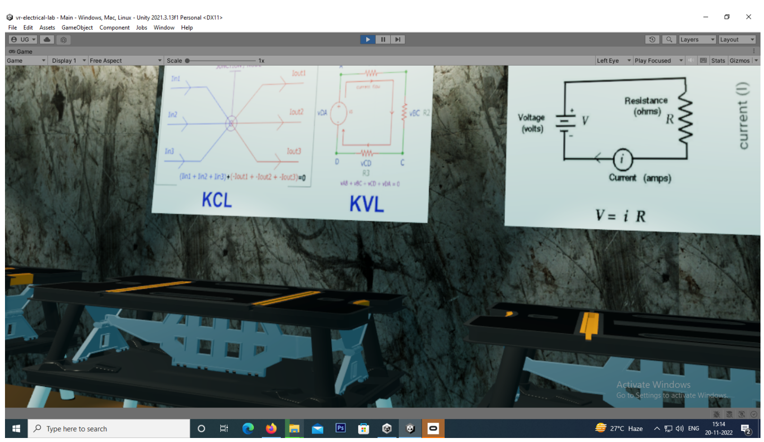

Fig. 4.1 Informative charts in the virtual laboratory environment.

4.1 Testcases

We have listed a few test cases and attached the corresponding screenshots to verify that our application works as intended from a user’s perspective:

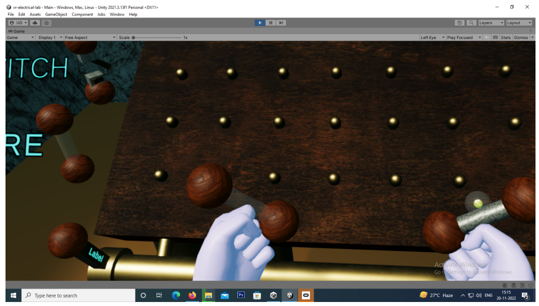

Fig. 4.2 User can pick up objects from both of his hands, and attach them on the board.

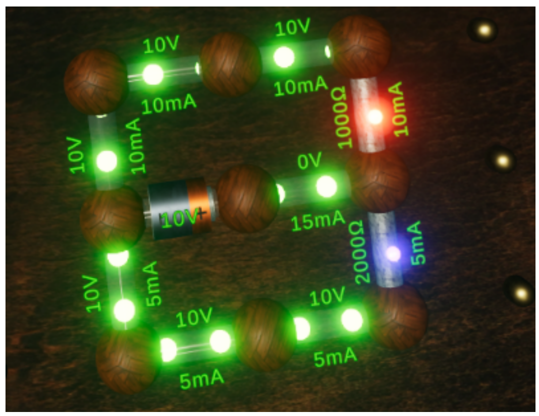

Fig. 4.3 Kirchoff's Current Law is verified in our software. *Kirchhoff's Current Law states that the total current entering a junction or a node equals the charge leaving the node as no charge is lost.*

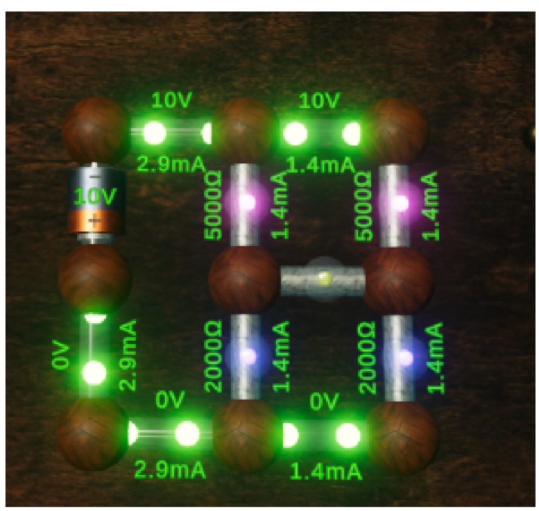

Fig. 4.4 The principle of Wheatstone Bridge works as expected in our software. *The Wheatstone bridge works on the principle of null deflection, i.e., the ratio of their resistances is equal, and no current flows through the circuit.*

Chapter 5

Future Work

There are a number of features that we see a prospect of adding to our project in the future. Some of them are:

1. Support for various components like capacitors, inductors, variable resistances, etc. could be added. This will allow users to simulate more complicated circuits. Currently there is support for only linear-resistances and other basic components.

2. Make the room look more like a laboratory by adding more types of equipment that might need to be modeled separately (function generators, oscilloscope, etc.).

3. Enable saving and loading circuits and multi-user support so that the user can resume building or share their creations. This can help tremendously in an educational setting.

4. Exam mode can be added to allow exams and evaluations to be conducted

5. We can make the application more user-friendly by showing a walkthrough when the user starts the software and tooltips over each component.

References

[1] "Circuit-lab: Circuit simulation and schematics.." https://www.circuitlab.com/.

[2] "Autodesk circuits.io.." https://library.io/.

[3] "Dc/ac virtual lab – dcaclab.com." https://dcaclab.com/.

[4] "Partsim -circuit simulation made easy." https://www.partsim.com/.

[5] "The vr chemistry lab." https://www.oculus.com/experiences/quest/3919613214752680/.

[6] "Onlabs: A virtual lab at your hands!." http://snf-858823.vm.okeanos.grnet.gr/onlabs/.

[7] "Metal (multimodal experience testbed and laboratory)." https://www.vrac.iastate.edu/facilities/metal/.

[8] J. Kim and W. Hyung, "Design and modelling immersive game contents system for virtual reality technology," pp. 226–232, 04 2014.