Engineering Education with Extended Reality (XR): Final Comprehensive Report

Joren Cruz, Diego Tapia, Daniel Wang

Activity Report

Abstract—This VR tool incorporates interactive simulations in attempt to facilitate a hands-on learning experience that enhances comprehension and retention of intricate Electrical and Computer Engineering principles. Users will be able to manipulate components, assemble circuits, and observe the behavior of electrical systems in real-time, offering a practical understanding that goes beyond textbook learning. Furthermore, the tool is to include features like guided tutorials, problem-solving exercises, and immediate feedback mechanisms to support and challenge students at various learning stages.

Index Terms—Virtual Reality, VR Learning, Electrical Engineering, Immersive Learning Environment, Interactive Simulations, Unreal Engine 5, PySpice, Socket Programming.

1 THE PROBLEM

ENGINEERS are important and necessary workers, who with their high intellect and strong technical abilities, create significant difference for the masses. One significant challenge in this context is the insufficient number of students who contemplate engineering as a viable career path, and even fewer who persist in pursuing it over the long term. The aim of this project is to provide an accessible and enjoyable tool, designed to captivate the interest of potential students.

2 INSPIRATION

VR immersive learning technology is already being used in various fields, especially in the Medical and Aeronautical fields [1]. From nurses and paramedics to aircraft maintenance engineers, VR provides a safe and isolated environment to practice and nail down simulated protocols that may be too expensive to run in real life. With XR we can provide the same concept. Rather than replacing the classroom or a laboratory session, we have a tool that helps students learn at their own pace.

3 DESIGN CONSTRAINTS

- **Time Costs:** A fixed timeline, such as a semester or academic year, may limit the depth of development and testing that can be achieved
- **Technology Limitations:** Availability and compatibility of VR hardware and software may impose constraints on the project's design and functionality.
- **User Access and Acceptance:** Ensuring that students have access to VR equipment (headsets, computers) can be challenging, especially if the hardware is not readily available to all students. The project also heavily relies on user adoption. Resistance to using VR technology or unfamiliarity with the medium can be a challenge.
- **Educational Integration:** Aligning the VR lab with the existing ECE curriculum and gaining buy-in from instructors.

- *Joren Cruz*
  E-mail: jcruz9@albany.edu,
- *Diego Tapia*
  E-mail: dtapia@albany.edu,
- *Daniel Wang*
  E-mail: dwang9@albany.edu,
  University at Albany, State University of New York.

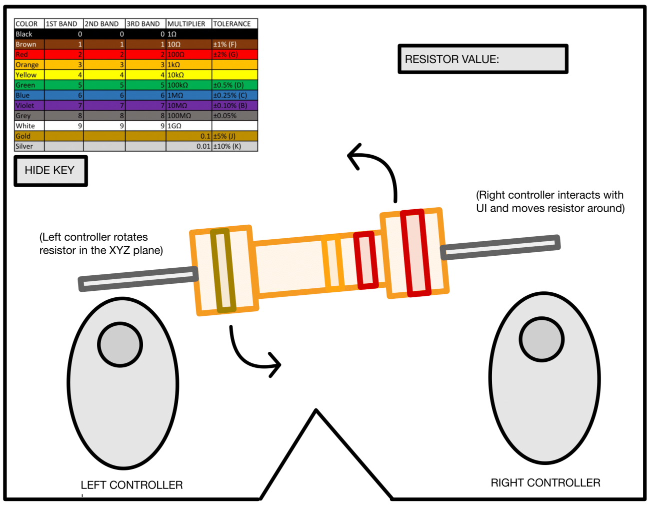

Figure 1. Example of what everyone would see vs. what the user would see.

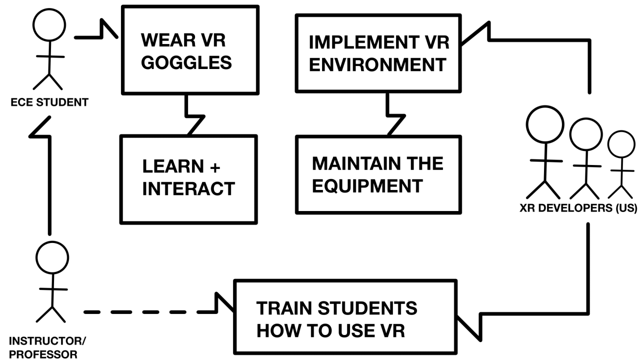

Figure 2. Use case diagram with students, instructors, and developers labeled

4 Proposed Solution

Our current solution for the presented issue is the development of an XR program. Implementation of said program for wearable technologies is implied.

4.1 Engineering Diagram

A visual example of our first prototype is included in Figure 1.

4.2 System Users

This VR lab is designed for new or introductory ECE students, the primary objective is to offer an immersive and accessible learning experience. Upon entering the virtual environment, students can explore and learn about the resistance values of resistors through interactive tutorials, zooming in on components, and practicing decoding resistance values based on color bands. Immediate feedback and progress tracking are integrated into the system to enhance the learning process. As shown in Figure 2, we as the developers are encouraged to instruct students and instructors on the proper utilization of the equipment and ensure its optimal functionality throughout the process.

4.3 Functional Requirements

1) **Headset and Tracking:** High-quality headset for immersive 3D visuals and accurate tracking of user's heads and hands.
2) **Working XR Environment:** Functional developer environment to test experimental code.
3) **Documentation and Tutorials:** Provide comprehensive documentation and interactive tutorials to help users get started and troubleshoot issues.
4) **User Interface (UI) and Interaction:** User-friendly menus and interfaces within the VR environment. Options for interacting with objects and menus in VR, such as gaze-based selection or hand gestures.

4.4 Non-Functional Requirements

1) **Performance:** The VR lab should provide a smooth and responsive user experience, with minimal latency or lag during interactions, to ensure that learning activities are not hindered by technical issues.
2) **Accessibility:** The VR lab should be designed to be accessible to all types of users to ensure equitable access for all students.
3) **Feedback and Metrics:** Collect user feedback and performance metrics to continu-

Factors		Alternative 1		Alternative 2		Alternative 3		
Criteria	Weight	QR Code	Weighted	Virtual Learning	Weighted	D-to-P Integration	Weighted	
Usability	8	4	3.2	8	6.4	6	4.8	
Cost	1	8	6.4	5	4	7	5.6	
Accessibility	3	10	8	3	2.4	3	2.4	
Time consumption	3	8	6.4	6	4.8	3	2.4	
User interactivity	6	3	2.4	9	7.2	8	6.4	
Educational Impact	10	5	4	9	7.2	9	7.2	
	31	38		40		36		RAW SUM
			30.4		32		28.8	WEIGHTED SUM

Figure 3. Decision Matrix

ously improve the VR experience. Implement analytics to monitor user behavior and performance metrics.

4.5 Justification

The development of a Virtual Reality lab for ECE students is well-justified by its capacity to significantly enhance the learning experience, bridging the gap between theory and practice through immersive, hands-on simulations. This technology not only offers accessibility and flexibility but also fosters engagement, motivation, and improved retention rates among students, preparing them for what to expect in this field. In our decision matrix, as illustrated in Figure 3, we evaluated three alternatives based on criteria such as cost, educational impact, usability, and more. The VR lab for calculating resistance values happens to be the most favorable option, with a weighted score reflecting its balanced performance across these criteria. Hence we highly recommend proceeding with this alternative, as it aligns with our project's goals and priorities.

5 Initial Prototype

5.1 Semester Goals

Our semester objective is to achieve a functional alpha version of the project, enabling us to gather user feedback and assess their experience. This will allow us to gauge user responses and determine the potential directions for the final product.

5.2 Year Goals

Our ultimate objective is to create and implement the first VR instructional tool for Electrical and Computer Engineering (ECE) students at the University at Albany. We'll provide an immersive environment for students to learn the ins and outs of a banded resistor all at their own pace. We hope to expand this idea to other components (capacitors, inductors, transistors) and other areas of ECE as well (visualization of EMF and radio frequency signals) that would be harder concepts to grasp on traditional 2D educational media.

6 Physical Design

6.1 System Architecture

A workflow block diagram for visualization is shown in Figure 4.

6.2 System Components or Subsystems

All components and subsystems are listed out and given a brief description.

- **VR Headset:** We use Meta Quest 2 because of its budget friendly option. It is also used to test and run the VR environment.
- **Unreal Engine 5:** Our choice of game engine development. With the new release of Version 5 and the introduction of nanite, the engine optimizes visuals which align with our initial objective for immersion. The internal visual code Blueprints also makes it easier for beginner coders.
- **Software Development:** Back-end Python Server which receives and recognizes electrical components and builds a netlist, runs the circuit analysis, and then sends back the computation. PySpice library is in the Python language which is different from the Unreal Engine native language.
- **3D Component Design:** We use Autodesk Fusion 360 to design our electrical components.

7 Budget

Figure 5 lists the main hardware and software recommended to replicate our project so far. The Meta Quest 2 is an affordable and widely supported hardware. The link cable is recommended but there may be other options to connect the VR headset with the development environment. The Autodesk Fusion 360 application is recommended as a free education

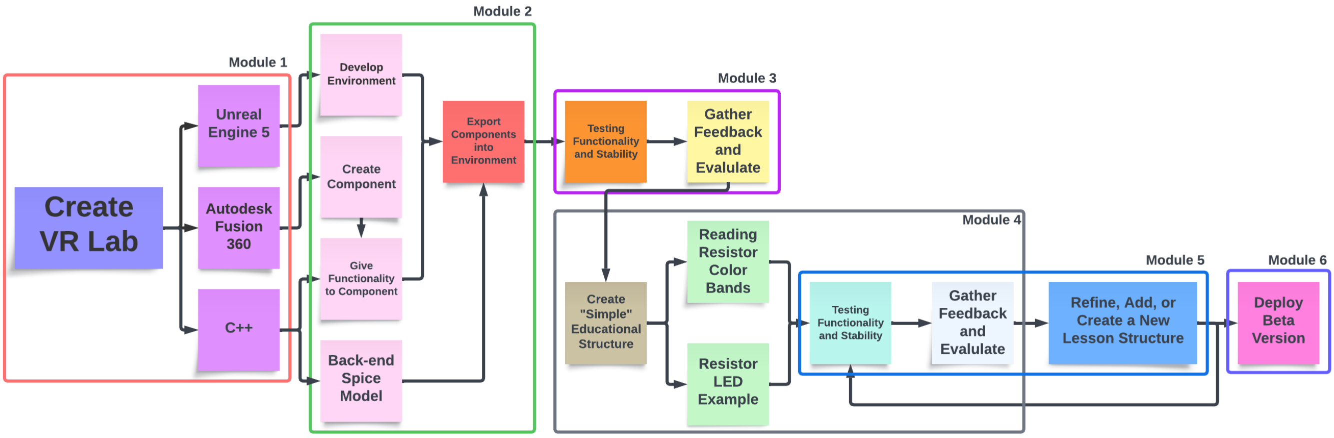

Figure 4. Workflow block diagram

plan is offered to students. The Unreal Engine 5 application is used as it provides powerful tools for VR development due to its recent update to version 5.3. It is worth mentioning, in the case of our project, we already had access to our headset device, resulting in no net costs.

8 Verification Testing

8.1 User Interface

The User Interface (UI) needs to be able to take in information from the user inputs. The goal was to take in electrical component specifications i.e. component names, properties, and their ”node” connections (how they'd be oriented and connected in a circuit or on a breadboard). In Figure 6. shows the final iteration in which the user can select their choice of electrical components from the menu. The example shows the resistor component which has options to select all component specifications as mentioned above. After the ”Add” button is selected, the component is sent to the server (built on the netlist). When the ”Compute” button is selected, the server is told to run the circuit analysis. This feature is 80% as the bulk of the work is completed and is functioning as intended. The menu can be expanded on with a larger library of components and can be made easier to navigate with more organization.

8.2 Object Interaction

In the current state of this tool, there are various components which have already been imported into the environment. The goal behind these components is to simulate real-world engineering tasks. Manipulation of resistors and capacitors is fully functioning, allowing users to pick them up and place them anywhere within the play space. In order to implement this, each asset much first be created using CAD Software, such as Fusion360. Once the component has been created, we must import it into Unreal Engine, which requires proper configuration to create a mesh for our components. Lastly we implement proper hit-boxes and collision areas to each base of our electronics, so that any instance spawned into our environment retains the same physics within the environment. An example component can be seen in Figure 7.

8.3 Socket Programming and SPICE Implementation

We use Transmission Control Protocol (TCP) as our chosen standard for transmitting data and messages across networks. Our approach involves establishing TCP connections to enable direct communication between a Python server and Unreal Engine 5. Unreal Engine 5 conveniently offers a TCP Socket Plugin, accessible through the Unreal Engine Marketplace, which facilitates communication with TCP servers entirely in blueprints as depicted in Figure 8.

By utilizing socket programming to create communication between the server and the game, we've successfully integrated SPICE

Items	Price (MSRP)	Date	Description	Links
Meta Quest 2 (128GB)	299.99	11/8/2023	XR Device	https://www.meta.com/quest/products/quest-2/
Link Cable	79.99	11/8/2023	Connects to PC	https://www.meta.com/quest/accessories/link-cable/
Autodesk Fusion 360		11/8/2023	Create Models	https://www.autodesk.com/products/fusion-360/personal
Unreal Engine 5		11/8/2023	Dev Environment	https://www.unrealengine.com/en-US/unreal-engine-5

Figure 5. Bill of Materials

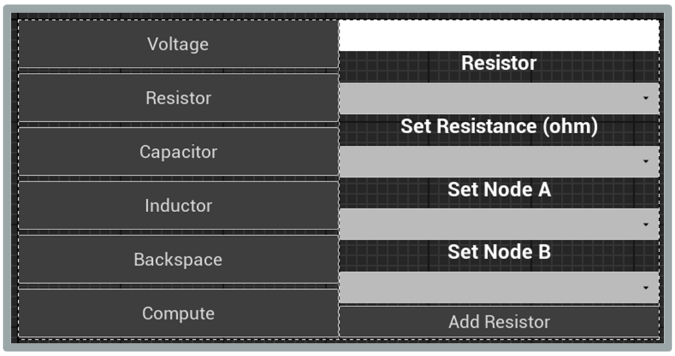

Figure 6. Final iteration of the component widget menu

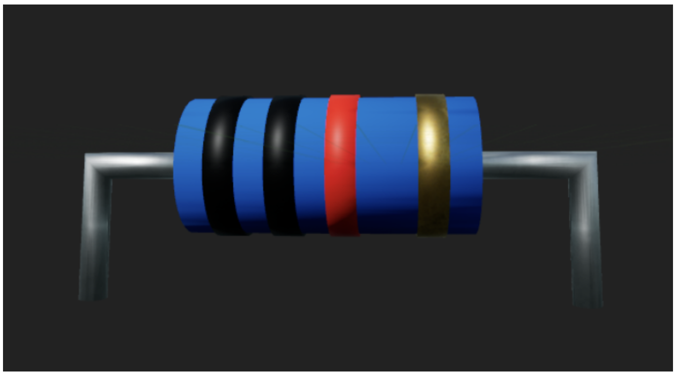

Figure 7. Image of a 1000 Ohm resistor, which changes color based on given impedance value

into our virtual learning environment. Our preferred approach involves using PySpice, a Python module designed to interface with SPICE engines like Ngspice and Xyce circuit simulators. PySpice serves as a powerful tool for electronic circuit simulations, offering a Pythonic interface for circuit modeling and analysis. Additionally, it supports a wide range of simulation types, including DC, AC, transient, and others.

Here's a simplified breakdown of the process:

- **Python Server:** The Python server will be set up to listen for incoming TCP connections. This server will handle the communication with the Unreal Engine client.
- **Unreal Engine 5 Client:** Within the project, you'll utilize the TCP Socket Plugin to initiate a connection with the server. Data transmission will be handled using blueprints. To set up, create a Blueprint Actor inheriting from TcpSocketConnection, drop it into the level, and use it.
- **Data Exchange Protocol:** In this protocol, the game transmits input data using the UI (user interface) Widget to the server via TCP, formatted as a netlist. Upon reception, the server performs analysis and integrates the data into its list. Subsequently, when the user triggers a computation request, the server processes the accumulated data and returns the computed output to the game. This structured protocol ensures smooth transmission and processing of data between the game and server components as shown in Figure 9.

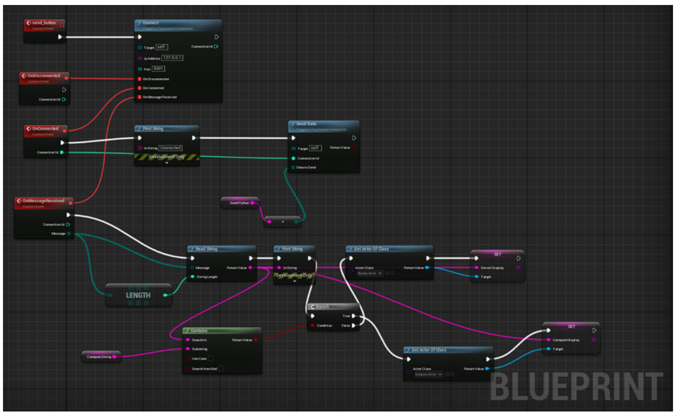

Figure 8. UE5 utilizing TCP Socket Plugin

9 Future Work

Listed below are a select few items we would like to see implemented, given the project were

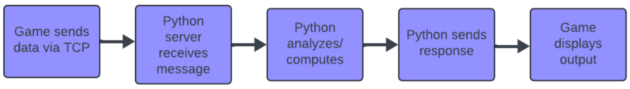

Figure 9. How the game and the server communicate with each other

to continue into another build:

- **Realistic Environment:** Source detailed 3D models and textures for realism and incorporate spatial audio for enhanced immersion
- **Breadboard Connectivity:** Developing a dynamic actor that adjusts an internal value automatically as components are connected to the circuit, enabling real-time calculations using PySpice.
- **Guided Practice:** Create a step-by-step lesson, in which the user is shown how an object interacts. Afterwards, the user will be tested in order to proceed to the next lesson.
- **FETs and BJTs:** Integrating Field-Effect Transistors (FETs) and Bipolar Junction Transistors (BJTs) into the environment can offer hands-on learning and introduce students to the fundamentals of transistors and semiconductor devices.
- **Gates and Switches:** Integrate 3D models of gates and switches and their functionality.
- **Bode Plots:** Integrating Bode plots into a UI within a game environment to introduce students to frequency response analysis and signal processing concepts.
- **Advanced Topics Exploration:** This will introduce advanced electrical engineering concepts such as digital logic circuits, operational amplifiers, and microcontrollers.

10 Conclusions

With the original intentions to create an XR lab environment, it became apparent that the implementation of circuit analysis software was a crucial step before further development in another area. Prioritizing functionality before appearance made us pivot in terms of goals and lead times. A back-end python server was developed to receive and recognize electrical component properties, create a net-list in real time, run a desired type of circuit analysis, and communicate important data regarding the user made circuits with the game engine environment. A front-end was needed in order to accept user inputs. For our case, this was implemented in the form of a UI menu to set and send the desired electrical component properties to the server in a properly formatted list. This was achieved within the '23-'24 academic school year at UAlbany. With room for more development and improvement, the design and implementation of the initial analysis application should in turn accelerate designs in other areas and allow future iterations of this project to focus more on appearance rather than functionality.