Visualizing Fundamentals of Magnetic and Electric Fields in Virtual Reality

Mario Manalu

July 2021

1 Introduction

In the first half of the 19th century, Michael Faraday studied magnetism and electricity. He hypothesized that a changing magnetic field can generate a current. He tested his hypothesis by conducting numerous experiments, most notably the coil and magnet experiment. He created a coil by wrapping a paper cylinder with wire, connected the coil to a galvanometer, and moved a magnet back and forth inside the coil. He noticed that the galvanometer showed there was a current induced when the magnet was in motion.

Faraday theorized that a current was induced because there were magnetic lines of force around the magnet. He demonstrated the existence of these lines of force by conducting the iron filings experiment. He poured tiny pieces of iron filings to coated sheets of paper placed on top of bar magnets. The iron filings were attracted to form a shape that resembles that of the magnetic lines of force. The iron filings' formation showed that the magnetic lines of force were curved and not straight as Newton conceptualized. After this experiment, Faraday began to visualize magnetic field lines as a collection of lines, each of which has a certain direction. These visualizations have transformed Faraday's ideas into more precise mathematical and physical concepts that provide more insights to people who study magnetic and electric fields.

There is one problem that new learners face in learning magnetic and electric fields. They are presented with two-dimensional visualizations when, in fact, these fields are three-dimensional. These visualizations are prone to misinterpretation. I aim to tackle this problem by creating immersive experiences in visualizing electric and magnetic fields in Virtual Reality. I harnessed the power of GPU, computer graphics, and virtual reality technologies to create a new way of learning about the two fields that make use of physical movement and gestures in a fun and engaging way. I developed four exhibits, each illustrates magnetic and electric fields in different scenarios. **The Simple Moving Magnet Exhibit** visualizes the magnetic and electric fields around a bar magnet as it moves back and forth. **The Interactable Magnet Exhibit** allows users to directly interact with the magnet by holding it and seeing the magnetic and electric fields generated by the magnet. **The Coil and Magnet Exhibit** is a recreation of Faraday's experiment in which users can move a magnet inside a coil to generate a current that will turn on a light bulb. The last exhibit, **The Two Magnets Exhibit**, visualizes the magnetic field that fill in the space between two bar magnets.

The goal of all the exhibits in this project is to give the user better visualizations of magnetic and electric fields. I realized that there are several layers of complexity in how these fields should be described. For instance, calculating the magnetic field at a point right at the pole of the magnet would give rise to division by zero. Thus, I use approximations in the calculations. In particular, I treat a magnetic field of a bar magnet by using the same equations as I would for the electric field of a pair of oppositely charged particles.

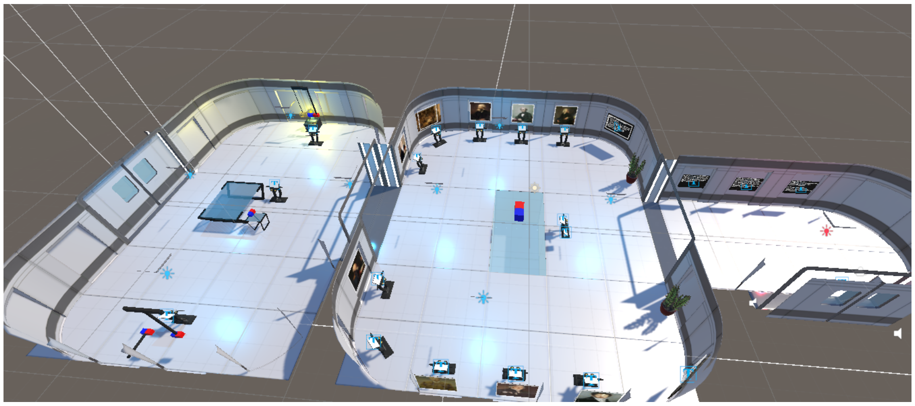

Figure 1: A picture of the virtual world from above.

In the next four sections of my paper, I describe the details of each of the four exhibits as well as the process of designing and developing them. In the next section entitled **Rendering the Fields**, I explain the process of rendering the fields. In the following section entitled **Game Instructions**, I walk through how one can engage in the virtual experience using the HTC Vive VR headsets. Finally, I reflect on my learning experience in working on this project and share my future visions on how this project can be improved.

2 The Simple Moving Magnet Exhibit

The Simple Moving Magnet Exhibit is the first exhibit that a user encounters when entering the virtual world. The purpose of this exhibit is to give the user a first dive into the theory of magnetic and electric fields. In this exhibit, the user is presented with a magnet that oscillates along the x-axis. As the magnet moves, the user can see a dynamically-rendered field that is computed based on the position of the magnet. The key concept illustrated in this exhibit is how and why the vector at a certain point in three-dimensional space around the magnet change its direction and magnitude when the magnet moves from one

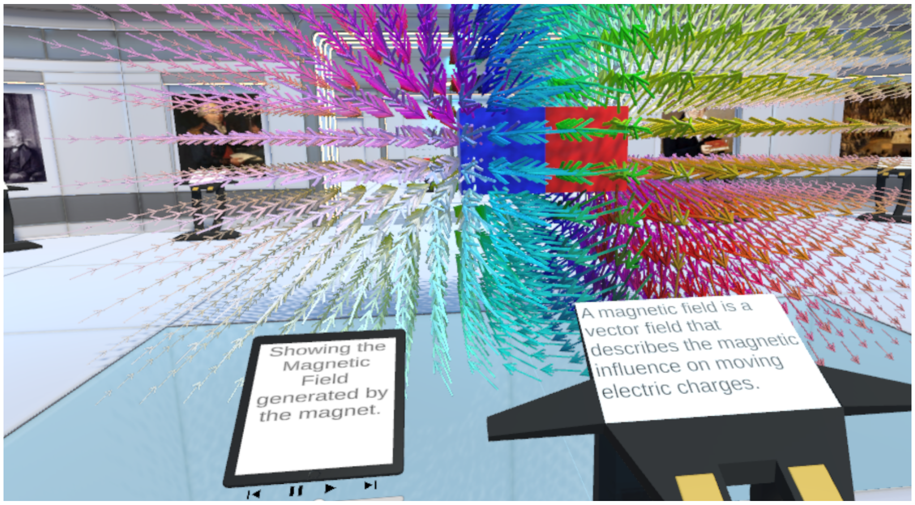

Figure 2: A picture of the Simple Moving Magnet Exhibit.

point to another.

2.1 Description

The user is provided with a tablet and a panel to help them study the key concept. The user can choose to see either the magnetic field or the electric field by pressing a button on the tablet. The tablet has a pause button which, if pressed, stops the magnet. This feature allows the user to observe the field more closely as they can pay attention to the vectors that are away from the magnet, which may not be visible when the magnet is in motion. The tablet is also equipped with a slider that the user can use to position the magnet as they wish. If the slider is slid to the left, the magnet will move in the same direction. The feature is implemented to encourage the user to use physical movement to visualize the field, creating a more immersive learning environment.

While a field is displayed, the user can go to a nearby panel to have a read on the fundamentals of the displayed field. The panel has two buttons that can be used to navigate through the pages. One can think of the panel as a

Figure 3: Pressing the play button on the tablet.

textbook containing two short chapters on magnetic and electric fields. The panel is placed in front of the moving magnet so that the user can identify the name, definition, and formula of the displayed field. The panel minimizes the need to take off VR headsets to read about the fields and wear them again to see the visualizations. The benefit of having such a panel is that the user can have the various learning resources they need to study the two fields in one place.

2.2 Fundamentals of Magnetic and Electric Fields

In this section, I will present a brief introduction to the fundamental concepts of magnetic and electric fields. These fundamentals motivated the emergence of the study of vector field in mathematics about 60 years after 19th century scientists started drawing field lines to visualize forces around magnets.

Definition 2.1. A magnetic field is a vector field that describes the magnetic influence on moving electric charges [1].

A magnetic field can be represented by a collection of lines of force that are continuous. The forces emerge from north-seeking magnetic poles and enter south-seeking magnetic poles. Unlike electric charges which can be isolated, the two magnetic poles always come in a pair. When a bar magnet is broken, two new bar magnets are obtained, each with a north pole and a south pole. In other words, magnetic “monopoles” do not exist in isolation, although they are of theoretical interest. Magnetic fields are created by electric currents in the space around where the currents flow. Currents which do not change with time make constant magnetic fields which we call DC fields. Currents which change sign in a regular manner with time are called alternating currents or AC. Magnetic fields play an important role in our society. They generate currents that power a lot of electronic appliances.

Definition 2.2. An electric field is the physical field that surrounds electrically-charged particles and exerts force on all other charged particles in the field, either attracting or repelling them [2].

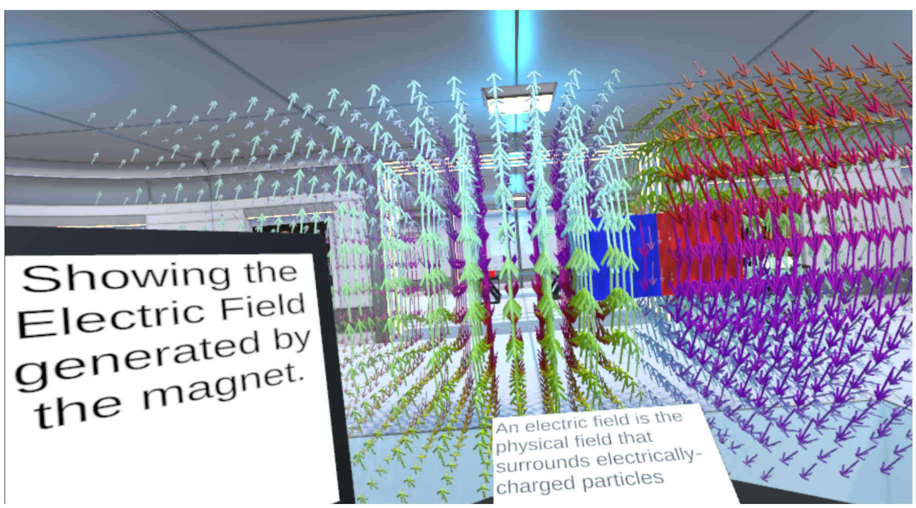

Figure 4: A first person point of view of an electric field.

It is a vector field associated with the Coulomb force experienced by a charge at a point. If the field is created by a positive charge, the electric field will be in radially outward direction. Otherwise, the electric field will be in radially inwards direction. Electric fields originate from electric charges, or from changing magnetic fields. The formula that describes the electric field experienced by a point charge is

$$E = \frac{F}{q}, \tag{1}$$

where F is the force experienced by the charge q [2].

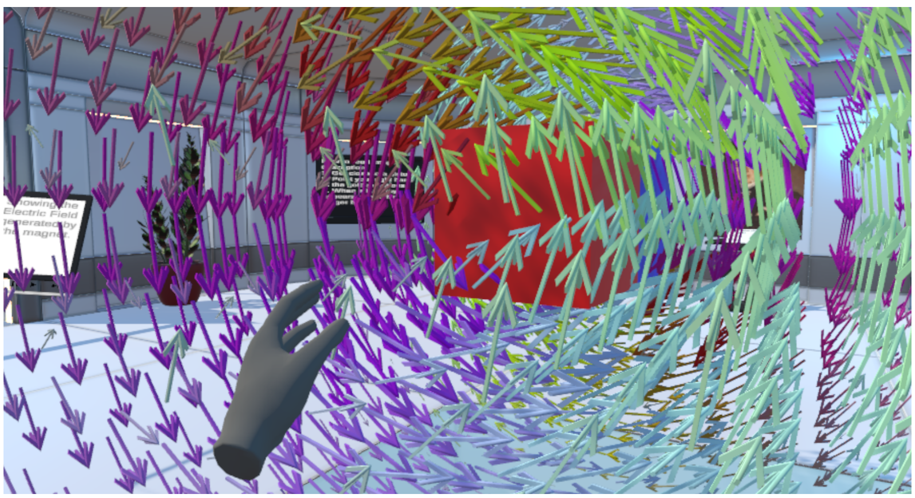

Figure 5: A picture of an electric field from the inside of the field.

2.3 Development

I began the development of this exhibit by creating a magnet prefab.

Definition 2.3. A prefab in Unity is a special type of component that allows fully configured GameObjects to be saved in the project for reuse [3].

I created two cubes, colored one cube red and the other blue, and stick them together as a prefab. I decided to make a magnet prefab so that I can reuse the prefab instead of making a magnet from scratch for future virtual experiences. Prefabs reduce time spent on not only creating objects from scratch but also modifying objects. If the prefab is modified, the modification is applied to all of the prefab instances across all scenes. Once the prefab is created, adding a magnet into the scene can be done by dragging the prefab into the hierarchy window.

The second step in developing this exhibit is adding a game object of type *VectorField* into the scene. The *VectorField* object reads the charges and positions of the two poles in each frame. It uses the two inputs to render a field of the user's choosing. A complete walk through of how the fields are rendered will be given in the section entitled **Rendering the Fields**. The size of the rendered field is configured to be larger than the magnet itself because larger visualizations are easier to understand and helpful in illustrating magnetic force at a point far from the magnet.

The next step is writing the *MoveMagnet.cs* and *GUIPanel.cs* scripts. These scripts enable the magnet to move if the play button on the tablet is pressed. Else, the magnet is paused and the slider is activated, allowing the user to move the magnet by moving the slider. The two scripts communicate with each other through the UI slider. The UI slider acts as an interface that sends a *float* value based on the position of the slider within the interval. When the magnet is not paused, the *Update()* function in the *GUIPanel.cs* script increments the value of the slider by 0.05 in each frame. The *MoveMagnet.cs* script reads in the slider value and computes it into a cosine function that returns the x-coordinate of the magnet displayed in the following frame.

3 The Interactable Magnet Exhibit

The purpose of the Interactable Magnet Exhibit is to allow the user to interact with the magnet in a way that gives much more freedom than the first experience. In this exhibit, the user can move the magnet in any direction as

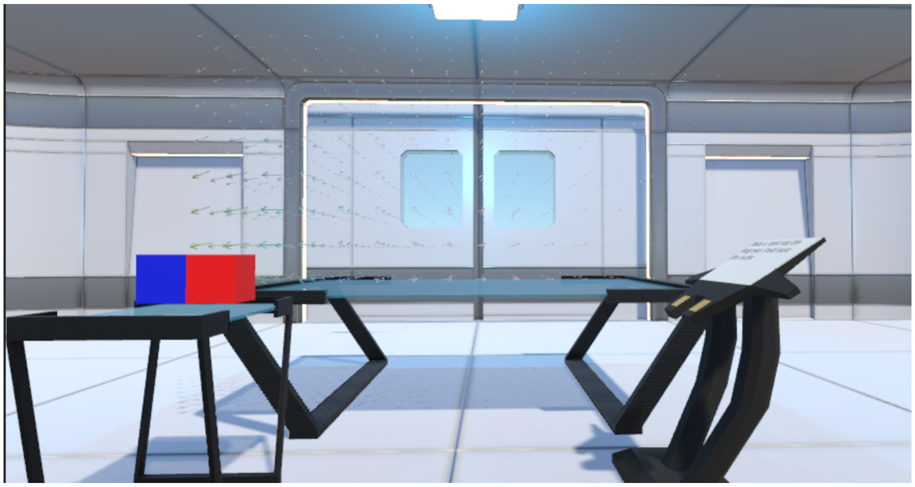

Figure 6: A picture of the Interactable Magnet Exhibit.

opposed to moving it along the x-axis only. The idea behind the creation of this exhibit is once the user has a basic understanding of magnetic and electric fields, they can try to move, rotate, and flip the magnet to see the different visualizations of the magnetic or electric field.

3.1 Description

The user is presented with a smaller bar magnet placed on a small table. A *VectorField* game object sits on a big table to the left of the small table. The user can grab the magnet with either their virtual left or right hand. Once the magnet is in the user's virtual hand, they can approach the bigger table where the field of their choosing is located.

There are three ways in which the user can move the magnet and observe a change in the field. The user can wave the virtual hand that is holding the magnet. This way of moving the magnet is recommended when the user chooses to visualize the magnetic field. When the user wants to see the electric field, it is recommended that the user holds the magnet and touch the left trackpad to move the virtual character or touch the right trackpad to rotate the character's head around. The differences in the recommended ways of moving the magnet arise because, in theory, the magnet generates an electric field only when it is moving. In practice, however, we can only move our hands for a few seconds before stopping. It can be observed that as soon as the user stops moving their virtual hand, the magnitude of the electric forces at all points in the field is zero. While the visualization is theoretically correct, there is not much to see in the short few seconds we move our hands. A continuous movement such as translation or rotation of the virtual character in any direction keeps the electric field on display long enough.

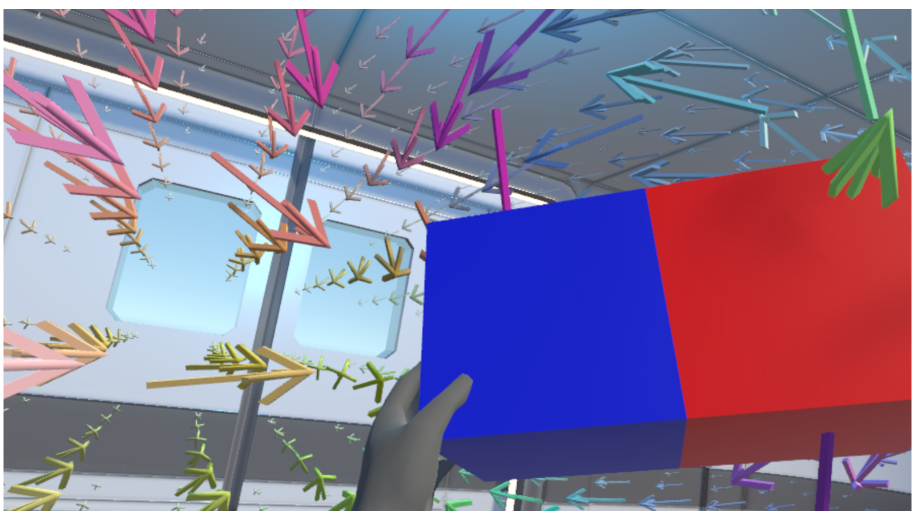

Figure 7: The user is holding the magnet.

There is a common problem with visualizing the magnitudes of magnetic force in textbooks. It is that there is no clear indication of the existence of magnetic force at a point far from the magnet. These kinds of visualizations can be confusing as they do not convey the nature of magnetic force in real life. This exhibit uses the affordances of virtual reality to let the user gain an intuition of the magnitude of magnetic force using a magnet and their virtual hands. When the user is located within the field, the process in which the arrows change their magnitudes becomes more visible. It can be observed from the inside of the field that even when the magnet is far from a particular arrow, the arrow does not completely disappear but appears in a minuscule size. The minuscule size of the arrow implies that there is still some magnetic force at the point despite having a relatively small magnitude.

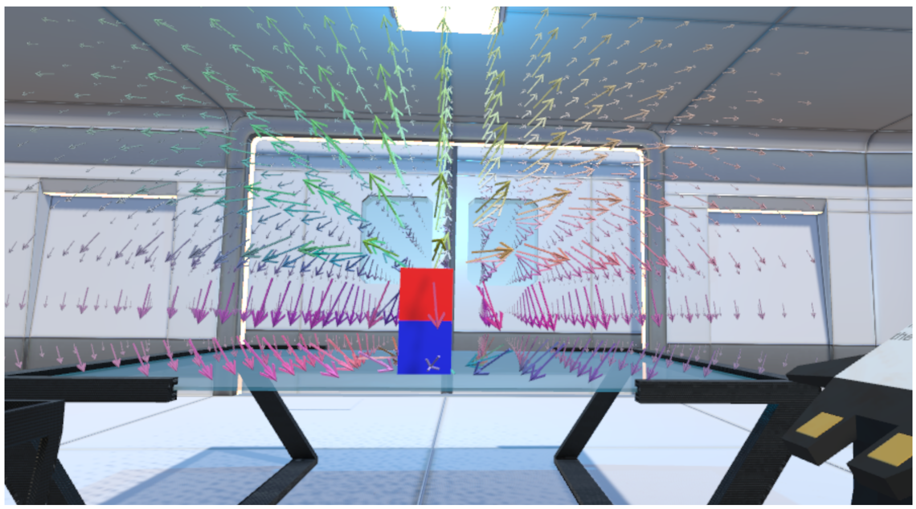

Figure 8: The magnetic field generated by the magnet when it is placed vertically on the table.

A similar kind of problem can be found in the visualizations of electric field too. It is often not clear whether a magnet is moving or not when it generates an electric field in a visualization. One may misunderstand the visualization and think that an electric field can be generated regardless, which is not correct. In this exhibit, the electric field gets rendered only when the magnet moves. As a result, the visualizations of electric field conveys the theoretical idea more accurately.

3.2 Development

I began developing this exhibit by dragging the magnet prefab created in the past into the scene. I decreased the size to mimic the size of an object that can be held. Then, I downloaded and imported a free Unity asset containing home furniture prefabs from the asset store. I created two instances of the table prefab and resized them such that one is significantly bigger than the other. The idea of having a smaller table came about because a physics component that applies gravity is attached to the magnet. The small table allows the user to pick the magnet from a convenient height, minimizing the need to pick up the magnet from the floor.

Once the objects are set up, I added another *VectorField* game object into the scene. I set the length and width of the field to be about the same as the length and width of the big table. An important lesson that I learned from developing this exhibit is that the *VectorField* game objects cannot use the same material. Thus, I duplicated the material used by the *VectorField* game object in the previous exhibit, renamed it, and attached it to the second *VectorField* game object.

The last but most important steps in developing this experience is attaching the XR Grab Interactable component to the magnet and attaching the XR Interactor component to the hands. The former component allows the game object it is attached to be picked up by the virtual hands. The latter component allows the hands to pick up game objects containing the former component.

4 The Coil and Magnet Exhibit

The third exhibit is grounded on Faraday's coil and magnet experiment in 1831. He discovered electromagnetic induction when carrying out this experiment. The explanation behind electromagnetic induction is summarized by Faraday's Law.

Definition 4.1. Faraday's Law states that the magnitude of the electromotive force induced in a circuit is proportional to the rate of change of the magnetic flux that cuts across the circuit. [4].

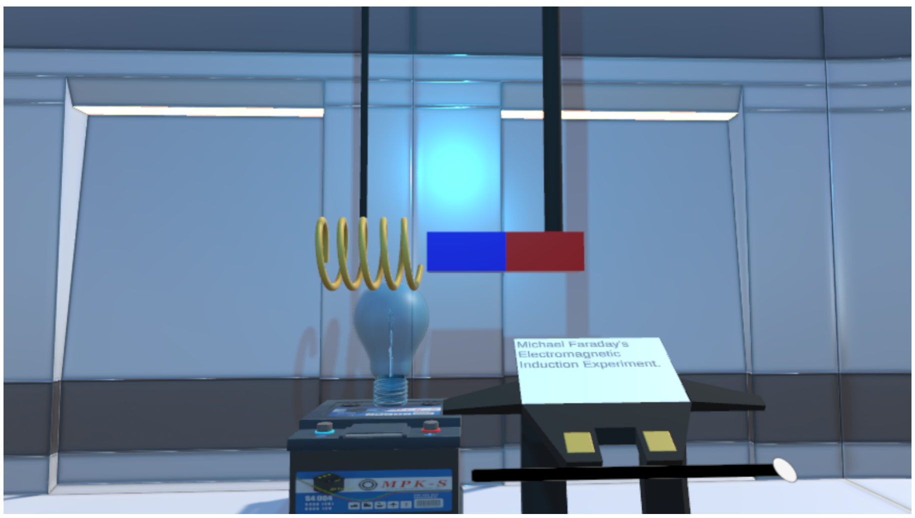

Figure 9: A picture of the Coil and Magnet Exhibit.

4.1 Description

The purpose of this exhibit is to recreate a scenario in which the user can carry out an experiment that mimics Faraday's coil and magnet experiment. The difference between this exhibit and Faraday's experiment is that Faraday used a galvanometer whereas the exhibit uses a light bulb to indicate whether there is a flow of current or not. I believe that doing an experiment that a scientist like Michael Faraday did brings more excitement in learning about electric field. Playing with the Coil and Magnet Exhibit may also give the user an insight to the reasons why Faraday thought what he was thinking or a deeper understanding of how a changing magnetic field generates an electric field.

The user is presented with six game objects in this exhibit: a magnet, an electric field, a coil, a light bulb, a battery, and a panel. The only game object that the user can interact in this experience is the panel. The panel has a slider that controls the position of the magnet in the scene. If the slider is slid to the left, the magnet will slide to the left. The key idea illustrated in this experience is as follows. When the magnet moves, there is a change in the magnetic field. The change in the magnetic field generates an electric field and results in a flow of current that turns the light bulb on.

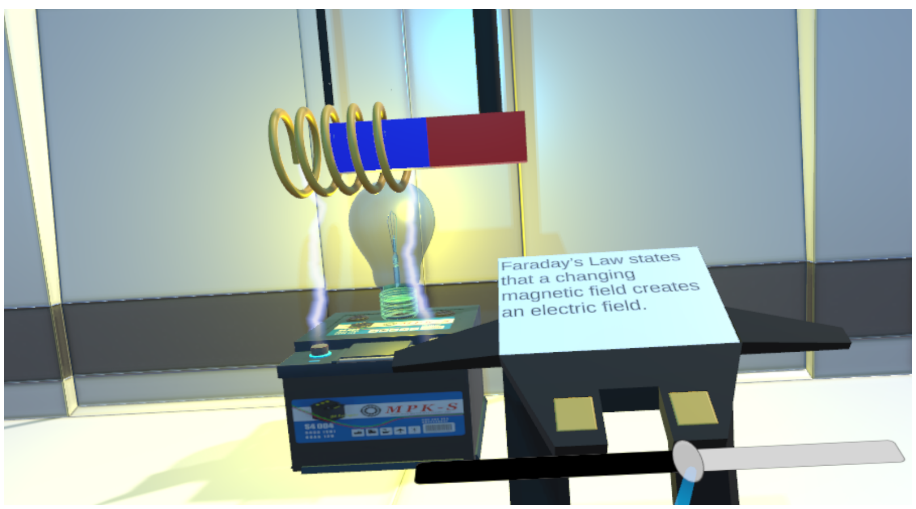

Figure 10: When the magnet moves, a current is induced, and the light bulb turns on.

4.2 Development

I started creating this exhibit by making a coil model in Blender first. When the design was finished, I exported the file in the .fbx format and imported it into Unity. I assigned a material to give the coil a color.

I continued with adding another *VectorField* game object into the scene.

This time I scaled the size of the arrows up up to give better visualizations. I also scaled the size of the field down and positioned them in such a way that shows that the arrows of the electric field "come out" of the coil.

Next, I downloaded some assets from the asset store and Sketchfab. I added a battery, a light bulb model, and two visual lightning effects into the scene. In designing the experience, I pictured that the lightning effects would give an impression of a flowing current that powers the battery, which in turn, powers the light bulb. The light bulb model does not come with a source of light attached to it. So, I created a spotlight game object as a child of the light bulb model. The spotlight object lights up the bulb when a conditional in the *GenerateLightning.cs* script evaluates to true.

Lastly, I wrote the *GenerateLightning.cs* script. The script keeps track of the position of the magnet in each frame. It saves the present position of the magnet and the past position of the magnet in the *Update()* function. If the present position is equal to the past position, then the magnet is not moving. Since the magnet is not moving, it deactivates the lightning effects and the spotlight object in the light bulb model. Else, the present position is not equal to the past position, which means that the magnet moves. Since the magnet moves, it activates the lightning effects and the spotlight that indicates a flow of current.

5 The Two Magnets Exhibit

The Two Magnets Exhibit presents a scenario in which the north pole of a magnet and the south pole of another magnet generate a magnetic field. The purpose of this exhibit is to visualize the changes in the field when a magnet is moving into or away from the other magnet. I find that visualizations of a magnetic field between two magnets in textbooks can be misleading. One can interpret that there is a constant number of magnetic field lines between two magnets. A key concept this exhibit illustrates is that the number of magnetic field lines is not constant.

5.1 Description

The user is presented with three game objects in this exhibit a dynamic magnet, a static magnet, and a panel. The dynamic magnet can be moved along the z-axis using a slider attached to the panel. When the dynamic magnet is moved away from the static magnet, more arrows will be rendered to fill in the space in between the magnets. The increase in the number of arrows illustrates that magnetic field lines are continuous and exists at any point in space.

Another key concept demonstrated in this exhibit is electrostatic force. The ideas behind it is summarized by Coulomb's Law .

Definition 5.1. Electrostatic forces are attractive or repulsive forces between particles that are caused by their electric charges [5].

Definition 5.2. Coulomb's Law states that the magnitude of the electrostatic force of attraction or repulsion between two point charges is directly proportional to the product of the magnitudes of charges and inversely proportional to the square of the distance between them [6].

In this exhibit, the positive point charge is represented by the north pole, and the negative point charge is represented by the south pole. The magnitude of the electrostatic force between the two poles is displayed on the panel. In observing the panel, the user is expected to notice that the magnitude of the electric force increases as the distance between the poles decreases as stated by Coulomb's Law.

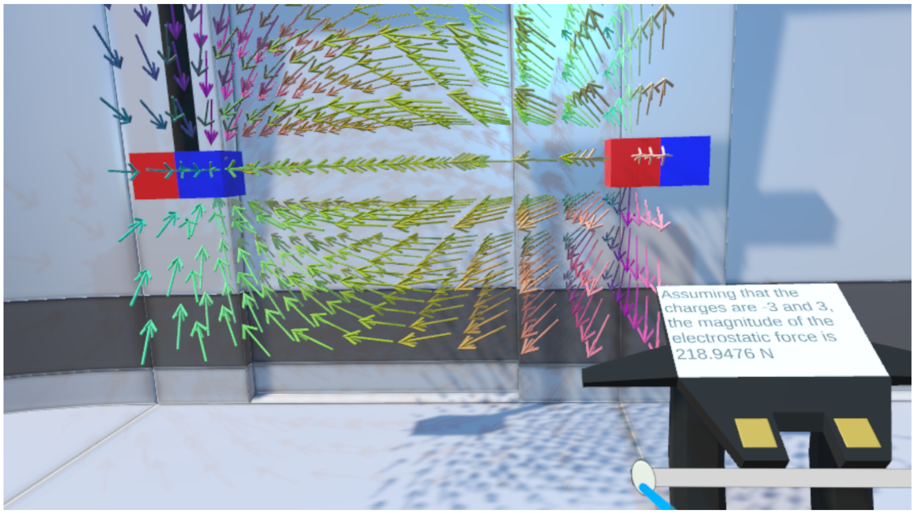

Figure 11: The magnitude of the electrostatic force exerted by the north pole to the south pole is 218 N.

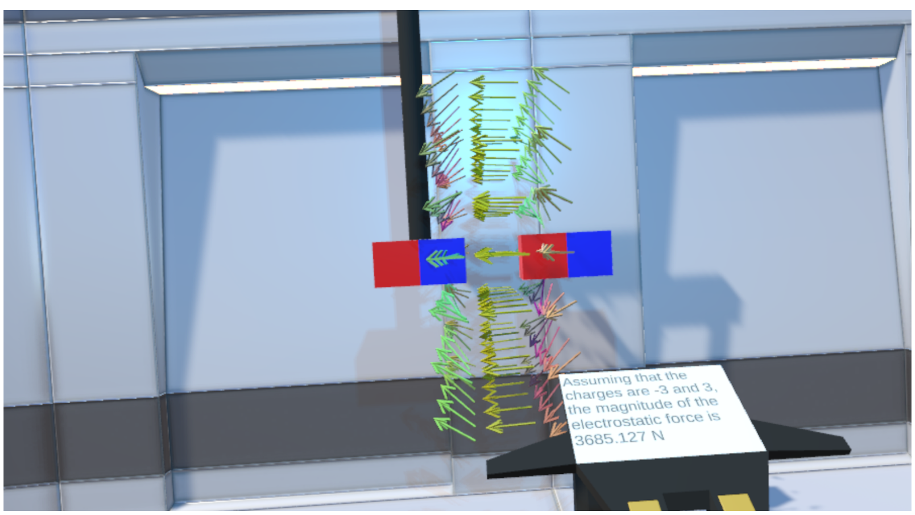

Figure 12: The magnitude of the electrostatic force exerted by the north pole to the south pole is 3685 N.

5.2 Development

I started creating this exhibit by adding two magnets prefabs into the scene. I fixed one of them to a wall and the other to a hollow cylinder that moves along the z-axis. I added a slider attached to panel that moves the dynamic magnet. When the slider is pressed, the user can move the dynamic magnet to move closer or away from the static magnet.

Once the magnets were set up, I added another *VectorField* object and scale it down enough to occupy the space between the two magnets when they are spaced out. While testing out the experience, I noticed a problem. When the magnets are close to each other, some arrows appear from some points behind the dynamic magnet. It is as if there exists some invisible charges that produce magnetic force behind the dynamic magnet.

I modified the *TwoMagnetsShader.shader* file to overcome the problem. I called the clip function inside the *ConfigureSurface()* function. The clip function controls how many arrows in the *VectorField* object gets rendered. It takes in the z-coordinate of the plane that divides the dynamic magnet into two equal parts. Then, it renders the arrows that are in front of the plane and discard the arrows that are behind the plane. As a result, more arrows get rendered as the distance between the two magnets increase and vice versa.

The last step in developing this exhibit was writing the *TwoMagnet.cs* script. The *Update()* function in the script calculates the magnitude of the electrostatic force between the magnets. The charges of the poles and the Coulomb's constant are hard-coded in the script. The positions of the poles in every frame are read in via the transform component of the magnet prefabs. The function uses the positions to find the distance between the poles. When the distance is calculated, the electric force is computed using the formula

$$|F| = \left| \frac{k \cdot q_1 \cdot q_2}{r^2} \right| \tag{2}$$

where k is the Coulomb constant, q_1, q_2 are the charges of the poles, and r is the distance between the poles. Once the magnitude is computed, it sends the the result in the *float* number data type to the display panel.

6 Rendering the Fields

The GPU takes care of the majority of the work in creating the fields. The CPU is only responsible for creating and giving references to buffers that the GPU uses to store data. I use the GPU to do the computation because the CPU could not handle the amount of computation needed without lagging. I describe the process of rendering the fields in the following subsections.

6.1 Rendering the Magnetic Field

There is a fair amount of work done by the *Magnetic.cs* script before the rendering process starts. The script reads in the position of the magnet in each frame and send it to the GPU along with the charges of the poles. The process of rendering the magnetic field begins in the *Update()* function in the *VectorField.cs* script. In the body of the *Update()* function, the *CalculateVectors()* function is called. The *CalculateVectors()* function set up all the buffers for storing data on the GPU side and dispatch the kernel function. Here, the kernel function refers to the function that renders the magnetic field.

When the kernel is dispatched, the work is passed on to the *VectorCompute.compute* script that calls the *Magnetic()* function in the *FieldLibrary.hlsl* script. The body of the *Magnetic()* function computes the magnetic field vector from a point charge and store the result into the appropriate buffer.

When the kernel finishes computing the vectors from all points in the field, the work is passed on to the *VectorDisplay.compute* script. In this script, the vectors are normalized because, theoretically, the size of the vectors that are closer to the magnet are significantly bigger than the vectors that are further away. After the normalization process, the vectors get rendered in the scene.

6.2 Rendering the Electric Field

The process of rendering the electric field is the similar to that of the magnetic field. The difference is the *VectorField.cs* script dispatches the kernel function that renders the electric field. When the kernel is dispatched, the worked is passed on to the *VectorCompute.compute* script. The *Electric()* function in the script computes the vectors from all points in the field. Lastly, the vectors get normalized and rendered in the scene.

7 Game Instructions

In this section, I provide instructions on how to move around the virtual space and interact with the game objects inside it. These instructions will use the HTC Vive controllers, which I used when developing the experience, as an example. I developed my virtual reality using the OpenXR platform to support various kinds of hardware and equipment without additional limitations. I expect that the following instructions apply to other kinds of controllers.

7.1 User Movement

There are two kinds of movement in my virtual reality: position and rotation. Position refers to the position of the virtual character at a specific frame whereas

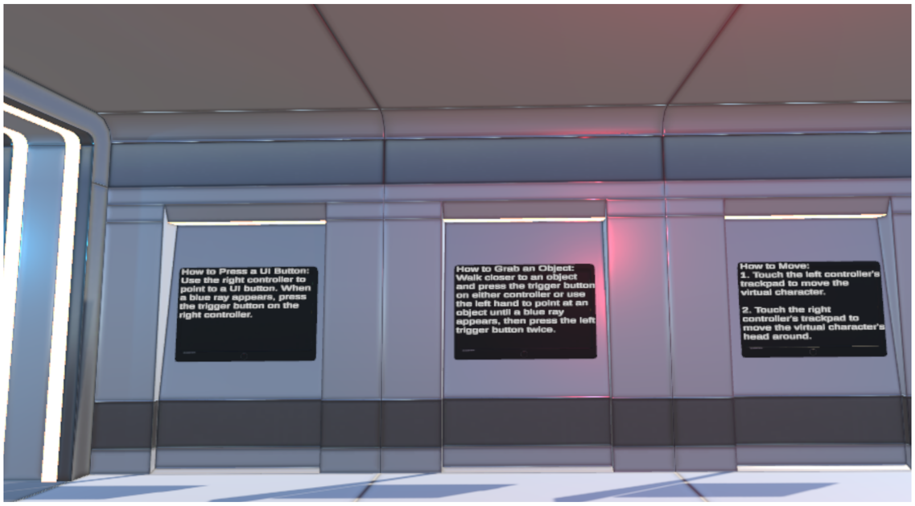

Figure 13: The first frame shows the user instructions on how to move and interact with game objects.

rotation refers to the virtual character's head rotation. The position movement is implemented to enable users to play in a relatively small space in reality whereas the rotation movement is implemented to minimize the need to rotate the user's head when looking at an object in the virtual reality.

The left controller's trackpad can be used to move the virtual character. To move forward, one can touch the upper part of the trackpad. Similarly, touch the bottom part of the trackpad to move backwards. The benefit of implementing this movement is that the user can "walk" about the virtual world without actually walking in the real world.

The right controller's trackpad can be used to rotate the head of the virtual character. To rotate right, one can touch the right part of the trackpad. To rotate left, touch the left part of the trackpad. The character's head can only rotate right or left. To rotate the head upwards or downwards, the user needs to rotate their own head. I placed all of the game objects at eye level to minimize the need to move the user's head up and down. This practice is considered a good game design in general.

7.2 Grabbing Interaction

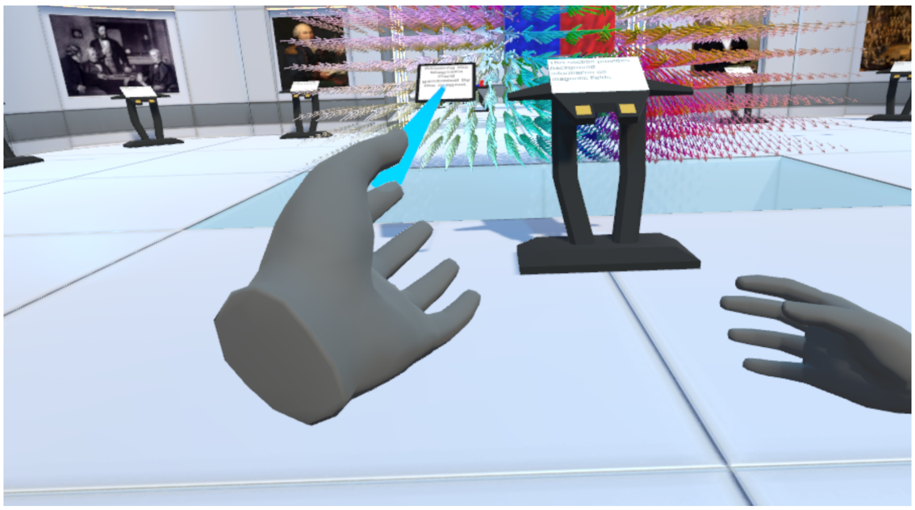

Figure 14: Grabbing the tablet from afar.

There are two game objects that the user can grab with their virtual hands: the tablet and the interactable magnet in the second exhibit. Both of these objects are designed in a way that makes learning more engaging as they motivate the user to move their own hands and be more active. In designing the grabbing interactions, I decided to enable either hand to grab game objects from within arm's reach but only use the left hand to grab objects from afar.

The interactable magnet can only be grabbed from within arm's reach whereas the tablet can be grabbed from afar too. To grab from within arm's reach, the user can move closer to the object and press the trigger button on the desired hand controller. To grab from afar, the user can lift their virtual left hand up and point it towards the tablet until a blue ray appears. The blue ray indicates that the tablet can be pulled into the user's left hand. While the blue ray is in display, press the trigger button on the left hand controller twice to grab it from afar.

7.3 UI Press interaction

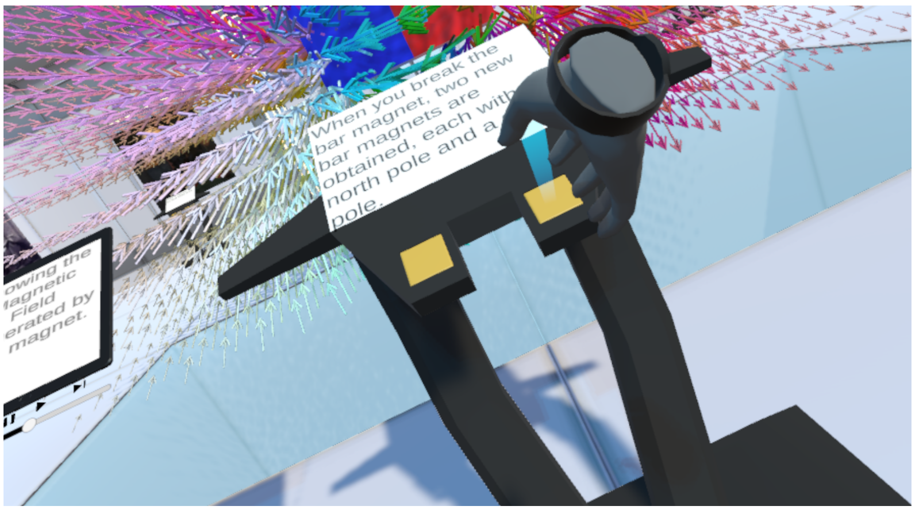

Figure 15: Pressing a button with the right hand.

The last kind of user interaction with game objects is pressing buttons. There are several buttons in the game. There are buttons on the tablet and the control panels. Interactions with these UI buttons are made possible by the XR Ray Interactor component. To press a button, point the right virtual hand to a button until a blue ray appears. The blue ray indicates that the button can be pressed. While the blue ray is in display, press the trigger button on the right controller.

8 Conclusion

Humans have always used visualizations to communicate abstract and concrete ideas. Visualizations help transform words that may be hard to digest into images that are easier to comprehend. In this project, I learned that the

Virtual Reality technology has the potential to render visualizations that convey theoretical ideas more accurately.

I reflect on my learning experience and share my visions for future work in the following subsections.

8.1 Personal Reflection

My experience in studying the Virtual Reality technology was both challenging and rewarding. It is challenging because the learning resources are not updated fast enough to keep up with the technology that is developing at a fast pace. There were software updates and new version releases every week that make online tutorials and documentations irrelevant and obsolete. Other challenges include the various tools and programming languages I needed to learn to develop a basic virtual reality. For instance, I had to learn to program in HLSL and ComputeShader to use the GPU for faster computation. These languages can be frustrating at times because they do not recognize compile-time errors. If there are errors, then the errors will be detected at run time. Sometimes, they cannot detect errors and just quit the application. Once I got a hang of these two languages, it was a smooth sailing.

I learned about how powerful the GPU really is in working on this project. In the previous iterations of the project, I had it compute more than 1331 triple integrals in the span of one frame, which is approximately 0.2 seconds. I knew the GPU is designed to do heavy-computation, but this was new to me. This was a revelation. Looking back, this was the fruit of all the hours I spent on learning HLSL and ComputeShader.

I also enjoyed designing the virtual world. I think anyone can be as creative as they would like to be when it comes to designing a virtual world. I decided to go with the futuristic theme in my virtual world because it suits the purpose of the project. I prioritize simplicity in my design process. For instance, there is a lot of empty space for the user to walk around between objects. I designed the virtual world in that way to give the user a lot of space to be active.

8.2 Future Work

I originally envisioned that it would be great if the user could teleport rather than "walking". I was hesitant to implement teleportation at the beginning, but I was convinced that "walking" can give the user motion sickness. I also think that adding hand animations would give a more realistic look to the hands. For instance, when the trigger button is pressed, the index finger of the left virtual hand would mimic the user's left index finger. I animated the left hand in the middle of the project and thought that I could animate the right hand later. Then, I got distracted with new ideas and had no time left to animate the right hand. At the end, I discarded the animations on the left hand.

I can imagine the virtual world having more exhibits of experiments by scientists like Ampere, Oersted, Volta, Coulomb, and many other scientists. My ultimate goal is to have the virtual world be a laboratory in which the user can carry out notable experiments that build our understanding of magnetism and electricity today.

9 Acknowledgement

I would like to express my gratitude to Professor French for supporting me with my ideas, giving constructive feedback, and spending a few hours pair-programming with me. I would like to thank my research peers; thank Mingyang and Harper who volunteered to try out my experiences and gave feedback; thank Harper for explaining how textbook visualizations can be misleading, downloading pictures for the scene, and writing a summary of the pictures; thank Will for writing the majority of the compute shaders and for sharing Unity tips and tricks that saved me a tremendous amount of time. I would like to thank all game developers and 3D designers who made their work available for download at no cost on the Unity Asset Store and Sketchfab. I credit their work in the *Citation.cs* script. Lastly, I would like to thank Grinnell College for funding my Virtual Reality research project.

10 References

[1] Feynman, Richard P.; Leighton, Robert B.; Sands, Matthew (1963). The Feynman Lectures on Physics.

[2] Britannica, T. Editors of Encyclopaedia (Invalid Date). Electric field. Encyclopedia Britannica. https://www.britannica.com/science/electric-field

[3] Unity, Contributors of Unity (2021, August 3). Prefab. Unity Documentation. https://docs.unity3d.com/Manual/Prefabs.html

[4] Britannica, T. Editors of Encyclopaedia (Invalid Date). Faraday's law of induction. Encyclopedia Britannica. https://www.britannica.com/science/Faradays-law-of-induction

[5] Helmenstine, Anne Marie, Ph.D. (2020, August 25). Chemistry Definitions: What are Electrostatic Forces? https://www.thoughtco.com/definition-of-electrostatic-forces-604451

[6] Helmenstine, Anne Marie, Ph.D. (2020, August 25). Coulomb's Law Definition in Science. https://www.thoughtco.com/definition-of-coulombs-law-604963