Revealing Complex Patterns from Quantum Physics Simulations

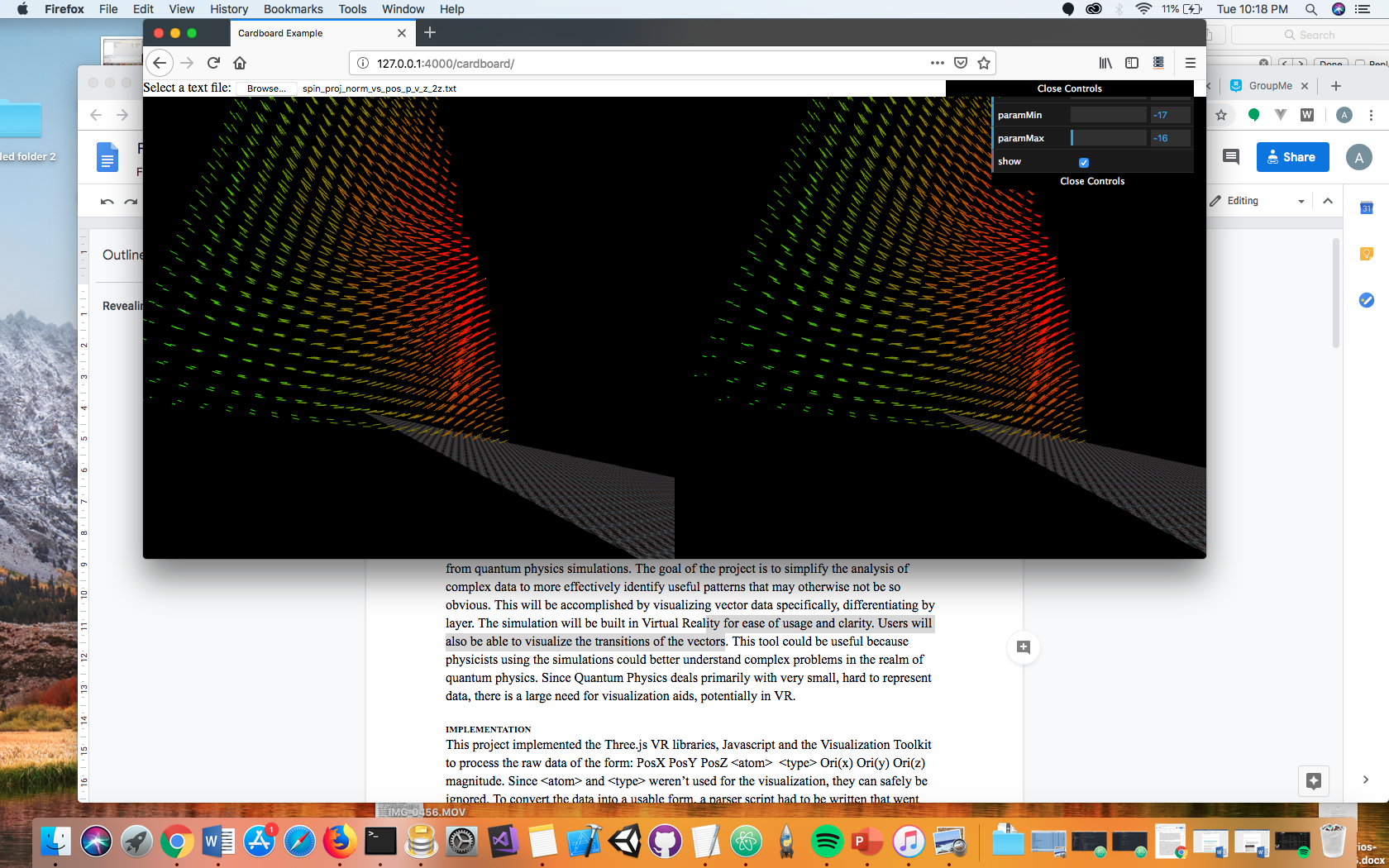
Enrique Rosen (rosen.258), Arturo Barrios(barrios.19)

**ABSTRACT**

The goal of our project is to visualize large quantities of data from quantum physics simulations and also to show how the vectors transition. This will be accomplished using Three.js in VR to simplify the visualization of the quantum physics data. The computer graphics problem we will address is how to visualize our data, which contains extreme magnitudes. Also, providing information to the user on where vectors from chosen regions transition to.

**INTRODUCTION**

For this project, the team thought it would be interesting and useful to visualize data taken from quantum physics simulations. The goal of the project is to simplify the analysis of complex data to more effectively identify useful patterns that may otherwise not be so obvious. This will be accomplished by visualizing vector data specifically, differentiating by layer. The simulation will be built in Virtual Reality for ease of usage and clarity. Users will also be able to visualize the transitions of the vectors. This tool could be useful because physicists using the simulations could better understand complex problems in the realm of quantum physics. Since Quantum Physics deals primarily with very small, hard to represent data, there is a large need for visualization aids, potentially in VR.



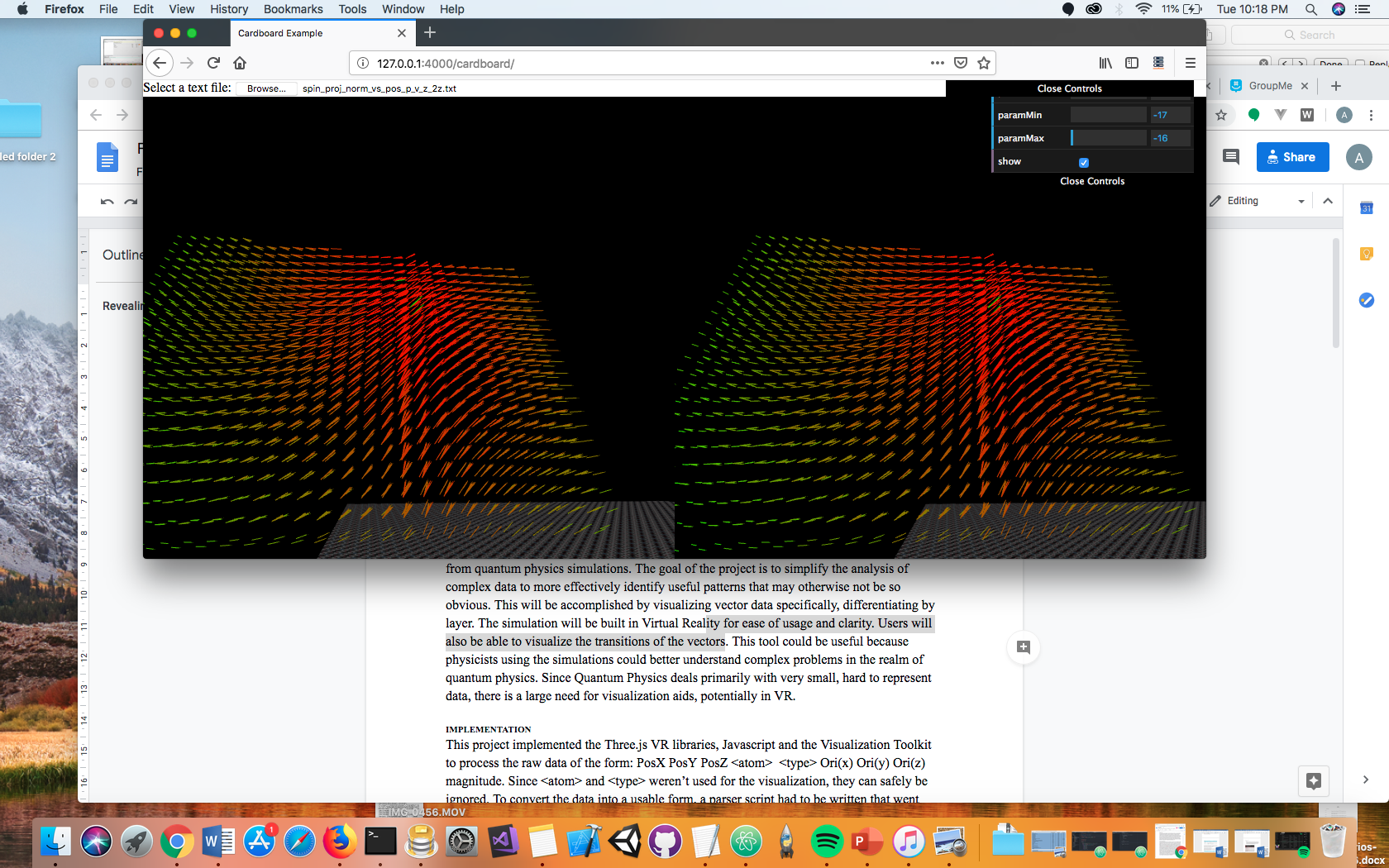
(subset of the data in the range of [-17,-16])

**RELATED WORK**

The goal for this project was to develop a VR project that allowed users with knowledge in Quantum physics to be able to better understand data from quantum physics simulations. More specifically, to be able to find complex and hidden information like the transition of vectors to other layers.

**DESIGN METHODS**

For this project, the team decided to give the vectors a color value between green and red. We chose the vector’s color value based on its magnitude to allow the user to better visualize the transitions going on in the relatively dense data. Furthermore, we chose a dark background because we thought it’d be the best color to allow the user to see the vectors clearly. Additionally, we added a floor to show where the vectors start. The users knowing where the vectors start is important to understanding the patterns that they’re recognizing when looking at the vectors. Finally, the team decided to allow the user to select the range of data that they want to visualize. Since there’s an overwhelming amount of data in this dataset, the team thought it’d be best to give the user an option to view a subset of the dataset. This way, lag is reduced and the user isn’t overwhelmed with the amount of data from the entire dataset. In the image below, you can see that almost all the data is seen by the camera. The user doesn’t have to scroll very far to the left or right to visualize all the data. If we had loaded all the data, the user would not be able to easily visualize all of the data which would effect the user’s experience when trying to find interesting patterns within the data.

****

(subset of the data in the range of [-17,-16])

**IMPLEMENTATION**

This project implemented the Three.js VR libraries, Javascript and the Visualization Toolkit to process the raw data of the form: PosX PosY PosZ <atom> <type> Ori(x) Ori(y) Ori(z) magnitude. Since <atom> and <type> weren’t used for the visualization, they can safely be ignored. To convert the data into a usable form, a parser script had to be written that went through the entire data file and stored the relevant information within the program. For each line of data, an arrow was created on screen representing the vector. The orientation parameters determined the direction of the arrow, the magnitude represented the length of the arrow and the position parameters determined its location on-screen. To make the visualization more clear, a logarithmic scaling algorithm was implemented that produced a number between 1 and 100 from the input, making it easier to be shown clearly. The magnitude was also used to determine the color of the arrow, going from green to red the higher it was.

Since Three.js makes it easier to create shapes and objects, it was selected over standard WebGL. The major computer graphics problem that this project addressed was the visualization of data with extreme magnitudes. The way this problem was addressed was by scaling the data down to a fixed range of values. It was not immediately apparent the best way to scale the data. This was solved by testing different scaling formulas for the optimal result. This was tested with a few thoughts in mind: First, the magnitudes in the raw data ranged from to , Secondly, the scaled values should be from 1 to 100 for convenience and clarity, Third, linear scaling would not create appropriately-scaled values. Ultimately the following equation was produced:

A challenge that had to be addressed was the number of data points that had to be added to the visualization. Before starting, the team had no idea how much data itself had to be parsed. This means that the team had to accommodate for large inputs of raw data. Trying to display all the roughly 200,000 vectors would simply be impossible to do within three.js. This was resolved by adding in the ability to specify specific regions of the data, so far less would have to be on screen at a time.

**RESULTS**

One of the metrics by which the success of our project was measured was by the clarity of the visualization. We would be able to see this by whether or not the vectors with differing magnitudes are clearly distinguishable from each other. Early on this wasn’t accomplished very well. Despite having a satisfactory scaling algorithm for the magnitudes, simply changing the length of the corresponding arrow drawn wasn’t sufficient. Since all the arrows were the same color and there were so many of them, it always resulted in a visualization where it was difficult to tell two vectors apart. Secondly, originally balls were displayed at the origin of each arrow. This was implemented ideally to improve clarity. The team met with Oscar Rosen, a PhD in Chemical Engineering and a specialist in data visualization, so his expertise was invaluable in improving the project. He suggested mapping the arrow color to the scaled magnitude, starting off with green at the lowest, and getting more red as it approached its maximum. He also strongly advised against displaying the balls at the origin of the arrows, how ultimately it was just more clutter in the visualization and expensive to create/display. The color-changing was implemented by changing the input RGB values for creating the arrows. The higher the number, the More R and less G it had. Both of these changes to the project vastly improved the clarity of the visualization. Now looking at certain regions, based on colors and orientation changes of the vectors made it far clearer to spot a transition.

**CONCLUSION**

In conclusion, We feel like this small project could potentially help physicists discover new patterns in quantum physics which could lead to a better understanding of an incredibly complex and exciting field. The universe is a vast and beautiful space, and there’s so little we currently understand about how it operates. Visualizations such as these help us unlock life’s greatest mysteries. It lets us discover who we really are and what we can endeavor to be.