My program visually represents the mathematics of a charged particle's motion near a magnet.

The idea to create this program came from both a previous student's project about the conservation of momentum and a program I used in high school physics about the motion of a test charge in an electric field called 'electric field hockey'. I took Physics C Electricity and Magnetism last year in high school, and I am currently a freshman in Calculus 3, with an intended major of electrical engineering. I had a fairly good understanding of the mathematics and physics I was trying to explain, but not quite the depth needed to make a useful program.

To complete this project, I learned how to use VPython, an entry level programming language that was perfect for this task. Having no previous experience in any programming language, I indeed faced a steep learning curve in this honors computer science class of predominantly sophomores.

I began this class trying to extract basic facts and learn how to access the class repository. With the help of my child prodigy classmates and the mentors, I eventually learned how to access the files and programs I needed to start learning. I started out by reading the class webpage about logic, working with DPGraph, and attending all of the class lectures. When it was time to start working on the final project, Alec Mori was especially helpful in downloading winzip, Python, and VPython on my laptop. I came in early and stayed after class on several Fridays to learn basic VPython from Gillian, who was very patient with my ignorance. After being directed to the given examples, I learned how to use Visual Python from the bouncing ball, dipole, and cross product sample programs. I isolated useful pieces of each code to add to my program, and I learned basic coding techniques such as 'while true' infinite loops, importing controls, setting variables, and 'if' statements.

As my project took shape and I gained a better understanding of the tools I had to work with, I realized that I was not going to be able to fulfill all of the the details my original project proposal. I tried for hours to make a click and drag vector be the user input initial velocity, but after isolating a drag vector from the cross product sample program and setting the axis as an initial velocity, I found that the program would not allow such a changing input. After break, I learned how the user could simply enter the initial velocity values, which might not be as intuitive, but it worked very well.

I ran into the odd problem of the test charge speeding up due to normal acceleration. Since a magnetic field does no work on a particle, this was a flaw in the visualization. It was greatly reduced however when the magnetic field was reduced or the test charge mass was increased.

I then decided that a time varying magnetic field would be very difficult for the user (and me) to comprehend, and not useful to explain with a program. I then decided to make the magnetic field constant with respect to time and varying with respect to position, since the force of a magnetic field depends on the distance to the magnet. For simplicity, I made the magnet face the observer and vary inversely with respect to z, and constant across x and y. The resulting decrease in magnetic field strength sufficiently counteracted the speeding up of the particle, and I ended up with a respectable finished product.