Introduction

Deciding what kind of project I wanted to do for Hypergraphics has been an interesting path. Suffice it to say that after talking with Professor Francis, I decided to combine my project in this class with my current research with Professor Giannetta. There I work with him on nuclear magnetic resonance (NMR). In my research, there is not a lot of visualization; the closest we come to visualization is adding a particular element to a circuit and graphing a function of wave frequency and reflected amplitude in order to find the resonance of the element we added. While this is useful, it does not allow someone to qualitatively grasp the concepts of NMR at the atomic level. This is critical to understanding application of NMR, most notably magnetic resonance imaging (MRI) and NMR Spectroscopy. My NMR project will be focused on visualizing the basic math used in NMR through a real time interactive computer animation (RTICA). This RTICA will visually show atoms changing orientation during a simulation of NMR, much as what happens when actual NMR is performed. This combination of activities complements my knowledge and understanding of NMR and my project will enable a general audience to understand how NMR works to some degree.

Project Description

My project will be written in PythonOGL. I will use the classical model of the magnetic dipole moment. Unlike in quantum mechanics, we will know the direction the magnetic dipole moment of each nucleus at all times. There are two kinds of NMR: continuous wave and pulsed. As intuitively understanding continuous wave NMR is easier, my project will show continuous wave NMR at the atomic level.

The program will generate an array of different elements, each with a different gyromagnetic ratio. Each nucleus will have a magnetic dipole that will point in a random direction. The user will be able to choose the orientation of an external magnetic field and the magnetic dipole moment of each atom will line up either parallel or anti-parallel to that field. Radiation of changing frequency will irradiate the array, one row or column at a time and the user will see each nucleus that is parallel to the magnetic field absorb a photon of its Larmar frequency and flip to the higher energy anti-parallel state. In addition, a simple graph of frequency and absorption will be shown so that the user can visually make the connection between photon absorption and flipping the dipole moment of the nucleus.

The second part of my project will be a game where the user uses NMR to find what element is in each location in the array. The user will be doing the same thing as before, except that the user will not see which atoms absorb radiation and flip dipole moments. The user will then enter a prediction into the program and the program will indicate which positions in the array that the user correctly determined. The interesting part about such a game is realizing that not every nucleus will initially be aligned parallel to the magnetic field and, hence, will not absorb radiation and be detected. This would mean that the user might have to radiate each row or column multiple times.

Project Timeline

F9: Have PythonOGL working on my computer and have an algorithm to manage the array of elements and magnetic field, finish setting up my webpage, make sure I know how svn works

F10: Have the elements absorbing the correct frequency of radiation and work on flipping elements

F11: Have my graph of absorption for each row and column working in two parts: last run results and total results

F12: First part of Project finished, hide graphics of nuclei flipping for game mode

F13: Integrate checking component into game

F14: Work on documentation

F15: Finish project, incorporate additional upgrades if time warrants

Project Goal

Although my program will not be using actual molecules and compounds, and thus concepts like chemical shift cannot be addressed, I would like my project to be a useful tool to illustrate how NMR works and provide a fun game for how to interpret data from an MRI or from NMR spectroscopy.