ElectronVisualized: Project Scope

Date:	May 5, 2022
From:	John Seong
Subject:	ElectronVisualized

Project Overview

An interactive module that visualizes the **electron configurations** of different elements. Displayed in an orbital format, based upon the internal calculations performed that involves the famous Dirac equation, which describes the shape and structure of the electrons in terms of the wavefunction.

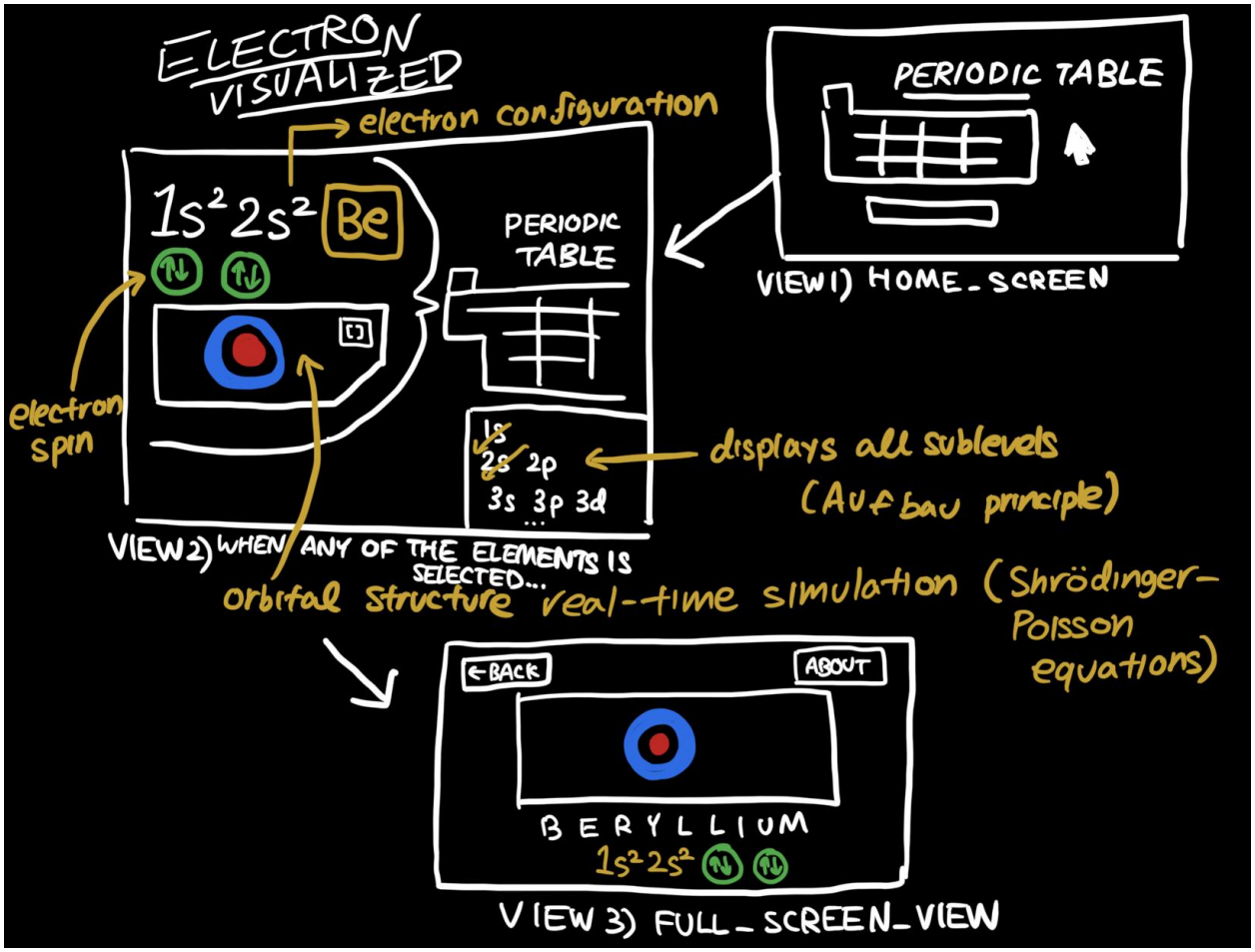
Users

1. **Teachers** and **Professors** will be able to give a better understanding of the orbital theory by providing students with a visual representation.
2. **Students** often struggle with chemistry due to the sheer amount of theoretical background that is needed to understand the course material. Using this interactive module, students will be better able to grasp the different concepts such as sp^3 hybridization with more ease by using techniques such as visualization.

Check out these Links:
https://levelup.gitconnected.com/create-your-own-quantum-mechanics-simulation-with-python-51e215346798
https://backend.orbit.dtu.dk/ws/portalfiles/portal/130802447/main.pdf

Product Features

Program Mockup Sketches



Feature Priority List

1. **MATH AND PHYSICS SIMULATION**: Implement the equations using NumPy and Cython (mainly based upon modern physics) — namely the Schrodinger-Poisson equation and the de Broglie-Bohm equation.
2. **GRAPHICS AND SHADERS DEPARTMENT**: Use particle generating library to simulate the angular momentum
3. **USER INTERFACE**: Prettify UI elements using the graphic library within the module that I am using (whether it is PyGame or HTML Canvas)

Version Guide

1. **Version 1:** Implement OOP and diversify the folder-file structure so it suits the purpose
2. **Version 2:** Import a shader and particle library to visualize the nature of electron movements
3. **Version 3:** For the math and physics part, implement the equations in the most efficient manner possible using different tools and extensions available which include: Cython (C and Python combined), NumPy, etc.
4. **Version 4:** Make the UI intuitive as much as possible, as well as add instruction and a documentation/guide

Schedule Milestones

Version 1: *May 5, 2022 - May 6, 2022*

- Project Planning: *May 5, 2022*
- Folder/File Structure: *May 6, 2022*

Version 2: *May 6, 2022 - May 16, 2022*

- Find and import all libraries related to visualizing the angular momentum and the distribution of the electrons: *May 6, 2022*
- Write all the necessary scripts related to graphics: *May 7, 2022 - May 16, 2022*

Version 3: *May 17, 2022 - May 25, 2022*

- Implement all the equations needed so that the program behaves according to the nature of real-life physics: *May 17, 2022 - May 25, 2022*

Version 4: *May 25, 2022 - May 30, 2022*

- Prettify the user interface: *May 25, 2022 - May 27, 2022*
- Write the documentation and manual: *May 27, 2022 - May 30, 2022*