OVERVIEW

- List the beginning of quantum theory. The quantum origin. With Planck and Bohr and all the stuff they discovered, I will be turning prompts into code visualizations for the quantum process
- If I was explaining quantum theory to someone and wanted to code projects that would sufficiently explain quantum, in the sections “quantum origin” and “quantum theory”. How many visual coding projects would be sufficient and what would they be

QUANTUM ORIGIN

1. Code a black body furnace, it should be connected to quantum and easy to understand as it is the very first step, visualize the problem that Planck encountered. Code it in html…… Should this animation go directly into the resonator discovery or is there something in between ? -> BlackBody.html
2. Code (in html) a visual of 1d standing waves to connect the blackbody uv problem into the resonator discovery (From here on whenever I say to code, it should be in html) -> LightWaves.html
3. Code an html visualization for a resonator that has different energy inside and out, but the energy has the same frequency inside and out -> Resonator.html
4. I believe that 3 things are left to visualize the origin of quantum, quantum of action, Bohrs Orbit, and De Brogile waves. Are any of these closer related to Quantum theory? Or are they still part of the origin ? -> BohrsOrbit.html

QUANTUM THEORY

1. Does Quantum Theory begin with quantum of action?
2. Code a quantum of action visual showing the energy packets moving from the atom -> QuantumOfAction.html
3. Code the failure of modern physics, blackbody radiation uv. -> UVCatastrophe.html
4. Code the wave particle duality -> PhotoElectric.html
5. Code matter as waves, include a particle speed slider and show the visualization from classic to quantum behavior -> MatterWaves.html
6. What is remaining to add to the Quantum Theory folder?
7. Code the uncertainty principle -> Uncertainty.html
8. Code the wave function -> AtomicOrbitals.html
9. Code an animation of Schrodingers equation -> SchrodingersWave.html
10. Now code an animation of measurement and wave function collapse -> WaveCollapse.html
11. Code a double slit wave animation, it should be a quantum wave of matter that branches off from the left and passes through 1 slit in the middle and the through 2 more slits. Place dots at where the wave hits the last wall, make sure the slits are visible. The wave should come from the left center of the wall and pulse, expanding out until it reaches slit number 1 in which all the dots that didn’t make it through stick to the wall and only the wave in the center continues, then it should pass through a double slit where the same

thing occurs, finally it should hit a wall, showing us where the dots ended up……..2nd prompt: No, please fix some parts of it. The waves that touch a wall should be absorbed by the wall and leave points, the waves that pass through should be visible…. 3rd prompt: Please keep the wave visible, it should be a wave of thousands of dots that either pass through or get stuck to the wall, but please make sure it is visible…. Add some sliders to adjust the speed and slit hole size -> DoubleSlit.html

12. Connect that double slit experiment into quantum computing, visualize the qubits….. make the wave part of this code better…. Please include 2 slits, one |0> and the other |1> keep track of how many particles go through each gate and mark superposition zone and collapse zone….. add a speed slider -> QubitSlit.html

QUANTUM COMPUTING

1. What are some programs I could make that involve the behavior of qubits and qudits?
2. Build a quantum sphere calculator. It should involve qubits inside of a sphere and the user should have buttons for every gate. First make a program with 1 sphere then write a program with 2 spheres and add the 2 qubit gates -> QuantumSphere.html, TwoQubitSpheres.html
3. Build a qudit playground -> QuditPlayground.html
4. I want you to make one final program that explains exactly how quantum computing works, show 100 spheres on screen and have a qubit in all of them, and have a "estimated value" for each of them as we can't see the qubits, they should always be moving back and forth between 1,0, and superposition. When collapsed, show a number at the bottom which is all their values in binary bits. Convert that to a base 10 number and add it onto a graph. The user can collapse as many times as they like and when they choose to do so, another base 10 number will be measured and added to the graph. Add an auto mode that collapses every second or so, if the user does not want to keep pressing the collapse button -> CollapsingQubits.html
5. Create a calculator. A user inputs a value for either |0> or |1> and the bot should reply with the other number. -> QubitCalculator.html
6. Create a problem solving quantum tool -> QuantumSearchTool.html
7. Create an environment for qubits to move around freely -> QuantumEnvironment.html
8. Create an environment for 10 qudits to move around. These qudits should have 10 dimensions -> TenQuditsWithTenDimensions.html
9. Optimize a search algorithm, using qubits -> QuantumSearchSimulator.html
10. Write an html code that should be added to one of the 5 sections and tell me which section it should go in -> CHSHBellTest.html

QUANTUM AI

1. Create an qubit battle Royale. Start with 100 qubits and give them 5 seconds to randomize and then collapse. All of the ones that say 0 are eliminated. Go until there’s only 1 qubit left -> QuantumBattleRoyale.html
2. Create a game with a Q learning qudit with 6 dimensions. there should be a spinning bar and 6 places to stand. If the spinning bar touches the qudit, a new geno qudit takes over. The goal at the start is randomly picking 1-6 and then over time Q learning to learn how to avoid the bar….. do not reset the bar every time a new generation starts, the bar should spin no matter what happens with the qudit -> QuditSpinner.html
3. Create an electron battle Royale, it should work like quantum of action, there should be 10 different hydrogen atoms in a circle, all with 1 electron spinning around it, when it moves closer to a layer it emits a light photon, if that releasing energy hits another electron, that atom is eliminated, as such, the electrons can jump to other atoms if they are close enough -> ElectronBattle.html
4. Create a game for 100 qudits with 100 dimensions. All qudits pick a dimension 1-100 and the dimension with the most qudits wins. It should start with 1 per dimension. Then split the game up again. If x qudits were on the winning dimension, make x dimensions and split them all up to their own dimensions. Run the code again with less dimensions, this time it’s whoever’s on a dimension by themselves wins the game. Finally there should be a replay button that restarts and lets the qudits learn how to strategize…… Make it a 10 x 10 grid with little red points in each of them. Then let there be a simulate button which goes to the next step of the game -> QuditBattleRoyale.html
5. Create an entanglement simulator. 2 qubits choose their gates at set intervals, if they both end up being entangled, reward them +1, if they do not end up entangled, reward them -1. They should not know what the other qubit chose -> EntanglementSimulator.html
6. Build a qubit maze, the maze should allow the qubit to pick two paths to walk down in a constant superposition state. It should start in the center with 4 paths available, only 1 should reward the qubit +1, the others should reward it -1. Using that system, it should find the exit fairly easily. The maze should be auto generated but have tons of paths and options, not just 4 paths. The agents should start in the center, once an agent fails, the next generation starts. -> QuantumMaze.html
7. Create a survival ai game where the agent has to constantly activate gates in order to survive from a qudit that can move anywhere……The code is supposed to be a qubit In a quantum sphere that is constantly changing its x,y,z values by quantum gates to avoid a qudit…… Both the qubit and qudits should be AI agents -> QuantumQubitChaseDemo.html

QUANTUM PROGRAMMING

1. What are 3 algorithms that can be optimized when coded using Qsharp?
2. Build Shor’s algorithm in Q#….. code all of it in qsharp…. This has an error, quantum arithmetic package doesn’t exist….. the code works but not how I would like it to run….. you created an infinite loop it’s going to 11 every time. -> ShorsGemini.qs

3. Now in Q sharp, code Grover’s search algorithm…. Show each step of the search -> GroversSearch.qs
4. Build Hamiltonian Simulation in Q# -> HamiltonianSimulator.qs