Young Photonics Congress 2025 (ICFO). Castelldefels (Barcelona), March 2025

Photonic Quantum Computer Simulator

Pineda Salgado, Freddy[1]; Laplana Mora, Teo[1]; Pizarro Castells, Joel[1]; Pérez García, Francesc ;[2]
[1] 5th Grade student at Pompeu Fabra High School
[2] Technology and Science teacher at Pompeu Fabra High School
Avinguda Fèlix Duran i Canyameres, 3. 08760, Martorell, Barcelona

1- INTRODUCTION

We've created an interactive web tool that demonstrates the principles of quantum computing. It uses 3D visualization with Three.js to represent Bloch Spheres. Users can apply quantum gates (Hadamard, Pauli-X, Pauli-Z), observe different state transitions and measure qubit probabilities.

2- MATERIALS & SOURCES

- HTML: The standard language for website creation.

- JavaScript: A programming language mainly used to make dynamic websites.

- Three.js: A JavaScript library to render 3d graphics.

3- TERMINOLOGY

- Qubit: Basic unit of Quantum Information

- Bloch Sphere: Visual representation of a Qubit

- Quantum Gate: Basic quantum circuit operating on a Qubit

- Theta (θ) and Phi (φ): The angles of the Qubit Vector represented on the Bloch Sphere

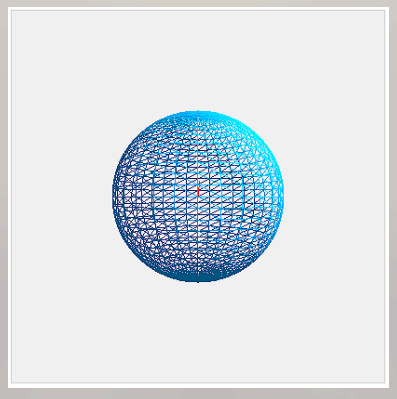
4- METHODS

The first step is to define the vectors Theta and Phi. We do that by generating a random number and multiplying it by Pi, or 2*Pi on Phi's case.
After that, we get the probability of the Qubit giving us a Zero 0 by getting the Cosine of Theta divided by 2. We do the same for the probability of getting a 1, but instead of a Cosine, we use a Sine.

Secondly, the vectors on the Bloch Sphere are represented. A sphere geometry with a radius of 1, and a height and width of 32 is created. The material for the sphere, specifying the color and that the wireframe is visible, is created. These variables are applied to the mesh creation function.
To represent the vectors, an array with the vertices of the line defined is created. The vertices are defined using this formula.

```
const x = Math.sin(theta) * Math.cos(phi);
const y = Math.sin(theta) * Math.sin(phi);
const z = Math.cos(theta);
```

After defining the geometry using the vectors, and the material, the line can be displayed on the screen.



The last part of the code are the Quantum Gates. Firstly, we define the calculations each gate does:
- Hadamard Gate: Makes the probability of having a 1 or a 0 be 50 percent for each one.

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

```
theta = Math.PI / 2;
phi = 0;
```

- Pauli-X Gate: Exchanges the probability of the qubit being a 1 for the probability of the qubit being a 0.

$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

```
theta = Math.PI - theta;
```

- Pauli-Z Gate: Subrtacts Phi's angle to 360. The probability stays the same.

$$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

```
phi = (phi + Math.PI) % (2 * Math.PI);
```

After applying any gate, or generating a new Qubit, the Bloch Sphere is updated acordingly.

5- RESULT

After creating the code, it is applied to the website in conjunction with HTML.
Thanks to this project, we've been able to learn about Quantum computing and Qubits, while applying our knowledge in website creation.

The final result can be observed at: https://pompeu.neocities.org/quantic