**Quantum Random Number Generator (QRNG) & Visualization**

**Presentation Writeup**

**1. Introduction**

In today's world, randomness plays a crucial role in fields like **cryptography, AI, simulations, and gaming**. Traditional random number generators (RNGs) either use **mathematical algorithms** (pseudo-RNGs) or rely on classical physics (e.g., thermal noise). However, these methods are not truly random.

**Quantum Random Number Generators (QRNGs)** leverage the **inherent uncertainty of quantum mechanics** to generate **truly unpredictable numbers**, ensuring the highest level of security and randomness.

Our project implements a **QRNG using two approaches**:

1. **Camera Noise-Based QRNG** – Extracts randomness from live pixel fluctuations.
2. **Quantum Tunneling-Based QRNG** – Uses a quantum simulation to generate random numbers.

Additionally, we have **five graphical representations** to visualize the QRNG output in real-time.

**2. Working of QRNG**

**2.1. Camera Noise QRNG**

1. The laptop's camera captures live video frames.
2. Each pixel has an intensity value (brightness), which fluctuates due to noise.
3. This noise is extracted and converted into binary (0s and 1s).
4. The sequence forms a truly random number.

**2.2. Quantum Tunneling QRNG**

1. We simulate **quantum tunneling**, where a particle has a probability of passing through a potential barrier.
2. The outcome is **random** due to quantum mechanics.
3. Each tunneling event is mapped to **0 or 1**, creating a stream of random numbers.

**3. Visualizations**

To make the QRNG process interactive and understandable, we created **five graphical representations**:

**1. Live Camera Noise Visualization**

* Displays the raw camera feed and highlights noisy pixels used for randomness extraction.

**2. Probability Distribution Graph**

* Shows the frequency distribution of generated random numbers (0–255).
* Helps verify that the numbers are uniformly distributed.

**3. Binary Entropy Stream**

* Displays the **real-time binary output** (0s and 1s) from the QRNG.
* Ensures there are no patterns, making it truly random.

**4. Quantum Tunneling Simulation**

* A **3D visualization** of particles hitting a barrier.
* Some particles pass through due to tunneling (random event).

**5. 3D Particle Flow Representation**

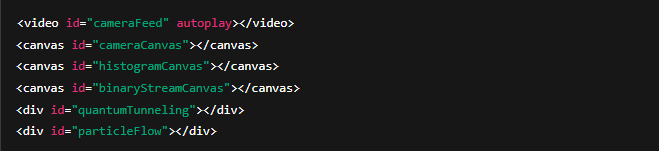
* Uses **Three.js** to show a cloud of moving particles.
* Demonstrates quantum-based randomness in a visual manner.

**4. Code Explanation**

The QRNG implementation consists of **HTML, CSS, JavaScript, and Three.js**. Below is a brief explanation of the key parts:

**4.1. HTML Structure**

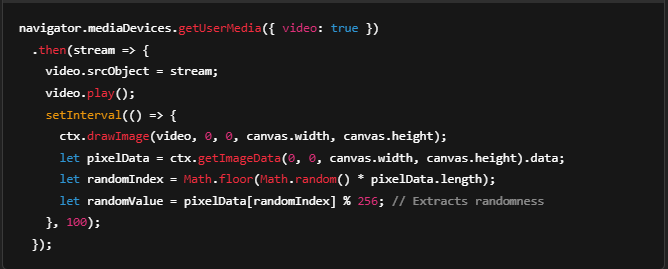
* Contains sections for each visualization.
* Includes a **video element** for camera noise and **canvas elements** for graphs.



**4.2. JavaScript (QRNG Logic)**

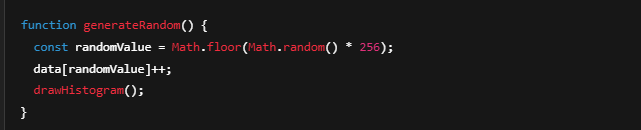
**Camera-Based QRNG**

* Captures pixel data from the camera and extracts **random values**.



**Quantum Tunneling Simulation**

* Uses **Three.js** to animate a wave function approaching a potential barrier.



**Probability Distribution Graph**

* Updates the graph dynamically to reflect randomness.

function generateRandom() {

const randomValue = Math.floor(Math.random() \* 256);

data[randomValue]++;

drawHistogram();

}

**5. Use Cases of QRNG**

QRNGs have numerous applications, especially in areas requiring **high security and unpredictability**:

| **Application** | **Description** |
| --- | --- |
| **Cryptography** | Used to generate **secure encryption keys** for banking, government, and military applications. |
| **Artificial Intelligence** | Enhances stochastic models and ensures unbiased machine learning datasets. |
| **Scientific Research** | Used in **Monte Carlo simulations**(A Monte Carlo simulation is a mathematical technique that simulates the range of possible outcomes for an uncertain event.) and quantum experiments. |
| **Online Gaming & Gambling** | Ensures fair, **unmanipulable random results** in online betting and casinos. |
| **Lottery Systems** | Generates tamper-proof numbers for **lotteries and random draws**. |

**6. Conclusion**

* **Traditional RNGs** are often predictable or rely on classical physics, making them **vulnerable** to attacks.
* **QRNGs**, on the other hand, use quantum physics to provide **true randomness**, making them ideal for **high-security applications**.
* This project demonstrates QRNG using **camera noise and quantum tunneling**, **visualized through graphs and 3D simulations**.
* The **interactive visualizations** help in understanding the randomness generation process.

**Future Improvements:**

* Implement a **hardware QRNG** using photonic quantum devices.
* Optimize the tunneling simulation using **GPU acceleration** for better realism.
* Integrate QRNG with **real-time encryption applications**.

**Why is QRNG Better Than Traditional Encryption?**

Traditional encryption (AES, RSA) relies on **pseudo-random number generators (PRNGs)**, which are **algorithm-based** and can be predicted with enough computational power. **Quantum Random Number Generators (QRNGs)**, however, use **quantum phenomena** to produce **truly random, unpredictable numbers**, making them far more secure.

**Advantages Over PRNGs**

| **Feature** | **QRNG** | **PRNG** |
| --- | --- | --- |
| **True Randomness** | Yes (Quantum-based) | No (Algorithm-based) |
| **Predictability** | Impossible | Possible with enough computation |
| **Security** | Physically secure | Can be compromised if the algorithm is exposed |
| **Periodicity** | Never repeats | Eventually repeats |

**Why Isn't QRNG Widely Used?**

1. **High Cost & Hardware Needs** – QRNGs require **specialized quantum hardware**, while PRNGs are software-based.
2. **Slower Speed** – Current QRNGs generate random numbers **slower** than PRNGs.
3. **Integration Issues** – Most encryption systems are designed for **PRNGs**, making QRNG adoption difficult.
4. **Limited Availability** – QRNG technology is mainly used in **government, military, and research labs**.

**Future of QRNG**

As **quantum technology advances**, QRNGs will become faster and more affordable, leading to their use in **secure banking, government communications, and post-quantum cryptography**.

a