Optimal NLP: Explaining the Fusion Collider 3D Application Using Inputs and Outputs

Need a Python code for a Fusion Collider 3D application physics simulation program that visualizes and models particle collisions, particularly focusing on nuclear fusion reactions and high-energy particle collisions. Inputs and outputs:

Core Concept

This application should simulate particle collisions in a 3D environment, allowing users to explore different types of nuclear and particle physics interactions through an interactive interface.

Inputs

1. **Collision Mode**
    - Users select from different collision types: Nuclear Fusion, Proton-Proton, Lead-Lead, Proton-Lead, or Electron-Positron
    - Each mode represents different physics scenarios (from low-energy fusion to high-energy CERN-style collisions)
2. **Energy Parameters**
    - Energy scale selection (keV, MeV, GeV, TeV)
    - Energy value slider to set the specific collision energy
    - The energy determines collision probability and outcomes
3. **Particle Selection**
    - For fusion mode:
        - Selection of atoms/isotopes from the periodic table (H, H-2, H-3, He, Li, etc.)
        - Each atom has properties (atomic number, mass, neutrons)
    - For CERN collision modes:
        - Selection of particles (proton, antiproton, electron, positron, lead ion)
        - Each particle has properties (charge, mass)
4. **Collision Geometry**
    - Approach angle (0° to 180°)
    - Controls the trajectory of colliding particles
5. **Control Commands**
    - Start, Stop, and Reset buttons to control the simulation

Outputs

1. **3D Visualization**
    - Real-time 3D rendering of the collision process
    - Visual representation of particles with appropriate colors and sizes
    - Animation of particles approaching, colliding, and resulting products
2. **Reaction Information**
    - Equation of the reaction (reactants and products)
    - Details of atomic numbers and masses involved
    - Probability of successful fusion or interaction
3. **Particle Emissions**
    - List of emitted particles (neutrons, protons, electrons, gamma rays)
    - Visual representation of emission patterns
4. **Energy Analysis**
    - Energy released in the reaction (in MeV)
    - Conversion to Joules and TNT equivalent for context
    - For CERN collisions: center-of-mass energy calculations
5. **Data Presentation**
    - Text displays showing reaction equations
    - Numerical data on probabilities and energy values
    - Color-coded visualizations of particles and outcomes

Data Flow

1. User configures the collision parameters (mode, particles, energy, angle)
2. Upon starting the simulation:
    - The physics engine calculates trajectories and collision outcomes
    - The 3D visualizer renders the particles and their movements
    - If collision occurs, the physics engine determines reaction products and energy release
3. Results are displayed in both the 3D visualization and text panels
4. User can modify parameters and run new simulations

The application bridges complex physics concepts with intuitive visualization, allowing users to explore nuclear and particle interactions through a user-friendly interface rather than complex mathematical equations.

Non-Optimal Natural language:

The Fusion Collider 3D application allows users to simulate and visualize particle collisions by selecting collision types (fusion or high-energy), particles (from hydrogen isotopes to lead ions), energy levels, and approach angles as inputs. The system then outputs a real-time 3D visualization of the collision, complete with reaction data showing the resulting particles, energy release calculations, and probability statistics based on accurate physics models.