VR Development Task: Radiation Interaction

Deadline: 7:00 PM, 13th June (IST)

---

1. Task Overview

Your task is to create a virtual reality (VR) scene where the user can interact with three distinct radiation sources and use a detector to identify the type of radiation emitted by each. This exercise focuses on demonstrating the unique properties of Alpha, Beta, and Gamma radiation in a practical, observable way.

---

2. Core Components Needed

You will need the following 3D models and UI elements in your scene. You are **free to use any simple placeholder 3D models** that visually represent these objects (e.g., simple spheres, blocks, cylinders with labels).

- **Radiation Sources (3x):**
  - One source labeled "Alpha Source"
  - One source labeled "Beta Source"
  - One source labeled "Gamma Source"
- **Radiation Detector (1x):**
  - A single handheld object that the user can pick up.
- **UI Text Element (1x):**
  - A simple text display (e.g., a floating panel) to show detected radiation type.

---

3. Interaction Flow & Behavior

The core of the interaction involves the user manipulating the detector and observing specific real-time feedback for each radiation type.

- **Scene Setup:**
  - Place the three "Radiation Source" models in distinct, easily accessible locations in the VR scene.
  - Place the "Radiation Detector" model within easy reach of the user (e.g., on a virtual table).
  - Ensure the UI text element for displaying the detected type is visible but initially blank (e.g., "Detected Type: ---").
- **User Action: Grabbing the Detector:**
  - The "Radiation Detector" must be a **grabbable** object. The user should be able to pick it up and hold it using their VR controller (e.g., Meta XR SDK grab interaction).
- **System Action: Detecting Proximity and Triggering Effects:**

    - When the user moves the "Radiation Detector" **close to** (within a defined detection radius of) one of the "Radiation Sources", the system should trigger distinct visual and auditory effects *specific to that radiation type*.
    - **For "Alpha Source":**
        - **Sound:** Play a unique, low-frequency, impactful sound (e.g., a dull thud or heavy pop).
        - **Particle Effect:** Emit thick, slow-moving, relatively large particle effects from the source, traveling a very short distance.
        - **UI Update:** The UI text element updates to "Detected Type: Alpha".
    - **For "Beta Source":**
        - **Sound:** Play a distinct, medium-frequency, rapid crackle or hiss.
        - **Particle Effect:** Emit medium-sized, fast-moving particles from the source, traveling a moderate distance.
        - **UI Update:** The UI text element updates to "Detected Type: Beta".
    - **For "Gamma Source":**
        - **Sound:** Play a unique, high-frequency, continuous hum or subtle shimmering sound.
        - **Particle Effect:** Emit very thin, extremely fast-moving, beam-like or wave-like particle effects from the source, traveling a long distance.
        - **UI Update:** The UI text element updates to "Detected Type: Gamma".
- **Leaving a Source:**
    - When the "Radiation Detector" is moved **away from** any source (outside the detection radius), the specific particle effects should stop, the unique sound should fade out, and the UI text element can either clear or revert to a "No Source Detected" message.

---

4. Technical Implementation Notes (C# Script Focus)

- You will primarily use a C# script (e.g., attached to the Detector object or a central game manager) to manage the interaction logic.
- **Real-time Updates:** Ensure effects and UI update smoothly and immediately as the detector moves.

---

5. Important Considerations

- **Simplicity:** Focus on making the interaction functional and clear, rather than overly complex visuals.
- **Model Choice:** Feel free to use simple cubes, spheres, or cylinders for the sources and detector. What's important is the *behavior* you program.

- **Clarity:** The core demonstration should be unmistakable: bring detector near source X, see/hear specific effect X, UI shows X.
- **Testing:** Ensure the detection radii are well-calibrated for easy and clear interaction.

Good luck with the task!