**Rage Room with VR Environment: Step-by-Step Process**

Task 1: Set Up Unity Project & Configure VR Environment

- Created a new Unity 3D project.

- Installed XR Interaction Toolkit for VR functionality.

- Set up XR Rig with VR controllers for player movement and interaction.

Task 2: Create the Ground Plane

- Added a Terrain object for a larger ground area.

- Scaled the terrain to fit the player’s movement.

- Applied a texture (e.g., concrete) to the terrain to match the rage room theme.

Task 3: Add a Skybox

- Selected a Skybox from Unity’s presets.

- Customized the skybox for a more intense and immersive environment.

Task 4: Add Environment Objects

- Imported 3D assets such as crates, barrels, and destructible objects.

- Placed these objects around the room manually for interaction.

- Added rigidbody components to make objects react to physics.

- Applied colliders (box, mesh, or sphere) to all interactive objects for proper collision detection.

- Textured objects using appropriate materials to fit the rage room theme.

Task 6: Configure Lighting and Shadows

- Added a Directional Light to simulate overhead lighting.

- Enabled shadows for all major objects to improve visual realism.

- Adjusted light intensity and shadow settings for a balanced indoor effect.

Task 8: Implement Basic VR Interaction

- Added XR Grab Interactable components to the grabbable objects (e.g., hammers, plates).

- Configured the XR Grab Interactable component to handle grab and release functionality.

- Fine-tuned object properties like mass and drag for realistic physics behavior when grabbed or thrown.

Task 9: Write the VR Interaction Script

- Created a custom C# script to manage VR interactions:

- Managed grabbing, holding, and releasing objects.

- Handled collision detection between objects.

- Ensured objects react properly to physics when dropped or thrown.

Challenges Faced:

- Initially faced problems with collision detection, resolved by adjusting colliders and rigidbody properties.

- Texturing the objects to match the theme was challenging but was eventually achieved after several trials.

Output :

