Experiment No. 8
Include animation and interaction in immersive environment
Date of Performance:
Date of Submission:
Marks:
Sign:

Experiment No. 8

Aim: Include animation and interaction in immersive environment

Theory:

1. **Animating Objects:**
   - Identify objects in the environment that you want to animate, such as doors, drawers, or rotating fans.
   - Create animations for these objects using Unity's Animation or Animator system.
   - For example, you can create an animation for a door opening and closing, a drawer sliding in and out, or a fan rotating.
   - Attach the animations to the corresponding objects in the scene.
2. **Interactive Objects:**
   - Identify objects in the environment that the player can interact with, such as switches, buttons, or levers.
   - Implement interaction scripts that respond to player input (e.g., clicking or touching) to trigger actions.
   - For example, you can create a script that toggles the lights on and off when the player clicks a light switch or adjusts the volume when the player interacts with a stereo.
   - Attach these interaction scripts to the interactive objects in the scene.
3. **Player Interactions:**
   - Implement player interactions with objects by detecting input events from VR controllers or mouse clicks.
   - Use Unity's input system to detect when the player interacts with objects in the environment.
   - Based on the input, trigger animations, change object states, or perform other actions as needed.
4. **Feedback and UI:**
   - Provide visual and auditory feedback to the player when interacting with objects.
   - Display UI elements such as tooltips or prompts to inform the player about available interactions.
   - Use sound effects or particle effects to indicate successful interactions or changes in the environment.
5. **Testing and Iteration:**
   - Test the interactions and animations in the environment to ensure they work as intended.
   - Iterate on the design based on feedback and playtesting to improve the overall experience.
   - Fine-tune the timing, responsiveness, and visual/audio feedback to make interactions feel intuitive and satisfying.

```
using UnityEngine;

public class DoorInteraction : MonoBehaviour
{
  private Animator doorAnimator;

  void Start()
  {

      doorAnimator = GetComponent<Animator>();
    }

    void Update()
    {
      // Check for player input to trigger the door animation
      if (Input.GetButtonDown("Fire1")) // Change "Fire1" to the appropriate input axis for
your setup
      {
        doorAnimator.SetTrigger("Open");
      }
    }
}
```

1. **Setting up Animation Parameters:**
   - In the Animator window, create a trigger parameter named "Open".
   - Create a new transition from any state to the door open animation state and set the condition to trigger when the "Open" parameter is true.
2. **Testing:**
   - Run the scene and interact with the door using the specified input method (e.g., mouse click or VR controller trigger press).
   - The door should animate open when triggered by the player input.
3. **Additional Enhancements:**
   - You can add sound effects or particle effects to accompany the door animation to make it more immersive.
   - Consider adding collision detection to prevent the door from opening when obstructed by other objects.
   - Experiment with different animation techniques and interaction methods to create a more engaging experience.

Conclusion:

How do you ensure seamless integration between animations and user interactions within the immersive environment? What Unity components or mechanisms facilitate this interaction?