Use chest

```
using UnityEngine;

public class UseChest : MonoBehaviour
{
    [Header("Chest Interaction Settings")]
    public GameObject handUI; // UI prompt to display when in reach.
    public GameObject objToActivate; // Object to activate when the chest is opened.

    private bool isPlayerInReach = false; // Tracks if the player is within interaction range.
    private Animator chestAnimator; // Reference to the chest's Animator component.
    private BoxCollider chestCollider; // Reference to the chest's BoxCollider component.

    private void Start()
    {
        // Initialize components and set initial states.
        handUI.SetActive(false); // Ensure the hand UI is hidden initially.
        objToActivate.SetActive(false); // Ensure the object to activate is disabled initially.

        chestAnimator = GetComponent<Animator>();
        chestCollider = GetComponent<BoxCollider>();
    }

    private void OnTriggerEnter(Collider other)
    {
        // Detect if the player enters the chest's trigger zone.
        if (other.CompareTag("Reach"))
        {
            isPlayerInReach = true;
            handUI.SetActive(true); // Display the hand UI prompt.
        }
    }

    private void OnTriggerExit(Collider other)
    {
        // Detect if the player exits the chest's trigger zone.
        if (other.CompareTag("Reach"))
        {
            isPlayerInReach = false;
            handUI.SetActive(false); // Hide the hand UI prompt.
        }
    }

    private void Update()
```

```
    {
        // Check if the player is in reach and presses the "Interact" button.
        if (isPlayerInReach && Input.GetButtonDown("Interact"))
        {
            OpenChest();
        }
    }

    private void OpenChest()
    {
        // Hide the hand UI and activate the associated object.
        handUI.SetActive(false);
        objToActivate.SetActive(true);

        // Trigger the chest's open animation and disable its collider.
        if (chestAnimator != null)
        {
            chestAnimator.SetBool("open", true);
        }

        if (chestCollider != null)
        {
            chestCollider.enabled = false;
        }
    }
}
```

Key Pickup

```
using UnityEngine;

public class KeyPickupHandler : MonoBehaviour
{
    public GameObject interactionPrompt; // UI to indicate the player can pick up the key
    public GameObject keyInventorySlot; // Object representing the key in the player's inventory

    private GameObject keyObject; // Reference to the key object
    private bool isWithinRange = false; // Tracks if the player is close enough to interact

    void Start()
    {
        // Disable UI and inventory slot display initially
        interactionPrompt.SetActive(false);
        keyInventorySlot.SetActive(false);
```

```
        // Assign the key object
        keyObject = this.gameObject;
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Reach"))
        {
            isWithinRange = true;
            interactionPrompt.SetActive(true); // Show prompt when player is nearby
        }
    }

    void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("Reach"))
        {
            isWithinRange = false;
            interactionPrompt.SetActive(false); // Hide prompt when player moves away
        }
    }

    void Update()
    {
        // Handle key pickup interaction
        if (isWithinRange && Input.GetButtonDown("Interact"))
        {
            interactionPrompt.SetActive(false); // Hide the interaction prompt
            keyInventorySlot.SetActive(true); // Display the key in the inventory
            keyObject.GetComponent<MeshRenderer>().enabled = false; // Hide the key object
        }
    }
}
```

Main Menu

```
using UnityEngine;

public class LanternPickup : MonoBehaviour
{
    private GameObject currentItem;
```

```
public GameObject handInteractionUI;
public GameObject lanternObject;

private bool isPlayerNearby;

void Start()
{
    currentItem = gameObject;

    handInteractionUI.SetActive(false);
    lanternObject.SetActive(false);
}

private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Reach"))
    {
        isPlayerNearby = true;
        handInteractionUI.SetActive(true);
    }
}

private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Reach"))
    {
        isPlayerNearby = false;
        handInteractionUI.SetActive(false);
    }
}

void Update()
{
    if (isPlayerNearby && Input.GetButtonDown("Interact"))
    {
        HandleLanternPickup();
    }
}

private void HandleLanternPickup()
```

```
    {
        handInteractionUI.SetActive(false);
        lanternObject.SetActive(true);
        StartCoroutine(RemoveItemAfterDelay());
    }

    private IEnumerator RemoveItemAfterDelay()
    {
        yield return new WaitForSeconds(0.01f);
        Destroy(currentItem);
    }
}
```

End game

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour
{
    [Header("Scene Names")]
    public string gameSceneName = "Game"; // Name of the scene to load when starting the
game.

    /// <summary>
    /// Loads the game scene to start the game.
    /// </summary>
    public void B_LoadScene()
    {
        if (!string.IsNullOrEmpty(gameSceneName))
        {
            SceneManager.LoadScene(gameSceneName);
        }
        else
        {
            Debug.LogWarning("Game scene name is not set. Please set it in the inspector.");
        }
    }

    /// <summary>
    /// Quits the application.
    /// </summary>
    public void B_QuitGame()
    {
```

```
        Debug.Log("Quit button pressed. Application will exit.");
        Application.Quit();

#if UNITY_EDITOR
        // Ensures the quit function works during testing in the editor.
        UnityEditor.EditorApplication.isPlaying = false;
#endif
    }
}
```