```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

//Edited version of code from tutorial by VeryHotShark on youtube

public class HoverUIController : MonoBehaviour
{
    [SerializeField] private TextMeshProUGUI hoverText;

    public void SetHoverMessage(string message) {
        hoverText.SetText(message);
    }

    public void ResetUI(){
        hoverText.SetText("");
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

//Edited version of code from tutorial by VeryHotShark on youtube

public class InteractionController : MonoBehaviour
{
    [Header("Data Objects")]
    [SerializeField] private InteractionData interactionData;
    [SerializeField] private InteractionInputData interactionInputData;
    [SerializeField] private HoverUIController hoverUIController;

    [Space]
    [Header("Ray Settings")]
    [SerializeField] private float rayDistance;
    [SerializeField] private float rayRadius;
    //[SerializeField] private LayerMask interactableLayer;

    private Camera p_cam;
    private bool p_interacting;

    void Start(){
        DontDestroyOnLoad(gameObject);
    }

    void Awake() {
        p_cam = FindObjectOfType<Camera>();
    }

    void Update() {
        CheckForInteractable();
        CheckForInteractionInput();
    }

    private void CheckForInteractable(){

        Ray _ray = new Ray(p_cam.transform.position,p_cam.transform.forward);
        RaycastHit hitInfo;

        //look straight ahead (with some wiggle room) for an interactable object

        bool hit = Physics.SphereCast(_ray, rayRadius, out hitInfo, rayDistance);
//, interactableLayer);

        if(hit){
            IInteractable _interactable = hitInfo.transform.GetComponent<IInterac
table>();

            if(_interactable != null && _interactable.IsInteractable){
                if(interactionData.IsEmpty() || !interactionData.IsSameObj(_inter
actable)){
                    //If there is a hit and the object hit is not already in cont
roller, update the item in the controller
                    interactionData.Interactable = _interactable;
                    hoverUIController.SetHoverMessage(interactionData.Interactabl
e.HoverMessage);
                } else {
                    //Ensures that hover message is up to date
                    hoverUIController.SetHoverMessage(interactionData.Interactabl
e.HoverMessage);
                }
            }
        } else {
            //if there is no hit, reset
            interactionData.Reset();
            hoverUIController.ResetUI();
        }
        Debug.DrawRay(_ray.origin,_ray.direction, hit ? Color.green : Color.red);
    }

    private void CheckForInteractionInput(){
        //on press, interact
        if(!interactionData.IsEmpty()){
            if(interactionData.Interactable.IsInteractable){
                if(interactionInputData.InteractPress){
                    interactionData.Interact();
                    interactionData.Reset();
                    hoverUIController.ResetUI();
                }
            }
        }

        interactionInputData.Reset();
    }

    public void UpdateText(){
        hoverUIController.SetHoverMessage(interactionData.Interactable.HoverMessa
ge);
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

//Edited version of code from tutorial by VeryHotShark on youtube

[CreateAssetMenu(fileName = "InteractionData", menuName = "Interactions/Interacti
onData")]

public class InteractionData : ScriptableObject {
    private IInteractable p_interactObj;

    public IInteractable Interactable {
        get => p_interactObj;
        set{p_interactObj = value;}
    }

    public void Interact(){
        p_interactObj.onInteract();
    }
    public bool IsSameObj(IInteractable newObj) => newObj == p_interactObj;
    public bool IsEmpty() => p_interactObj == null;
    public void Reset() => p_interactObj = null;


}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

//Edited version of code from tutorial by VeryHotShark on youtube

[CreateAssetMenu(fileName = "InteractionInputData", menuName = "Interactions/Inpu
tData")]
public class InteractionInputData : ScriptableObject
{
    public bool p_interactPress;

    public bool InteractPress {
        get => p_interactPress;
        set => p_interactPress = value;
    }

    public void Reset(){
        p_interactPress = false;
    }
}
```