하늘이 시간에 따라 움직이는 스크립트

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

public class SkyControl : MonoBehaviour
{    void Update()
    {
        RenderSettings.skybox.SetFloat("_Rotation", Time.time * 0.5f);
    }
}
```

키에 할당하는 기능 스크립트

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class keyController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "cars")
        {
            SceneManager.LoadScene("AnProjectClear");
        }
    }
}
```

총알 스크립트

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

```
public class bulletcontroller : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "animals")
        {
            Destroy(other.gameObject);
            Destroy(gameObject);
        }

    }
}
```

UI 스크립트

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;
using UnityEngine.InputSystem;
using UnityEngine.Events;
using System;

public class UI_InteractionController : MonoBehaviour
{
    [SerializeField]
    GameObject UIController;

    [SerializeField]
    GameObject BaseController;

    [SerializeField]
    InputActionReference inputActionReference_UISwitcher;

    bool isUICanvasActive = false;
```

```
    [SerializeField]
    GameObject UICanvasGameobject;


    private void OnEnable()
    {
        inputActionReference_UISwitcher.action.performed += ActivateUIMode;
    }
    private void OnDisable()
    {
        inputActionReference_UISwitcher.action.performed -= ActivateUIMode;

    }

    private void Start()
    {
        //Deactivating UI Canvas Gameobject by default
        if (UICanvasGameobject !=null)
        {
            UICanvasGameobject.SetActive(false);

        }

        //Deactivating UI Controller by default
        UIController.GetComponent<XRRayInteractor>().enabled = false;
        UIController.GetComponent<XRInteractorLineVisual>().enabled = false;
    }

    /// <summary>
    /// This method is called when the player presses UI Switcher Button which is the input action defined in Default Input Actions.
    /// When it is called, UI interaction mode is switched on and off according to the previous state of the UI Canvas.
    /// </summary>
    /// <param name="obj"></param>
    private void ActivateUIMode(InputAction.CallbackContext obj)
    {
        if (!isUICanvasActive)
        {
            isUICanvasActive = true;

            //Activating UI Controller by enabling its XR Ray Interactor and XR Interactor Line Visual
            UIController.GetComponent<XRRayInteractor>().enabled = true;
            UIController.GetComponent<XRInteractorLineVisual>().enabled = true;

            //Deactivating Base Controller by disabling its XR Direct Interactor
            BaseController.GetComponent<XRDirectInteractor>().enabled = false;
```

```
            //Activating the UI Canvas Gameobject
            UICanvasGameobject.SetActive(true);
        }
        else
        {
            isUICanvasActive = false;

            //De-Activating UI Controller by enabling its XR Ray Interactor and XR Interactor Line Visual
            UIController.GetComponent<XRRayInteractor>().enabled = false;
            UIController.GetComponent<XRInteractorLineVisual>().enabled = false;

            //Activating Base Controller by disabling its XR Direct Interactor
            BaseController.GetComponent<XRDirectInteractor>().enabled = true;

            //De-Activating the UI Canvas Gameobject
            UICanvasGameobject.SetActive(false);
        }

    }
}
```