

Лабораторна робота №8

Тема: Створення VR – сцени для Oculus Rift

Мета роботи: набути практичні навички з використання технології віртуальної реальності при створенні ігор

Теоретичні відомості: лек.3 по VR

Віртуальна реальність стає дедалі популярнішою. З'являється нове обладнання для занурення у віртуальний світ, розробляються захоплюючі ігри та програми.

Завдання на лабораторну роботу

1. Створіть VR-застосунок під керуванням ОС Android

Створимо в Unity3D просту сцену для одного з найпопулярніших шоломів віртуальної реальності **Oculus Rift** (рис. 1).



Рис. 1

Спочатку створимо в Unity сцену для 3D (рис. 2).

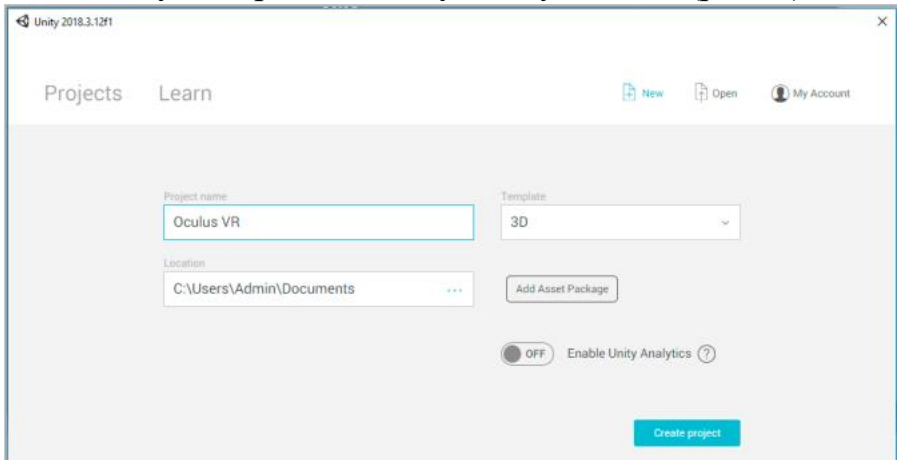


Рис.2

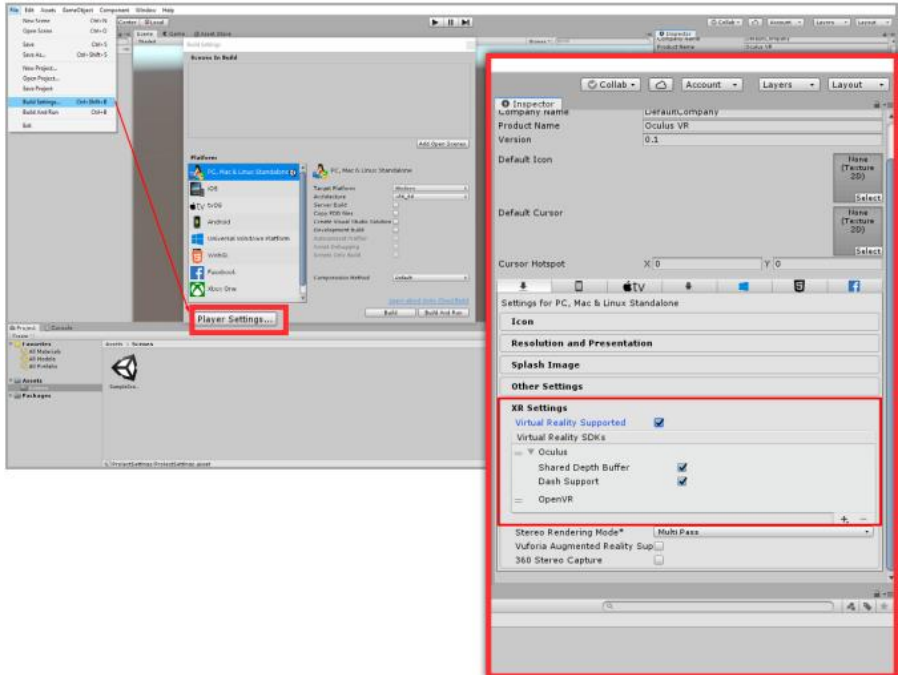


Рис. 3

Підготуємо проект для роботи із віртуальною реальністю. Переходимо **File => Build Setting => Player Settings** і у випадяючому меню **XR** ставимо позначку **Virtual Reality Supported** (рис . 3).

Тепер потрібно імпортувати асет для інтеграції **Oculus Rift** . Ви можете знайти його у папці *Oculus* у додаткових матеріалах. Цей асет можна завантажити з **Unity Asset Store** (рис. 4).

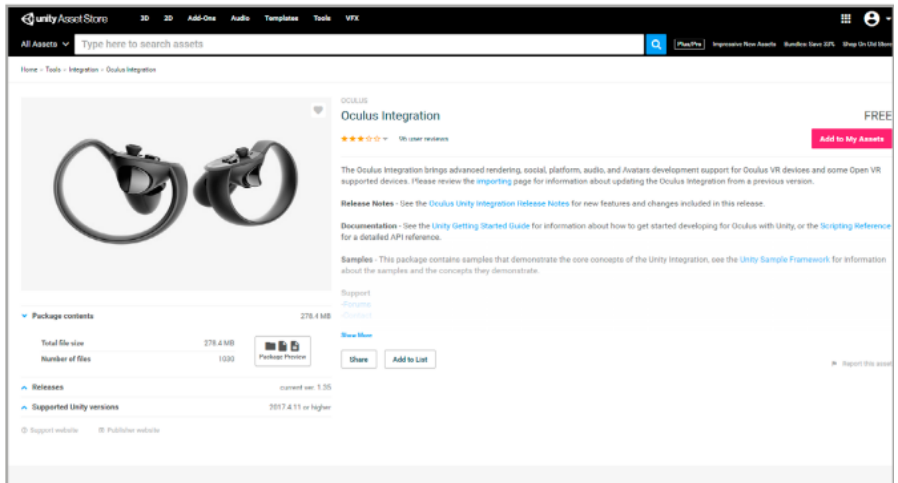


Рис. 4

Рекомендуємо встановлювати **Oculus Integration** з **Asset Store**, оскільки там є актуальна версія асета.

Після встановлення пакета слід імпортувати заготовку для нашої сцени. Вона знаходиться у папці **VRBasics** у додаткових матеріалах. Перейдіть **Assets => VRBasics => Examples => Scenes** і відкрийте сцену **demoStation**. Виглядає вона так (рис. 5):

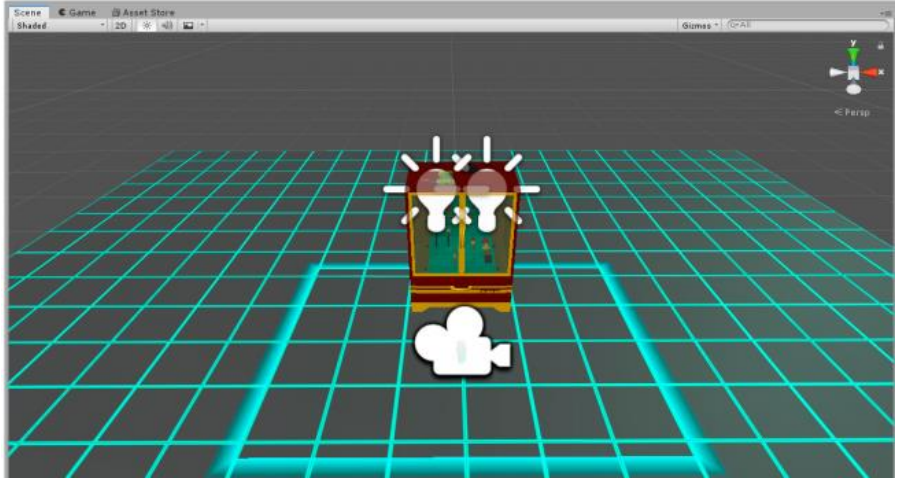


Рис. 5

На сцені знаходиться шафа з механічними перемикачами, кнопками та важелями. До них ми прив'яжемо різні події, які змінюватимуть світ навколо гравця (рис. 6).



Рис. 6

Давайте зробимо так, щоб після натискання на червону кнопку **skybox** день змінювався на ніч. Для цього нам знадобиться вміст папки **Day-Night Skyboxes** із додаткових матеріалів.

Тепер знайдіть у інспекторі червону кнопку. Нас цікавить її об'єкт **slider** (рис. 7).

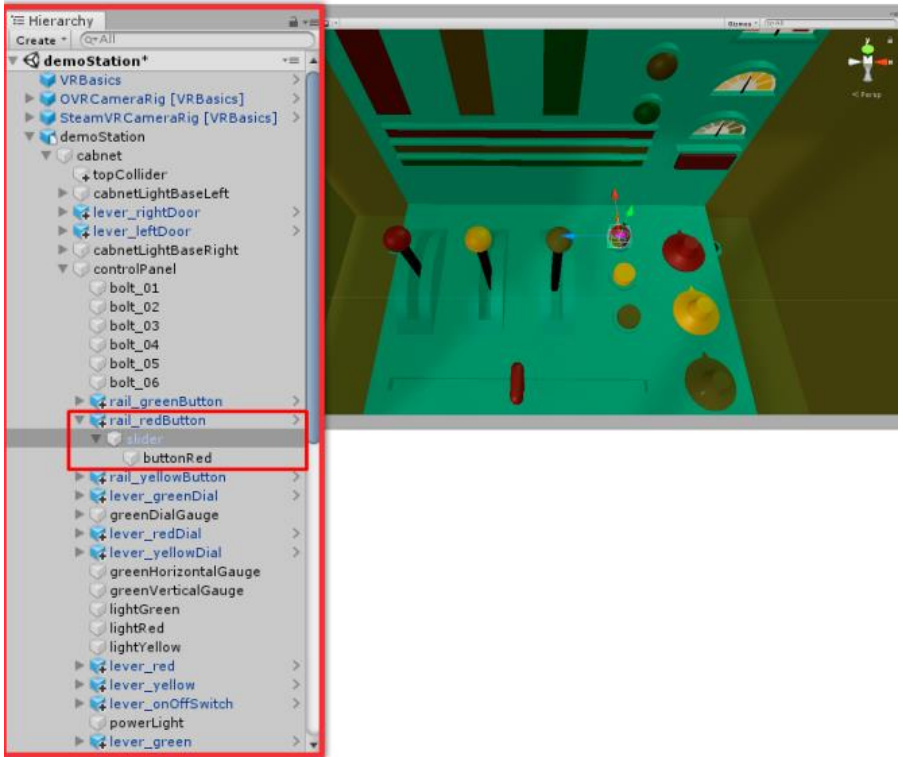


Рис. 7

У нього в компонентах є скрипт **VRBasics_Slider**. У ньому міститься значення **Position**, яке перевіряє, чи

кнопка натиснута. Якщо кнопка не натиснута, це значення дорівнює **0**, і якщо натиснута, то — **1** (рис. 8).

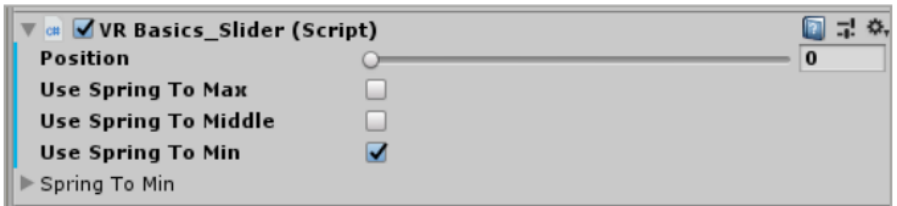


Рис. 8

Давайте прив'яжемо до об'єкта **slider** скрипт, який перевірятиме, чи кнопка натиснута, і змінювати **skybox** у разі її натискання. Скрипт *RedButton.cs* можна знайти у папці з додатковими матеріалами. Імпортуйте його в проект і перетягніть на **slider**.

Скрипт містить наступний код:

```

using UnityEditor;
using UnityEngine;

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

    }

    // Update is called once per frame
    void Update()
    {
        if(this.GetComponent<VRBasics_Slider>().
            position >= 1)
        {
            RenderSettings.skybox =
                (Material)AssetDatabase.
                LoadAssetAtPath("Assets/Day-Night
                Skyboxes/Materials/SkyNight.mat",
                typeof(Material));
        }
    }
}

```

У функції **Update()** ми перевіряємо, чи кнопка натиснута. У разі натискання ми змінюємо **skybox** на **SkyNight.mat** . Якщо хочете змінити його на інший матеріал, просто замініть шлях до нього. Найпростіший спосіб отримати шлях файлу в Unity – натиснути правою кнопкою на потрібний об'єкт у редакторі та вибрати **Copy Path** (рис. 9).

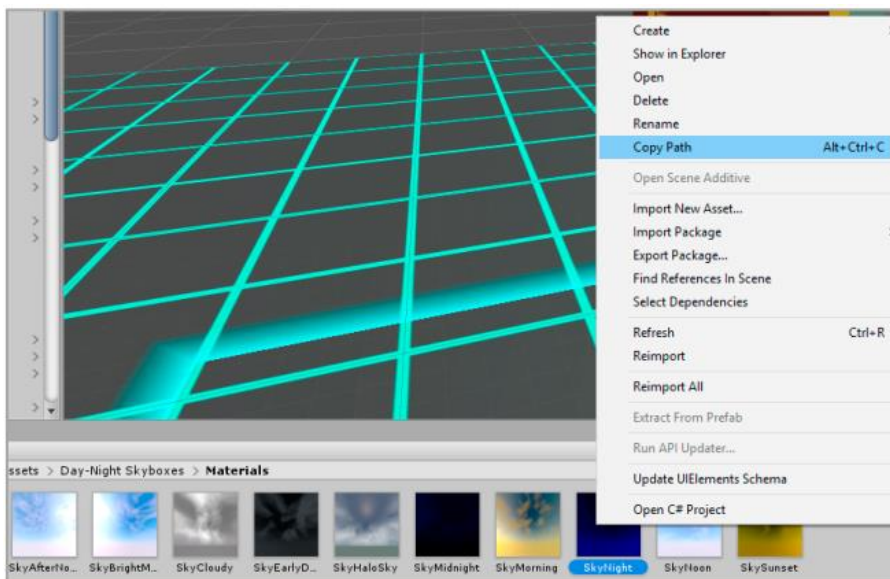


Рис. 9

Давайте спробуємо запустити гру і натиснути на червону кнопку, щоб перевірити спрацювання натискання (Рис. 10).

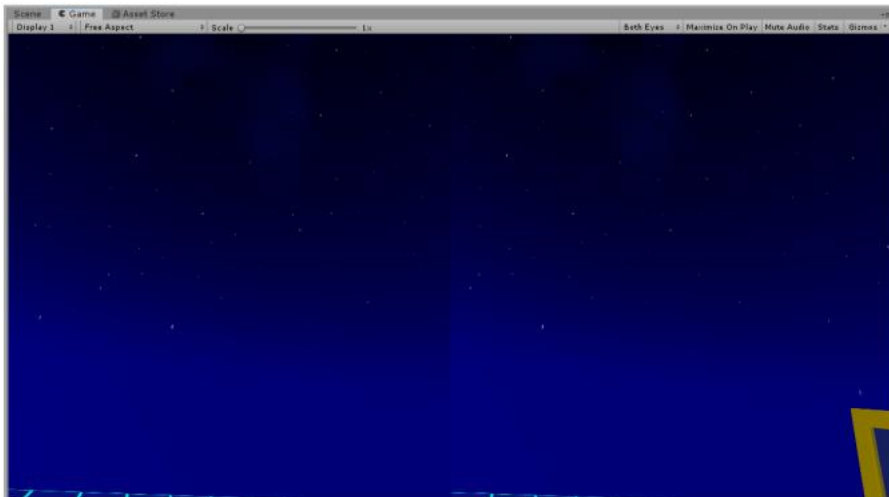


Рис. 10

Ви також можете змінити розташування самої шафи в залежності від вашого зростання, щоб було зручніше використовувати пульт (рис. 11).

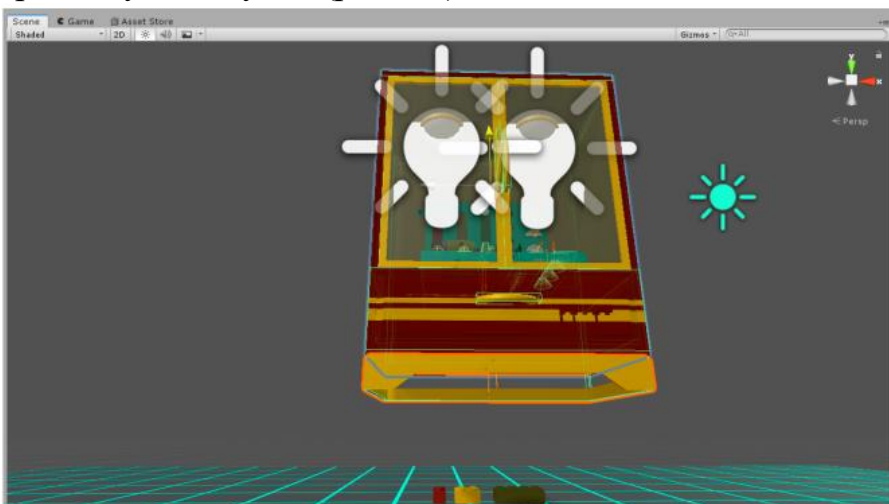


Рис. 11

Тепер прив'яжемо аналогічні скрипти до зеленої та жовтої кнопок. Скрипти *GreenButton.cs* та *YellowButton.cs* також можна знайти в папці з додатковими матеріалами. Відрізняються вони лише рядком із присвоєнням матеріалу skybox. Для жовтої кнопки:

```
RenderSettings.skybox = (Material)AssetDatabase.  
LoadAssetAtPath("Assets/Day-Night Skyboxes/  
Materials/SkyHaloSky.mat", typeof(Material));
```

Для зеленої кнопки:

```
RenderSettings.skybox = (Material)AssetDatabase.  
LoadAssetAtPath("Assets/Day-Night Skyboxes/  
Materials/SkyNoon.mat", typeof(Material));
```

Погляньте, як змінилося оточення при натисканні жовтої кнопки (рис. 12):

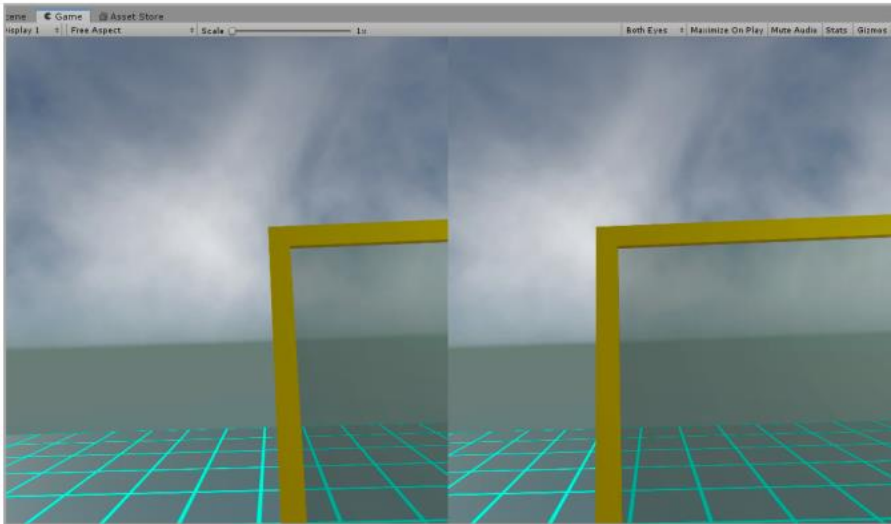


Рис. 12

А ось як змінилося оточення при натисканні зеленої (рис. 13):

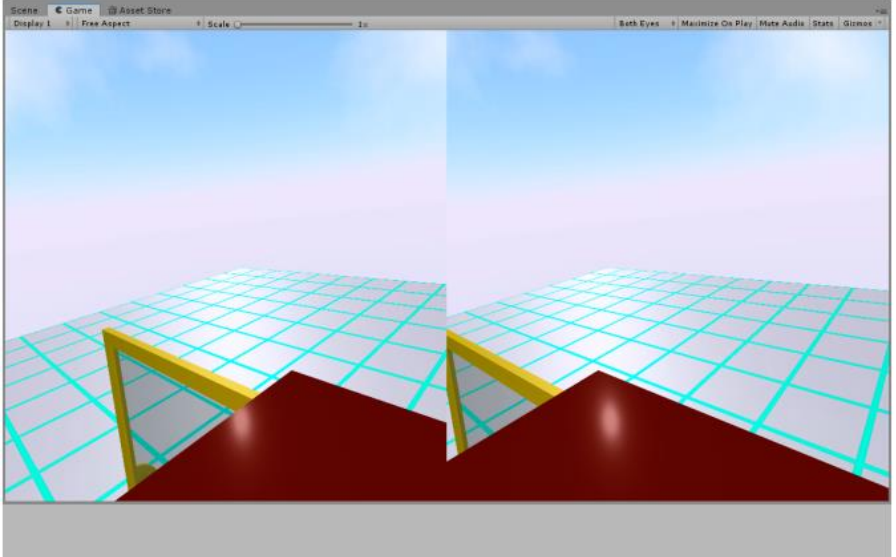


Рис. 13

Добре розширимо функціонал пульта, додаючи на сцену різні об'єкти при перемиканні важелів. Нехай це будуть дерева, ви зможете знайти дані об'єкти у додаткових матеріалах у папці **Trees**.

Додамо на сцену три **GameObject** і помістимо в кожен з них різні дерева. Залежно від кольору використаного важеля ми мінятимемо додані на сцену дерева.

Наприклад, об'єкт **RedTrees** (рис. 14).

За замовчуванням кожен об'єкт буде вимкнено в інспекторі (рис. 15).

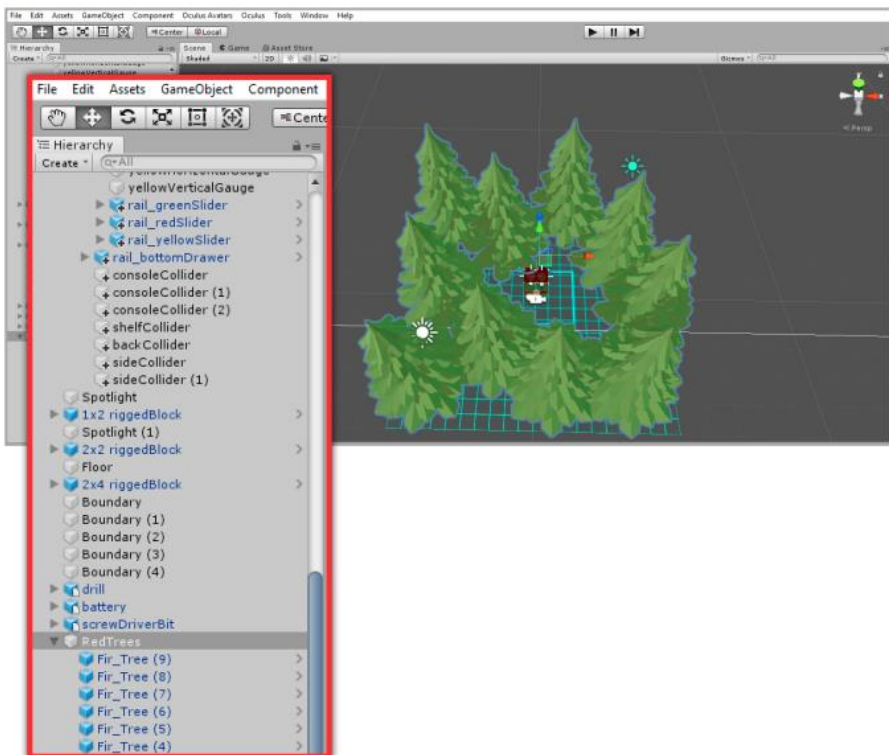


Рис.14.

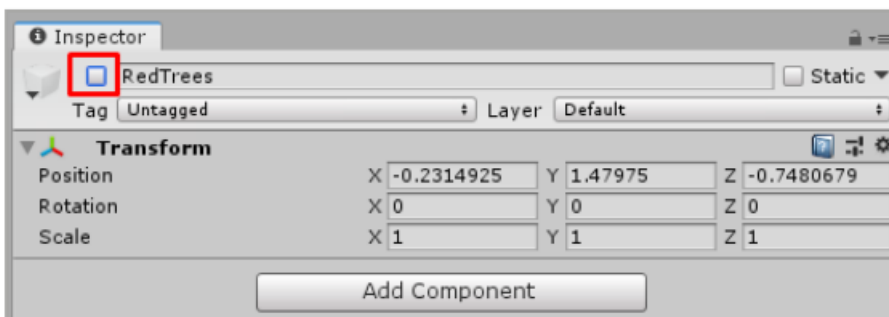


Рис. 15

Об'єкт **YellowTrees** виглядає так
(Рис. 16).

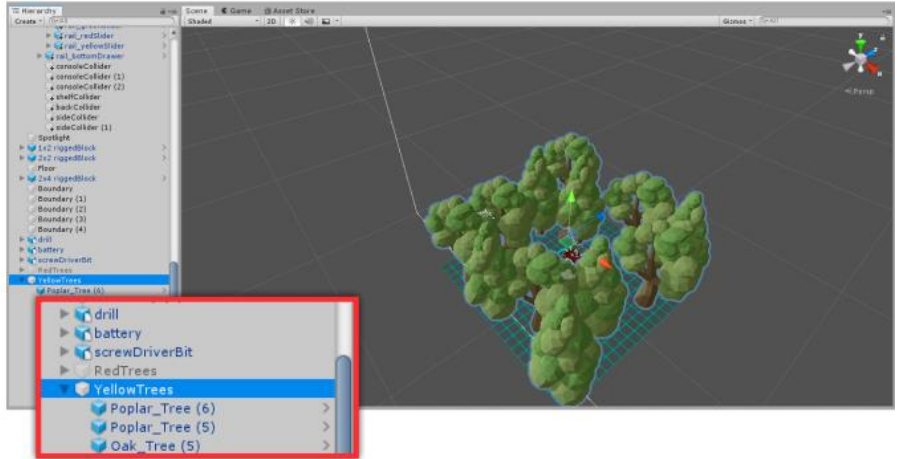


Рис. 16

І, нарешті, об'єкт **GreenTrees** (рис. 17):

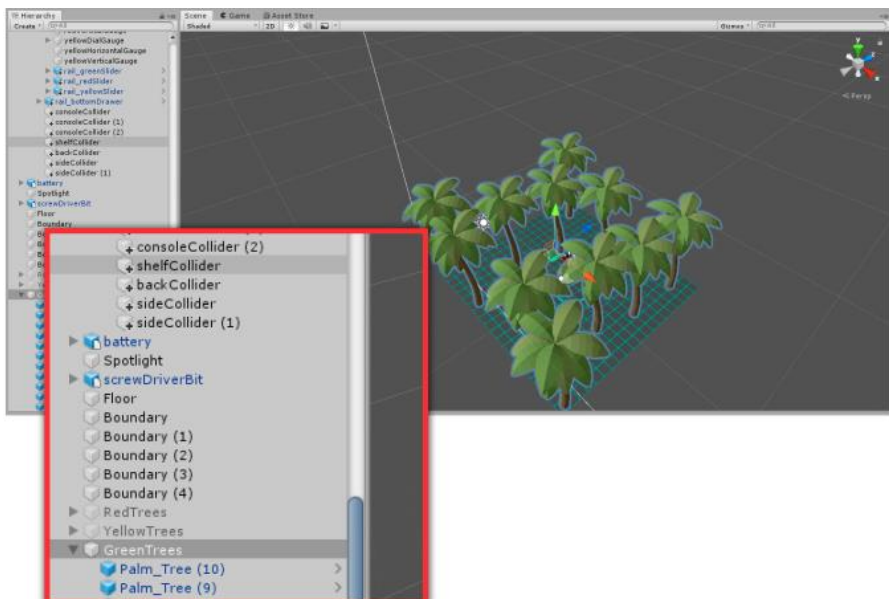


Рис. 17

Знайдіть на сцені будь-який із трьох важелів. Важелі містять об'єкт **Hinge** , який потрібно змінити. Для кожного важеля необхідно встановити значення скрипта **VRBasic_Hinge** у об'єкта **Hinge** , як у малюнку 18.

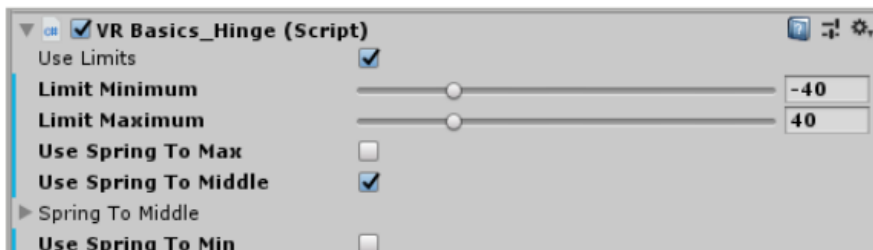


Рис. 18

Тепер важіль повертатиметься до середини. Перейдемо до скрипту. Як бачимо, важіль має змінну

angle , яка відповідає куту його нахилу. У папці з додатковими матеріалами розташовані скрипки.

ти *RedHinge.cs* , *YellowHinge.cs* та *GreenHinge.cs* .

Розглянемо скрипт для червоного важеля.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class RedHinge : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        if (this.GetComponent<VRBasics_Hinge>().angle >= 40)
        {
```



```

        FindInActiveObjectByName("GreenTrees").
            SetActive(false);
        FindInActiveObjectByName("YellowTrees").
            SetActive(false);
        FindInActiveObjectByName("RedTrees").
            SetActive(true);
    }
}
GameObject FindInActiveObjectByName(string name)
{
    Transform[] objs = Resources.
        FindObjectsOfTypeAll<Transform>()
        as Transform[];
    for (int i = 0; i < objs.Length; i++)
    {
        if (objs[i].hideFlags == HideFlags.None)
        {
            if (objs[i].name == name)
            {
                return objs[i].gameObject;
            }
        }
    }
    return null;
}
}

```

У **Update()** ми перевіряємо кут нахилу важеля. Якщо він дорівнює **40** , то за допомогою функції **FindInActive ObjectBy Name()** ми робимо неактивними об'єкти з деревами для зеленого та жовтого важеля, але активуємо об'єкт із деревами для червоного.

Скрипти для жовтого та зеленого важелів аналогічні. Їхня відмінність тільки в рядках, які визначають, який об'єкт буде видимим.

Давайте прив'яжемо всі три скрипти до відповідних **Hinge** (рис. 19):



Рис. 19

Таким чином, смикнувши за червоний важіль, на локації з'являться відповідні дерева (рис. 20):

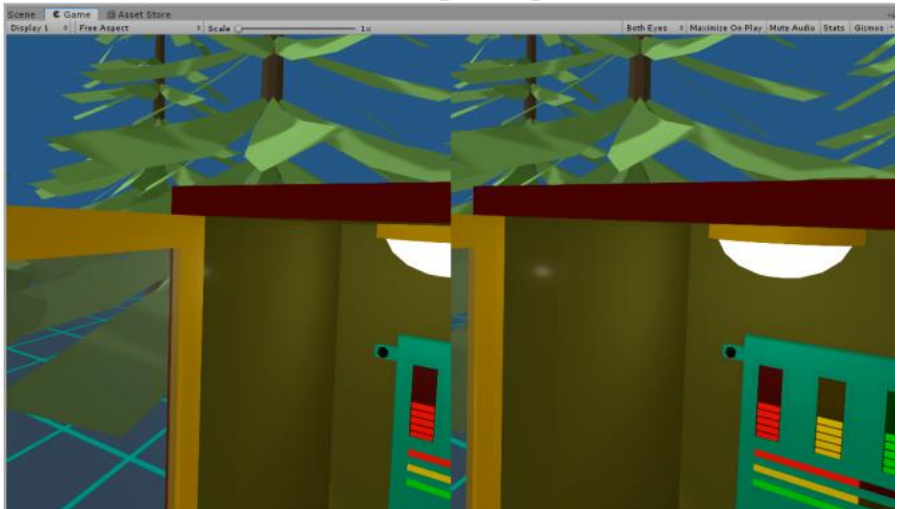


Рис. 20

Залишилося додати можливість перезапускати сцену після натискання на чорний перемикач. Знайдіть на пульті керування відповідний об'єкт **Hinge** (рис. 21).

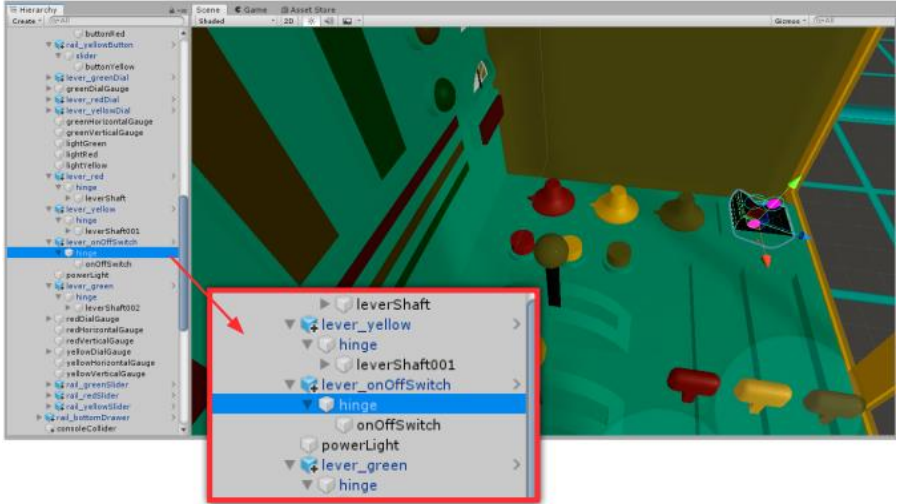


Рис. 21

При натисканні значення **angel** цього перемикача дорівнює **-34** . Імпортуємо в проект скрипт **Black Hinge** та прив'яжемо його до перемикача. Скрипт перевірятиме значення та перезапускатиме рівень, якщо **angel = -34** .

Робиться це за допомогою наступного коду:

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

public class BlackHinge : MonoBehaviour
{
```

```
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if (this.GetComponent<VRBasics_Hinge>().angle == -34)
            SceneManager.LoadScene(0);
    }
}
```

Таким чином, після натискання на чорний перемикач, сцена спорожніє (рис. 22):

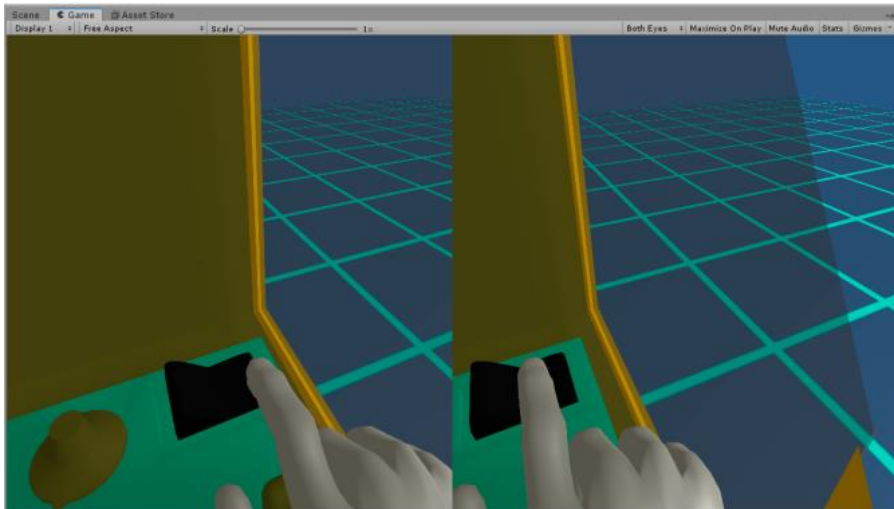


Рис. 22

Відмінно наша VR-сцена для **Oculus Rift** готова!

2. Закомітити файли та проект на гіт. Надайти доступ maintainer для @graf_ztu.