CSE 566 Virtual Reality, Spring 2020, Assignment 1: Basic VR

Tianao Wang 112819772

February 18, 2020

1 Google Drive Link

Google Drive Link

2 Background

(1) Unity version:
2018.4.14f1

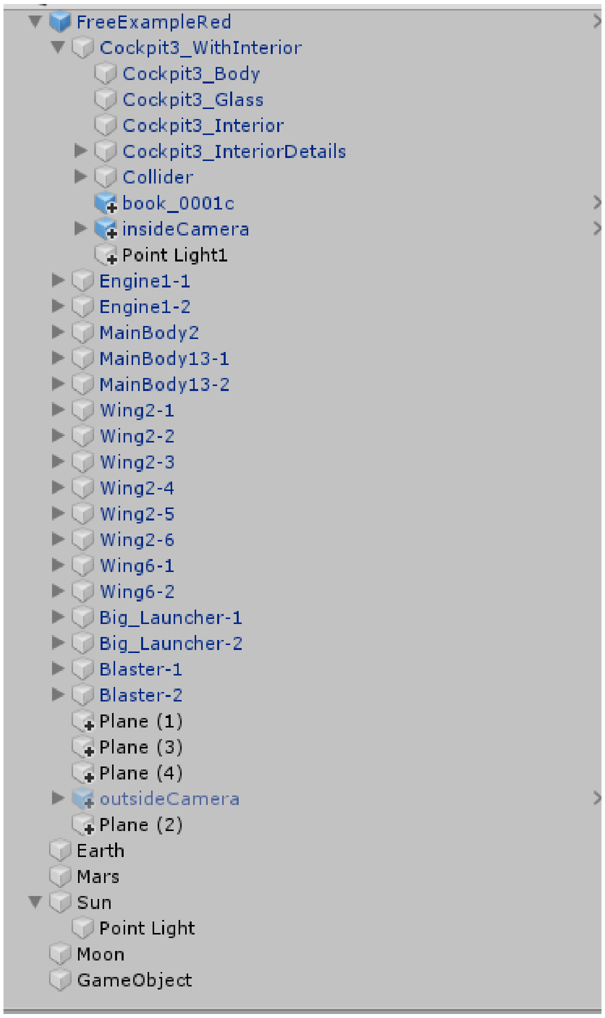
(2) Hardware used:
Oculus

(3) Directory hierarchy:
FreeExampleRed:The spaceship. There is a OVRCameraRig camera, a book and a point light in the spaceship's cockpit. Also, it has a OVRCameraRig camera outside the spaceship to get the outside scene. It orbits around the earth and the Sun. User can switch the camera by using the button on the Oculus controll. The light will be on when it is not illuminated by the Sun. There is also a switch on the Oculus controll to control the gravity of the spaceship. If the gravity is on, the book will fall into the ground. If the gravity is off, the book will gain a force from the Earth and the Sun, it will start floating into the air.

Earth:The earth. It orbits around the sun. It also has rotation.

Moon:The moon. It orbits around the earth and the sun. It also has rotation.

Sun:The Sun. There is a point light in sun to create light. It also has rotation.

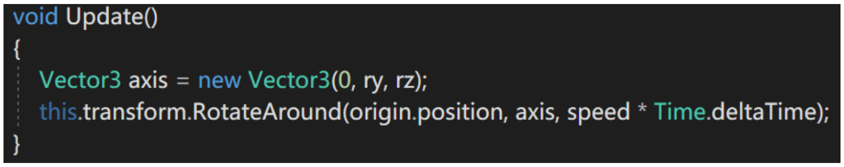


```
FreeExampleRed
  Cockpit3_WithInterior
    Cockpit3_Body
    Cockpit3_Glass
    Cockpit3_Interior
    Cockpit3_InteriorDetails
    Collider
    book_0001c
    insideCamera
    Point Light1
  Engine1-1
  Engine1-2
  MainBody2
  MainBody13-1
  MainBody13-2
  Wing2-1
  Wing2-2
  Wing2-3
  Wing2-4
  Wing2-5
  Wing2-6
  Wing6-1
  Wing6-2
  Big_Launcher-1
  Big_Launcher-2
  Blaster-1
  Blaster-2
  Plane (1)
  Plane (3)
  Plane (4)
  outsideCamera
  Plane (2)
Earth
Mars
Sun
  Point Light
Moon
GameObject
```

3 Implement

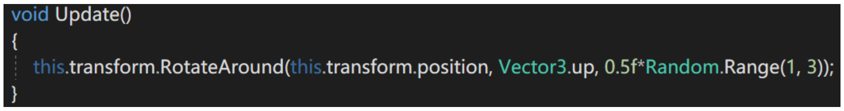
3.1 Scene

(1) Revolution. For this scene, I implement the orbit around function and rotation function. The orbit around function can make a object rotate around another object. I give a center, axis and speed to the outer object and use the RotateAround function to let it rotate around the center.



```
void Update()
{
    Vector3 axis = new Vector3(0, ry, rz);
    this.transform.RotateAround(origin.position, axis, speed * Time.deltaTime);
}
```

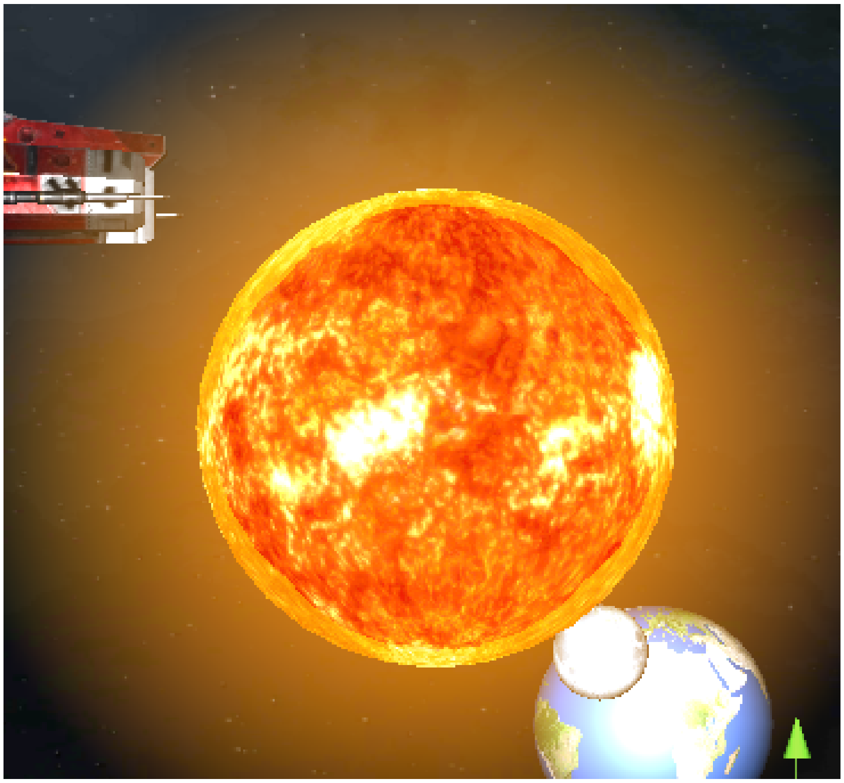
(2) Rotation. All the stars should be able to rotate by themselves. We just need to change a little bit code from Revolution by setting the axis as Vector3.up and setting the center as the object itself.



```
void Update()
{
    this.transform.RotateAround(this.transform.position, Vector3.up, 0.5f*Random.Range(1, 3));
}
```

3.2 Light

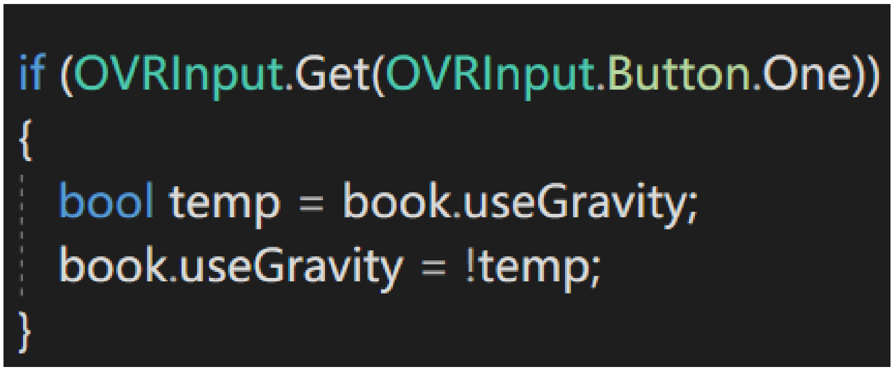
(1) The Sun has a point light. To make it more natural, I change the shader of the Sun to self-illumin and add a Halo to it.



(2) There is also a light in the spaceship. It will use the Vector3.Distance() function to compare the distance from the Sun to the Earth and the Sun to the spaceship. If the spaceship is not illuminated by the Sun. The light in the spaceship will turn on, otherwise it will turn off.

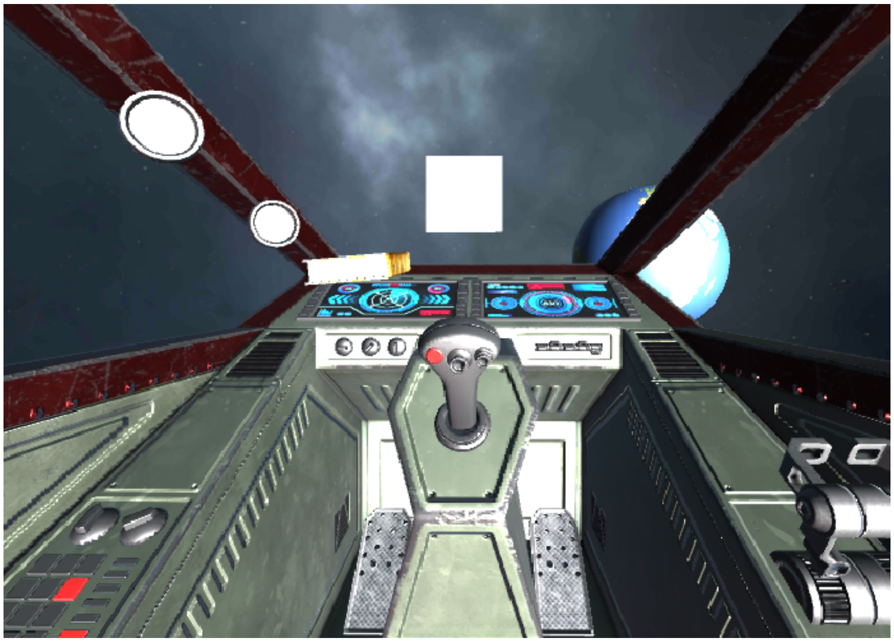
3.3 Gravity

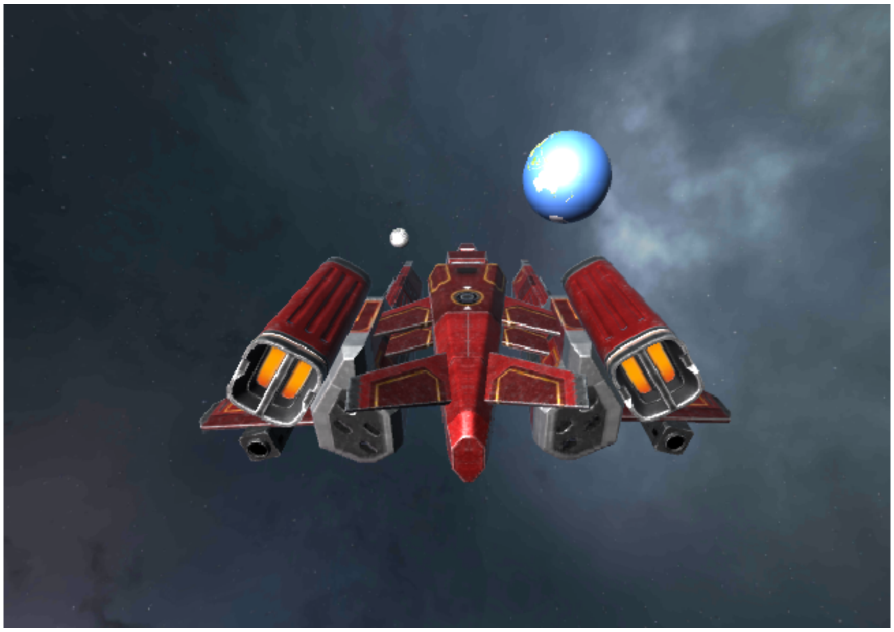
I add a book into the spaceship's cockpit. It has rigidbody and mesh collider components and select convex. I also add all the devices in the spaceship rigidbody and mesh collider components and select is Kinematic. This will make sure the book will not across the devices in the spaceship. There is some problems with the glass of the spaceship because the three pieces of glass are in total which means they can not be divided. However, unity can not dectect the collide happened inside the object. So I add 4 air walls to make sure the book will not float outside the cockpit. After all of these, I just need to use a button of the Oculus controller to change the attribute——use gravity. When it has gravity, it will fall down. In addition, when it doesn't have gravity, the Earth and the Sun will give it a force to make it float. The core code is shown below.



```
if (OVRInput.Get(OVRInput.Button.One))
{
    bool temp = book.useGravity;
    book.useGravity = !temp;
}
```

3.4 Camera





There are two cameras, one is inside and the other one is outside. User can switch the camera using the button of the Oculus controller. I met serveral problems when I tried to implement this function. I have tested the following ways to switch the cameras and finally I select one which looks best.

(1) Directly change two cameras. I use two OVRCamera and this works well when I test using my PC and keyboard. However, when I compile the program into Oculus helmet. The changing camera functions doesn't work well. The camera does change but it flicker back.

(2) Using two buttons. I also tried to use two buttons to controll the camera. I can successfully switch outside. However, I can't switch back into the spaceship.

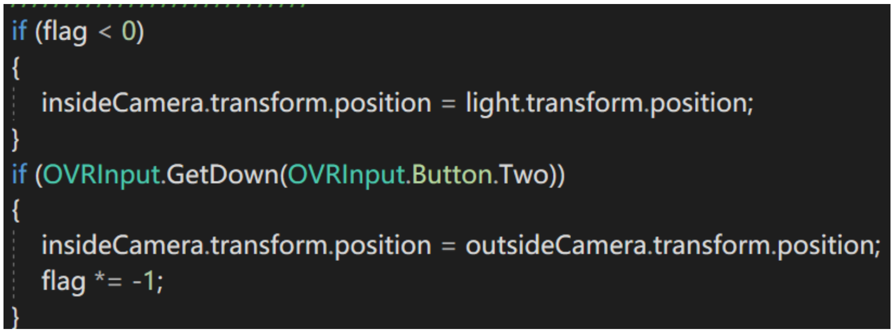
(3) Change the update function frequency. The update function will be called each frame. I change the detect frequency but it also doesn't work.

(4) Change the sensitivity of the controller. It may caused because the sensitivity of the controller is too high, so I decrease the value of sensitivity. However, I still can't fix it.

(5) I triede IEnumerator function to sleep for a while, but the camera still switch back after the sleep.

(6) Directly change the position of camera. Finally, I choose the solution which looks best. Directly record the positions of the two cameras and change the position of the working camera. By doing this, I can switch the camera successfully.

The core code is shown below.



```
if (flag < 0)
{
    insideCamera.transform.position = light.transform.position;
}
if (OVRInput.GetDown(OVRInput.Button.Two))
{
    insideCamera.transform.position = outsideCamera.transform.position;
    flag *= -1;
}
```

4 Reference

(1) Starfield Skybox,PULSAR BYTES, https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717

(2) SolarSystemPackage,Simon István https://assetstore.unity.com/packages/essentials/legacy-image-effects-83913

(3) HiRezSpaceshipsCreatorFree,Ebal Studios,https://assetstore.unity.com/packages/3d/vehicles/space/hi-rez-spaceships-creator-free-sample-153363

(4) Oculus Integration,Oculus,https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022

(5) Books,VIS Games4,https://assetstore.unity.com/packages/3d/props/interior/books-3356