CSE 566 Virtual Reality Spring 2020

Assignment 2: UI Interaction

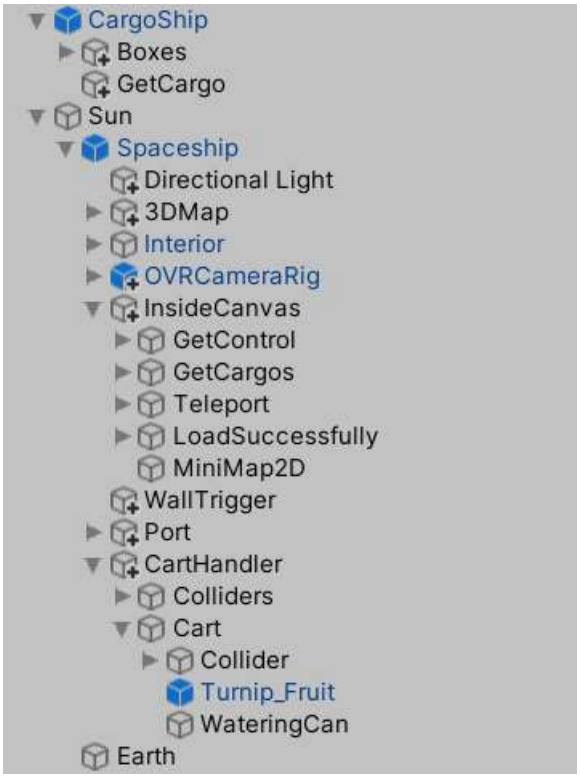
https://drive.google.com/open?id=1tEjxLptKhHDiHEd65QAikn3a3r8y9ycy

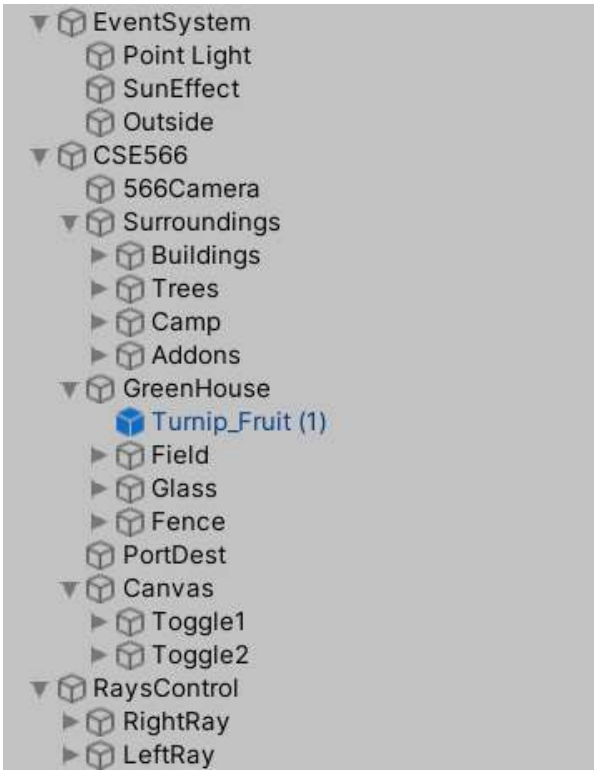
- **Name and Stony Brook ID:** Yao Li 112715069
- **Unity version:** Unity 2019.3.0f6 (64-bit)
- **Hardware used:**

  PC: Windows 10; Inter(R)Core(TM)i7-9750H @2.60GHz

  Device: Oculus Quest

- **Directory hierarchy**





The most basic objects in this scene is the Sun, the Earth, the spaceship, the cargo ship, CSE566 planet, Ray Control System and other effects.

- **Details on implementation:**
  - ***Actions***

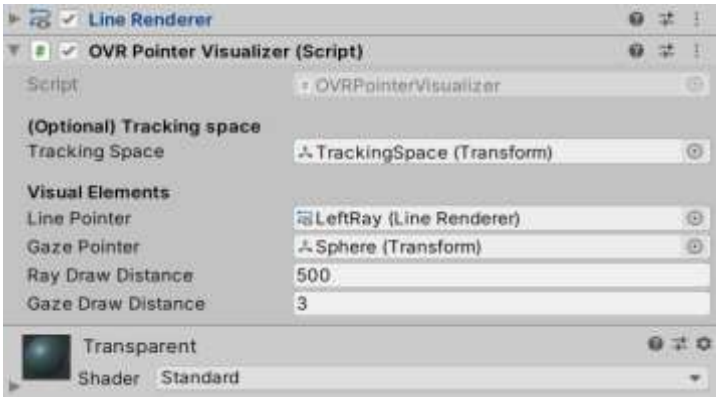
1. Ray Casting

The Oculus VR SDK provides a series of scripts to perform the ray casting: "OVR Pointer Visualizer" helps to set the visual elements by getting the tracking space value. The Line Render component can be used to set attributes like width and color.

"OVR Raycaster" performs a raycast using the worldSpaceRay in eventData.

"OVRRayPointerEventData " implements properties of the pointer, such as the eligibility of click and drag.

One left ray can be set with these scripts. For the right ray, set the right controller as active in "OVR Pointer Visualizer". Now we have one beam of ray from the left and right controller separately.



2. Steering

Steering mode can also be switched by pressing PrimaryHandTrigger button. OVRInput.Controller.LTrackedRemote() is used to set the controller. OVRInput.GetLocalControllerRoatation() is used to get the orientation of the spaceship. Quaternion.Slerp() is used to interpolates rotation between the original position to the real-time controller rotation status.

For the speed control, pressing Button B to move forward and A to accelerate by multiplying the speed value.

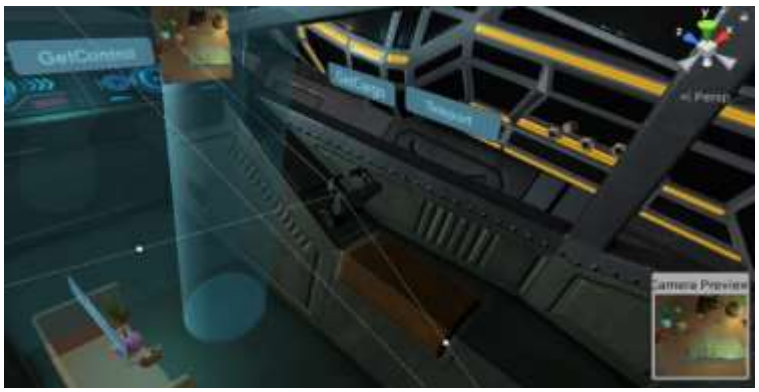
```
Quaternion CtrRotation;
Vector3 OrigPos = transform.position;

SetController();
CtrRotation = GetOrientation();

transform.rotation = Quaternion.Slerp(transform.rotation, CtrRotation, rotation_speed * Time.deltaTime);

transform.Translate(transform.forward * forward_speed * Time.deltaTime);
```

3. Camera Moving

I used a boolean value to turn on/off the moving mode by pressing Primary-HandTrigger button. While moving, the joystick can get the position in X-Z plane, thus the camera can move from front to end int the spaceship.

4. Pulling

Objects can be pulled only when they are pressed by primary index trigger. Then for per frame, the Lefthand Thumbstick can control the position of X and Y Axis, while the Righthand Thumbstick can control the position of Z axis (as we only need three dimensions). Once the trigger is pressed again, the velocity will be set to zero, thus disabling the pulling action.



```
Vector2 leftAxis = OVRInput.Get(OVRInput.RawAxis2D.LThumbstick);
Vector2 rightAxis = OVRInput.Get(OVRInput.RawAxis2D.RThumbstick);


direction.Set(-leftAxis.x, leftAxis.y, -rightAxis.y);
transform.Translate(direction * speed * Time.deltaTime, cam.transform);
```

5. Releasing Cargoes

For the four boxes on the cargo ship, first setting them as rigid body, then adding force on them while pressing the button. With the random direction of the force, the cargoes can be released into the space at certain speed.

6. Teleporting

I use two cylinders to be the transportation tubes. One is inside the spaceship, the other is on the CSE566 planet. Each one is a trigger. When moving the camera into the cylinder range, the player can press the button to teleport from one cylinder to another(and vice versa) by changing the camera position directly. Triggering this teleportation is implemented by putting a cube under the camera and setting it as a trigger.

7. Gardening

Two types of gardening missions are implemented in this assignment. First is harvesting, which means pulling the crop out from the soil. The second one is watering. When pulling the kettle above the target field, detecting the distance between the kettle and the field. If the distance is smaller than a certain value, then the field will change to a darker color, showing that the watering is completed.

```
private void Watering()
{
    float dist = Vector3.Distance(Kettle.transform.position, KettleDestPos.transform.position);
    if(Mathf.Abs(dist) < 3)
    {
        Object.Destroy(TargetField);
    }
}
```

- ***UI Interactions***

1. 3D Map

To construct a 3D map, I duplicate the whole system with their location information and rescale them to put inside the spaceship. After changing the flying speed accordingly, this small version can be served as a dynamic map to navigate, highlight and show the position as well.

2. 2D Map

I set a camera above CSE566 to get a bird view of the planet, then used a rendered texture to project the view to a raw image on canvas, finally put this raw image inside the spaceship to show the 2D map.

3. Buttons

- Get Control / Get Cargo / Teleportation : Each button can be turned on/off by clicking once the button. With the "Event Trigger" component, the function OnButtonClicked() is used to set them active or inactive.

- Load Successfully: Set active and showed when the cargo box is pulled inside.

- Task List: The UI element "toggle" is used to performed as showing task list. Once a task is finished, players can point towards it and press the PrimaryIndexTrigger to present the accomplishment of the task.

- ***Lighting Conditions*:**

  Point lights are set inside the universe and on the CSE566. Also there is directional light set in the spaceship.

  In the small 3D map, there are point lights on planets, spaceship and cargo ship, being highlighted with different colors at a different blinking speed.

- ***References:***

  Skybox: https://assetstore.unity.com/packages/2d/textures-materials/milky-way-skybox-94001

  Planets: https://assetstore.unity.com/packages/3d/environments/planets-of-the-solar- system-3d-90219

  Spaceship: https://assetstore.unity.com/packages/3d/vehicles/space/hi-rez-spaceships-creator-free-sample-153363

  CSE566 : https://assetstore.unity.com/packages/3d/environments/historic/low-poly-middle-ages-village-141446

  Kettle: https://assetstore.unity.com/packages/3d/garden-realistic-tools-68960

  Boxes: https://assetstore.unity.com/packages/3d/props/cardboard-boxes-pack-30695

  Cargo Ship: https://assetstore.unity.com/packages/3d/vehicles/space/oxar-light-freighter-82240

  Crops: https://assetstore.unity.com/packages/3d/vegetation/plants/cartoon-farm-crops-79777