LYROBOTIX

Notice

1.NOLO's coordinate system for position and rotation is different from that of OpenVR. Therefore, we need to adjust as follows. Source code of controller driver is available for your reference.

Adjustment of position data:

```
m_Pose.vecPosition[0] = data.pos[0];
m_Pose.vecPosition[1] = data.pos[1];
m_Pose.vecPosition[2] = -data.pos[2];//Z
```

Adjustment of rotation data:

```
m_Pose.qRotation.w = -data.rot_quat[0];//w
m_Pose.qRotation.x = data.rot_quat[1];
m_Pose.qRotation.y = data.rot_quat[2];
m_Pose.qRotation.z = -data.rot_quat[3];//z
```

2.We need to set up the center of rotation in OpenVR as follows. Source code of controller driver is available for your reference.

```
m_Pose.vecDriverFromHeadTranslation[0] = 0.000f;
m_Pose.vecDriverFromHeadTranslation[1] = 0.007f;
m_Pose.vecDriverFromHeadTranslation[2] = -0.073f
```

3.Under the circumstance of realizing throwing function, we can't keep providing data of velocity and angular velocity to OpenVR all the time, resulting violate vibration of the controller. We can provide speed and acceleration to OpenVR when pressing the trigger, while set the data as 0 when releasing the button. Source code of controller driver is available for your reference.

4.We do not recommend using rotation data of NOLO headset marker but the data of the third-party HMD when it comes to HMD rotation data.

5.This SDK comes with SteamVR Room Setup function. We can place the headset marker on ground and press its pair button. By doing this, we realize the SteamVR Room Setup function. In addition, the first-time SteamVR users need to do SteamVR Room Setup once.

6.Headset driver of SteamVR:

In case of being kicked out by some games:

```
Prop_ModelNumber_String："ViveMV"，
Prop_ManufacturerName_String："HTC"
m_Pose.shouldApplyHeadModel = false;
```

Controller driver of SteamVR:

Choices of controller models:

```
Prop_RenderModelName_String:"vr_controller_vive_1_5"
```

Vibration setting：

```
bool NOLOTrackedDevice::TriggerHapticPulse(uint32_t unAxisId, uint16_t usPulseDurationMicroseconds)
{
	int n = usPulseDurationMicroseconds/40;
	if (n>50)
		n = 50;
	}
	if (m_nId == 0) {
		set_Nolo_TriggerHapticPulse(NoloDeviceType::LeftControllerDevice, 50+n);
	}
	else if (m_nId == 1)
```

```
        {
                set_Nolo_TriggerHapticPulse(NoloDeviceType::RightControllerDevice,50+n);
        }
        return true;
}
```

7:Two ways of getting NOLO data: 1.Get; 2.Registration callback

```
NOLO_API    NoloData _cdecl get_Nolo_NoloData();
```

Or

```
NOLO_API     bool _cdecl registerNoloDataNotifyCallBack(noloDataNotifyCallBack fun , void * context);
```

The following function must be realized when NOLO is connected:

1.The connection condition and the battery status of NOLO device must be demonstrated on your software. Please make it convenient for users to check whether the device is working.

2.Calibrate the position by double clicking the system button(power button) of any controller.In other words, making the controller face directly to the base station and double click the system button on the controller, reset the position on the Yaw axis of the HMD, that is to ensure the positive direction of SteamVR directly faces to NOLO base station.

3.Must realize a rotation of 180 degree by double clicking the menu button of any controller. Source code of controller driver and the NOLO_OSVR_SteamvrDriver is available for your reference.

4.While using your software, please notify user to close the NOLO driver when your software monitored the NOLO driver running.

5.Please do not run this SDK in the background when user closes your software.

```
close_Nolo_Device();
```

6.Example

https://github.com/NOLOVR/NOLO-Windows-SDK/tree/master/Examples