ABA AIOT PVT LTD
Pratik Rohane

Teleoperation with VR Headset

OBJECTIVE

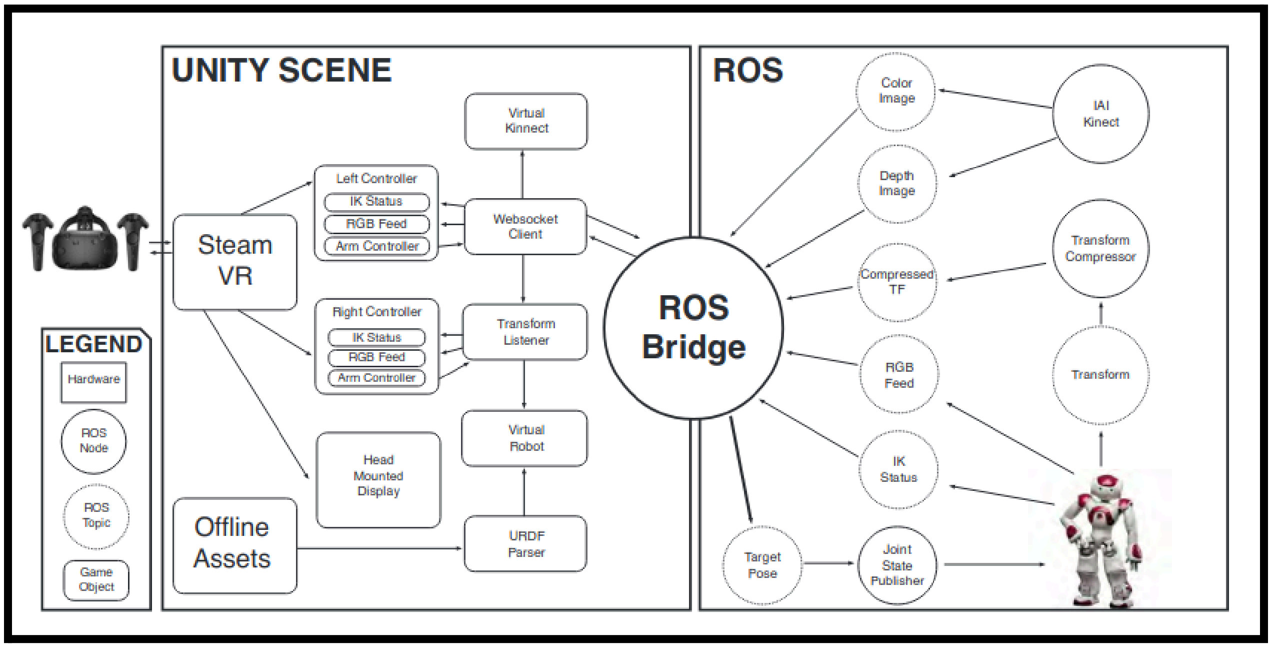
1. Effectively perceive the robot environment
2. Effectively control the robot

ROS Reality

ROS Reality is **an open-source, over-the-Internet teleoperation interface between any ROS-enabled robot** and any Unity-compatible VR headset.

Step1. Configuring the Unity Scene
Step2. Start the ROS node

ROS Reality Architecture



Sensory data (camera) sent from the Robot to user and Movement commands (touch controller) are sent from the user to Robot

ROS Reality Components

1. VR Headset (Unity Support)
2. Unity
3. ROS

Unity

- Android SDK
- Linux Build Support
- Oculus Integration
- XR Plugin Management
- Build Setting
    - Platform→ Android (for oculus quest)
    - Texture Compression: ASTC

ROS

- WebScoket Client
- Transform Listener

How to Set Up ROS and Unity

-Import plugins into Unity Project
-Generate C# code for the ROS msg files

Publishing a message from the Unity to ROS Topic:

-Create instance of ROS message in scripts
-Publish message to ROS topic

Subscribing to a ROS Topic from Unity:

-Extend ROSSubscriber class in scripts
-Override Async function to read and instantiate ROS message

Scene Rendering and Requirement

Required High-performance rendering system, strategy to develop that system is to stream sensor data to a powerful local computer, which builds and displays a local model of the world using Unity.

Virtual Scene in Unity

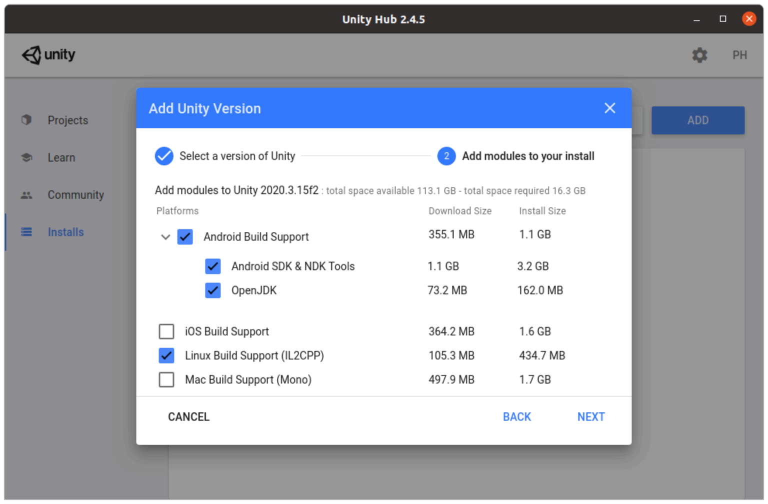
1. 3D model of robot: importing URDF description of robot
2. 3D point-cloud of the scene: Obtained by calibrating Kinect sensor mounted on the robot
3. Display the robot camera: high resolution live image of the environment forward to the robot

Installation

Unity

[Installing Unity Hub]

Install Android SDK and Linux Build Support in Unity



✓ Select a version of Unity — 2 Add modules to your install

Add modules to Unity 2020.3.15f2 : total space available 113.1 GB - total space required 16.3 GB

Platforms	Download Size	Install Size
✓ Android Build Support	355.1 MB	1.1 GB
✓ Android SDK & NDK Tools	1.1 GB	3.2 GB
✓ OpenJDK	73.2 MB	162.0 MB
☐ iOS Build Support	364.2 MB	1.6 GB
✓ Linux Build Support (IL2CPP)	105.3 MB	434.7 MB
☐ Mac Build Support (Mono)	497.9 MB	1.7 GB

WebSocket

Installing Rosbridge

```
sudo apt-get install ros-<rosdistro>-rosbridge-suite
```

Running Rosbridge

```
roslaunch rosbridge_server rosbridge_websocket.launch
```

Enable VR in Unity

[Unity VR Manual]

Transfer the model in Unity from ROS

Create a Launch file which start the file server and load urdf of robot

```
<launch>

   <include file="$(find
file_server)/launch/ros_sharp_communication.launch">
       <arg name="port" value="9090" />
   </include>

   <param command="$(find xacro)/xacro $(find
trolly_description)/urdf/trolly.xacro" name="robot_description"/>

</launch>
```

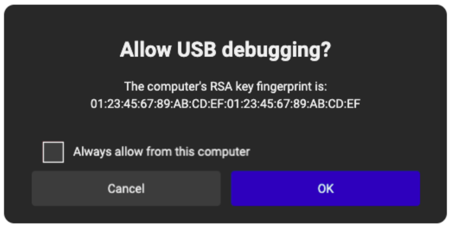
Set up Oculus Rift in Unity

Step1: Turn on Developer Mode of Oculus

Oculus Companion App: Settings > More Settings > Developer Mode

Step2: Connect Oculus Rift with USB

Allow USB Debugging



Allow USB debugging?

The computer's RSA key fingerprint is:
01:23:45:67:89:AB:CD:EF:01:23:45:67:89:AB:CD:EF

☐ Always allow from this computer

Step3: Enable VR support in Unity

Go to Edit > Project Settings > XR Plugin Management

- Install XR Plugin
- Select Oculus

Step4: Import Oculus Integration Plugin in Unity

- Got Window > Asset Store
- Search “Oculus Integration”
- Import “Oculus Integration”

CommandInvokationFailure: Unity Remote requirements check failed

Try these steps in-order:

1. In Unity, go to Edit -> Project Settings -> Editor, and set the "Device" to "Any Android Device"

2. Plug-in your phone.

3. Switch the connection type to USB tethering.

4. Enable developer mode (if already ON then switch it OFF before plugging in your phone)

5. Enable USB debugging (if already ON then switch it OFF before plugging in your phone)

When you turn on USB debugging your phone will ask to allow the connected PC to access the RSA, allow it.

6. Now, switch the connection mode to "USB for Transferring files/Android Auto", the phone will again ask to allow the computer to access the RSA, allow it.

7. In Unity, go to File -> Build Settings, then select your device (which will now be visible in the "Run Device" dropdown).

8. Now, open Unity Remote 5 on your device.

9. Hit play in Unity Editor.

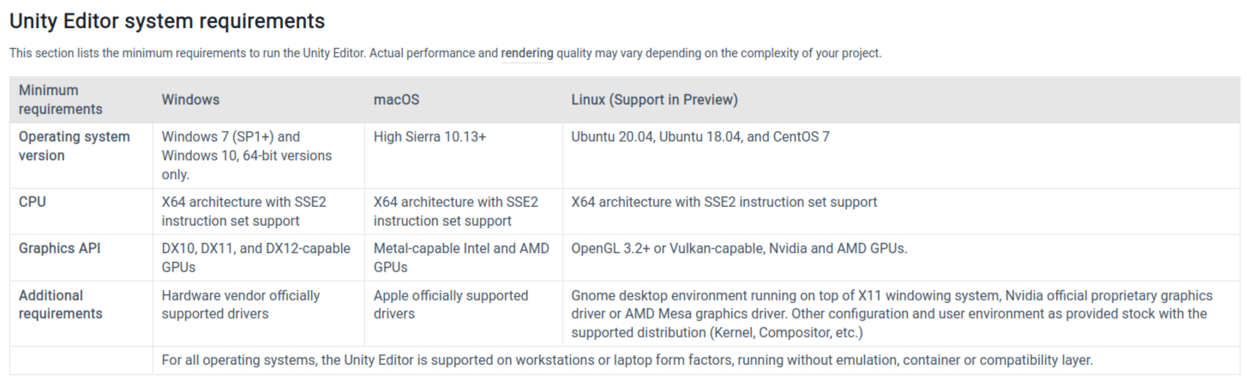
To enable Android

File>>Build Setting >> Android >> Select Android
Edit >> Project Setting >> Editor >>Unity Remote >> Device >> Any Android

System Requirement for Unity
Minimum requirement-

- Intel i5-4590 equivalent or greater.

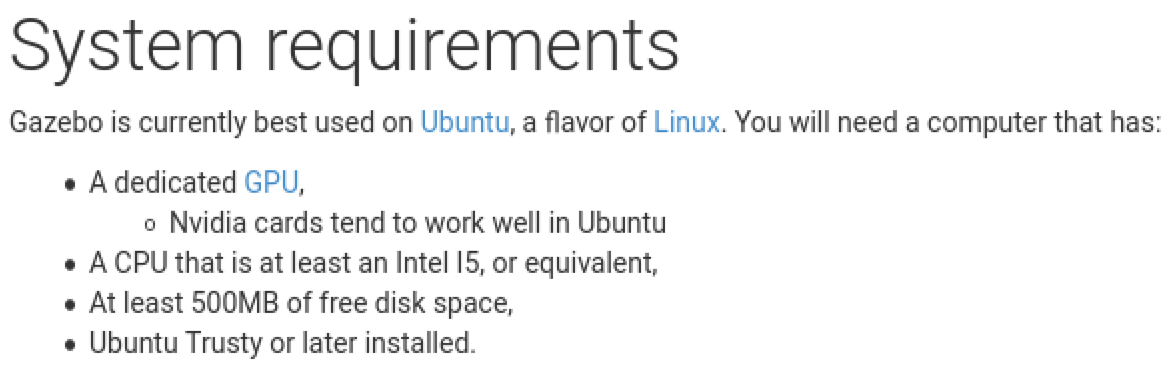
- NVIDIA GTX 970 or AMD R9 290 or greater.
- 8GB RAM or more.
- HDMI 1.3 and 3x USB 3.0 plus 1x USB 2.0



Unity Editor system requirements

This section lists the minimum requirements to run the Unity Editor. Actual performance and **rendering** quality may vary depending on the complexity of your project.

Minimum requirements	Windows	macOS	Linux (Support in Preview)
Operating system version	Windows 7 (SP1+) and Windows 10, 64-bit versions only.	High Sierra 10.13+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, and DX12-capable GPUs	Metal-capable Intel and AMD GPUs	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers	Apple officially supported drivers	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)
	For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.		



System requirements

Gazebo is currently best used on Ubuntu, a flavor of Linux. You will need a computer that has:

- A dedicated GPU,
  - Nvidia cards tend to work well in Ubuntu
- A CPU that is at least an Intel I5, or equivalent,
- At least 500MB of free disk space,
- Ubuntu Trusty or later installed.