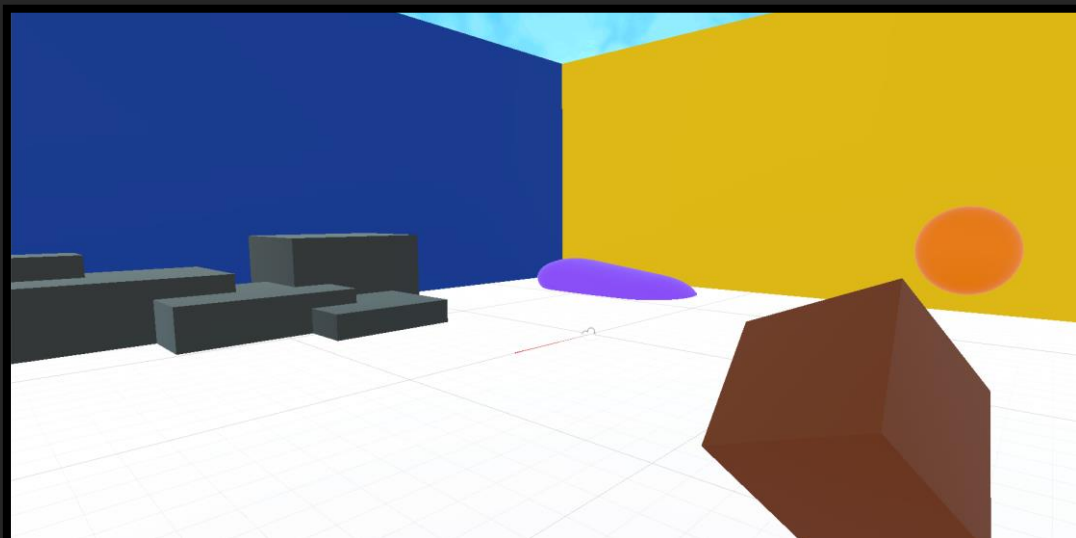


VR Teleport-package - Documentation

Content:

- [Introduction](#)
 - *What's a teleport?*
 - *Why teleport when I can move?*
 - *OK, you sold me on it. How do I use it?*
- [Getting Started](#)
 - *What do I need?*
 - *How to import it*
 - *Prepping the scene*
- [Using it](#)
 - *How to TP*
 - *Cancelling TP*
 - *Custom rotation*
 - *Undo*
 - *Controller visibility*
- [Issues](#)
 - *Solved issues*
 - *Unsolved but known issues*
 - *Unsolved and unknown issues*
- [Patch Notes](#)



Introduction

What's a teleport?

Teleporting is a fictional form of transportation, instantly transporting a person or object to a specific location – mostly used in science fiction media.

Why teleport when I can just move?

In the context of a game or simulation, moving fluidly would feel quite a bit better than a teleport and is more user-friendly, however when talking about VR this story is different.

Moving in a VR-application without moving in real life can make you feel pretty nauseous and disoriented. For this reason, teleporting is the best alternative when it comes to freely moving. Because you don't actually move, your position just changes.

Note: Moving in real-life (and simultaneously moving in VR) is fine so the Teleport-package allows that.

OK, I'm sold. How do I use it?

Well this document is about how to use it and how I built it, so just look at the next few pages, but in short:

- Use the controller's back-trigger to "shoot" at a location and you'll be instantly transported there.
- Move the stick to control your new central-rotation
- Cancel teleportation or undo it with the B-button
- You can't teleport onto walls or steep terrain
- You can still look and move around by using your physical body.

Getting started

What do I need?

- **Unity (newest version)**

Obviously you're going to need Unity, preferably the newest version you can get. The package was made using Unity version 2019.2 but it's probably functional unless major changes happen or Oculus' VR-package changes a lot.

- **Oculus VR Package**

Along with that, you're gonna have to import Oculus' VR-package from Unity's Asset Store (for free).

Note: Oculus' VR-Package is extremely important here, as you won't be able to test the app in VR, let alone use the VR-Teleport package as it's built for it.

- **Teleport VR-Package**

Import this package into Unity as well. It includes a SampleScene that isn't required for the package to work but I recommend using it as a test for getting VR to work.

- **An Oculus Quest unit**

Of course, to test VR, you're going to need a VR device. This package is made specifically with Oculus Quest in mind. I doubt it works with other devices.

- **A right Oculus Quest controller**

This package only has functionality for the right controller. None of the buttons will work but the left controller will register movement.

Prepping the Scene

When you've imported the Oculus Package and the Teleport VR-Package, you'll only need to drag the *TELEPORT_RIG*-prefab into the scene and follow the steps below. Alternatively, you could use the SampleScene provided in the Teleport VR-package.

Prepping the project

Before you're able to test anything, this is what I did to make the project work after setting up the scene. Follow this and you should be fine.

Use CTRL+SHIFT+B to open the Build-settings, Then follow these steps.

- Register the scene in the *build-settings*
- Switch platform to *Android*

Open the Player Settings

- Enter your Company name and Product name

Go to Other settings

- Remove *Vulkan Graphics API*
- Set the *Minimum API Level* to at least 19

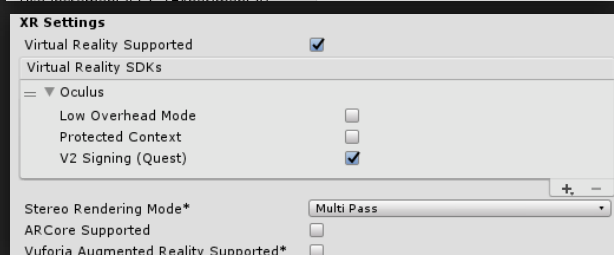
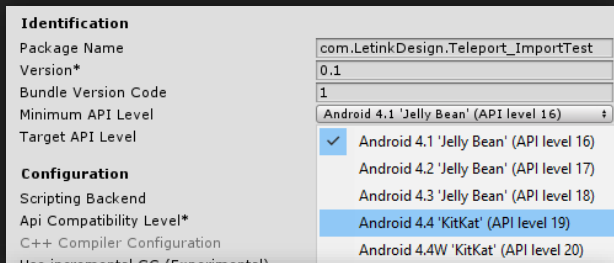
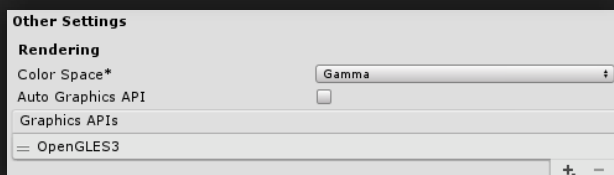
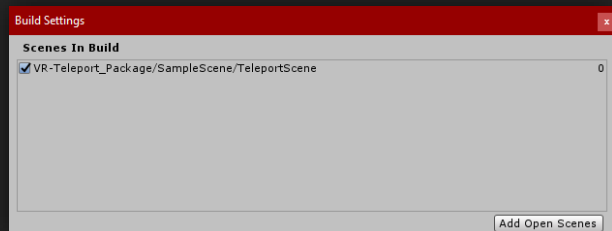
Open XR Settings

- Add the *Oculus SDK*
- Tick the *V2 Signing (Quest)* box

Put on the Oculus Quest

- Make sure that the Oculus Quest is connected to your PC and on.
- Make sure that the Oculus Quest has Development Mode enabled.

Go back to the Build-menu and select Build and Run.



Using it

How to Teleport

By pressing and holding the trigger on the back of the controller, using your index-finger, a line will appear from your controller -- thickening as it goes on.

If this line hits a surface, it should place an object with an arrow pointing forward. This object is called the TPTarget.

TPTarget shows you where you're about to teleport to. By letting go of the trigger you'll instantly be placed there.

The TPTarget, along with the line should be blue, but if it's red, it simply means that you're not allowed to teleport to that location because the floor is too steep. The maximum allowed amount of steepness can be adjusted in the *TP_Core component* (~0.98f is recommended)

Cancelling the Teleport-Laser

If you started pressing and holding the Teleport-trigger, but regret it, you'll be able to cancel the Teleport-Laser by pressing either A or B.

Custom rotation

The arrow in TPTarget is controllable. By holding the right-controller's stick in any direction the arrow will mimic the exact direction and hold it when you let go of the stick.

The arrow shows you what your next central rotation will be after teleporting. *For example: If you hold the stick downwards, you'll be facing the other way after teleporting.*

Undo

Once you've teleported, you might want to go back to a specific point. This is why all teleport-positions (and rotations) are saved and by pressing **B** you can go back 1 step at a time. It works almost exactly like Ctrl+Z in that sense.

Controller visibility

Don't like the controller being in the VR-app as well? Press the Right-Controller's stick to toggle visibility of both controllers.

Issues

Solved:

The TPTarget doesn't show up. What do I Do?

The 2 TPTarget-objects are most likely not assigned (properly). Check if your TP-Handler object has the TP_TargetPlacer object and the 2 TPTargets are both assigned. The 2nd one is the *disallowed* version.

Keep in mind that the TPTargets are not instantiated, which means they should be in the scene. It doesn't matter if they're disabled, though.

The line is red and TPTarget doesn't show up when I hit my floor

Check if the line goes through. If it does, that means that the object you're trying to hit doesn't have a collider or is on a layer not selected in the TP_Laser-component. Adjust either the floor's layer or the layermask.

I'm not allowed to TP to a specific floor (red line and TPTarget).

Chances are your floor's steepness is too high for the current max-steepness value. This value can be changed in the TP_Core component.

If this is not the case, there might be an issue of the object's normal not pointing up. Try it on a plane/terrain. If that does work, it might be an issue with your object.

Unsolved but known Issues

The TPTarget-Disallowed -object isn't properly oriented on walls or slanted terrain.

This is a known issue and I'm having some trouble fixing it, as I don't understand why it does what it does. It's not a big problem, though, as it doesn't break the application. It is slightly aggravating, though.

(Recreate: Try teleporting to different walls)

The TP-Laser doesn't turn blue when moving from steep-terrain to flat terrain. (*Doesn't happen when they're different objects*)

This is a known issue. This happens because the line-color updates every time a new object is hit or when the line starts missing. If you keep pointing at the same object, the line will never update – which means that it won't update even if that one object has both steep and non-steep angles. It doesn't have any impact on gameplay, luckily.

(Recreate: Try teleporting on top of a sphere and move from the outside to the top)

Patch Notes:

+ = Added functionality/bug fixes

* = Change

- = Removal of functionality

V1.0.1.

+Fixed an issue regarding the TELEPORT_RIG -prefab not being imported right, which resulted in the prefab being completely useless.

+Added Patch notes to the documentation. You're looking at them right now. Thanks!

V1.1.0.

+Button mappings are now available in the inspector, allowing you to set any Oculus button for any of the TP functions, however the regular Teleport-function is restricted to 1-Dimensional Axis (like the triggers on the Oculus Quest controller's back and side)

*Some scripts now have a bit more information/documentation in the form of pseudocode.

*The left controller object's MeshRenderer is now disabled by default. It won't activate through scripts – It's GameObject.activeSelf value can still change.

*The orange ball statue in the Sample Scene has moved down a small bit to make it easier to teleport to (from the gray block statue).

+Added some more pictures to the documentation.