Table of Contents

Quick Start

Please note that the code provided in this page is *purely* for learning purposes and is far from perfect. Remember to null-check all responses!

Breaking Changes Notice

If you've just updated the package, it is recommended to check the *changelogs* for information on breaking changes.

Notice

In this page, the fields, properties and methods of each type will not be explained. Every type has been fully documented in code, so please check the code docstrings or reference documentation to learn more about each type.

XRKeyboardManager

Setup

`XRKeyboardManager` is a singleton script to help manage the *Meta Quest Virtual Keyboard*.

First, follow the *getting started guide* to setup the Virtual Keyboard. Then, add an instance of `XRKeyboardManager` to your scene (Hierarchy -> right click -> XR -> XR Keyboard Manager) and fill the required fields. For more information, check the reference documentation for `XRKeyboardManager`.

Usage

To register a listener to the keyboard, call `SetListener(OVRVirtualKeyboard.ITextHandler listener)` with a reference to the `ITextHandler` listener. This will also cause the keyboard to appear.

To hide the keyboard, call `RemoveListener(OVRVirtualKeyboard.ITextHandler listener)` with a reference to the current listener. Ideally this should only be called by the listener itself.

`XRKeyboardManager` also contains some `UnityEvent` callbacks for when the keyboard is toggled. This can be, for example, used to toggle the visibility of controller visuals when the keyboard is toggled.

Input Field

`TextInputField` is a TextMeshPro-based input field that works with `XRKeyboardManager`.

Usage

Add an instance of the input field to a canvas *configured to work with Interaction SDK* by right-clicking on the hierarchy -> UI -> "XR Input Field".

You can get the content from the input field by either registering a callback to the `OnFieldSelected UnityEvent` or by directly accessing the `Text` property.

Voice Input

By adding the `TextInputFieldVoiceHandler` component to a `TextInputField`, you can have voice input for the field, toggled by a button. There is a readymade prefab with the voice input setup which can be created by right-clicking on the hierarchy -> UI -> "XR Input Field with Voice Input".

Make sure to setup *Meta's Voice SDK* and add a dictation service to the scene with the input field. The language used in the dictation service is used by the voice handler. Make sure the font you use supports the voice input language!

Camera Follower

`SimpleCameraFollower` is a script that makes the attached `GameObject` follow the player's camera with:

- Optional Y and Z positional offsets
- Configurable movement and rotation delays and
- Variable movement and rotation speeds

This script does not require much more configuration and can be used as-is, without additional setup.

Movement Controller

`SimpleMovementController` is a script that uses `OVRInput` and `RigidBody` physics to provide continuous movement and snap rotation for the player. Since it is a singleton, more than one instance of the script is not expected in a given scene.

The movement speed and rotation snap angle can be configured through fields of the same name. You can toggle movement completely through the `MovementEnabled` field. In fact, it is used by `XRKeyboardManager` to automatically stop movement when the keyboard is active.