VRIF Interaction Component for VR Builder

Table of Contents

1. Introduction
2. Requirements
3. Quick Start
4. Properties
    - Grabbable Property
    - Touchable Property
    - Usable Property
    - Snappable/Snap Zone Property
    - Lever/Wheel Property
5. Check Control Position Condition
6. Contact

Introduction

This add-on allows to use VR Builder together with VR Interaction Framework. A VR Builder process guides the user through the application, while VRIF provides the interactions in the scene.

Requirements

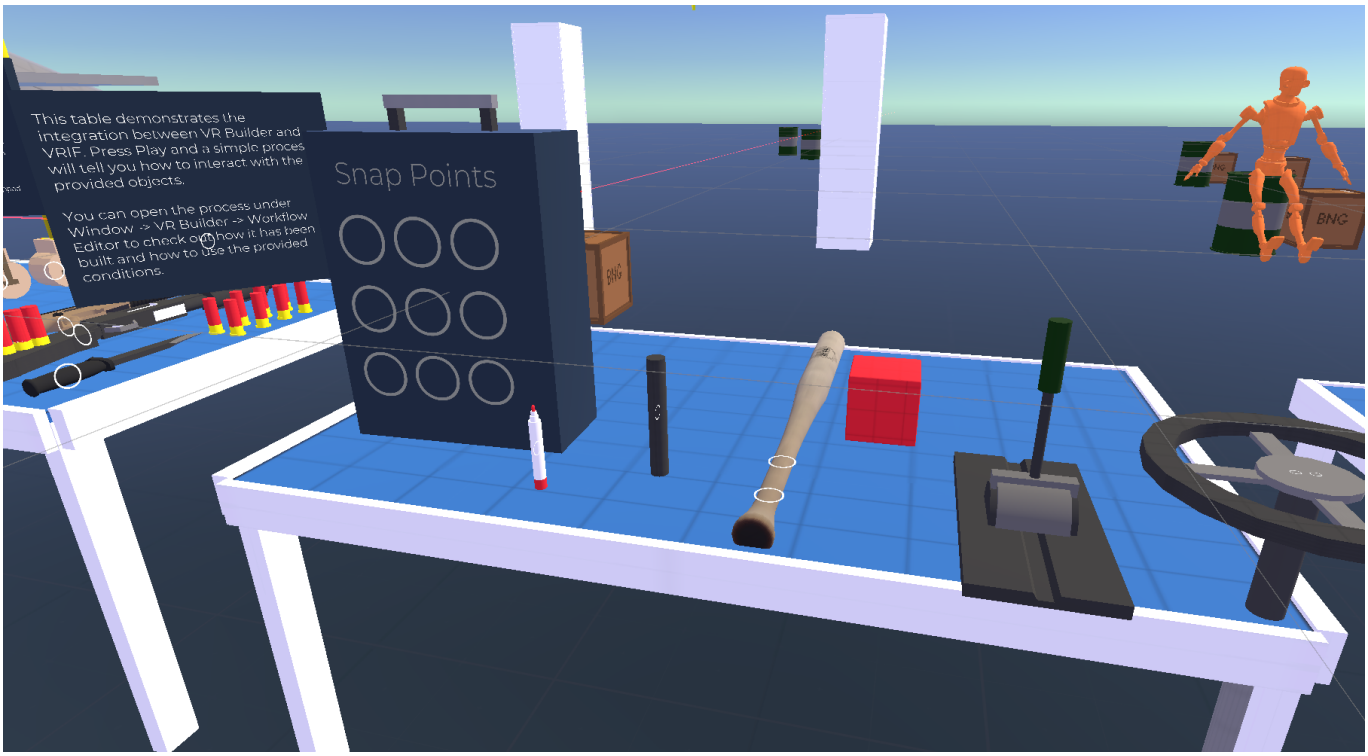
This add-on requires VR Builder version 2.1.0 or later to work.

It is recommended to disable the built-in XR Interaction Component to use this integration. You can do so in `Project Settings > VR Builder > Settings`.

Quick Start

If you are familiar with VR Builder and VRIF, the best way to start is probably the demo scene. A simple process will let you try all the integrated interactions one by one, and you will be able to check the process and the objects to see how they are configured. The first time you open the demo scene, you should do so from the menu: `Tools > VR Builder > Demo Scenes > VRIF Integration`. This will copy the process JSON in the StreamingAssets folder. The process will not work otherwise.

The VR Builder demo is limited to a single table in a copy of the VRIF demo scene. You will be instructed by the VR Builder process to interact with the objects one by one, thus testing every interaction provided.



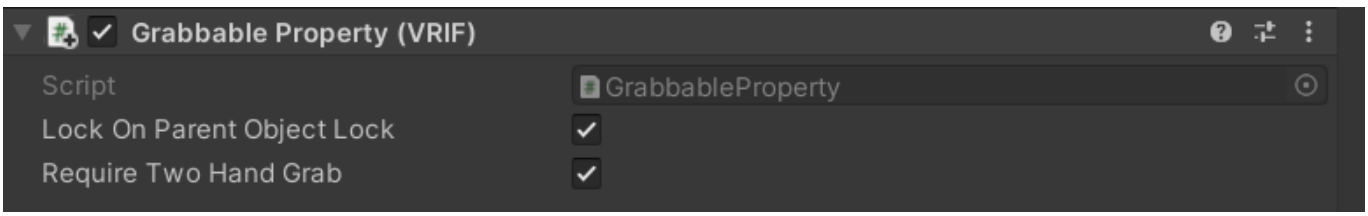
Properties

This integration includes the following properties. Most of those work with the standard VR Builder conditions and are replacements for similar properties in the built-in XR Interaction Component. Lever and Wheel properties are unique to this integration and allow to check the position on levers and wheels respectively thanks to a bespoke condition.

Grabbable Property

The included `Grabbable Property` allows to use the standard Grab Object and Release Object conditions with the VRIF grabbers and grabbables. Adding it to a game object will automatically add the `Grabbable` and `Grabbable Unity Events` components from VRIF.

Note that a rigidbody is not automatically added to the object, as the `Grabbable` component can be also used without in some cases. However, if you want a "standard" grabbable object, you should add one manually.



An additional feature of this implementation is the possibility to require a two hand grab by ticking the corresponding checkbox on the component itself. In this case, the object will count as grabbed only if grabbed with both hands. Note that this does not inherently change the behavior of the object, for example by preventing the user from grabbing it with one hand only. It only defines what VR Builder considers a valid grab on the object.

Touchable Property

Allows to use the Touch Object condition with VRIF.

Usable Property

This property qualifies the object as "in use" based on the `onTriggerDown` event on the `Grabbable Unity Events` component. It works similarly to the `Usable Property` in the built-in XR Interaction Component, with the difference that the object can always be used as long as it's grabbed (i.e. this property is not locked).

Snappable/Snap Zone Property

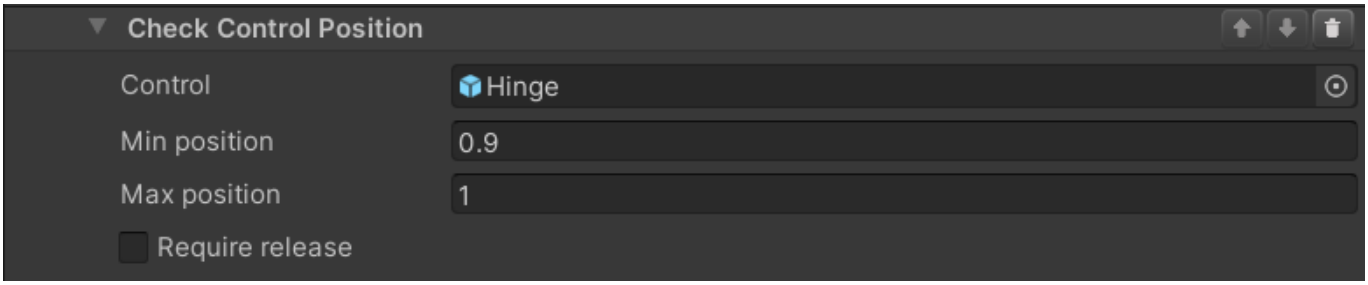
These allow to use VRIF snap zones in a VR Builder process. Note that snap zones in VRIF, contrary to the one in the built-in XR Interaction component, don't snap one specific object by default. The `Snap Zone Property` does not perform any automatic configuration on the snap zone itself. Please refer to the VRIF demos and documentation to configure the snap zones.

Lever/Wheel Property

These can be added to a game object with respectively the `Lever` or `Driving Wheel` component. The object can then be used in the `Check Control Position` condition.

Check Control Position Condition

Description



This condition triggers when a movable object like a lever or a wheel reaches a position within a range. The position is normalized from 0 (down) to 1 (up) for a lever, and from -1 to 1 for a driving wheel.

Configuration

- **Control**

  The object whose position we want to check. The game object referenced here should be the one with the `Lever` or `Driving Wheel` component, not necessarily the root object. Please manually add a `Lever Property` or `Wheel Property` instead of relying on the `Fix it` button. As they are implementations of the same interface, it is not possible at this time to automatically select the correct property.

- **Min position**

  The minimum position which will be considered valid by the condition.

- **Max position**

  The maximum position which will be considered valid by the condition.

- **Require release**

  If this is checked, the condition will not complete until the object is released. This will require the user to place the object in the correct position and release it instead of just moving it back and forth until something happens.

Contact

Join our official Discord server for quick support from the developer and fellow users. Suggest and vote on new ideas to influence the future of the VR Builder.

Make sure to review VR Builder if you like it. It will help us immensely.

If you have any issues, please contact contact@mindport.co. We'd love to get your feedback, both positive and constructive. By sharing your feedback you help us improve - thank you in advance! Let's build something extraordinary!

You can also visit our website at mindport.co.