

# Picking up objects (Basic)

This page's original source is here: <https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/VR/VRHowTos/UsingTouchControllers/>

## Prerequisites

[Set up an XR Rig](#)

[Basic Teleport \(covers input\)](#)

*Adding Motion Controller support to your UE4 projects can add a level of real sense of immersion and realism. One way to take this immersion and realism to the next level is add the ability to pick up objects placed in the world with your Motion Controllers. In the following document we will take a look at how you can add Motion Controller support your UE4 VR project.*

## Motion Controller Setup

In the following sections we will go over how to setup your Motion Controllers so that they can pick up and drop items that are placed in the level. For this guide, we will focus on using the just the LEFT controller.

## Component, Variable, & Event Setup

Before we can start to add nodes to the Event graph, we first need to create and set up a few Components and Variables. In the following section, add these variables to your variable list

Component / Variable Type	Name	Value
<b>PrimitiveComponentType</b>	HitComponent	
<b>Boolean</b>	IsHoldingObject	false
<b>Actor</b>	PickedUpActor	

You will also need to create two **Custom Events in your Event Graph** and name them the following:

Node Name	Value
PickUpObject	N/A
DropObject	N/A



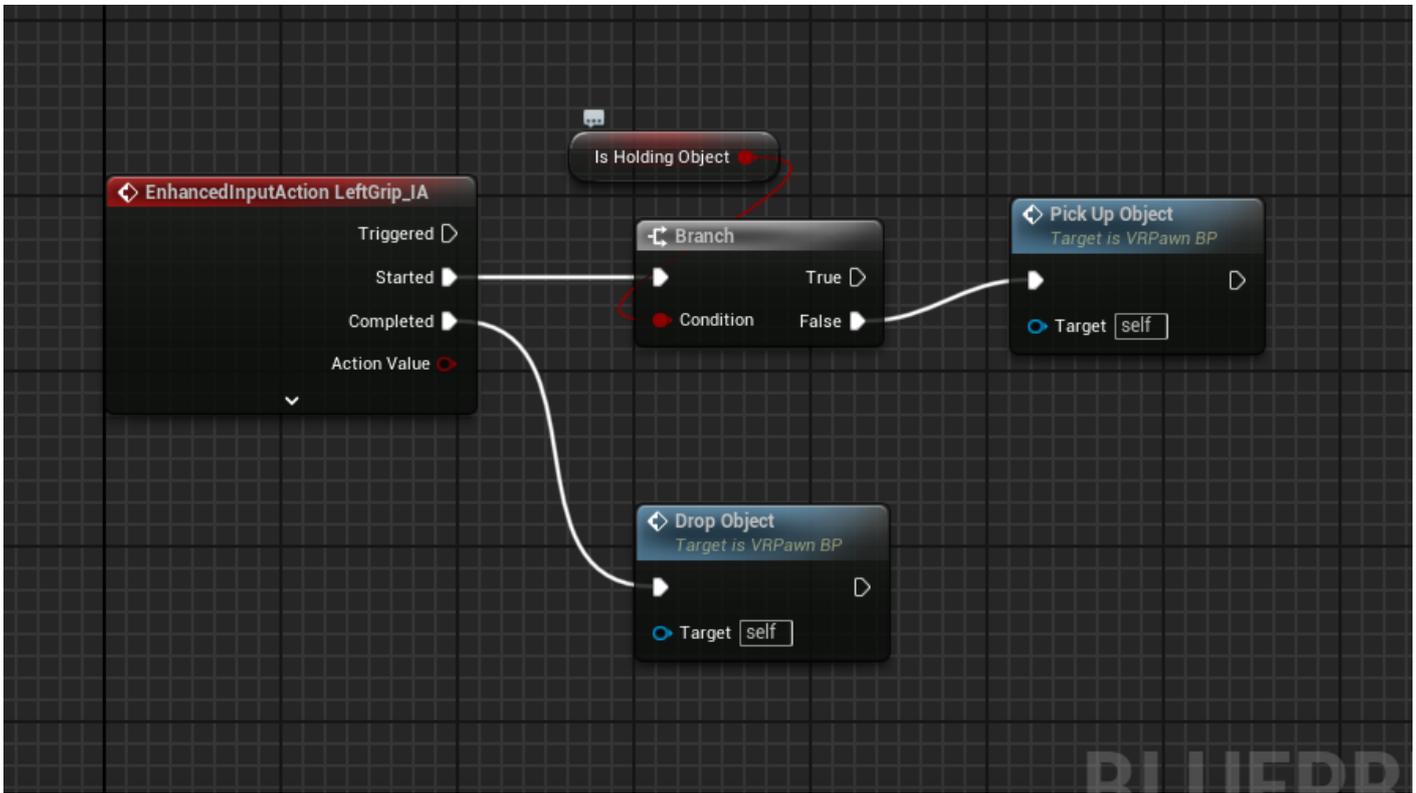
The names of these do not differentiate between Left or Right (and that is on purpose). At the end of the tutorial, I recommend adding "Inputs" to these events and feeding your controllers in. But don't do that now.

Finally, if you haven't already, as found in the teleport tutorial, you need to create two additional **Input Actions** for Left and Right grip actions. Be sure to assign these actions to your **Input Mappings**. Here is a link if you forgot how to set up input:

<https://scil-wiki.su.edu/books/unreal-engine/page/setting-up-basic-teleport-pt2>

## Holding and Dropping of Items

The Holding and Dropping of Items section call the different Custom Events that handle picking objects up and holding objects as well as dropping them. The user initiates this by pressing or releasing the Left Motion Controller Trigger. One thing to take note of here is that we are using a Branch statement to make sure that we can only pick one object up at a time. The reason this is important is we only want the user to be able to pick up one object at a time.



## Pick Up Object Event

The Pick-Up Object Event section handles all of the logic for finding objects that can be picked up and picking them up. While this is one function, it does have two parts that it can be broken down into. In the following section, we will take a look at these two parts and what they do.

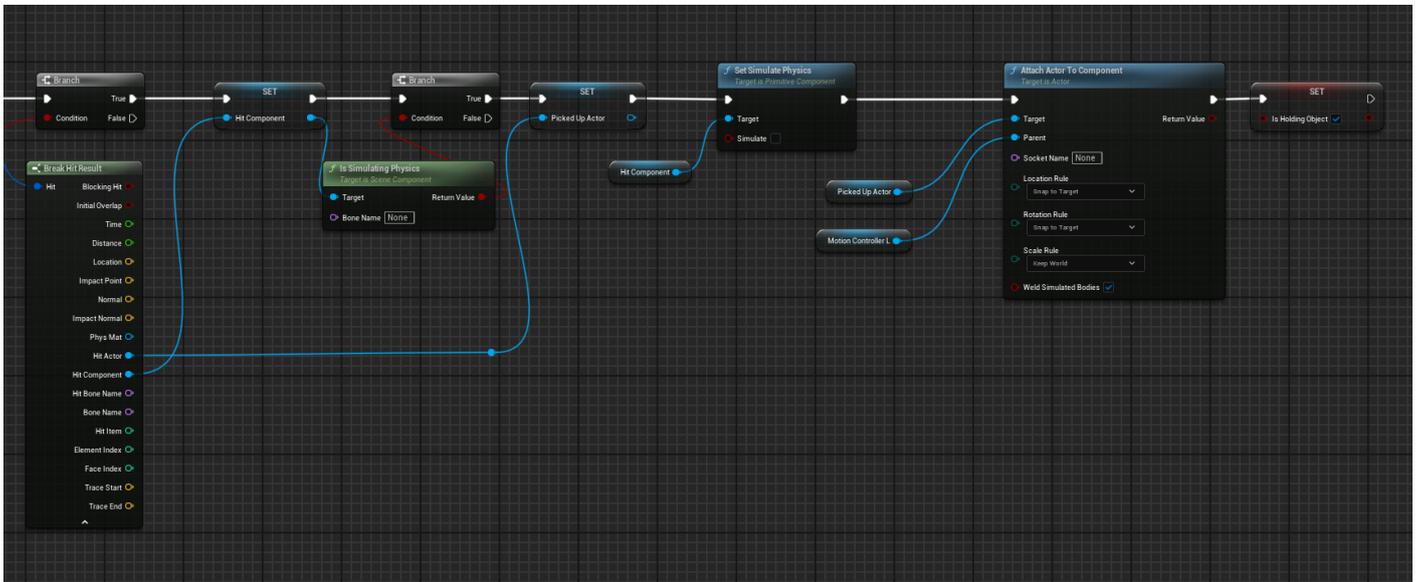
The first part of the Pick-Up Object Event deals with finding the object in the world that meets the requirements for being picked up. To do this, a ray is cast 1,000 CM into the world from the forward facing direction of the user's Left Hand Motion Controller. This ray is also told to ignore any object

that does not have an Object Type of Physics Body.



For testing purposes, the **Draw Debug Type** has been set to **Duration** allowing us to see the ray that was cast into the world. When you are ready to use this in production, make sure to set the Draw Debug Type to **None**.

The second part of the Pick-Up Object Event deals with what happens once we find an object we can pick up. When an object is found the **Break Hit Result** node is used to get more information about what we hit. In this case, we are using it to find out what Actor and Component were hit. Physics is then disabled on that object, and the object is then attached to our **Motion Controller**. Finally, we set the variable **IsHoldingObject** to true so that we can not pick something else up.



It may be difficult to see from the image above, but **Attach Actor to Component** has **Location Rule** and **Rotation Rule** set to **Snap to Target** and **Scale Rule** set to **Keep World**.

## Drop Object Event

The Drop Object Event section handles all the logic for dropping objects that have been picked up and also making sure that everything is reset, so we are ready to pick something else up. The first thing that the Drop Event does is check to make sure that the user is holding something that can be dropped. If they are holding something, then that object is then detached from their Motion Controller and has its physics re-enabled allowing it to fall to the ground. Finally the Hit Component and Picked Up Actors variables are cleared of any old data making sure they are ready to store the newly picked up object.



With the Blueprint now complete, we just need to add some objects to the world to pick up. Since the objects we are looking to pick up must have a physics body, you can just add a few Static Mesh and set their **Mobility** to **Movable** and also make sure to enable **Simulate Physics**.

## Where do we go from here?

You can add input parameters to your Inputs section of the Pickup and Drop events. These inputs you can fill with your motion controller. Then, the parameters can be used within the event itself by

replacing the motion controller. If you are stuck, see Wes in SCiL.

This covers basic grab interactions. For a more complete interaction system, consider using the **VR Expansion Plugin**: [VR Expansion Plugin](#)

---

Revision #5

Created 21 June 2024 15:12:23 by Wes

Updated 4 September 2024 18:11:51 by Wes