

# Upgrade Guide

If you're insane and decided to upgrade this asset in the middle of your project development, this guide is for you.

Remember that to upgrade LeanTouch you must first back up your project, delete the LeanTouch folder, then install the new version. This is because LeanTouch script and scene files get added and removed, and Unity's asset installer doesn't care about that, so you can end up with old and new files, which probably will cause errors that aren't my fault.

## Version 1.2.7

### **LeanOrbit\_\_ & LeanCameraOrbit\_\_ removed**

These components were removed in order to make the code more modular.

To do orbits, you should now make a root/pivot GameObject that has the LeanPitchYaw or similar component, and add your camera as a child, using LeanCameraDolly or just setting the transform.localPosition.z.

This change was important because as I added more orbit-related components, I found I was duplicating a lot of zoom and dolly functionality in order to satisfy every orbit variation, so separating all of these functionalities makes it easier to maintain.

## Version 1.2.2

### **LeanTouch.OnFingerHeldDown & OnFingerHeldSet & OnFingerHeldUp removed.**

These features were moved to the LeanFingerHeld component.

## Version 1.1.6

### **LeanSelect2D & LeanSelect3D removed**

These components were combined into the LeanSelect component.

The only setting you must change is 'Select Using' to either 'Overlap 2D' or 'Raycast 3D'.

## Version 1.1.5

### **LeanTouch.DragDelta removed**

This has been replaced with: `LeanGesture.GetScreenDelta();`

### **LeanTouch.SoloDragDelta removed**

This was removed because it's very game specific. You can get all fingers via `LeanTouch.Fingers` or `LeanTouch.GetFingers()` and perform similar logic yourself by comparing the size and reading the remaining finger's information.

### **LeanTouch.MultiDragDelta removed**

This was removed because it's very game specific. You can get all fingers via `LeanTouch.Fingers` or `LeanTouch.GetFingers()` and perform similar logic yourself by comparing the size and reading the combined finger information via `LeanGesture.GetScreenDelta(fingers)`

### **LeanTouch.TwistDegrees removed**

This has been replaced with: `LeanGesture.GetTwistDegrees();`

### **LeanTouch.TwistRadians removed**

This has been replaced with: `LeanGesture.GetTwistRadians();`

### **LeanTouch.PinchScale removed**

This has been replaced with: `LeanGesture.GetPinchScale();`

### **LeanTouch.OnFingerDrag removed**

This was removed because on real devices, fingers are very rarely not dragging. You can replace this with `LeanTouch.OnFingerSet`, which will get called every frame a finger is touching the screen. To replicate the old behavior, simply check if `finger.ScreenDelta` isn't `Vector2.zero`

## **LeanTouch.OnMultiTap removed**

This was removed because it's very game specific. It was re-implemented in the LeanMultiTap component, and accompanying demo scene.

## **LeanTouch.OnDrag removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and call Leangesture.GetScreenDelta(fingers)

## **LeanTouch.OnSoloDrag removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and check for one finger, then use finger.ScreenDelta

## **LeanTouch.OnMultiDrag removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and check for multiple fingers, then use OnGesture.GetScreenDelta(fingers)

## **LeanTouch.OnPinch removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and call Leangesture.GetPinchScale(fingers)

## **LeanTouch.OnTwistDegrees removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and call Leangesture.GetTwistDegrees(fingers)

## **LeanTouch.OnTwistRadians removed**

This was completely removed. You can instead hook into LeanTouch.OnGesture, and call Leangesture.GetTwistRadians(fingers)

## **LeanFinger.LastSnapshotDelta renamed**

Now LeanFinger.LastSnapshotScreenDelta

## **LeanFinger.DeltaScreenPosition renamed**

Now LeanFinger.ScreenDelta

## **LeanFinger.TotalDeltaScreenPosition renamed**

Now LeanFinger.SwipeScreenDelta

## **LeanFinger.ScaledTotalDeltaScreenPosition renamed**

Now LeanFinger.SwipeScaledDelta

## **LeanFinger.SwipeDelta removed**

This was removed because it's very game specific. You can instead use LeanFinger.GetSnapshotScreenDelta()

## **LeanFinger.ScaledSwipeDelta removed**

This was removed because it's very game specific. You can instead use LeanFinger.GetSnapshotScaledDelta()

## **LeanFinger.ScaledTotalDeltaMagnitude removed**

This was removed because it's very game specific. You can calculate this yourself by accumulating LeanFinger.ScaledDelta.magnitude

## **LeanFinger.GetScaledSnapshotDelta renamed**

Now LeanFinger.GetSnapshotScaledDelta;

## **LeanFinger.GetDeltaWorldPosition renamed**

Now LeanFinger.GetWorldDelta;