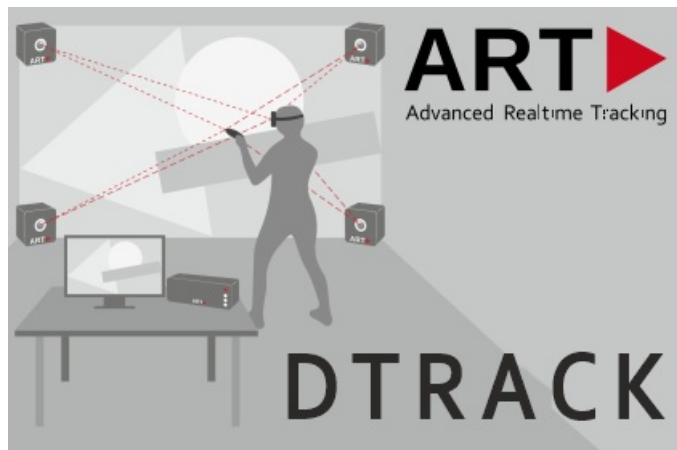


ART DTRACK Plugin for Unity Game Engine 2019 or later

Version v1.1.3

This is a component for Unity 2019 or later with the purpose of native integration of the Advanced Realtime Tracking (ART) DTRACK (DTrack2 or DTRACK3) realtime tracking solutions.



This Unity Asset provides access to DTRACK realtime tracking data, that is sent over network using UDP/IP datagrams. Each UDP packet contains one frame of tracking data including all outputs activated via the DTRACK software (see section **DTRACK Configuration**). This package currently supports the DTRACK Standard Body ('6d'), Flystick ('6df2') and Fingertracking ('gl') data formats.

Features

- Validated with Unity Editors 2019.4 to 2023.2 (but should work with any later Unity)
- Supports DTRACK room calibration modes 'Powerwall' and 'Normal'
- Supports tracking of 6DOF Standard Bodies and Flystick2, Flystick2+ and Flystick3
- Supports Flystick buttons and joystick/trigger (emitting Unity events)
- Supports Fingertracking, with 'Leap Motion Realistic Male/Female Hands' (by Storkplay)
- (Optionally) enables UDP connection through stateful firewall

Installing the ART DTRACK Plugin

There are several ways to get the ART DTRACK Plugin.

Unity Asset Store

The plugin is available (for free) in the Unity Asset Store <https://assetstore.unity.com/packages/tools/integration/art-dtrack-plugin-246990>.

After 'purchasing' you can install it via the Unity Package Manager:

1. Launch Unity Hub
2. Create/Open Unity project
3. Open Package Manager (*Window* → *Package Manager* → *Packages: My Assets*)
4. Download the package (button *Download*), then install it (button *Install*)

The plugin will be installed in your project's Assets folder:

/path/to/unity/projects/*MyUnityProject*/Assets/ARTDTrackPlugin/ .

OpenUPM

OpenUPM is an open-source service hosting lots of (UPM) Unity packages, all of them released as open-source. The ART DTRACK plugin is available (for free) via <https://www.openupm.com/packages/com.ar-tracking.dtrack>.

To install it, follow the instructions mentioned at the above website; roughly:

1. Launch Unity Hub

2. Create/Open Unity project
3. Open *Edit* → *Project Settings...* → *Package Manager*
4. Add a new Scoped Registry (corresponding to the above OpenUPM link), then click *Save* or *Apply*
5. Open Package Manager (*Window* → *Package Manager* → *Packages: My Registries*)
6. Choose *ART DTRACK Plugin* , then install it (button *Install*)

The plugin will be installed in your project's Packages folder:

/path/to/unity/projects/*MyUnityProject*/Packages/**ART DTRACK Plugin**/ .

GitHub

You can download a ready-to-use custom Unity package (*UnityDTrackPlugin-vX.X.X.unitypackage*) at

<https://github.com/ar-tracking/UnityDTrackPlugin/releases>.

In order to install the package, follow the steps below.

1. Launch Unity Hub
2. Create/Open Unity project
3. Import package (*Assets* → *Import Package...* → *Custom Package...*)

The plugin will be installed in your project's Assets folder:

/path/to/unity/projects/*MyUnityProject*/**Assets/ARTDTrackPlugin**/ .

Plain Sources

You can download or clone sources for this plugin at <https://github.com/ar-tracking/UnityDTrackPlugin>.

Updating from ART DTRACK Plugin v1.0.3 or later

IMPORTANT: To update an older plugin within an existing Unity project it's necessary to replace the entire directory

/path/to/unity/projects/*MyUnityProject*/Assets/**DTrack**/ or

/path/to/unity/projects/*MyUnityProject*/Assets/**ARTDTrackPlugin**/ , if present.

1. Launch Unity Hub
2. Open Unity project
3. Remove the directory, either manually or in Unity Editor (e.g. *Project* → *Assets* → *DTrack* → (right click) *Delete*)
4. Install the ART DTRACK Plugin (see above)

There's no need to adjust settings in already attached scripts (**DTrack**, **DTrackReceiver6Dof**, **DTrackReceiverFlystick**, ...).

Please note: The formerly (ART DTRACK Plugins v1.0.X) used 'Unity events' to notify pressed Flystick buttons were declared deprecated and will be removed in some future version of the plugin. Please don't use them in new projects (see section **Applying Flystick Data**).

Tip: If after an update existing button event settings (*Button Press Event X*, now denoted as *Button X Pressed Event Deprecated*) are not shown in Unity Editor, exiting and re-starting of the Unity Editor might help.

DTRACK Configuration

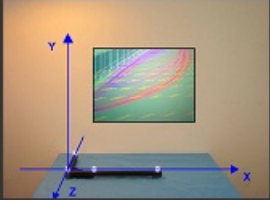
Find here a quick-start guide to DTRACK. For details, please refer to your *DTRACK3 User's Guide* and *DTRACK3 Programmer's Guide* , that is shipped with the DTRACK distribution. In this section we assume that the ART tracking system is properly set up and a room calibration was done. Further, a set of Standard Bodies and/or Flysticks and/or Fingertracking hands is calibrated.

Room Calibration

For general information about the DTRACK room calibration and room adjustment see the *DTRACK3 User's Guide* . Here we discuss details relevant for use with the Unity Engine.

DTRACK coordinates refer to a right-handed coordinate system, whereas Unity is using a left-handed coordinate system with the Y axis pointing up. The plugin provides two possibilities to convert DTRACK coordinates into the Unity world, corresponding to the two DTRACK calibration modes (setting *coordinate system*).

wand length [mm]
300,00



marker distances	expert
A (1-2) [mm]	240,0
B (1-4) [mm]	90,0
C (1-3) [mm]	150,0
height [mm]	0,0

coordinate system
Power wall

☐ re-calibration (without angle tool)

Show details

Transfer

Calibrate

Exit

The calibration angle tool which comes with your ART tracking system defines the coordinate system layout in your tracking area. It consists of four retroreflective or active markers mounted onto an L-shaped frame. The marker on top of the edge of this L-shape by default designates the origin of the DTRACK coordinate system.



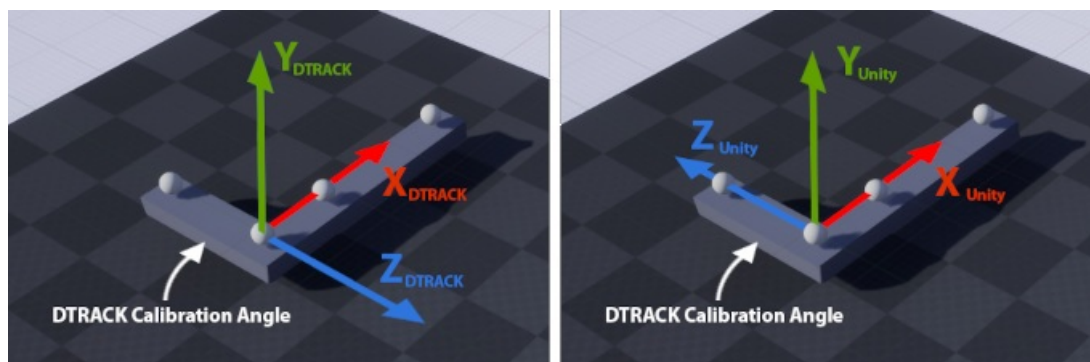
DTRACK Calibration Mode 'Powerwall'

Please note: It's recommended to use this calibration mode.

When using the DTRACK *Powerwall* calibration mode (see figure below), the long arm of this L-shape corresponds to the X axis and the short arm to the (-Z) axis. DTRACK coordinates refer to a right-handed coordinate system, so if the tool is placed flat on the ground with the markers pointing up, the Y axis points upwards.

The plugin transforms a right-handed position of a DTRACK 6DOF measurement to a left-handed Unity position by negating the Z axis, i.e.

$$(X_{\text{Unity}}, Y_{\text{Unity}}, Z_{\text{Unity}}) = (X_{\text{DTRACK}}, Y_{\text{DTRACK}}, -Z_{\text{DTRACK}}).$$



DTRACK and Unity coordinate systems in *Powerwall* mode

DTRACK Calibration Mode 'Normal'

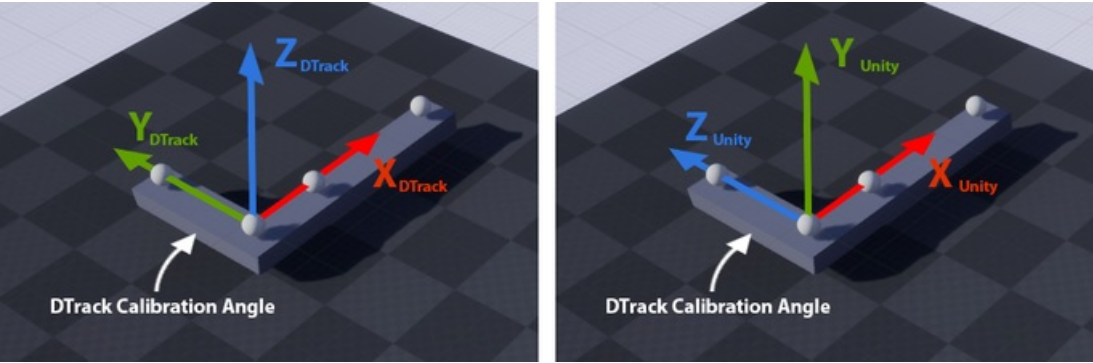
This was the only available mode in ART DTRACK plugin prior to v1.1.1.

Please note: It's recommended to prefer the 'Powerwall' calibration mode (see above).

When using the DTRACK *Normal* calibration mode (see figure below), the long arm of the L-shape corresponds to the X axis and the short arm to the Y axis. DTRACK coordinates refer to a right-handed coordinate system, so if the tool is placed flat on the ground with the markers pointing up, the Z axis points upwards.

The plugin transforms a right-handed position of a DTRACK 6DOF measurement to a left-handed Unity position by switching the Y and Z axes, i.e.

$$(X_{\text{Unity}}, Y_{\text{Unity}}, Z_{\text{Unity}}) = (X_{\text{DTRACK}}, Z_{\text{DTRACK}}, Y_{\text{DTRACK}}).$$



DTRACK and Unity coordinate systems in *Normal* mode

Setting Outputs

To configure the tracking data stream generated by DTRACK, execute these steps:

1. Open dialog *Output Settings* via menu *Tracking* → *Output* (DTRACK3) or *Settings* → *Output* (DTrack2), respectively
2. Activate a channel if needed
3. Fill in hostname or IP address, and UDP port number of the device receiving tracking data
4. Select outputs you are interested in (i.e., currently frame counter 'fr', 6DOF Standard Body '6d', Flystick '6df2' and Fingertracking 'gl' are supported)

Channel 1

Channel 2

+

☒ active

send to

my.unity.hostname.or.ip

☐ this computer

☐ multicast (224.0.1.0 - 239.255.255.255)

UDP port

5000

send data divisor

1

Identifier	Description
<input checked="" type="checkbox"/> fr	frame counter
<input type="checkbox"/> ts	timestamp
<input type="checkbox"/> st	system status
<input type="checkbox"/> 6dcal	number of calibrated bodies
<input checked="" type="checkbox"/> 6d	6DOF standard body
<input type="checkbox"/> 6di	6DOF inertial body
<input type="checkbox"/> 3d	3DOF marker
<input checked="" type="checkbox"/> 6df2	Flystick
<input type="checkbox"/> 6dmt2	Measurement Tool
<input type="checkbox"/> 6dmtr	Measurement Tool reference
<input type="checkbox"/> glcal	number of calibrated Fingertracking hands
<input checked="" type="checkbox"/> gl	Fingertracking hand

☐ act as router for tracking output

Exit

Identifying Body IDs

There are several ways to identify the ID numbers of DTRACK Standard Bodies, Flysticks and Fingertracking hands, as needed later to configure the Unity plugin. E.g. refer to column *ID* in dialog *Body Administration* via menu *Tracking → Body Administration* (DTRACK3) or *Settings → Body Administration* (DTrack2).

StandardFlystickMeasurement ToolFingertracking

number of bodies2

ID	Active	Reference	Name	Calibration	Filter	Delete	Reset
1	✓	<input type="checkbox"/>	AGT5	✔ Custom	default	✗	↺
2	✓	<input type="checkbox"/>		✔ Custom	default	✗	↺

↑

↓

FilterCalibration

Exit

Note that listed Flystick IDs are prefixed with a capital **'F'**, as well as listed Fingertracking hand IDs are prefixed with a capital **'H'**. When referencing Flysticks or Fingertracking hands from within Unity, this prefix must be removed.

StandardFlystickMeasurement ToolFingertracking

number of flysticks2

ID	Active	Model	Serial/Port	Calibration	Filter	Delete	Reset
F1	✓	Flystick2	00013	✔ Custom	default	✗	↺
F2	✓	Flystick2+	00005	✔ Custom	default	✗	↺

FilterCalibration

Exit

Fingertracking

When configuring ART Fingertracking hand devices with 3 thimbles, be sure to activate the *Simulate 5 fingers* setting.

Settings

number of hands2

selected hands

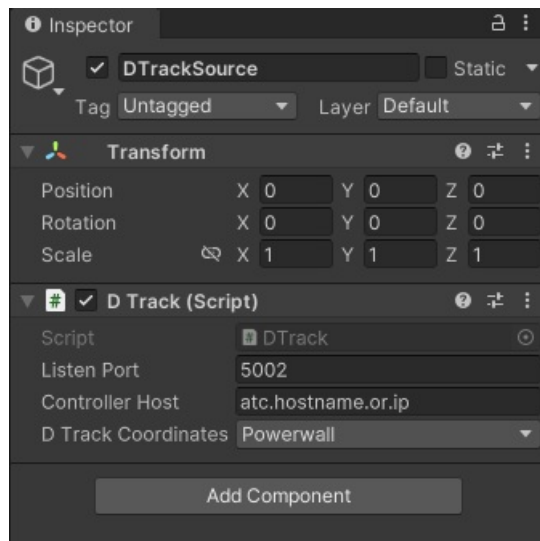
Hand ID	Name	Geometry	Simulate 5 fingers	Cal.
H1	FINGERTRACKING2 right Hand Target 1	KA 3AR rechts 1	✓ 3 -> 5	✔
H2	FINGERTRACKING2 left Hand Target 2	KA 3AR links 2	✓ 3 -> 5	✔

Plugin Configuration

Streaming position, rotation and button events data from DTRACK tracking systems to objects in your scene, requires appropriate network settings. In your scene add an *Empty* game object and give it a name, e.g. **DTrackSource**. To this object attach the **DTrack** script via *Add Component* → *Scripts* → *DTrack* → *DTrack*.

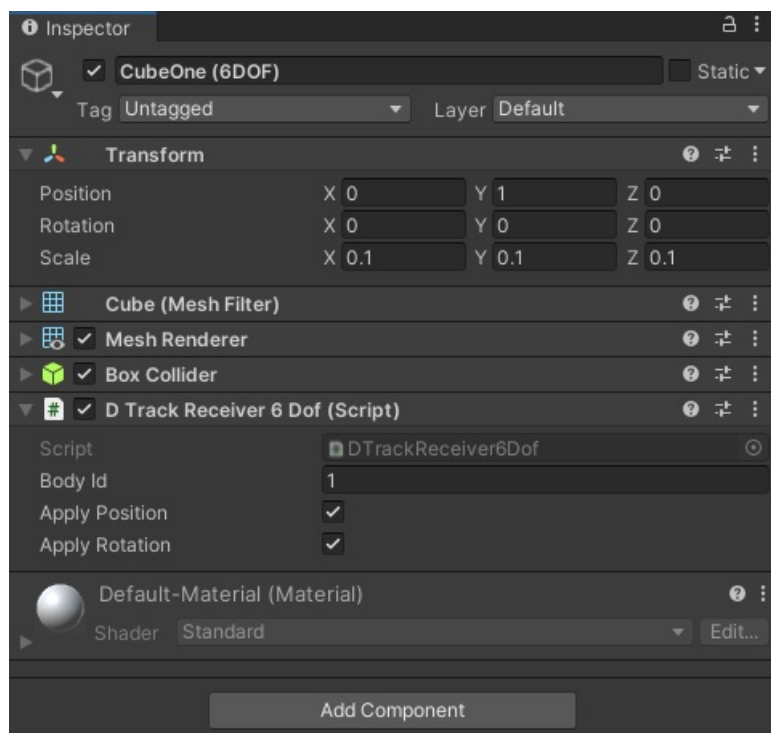
- Set *Listen Port* number matching the setting for DTRACK (see section **Setting Outputs**)
- Optionally set *Controller Host* with hostname or IP address of your ART Controller, if there is a 'stateful firewall' installed between Controller and Unity; normally that's sufficient to pass UDP packets with tracking data to the plugin
- Set *DTrack Coordinates* matching the calibration mode used in DTRACK (see section **Room Calibration**)

Note that position data in the DTRACK output stream have unit millimeters. The DTRACK Unity Plugin converts such values to unit meter.



Applying 6DOF Standard Body Data

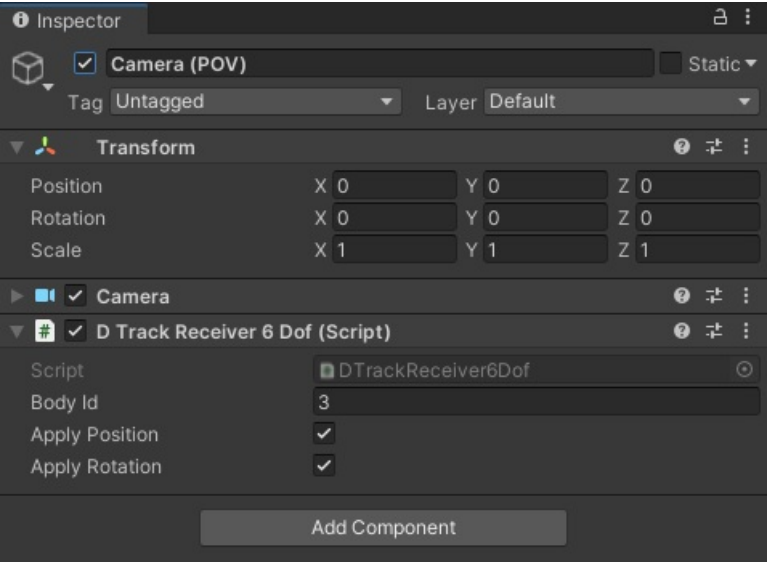
In your scene attach via *Add Component* the DTrack script **DTrackReceiver6Dof** to an object you want to receive positional and rotational data. In the **DTrackReceiver6Dof** mask type in the ID that was assigned to the Standard Body by DTRACK (see section **Identifying Body IDs**).



When the ART tracking system is running, you should now be able to see *Position* and *Rotation* data in the **Transform** box, as soon as you switch to *Play* mode.

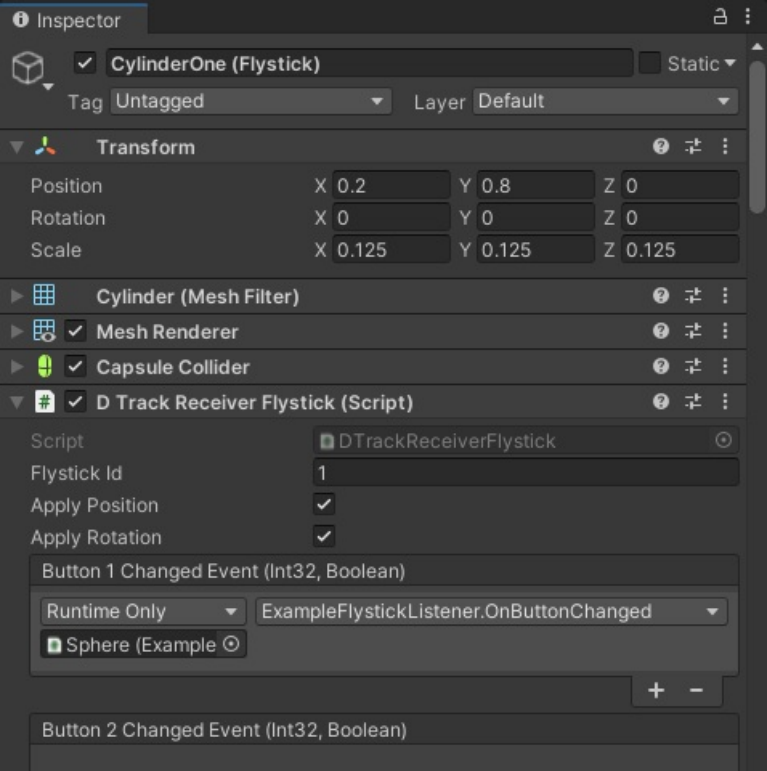
Applying 6DOF Standard Body Data to the Camera

For non-static, point-of-view cameras, you can attach a DTRACK receiver with positional and rotational data, e.g. of a 6DOF Standard Body.



Applying Flystick Data

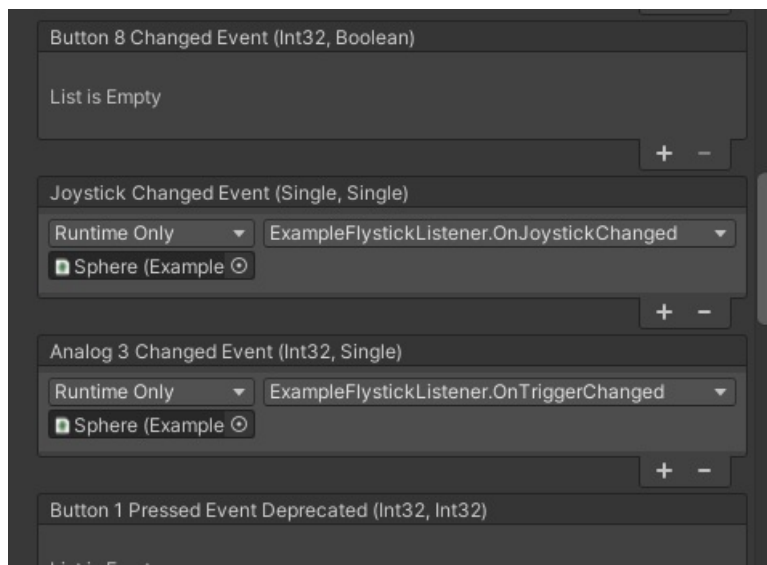
In your scene attach via *Add Component* the DTrack script **DTrackReceiverFlystick** to an object you want to receive positional and rotational data. In the **DTrackReceiverFlystick** mask type in the ID that was assigned to the Flystick by DTRACK (see section **Identifying Body IDs**, without prefixed capital 'F').



DTrack script **DTrackReceiverFlystick** can send 'Unity events' to announce changed Flystick buttons or joystick values to scripts attached to arbitrary game objects. It provides:

- 8 events on changed Flystick buttons corresponding to the (maximum) number of 8 buttons of Flystick2+ (*Button X Changed Event*); an event is invoked once every time a button is pressed or released
- One event on changed Flystick joystick values (horizontal and vertical) (*Joystick Changed Event*); the event is invoked every time one of the values has changed
- One event on changed trigger value (just available at Flystick2+) (*Analog 3 Changed Event*); the event is invoked every time the value has changed

Please note: The formerly (ART DTRACK Plugins v1.0.X) used 'Unity events' to notify pressed Flystick buttons (*Button Press Event X*) were declared deprecated and will be removed in some future version of the plugin. Please don't use them in new projects.

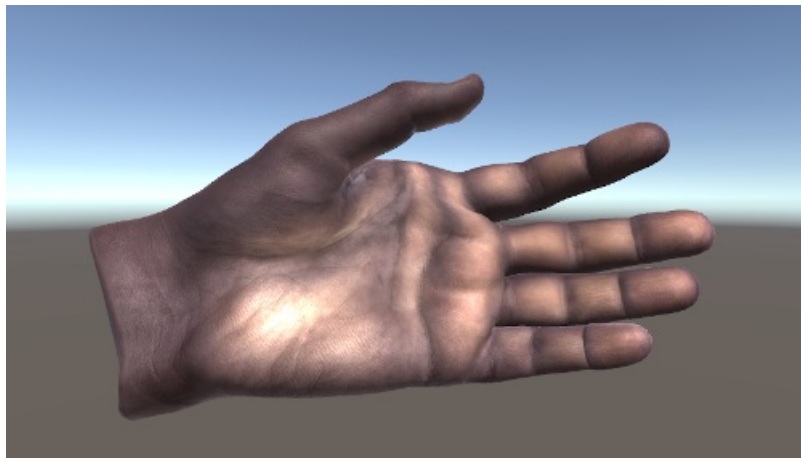


See scripts **ExampleFlystickListener** or **ExampleCameraNavigator** on how to receive these 'Unity events' in your scripts. To register an own listener routine first create an additional entry in the wanted *Event* element by pressing **+**. Then drag and drop the game object, that should receive the event, from the *Hierarchy* tree into the *Event* element. Finally choose the listener routine's name.

Applying Fingertracking Data

The DTRACK Plugin provides support of ART Fingertracking, so far adjusted for usage with 'Leap Motion Realistic Male/Female Hands' (by Storkplay, available at the Unity Asset Store), or hand models with equivalent rig and coordinate systems:

- <https://assetstore.unity.com/packages/3d/characters/humanoids/leap-motion-realistic-male-hands-109961>
- <https://assetstore.unity.com/packages/3d/characters/humanoids/leap-motion-realistic-female-hands-211090>

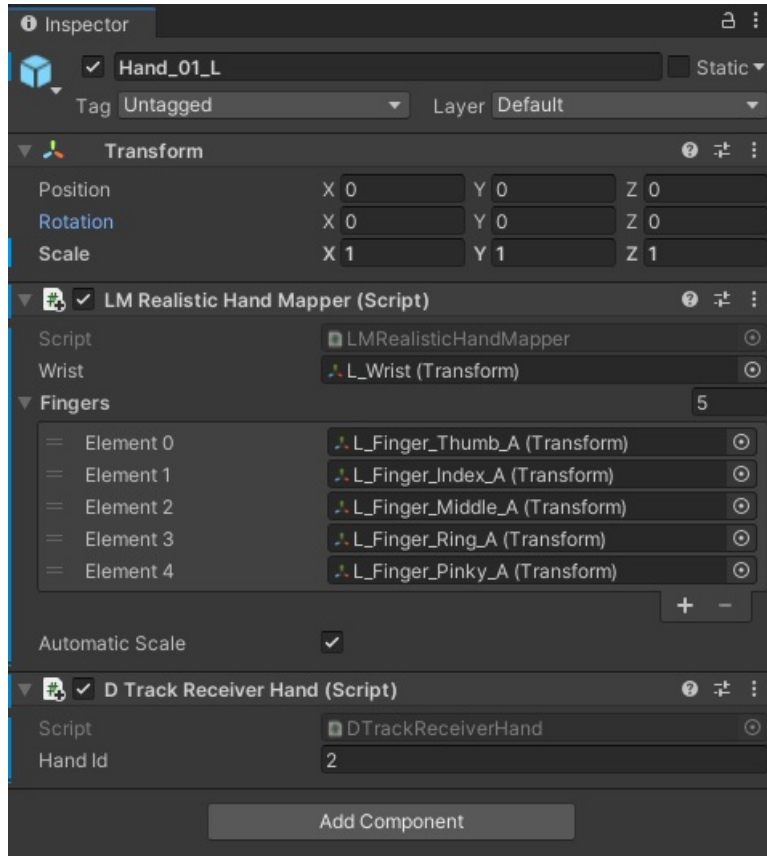


Add one of the *Prefabs* of your **LMRealisticFemaleHands** or **LMRealisticMaleHands** asset to your scene.

Within the root object of the created left or right hand (**Hand_0X_L** or **Hand_0X_R**) now two components with missing scripts appear. Remove these components via *Remove Component*, attach instead via *Add Component* the DTrack script **LMRealisticHandMapper**. It automatically will attach also a DTrack script **DTrackReceiverHand**.

In **DTrackReceiverHand** mask enter the *Hand Id* that was assigned to the Fingertracking device by DTRACK (see section **Identifying Body IDs**, without prefixed capital 'H').

In **LMRealisticHandMapper** mask first ensure that the number of elements in list *Fingers* is 5. Now enter links to the corresponding game objects of the hand, for *Wrist* (**L_Wrist** or **R_Wrist**) and the fingers' root joints (**L_Finger_Index_A**, ... or **R_Finger_Index_A**, ...). This can be done e.g. by dragging and dropping the objects from the *Hierarchy* tree into the **LMRealisticHandMapper** mask. The elements of *Fingers* have to be ordered as: thumb, index, middle, ring, pinky.



If the real size of the person's hand is differing too much from the hand model's size, it might be useful to modify the *Scale* factor in the **Transform** box of the hand's root object (**Hand_0X_L** or **Hand_0X_R**). If setting *Automatic Scale* is enabled, the hand mapper script tries to find a suitable value automatically.

Advanced Realtime Tracking GmbH & Co. KG
 Am Oeferl 6
 82362 Weilheim i. OB
 Germany

<https://ar-tracking.com>