

# Head Gesture Input for VR

## Version 1.0

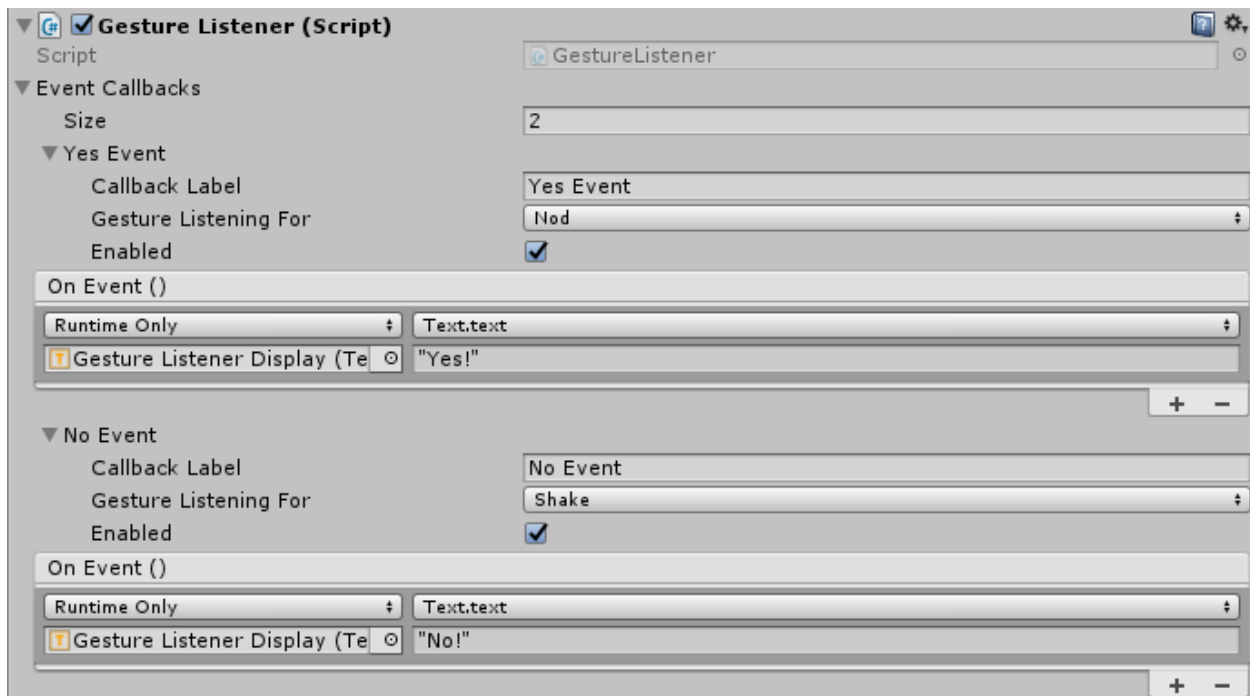
### Basic Usage

The Head Gesture Input system has two primary parts; the **Manager** and the **Listener**.

The **Manager**, behaves like a singleton, meaning that only one should exist per scene. This component should be attached to the player-vision camera, and will broadcast out input events to all **Listeners** when valid head gestures are detected.

As such, the **Listener** component should be added to any objects that you want to act when a head gesture happens. These components contain a list, wherein you define the callbacks you wish to occur for each type of event. These callbacks can be defined either in code, or in the inspector.

For example, here are two callbacks in a **Listener** that are set up to change the text of a text component:



Each callback has a field for its identifying label (NOTE: this label should be unique among all callbacks on the given **Listener**), the type of gesture it should listen for, whether or not it is currently enabled, and finally the methods to actually execute.

## Additional Methods

The **Listener** features the following methods:

```
public GestureEventCallback FindCallback(string label)
```

```
public void RemoveCallback(string label)
```

```
public void AddCallback(GestureEventCallback callback)
```

These methods are useful for modifying callbacks during runtime. Here, we see that the label given to a callback is used to later look it up, should we need to modify it through code, such as to enable or disable it. We can also add a callback by constructing a new `GestureEventCallback`, which has the following constructor:

```
public GestureEventCallback(string label, GestureType gestureType, bool isEnabled, GestureEvent gEvent)
```

`string label`: The unique label to give this callback.

`GestureType gestureType`: Denotes whether the callback is for a nod or a shake.

`bool isEnabled`: Denotes whether the callback is enabled or not.

`GestureEvent gEvent`: The actual method to execute upon receiving a gesture input.

The `GestureEvent` is simply an extension of the base `UnityEvent` class, which in turn is simply a method call that takes zero arguments.

The **Manager** does not feature any public methods, but does feature the following exposed parameters, which have been set to working values by default:

`public float DeltaThreshold`: The minimum difference in quaternion readings to trigger a gesture check.

`public float TimeThreshold`: The minimum amount of time that each gesture motion must be from the previous to be read as a gesture.

`public int GestureAmountThreshold`: The minimum amount of passing gesture checks that must occur to create a valid gesture.

For instance, increasing the `GestureAmountThreshold` to a value of 10 would require the player's head to change movement direction 10 times, each within `TimeThreshold` of the last change, in order to trigger a gesture and invoke the callbacks on the **Listeners**.

## Bug Reports

If you've found a bug, feel free to contact me at my email, [twfarro@wpi.edu](mailto:twfarro@wpi.edu).

Thank you for using this plugin, I hope it serves your needs well!