EasyTouch 5.X

New features

Index

What's new 3
Scripting 4
  Namespace 5
  New syntax 5
  Singleton 6
  Classes 6
Quick Gesture 7
  QuickTouch 8
  QuickTap 9
  QuickLongTap 9
  QuickEnterOverExit 10
  QuickSwipe 10
  QuickDrag 11
  QuickPinch 12
  QuickTwist 13
  Simple Action 14
Play Maker 14
  How to install 14
  Setup your scene 15
  Actions 16
  Transition 17

What's new in EasyTouch 5

- Add namespace HedgehogTeam.EasyTouch.
- Add Quick Gesture components.
- Add new syntax to use EasyTouch in Update method.
- EasyTouch class is now a singleton.
- Build in PlayMaker Add-on.
- You can choose to always send Swipe event during a drag.
- Add TV_OS
- Expose new touch data in Gesture Class (AltitudeAngle, AzimuthAngle, etc..)

Consult EasyTouch.PDF documentation for more detail on the existing features before 5.X

You can also visit our youtube channel : https://www.youtube.com/playlist?list=PL_xsQKiLfGUYpZt7uDepdadHo7GBGbozH

Scripting

EasyTouch 5 brings several new features for developpers

- Namespace
- New syntax
- Singleton
- New member for Gesture class

Namespace

Namespace : HedgehogTeam.EasyTouch

Easy Touch 5 introduces a namespace to avoid potential conflicts with another packages.

We must adding a using directive at the top of the file :
using HedgehogTeam.EasyTouch;

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using HedgehogTeam.EasyTouch;

public class TouchMe : MonoBehaviour {

	// Subscribe to events
	void OnEnable(){
		EasyTouch.On_TouchStart += On_TouchStart;
	}

	// Unsubscribe
	void OnDisable(){
		EasyTouch.On_TouchStart -= On_TouchStart;
	}

	// Unsubscribe
	void OnDestroy(){
		EasyTouch.On_TouchStart -= On_TouchStart;
	}

	// At the touch beginning
	public void On_TouchStart(Gesture gesture){

		// check that we have touched the sphere
		if (gesture.pickedObject == gameObject)
			gameObject.renderer.material.color = new Color( Random.Range(0.0f,1.0f),
    Random.Range(0.0f,1.0f), Random.Range(0.0f,1.0f));
	}
}
```

New syntax

New syntax

A new syntax allows to use Easy Touch in the Update method without having to subscribe to events.
This method retrieves the current gesture, and used the gesture class, as you would after receiving an event.
The class Gesture has a new member type type of EasyTouch.EvtType, allowing you to identify the current gesture.

However, this syntax is slower than management by events, because the data are placed in a queue and available frame by frame.

You can look at the example scene : **RTS_NewSyntaxe**

```
using UnityEngine;
using System.Collections;
using HedgehogTeam.EasyTouch;


public class NewSyntax : MonoBehaviour {

	// Update is called once per frame
	void Update () {

		// Get the current gesture
		Gesture current = EasyTouch.current;

		// Test the current gesture type or other properties from Gesture class
		if (current.type == EasyTouch.EvtType.On_TouchStart && current.pickedObject ==
gameObject){
			gameObject.GetComponent<Renderer>().material.color = new
Color( Random.Range(0.0f,1.0f),  Random.Range(0.0f,1.0f), Random.Range(0.0f,1.0f));
		}
	}
}
```

Singleton

Singleton

EasyTouch is now a singleton, if you use the new syntax or Quick Gesture component, it will be automatically created without having to add it manually in the scene.

Classes

Gesture

New members

The gesture class has expanded with new members (required Unity 5.3) from Unity Touch class.

altitudeAngle : Value of 0 radians indicates that the stylus is parallel to the surface, pi/2 indicates that it is perpendicular.
azimuthAngle : Value of 0 radians indicates that the stylus is pointed along the x-axis of the device.
maximumPossiblePressure : The maximum possible pressure value for a platform. If Input.touchPressureSupported returns false, the value of this property will always be 1.0f.
pressure : The current amount of pressure being applied to a touch. 1.0f is considered to be the pressure of an average touch. If Input.touchPressureSupported returns false, the value of this property will always be 1.0f.
radius : An estimated value of the radius of a touch. Add radiusVariance to get the maximum touch size, subtract it to get the minimum touch size.

radiusVariance : The amount that the radius varies by for a touch.
type : A value that indicates whether a touch was of Direct, Indirect (or remote), or Stylus type.

Update of two methods

Methods GetCurrentFirstPickedUIElement & GetCurrentPickedObject now has an optional parameter to indicate whether you want the current object compared to a 2-finger gesture

public GameObject **GetCurrentFirstPickedUIElement**(bool isTwoFinger=false)
public GameObject **GetCurrentPickedObject**(bool isTwoFinger=false)

EasyTouch

Static Methods GetCurrentPickedUIElement & GetCurrentPickedObject now has an optional parameter to indicate whether you want the current object compared to a 2-finger gesture

public static GameObject **GetCurrentPickedUIElement**(int fingerIndex,bool isTwoFinger)
public static GameObject **GetCurrentPickedObject**(int fingerIndex, bool isTwoFinger)

Quick Gesture components

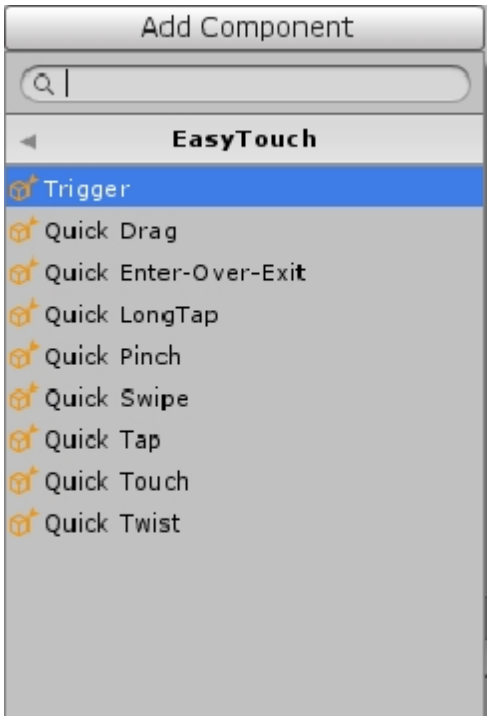
The quick gesture are components that allow you to trigger actions based on a gesture made on a gameobject.

No other script is required, each component is responsible to initialize Easy Touch, and setup it relative to the nature of gameobject (3d, 2D, UI Element, layer ...) and the quick component setting. **The GameObject must have a collider, except for the GUI element.**
A gameobject may possess one or more quickgesture component of different type, or of the same type.

Each component corresponds to a kind gesture
QuickTouch : Catch Touch_Start, Touch_Down, Touch_Up
QuickTap : Catch Simple_Tap, Double_Tap
QuickLongTap : Catch LongTap_Start, LongTap, LongTap_End
QuickenterOverExist : Special component to catch Enter , Over, Exit on a gameobject
QuickSwipe : Catch Swipe, Swipe_End
QuickDrag : Special component to perform a real drag
QuickPinch : Catch Pinch, Pinch_End
QuickTwist : Catch Twist, Twist_End

How to create

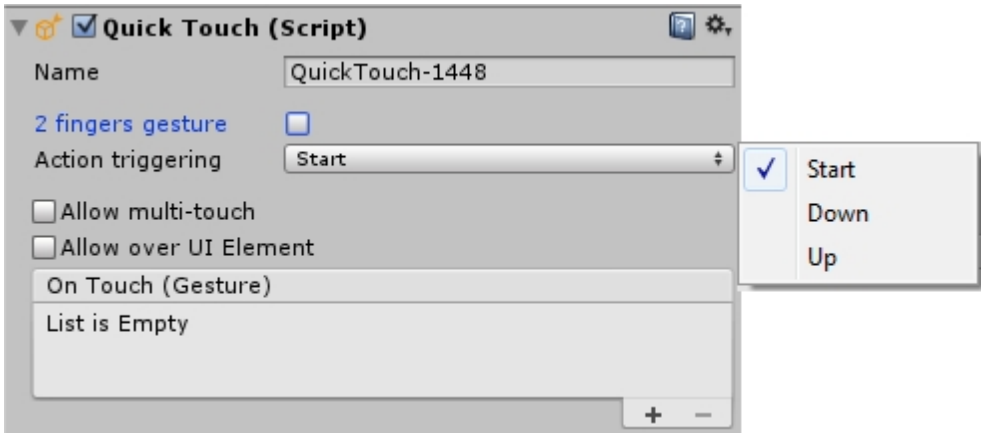
1- Click on "Add Component" button in inspector
2- Select EasyTouch
3- Select your QuickGesture



QuickTouch

QuickTouch

Allows the management of type touch gestures on a gameobject.



Setting

Name : A unique name that identifies the component
2 fingers gesture : This indicates whether the gesture should be done with 2 fingers

Action triggering : Triggering Phase relative to the gesture
Allow multi-touch : The component will be reactive in case of multi-touch on the owner
Allow over UI Element : The component will be reactive in case of touch through a UI Element

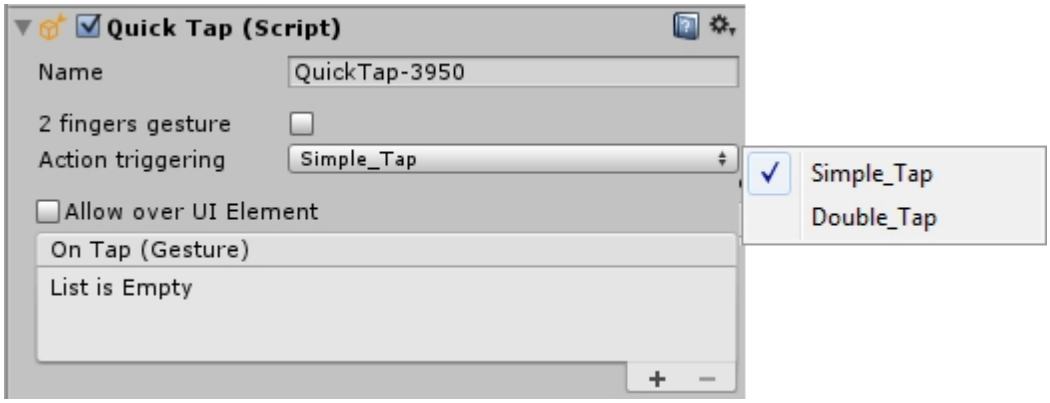
Event

On Touch : A UnityEvent that is invoked when when the component is triggered

QuickTap

QuickTap

Allows the management of type tap gestures on a gameobject.



Setting

Name : A unique name that identifies the component
2 fingers gesture : This indicates whether the gesture should be done with 2 fingers
Action triggering : Triggering Phase relative to the gesture
Allow over UI Element : The component will be reactive in case of touch through a UI Element

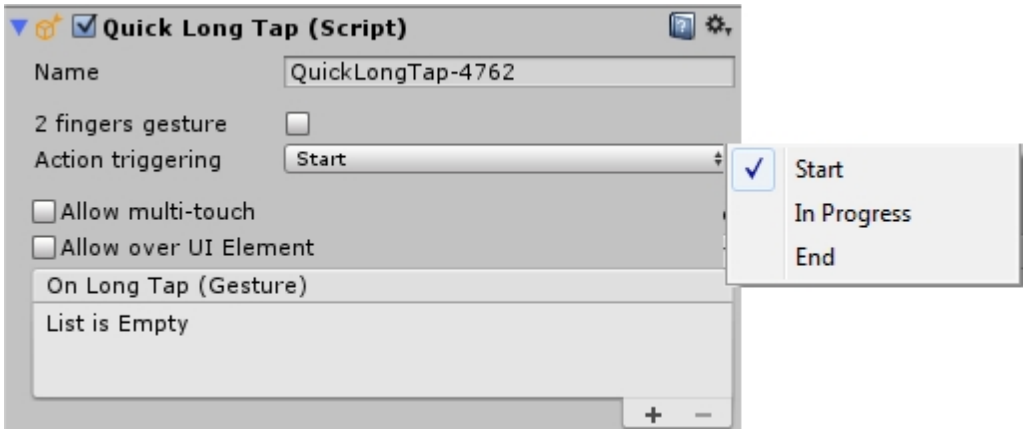
Event

On Tap : A UnityEvent that is invoked when when the component is triggered

QuickLongTap

QuickLongTap

Allows the management of type long tape gestures on a gameobject.



Setting

Name : A unique name that identifies the component
2 fingers gesture : This indicates whether the gesture should be done with 2 fingers
Action triggering : Triggering Phase relative to the gesture
Allow over UI Element : The component will be reactive in case of touch through a UI Element

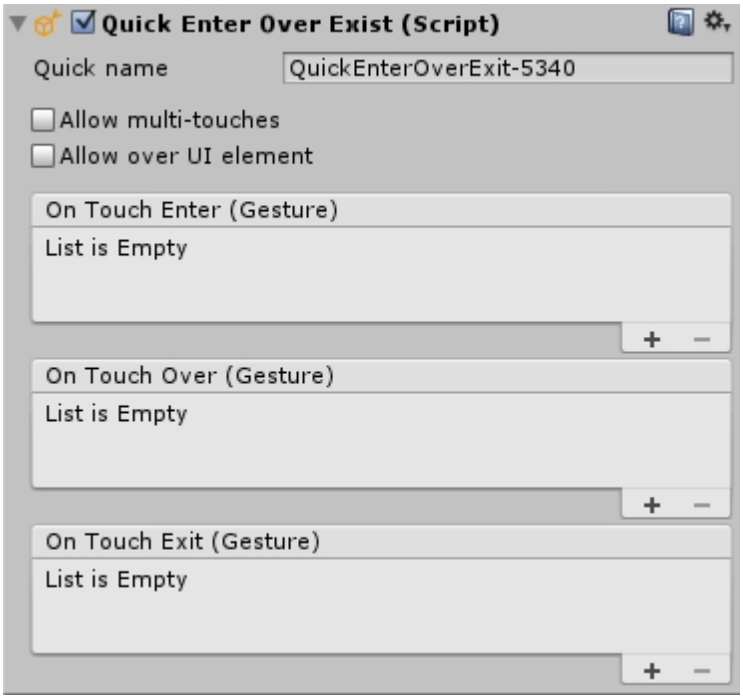
Event

On Long Tap : A UnityEvent that is invoked when when the component is triggered

QuickEnterOverExit

QuickEnterOverExit

Allows management of the different phase of a touch on a gameobject



Setting

Name : A unique name that identifies the component
Allow multi-touch : The component will be reactive in case of multi-touch on the owner
Allow over UI Element : The component will be reactive in case of touch through a UI Element

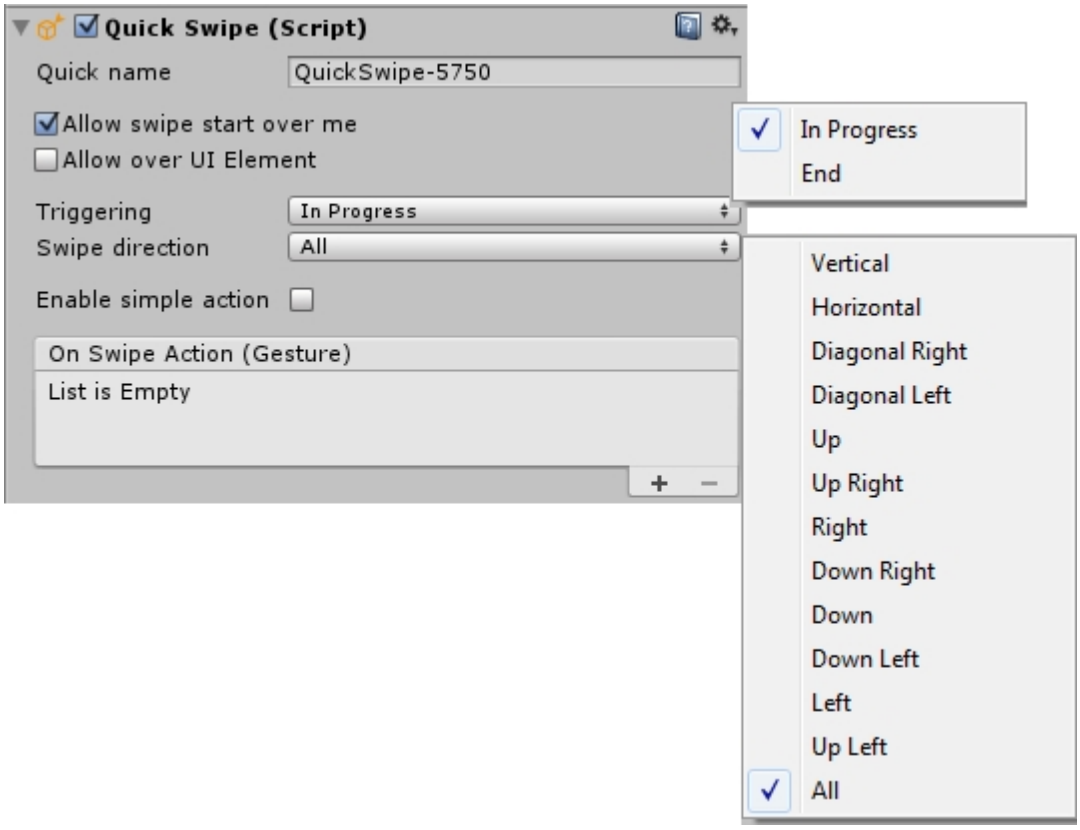
Event

On Touch Enter : A UnityEvent that is invoked when the touch enters the owner
On Touch Over : A UnityEvent that is invoked while the touch stays over the owner
On Touch Exit : A UnityEvent that is invoked when when it moves away

QuickSwipe

QuickSwipe

Allows the management of type swipe gestures for a gameobject.



Setting

Name : A unique name that identifies the component
Allow swipe start over me :Allow that the gesture start over owner
Allow over UI Element : The component will be reactive in case of touch through a UI Element
Action triggering : Triggering Phase relative to the gesture
Swipe direction : The direction of the swipe that triggers the event
Enable simple action : Activate elementary action

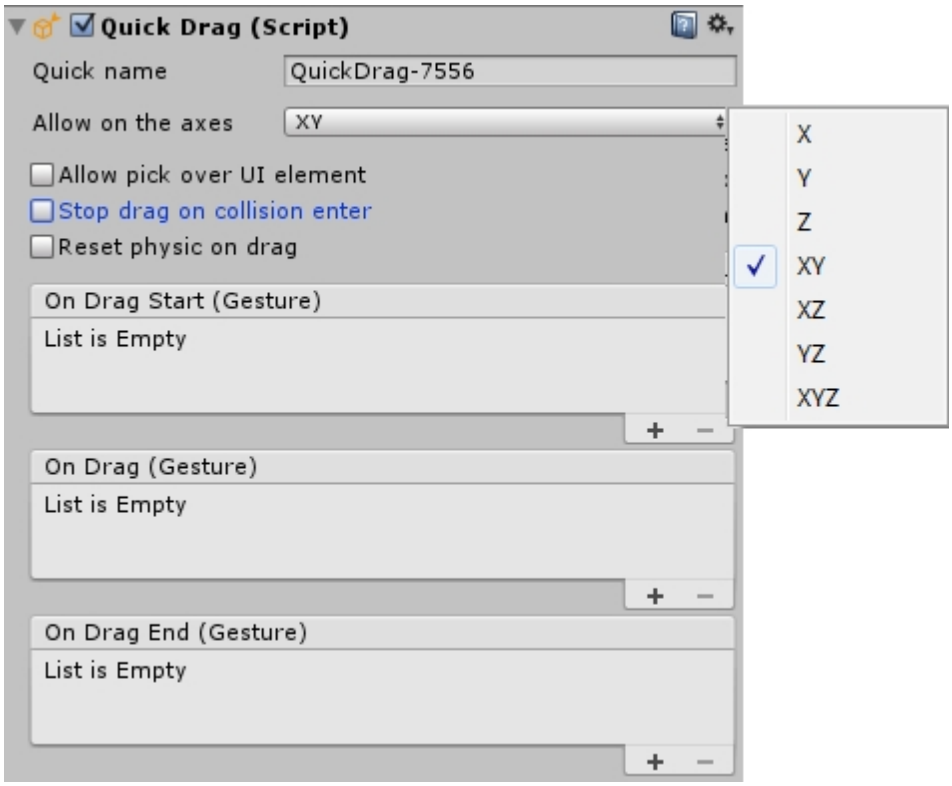
Event

On Swipe Action : A UnityEvent that is invoked when when the component is triggered

QuickDrag

QuickDrag

This component allows to drag gameobject who own it



Setting

Name **:** A unique name that identifies the component
Allow on the axes : The axes on which the drag will be allowed
Allow pick over UI Element **:** The component will be reactive in case of touch through a UI Element
Stop drag on collision enter : The curent drag action will not be active if the gameobject collides.
Reset physic on drag : The physic will be disabled during the action of drag

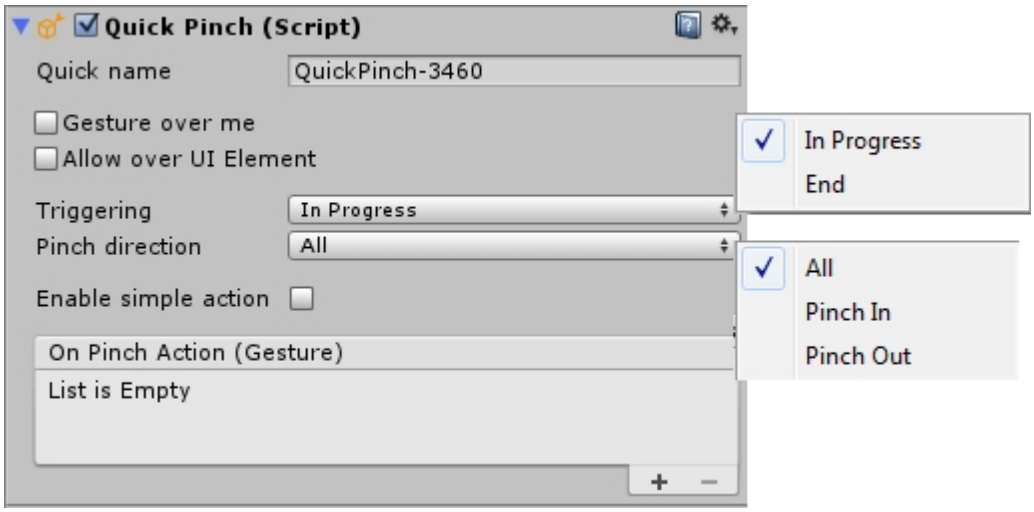
Event

On Drag Start **:** A UnityEvent that is invoked when the drag is begining
On Drag : A UnityEvent that is invoked while the drag is activated
On Drag End : A UnityEvent that is invoked when when the drag is end.

QuickPinch

QuickPinch

Allows the management of type pinch gestures for a gameobject.



Setting

Name : A unique name that identifies the component
Gesture over me : The geture must be
Allow over UI Element : The component will be reactive in case of touch through a UI Element
Action triggering : Triggering Phase relative to the gesture
Pinch direction : The direction of the pinch that triggers the event
Enable simple action : Activate elementary action

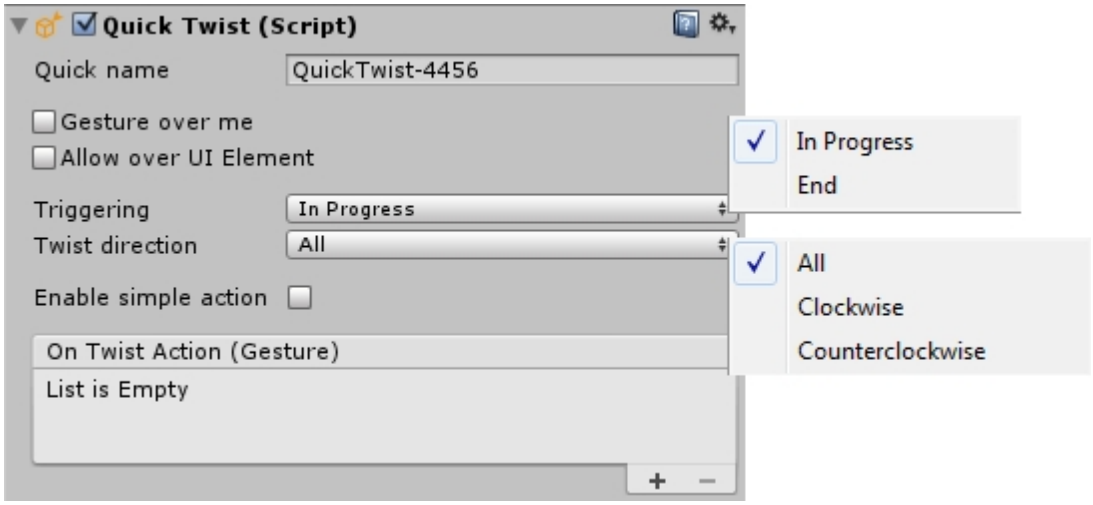
Event

On Pinch Action : A UnityEvent that is invoked when the pinch is triggered

QuickTwist

QuickTwist

Allows the management of type twist gestures for a gameobject.



Setting

Name : A unique name that identifies the component
Gesture over me : The geture must be
Allow over UI Element : The component will be reactive in case of touch through a UI Element
Action triggering : Triggering Phase relative to the gesture

Twist direction : The direction of the twisst that triggers the event
Enable simple action : Activate elementary action

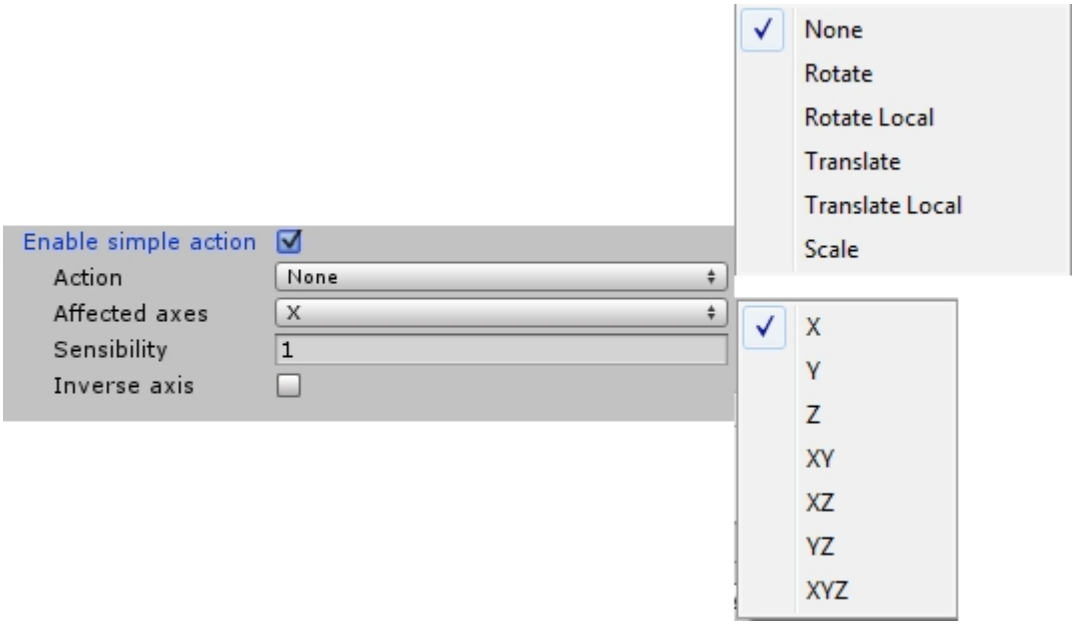
Event

On Twist Action **:** A UnityEvent that is invoked when the twist is triggered

Simple Action

Simple Action

It allows to apply a simple action on the owner based on gesture



Action : The action that will be applied
Affected axes : The axes that will be affected
Sensibility : The sensitivity with respect to gesture
Inverse axis : Inverse or not the action with respect to the axis

Add-on for Player Maker

EasyTouch 5 provides an integrated add-on for PlayMaker 1.8.

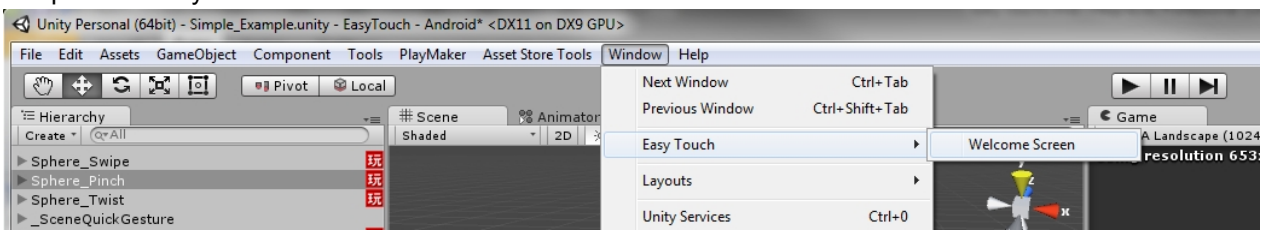
How to install
Setup your scene
Actions
Transitions

How to install

How to install

PlayMaker 1.8 must be present in your project before you begin the installation phase

1- Open the EasyTouch welcome screen



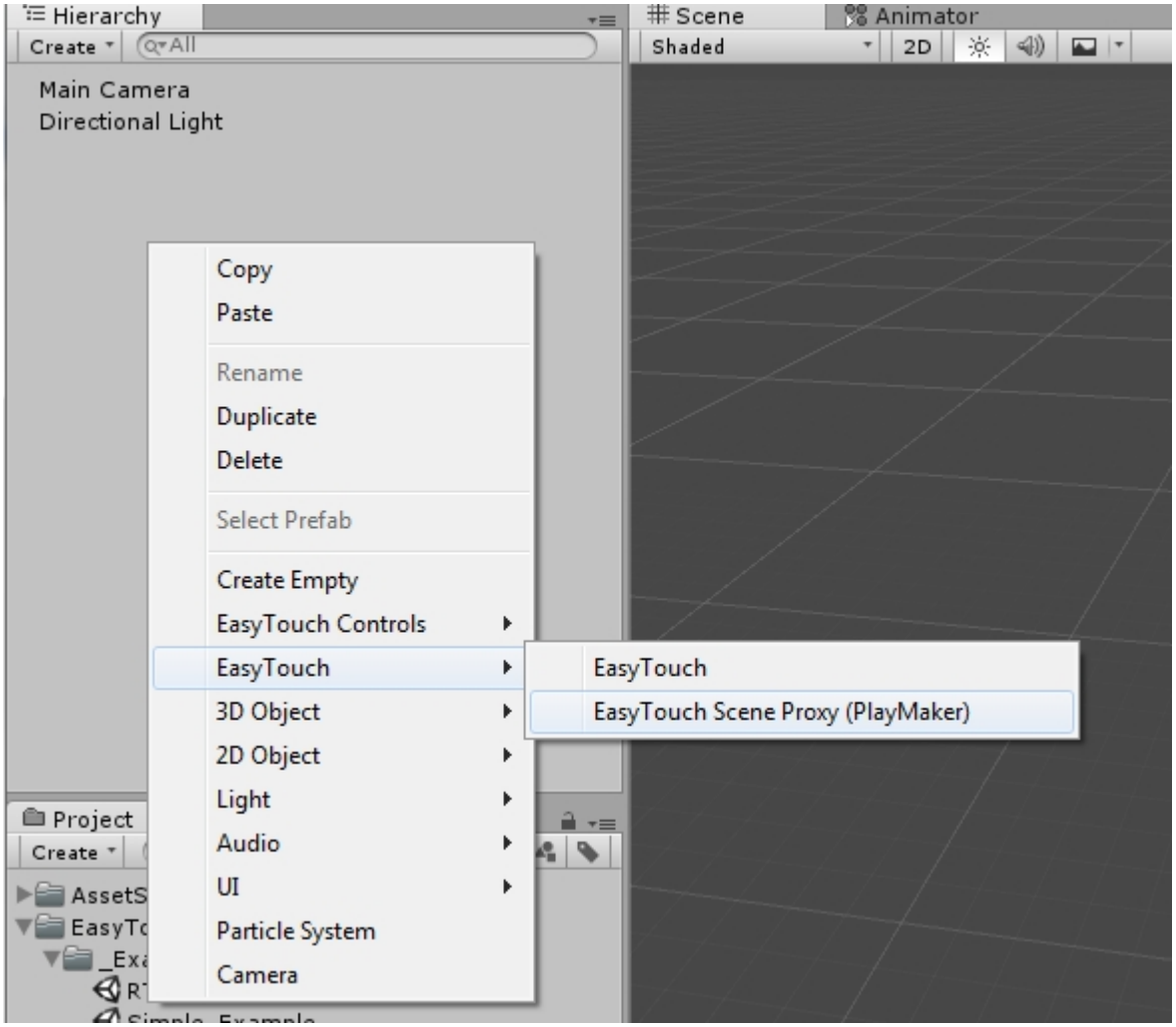
2- Click on : Install PlayMaker add-on

Setup your scene

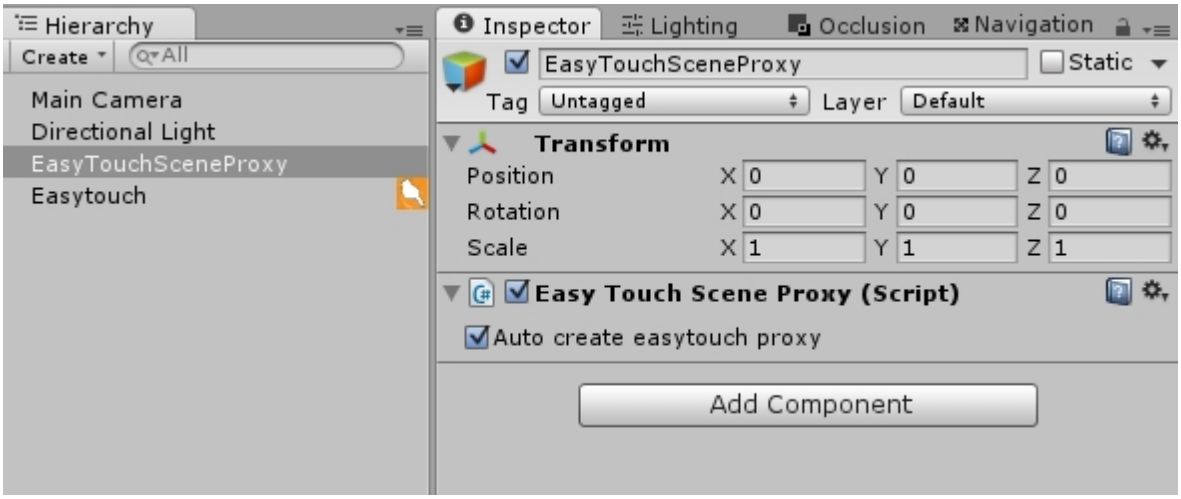
Setup your scene

You must add a the gameobject EasyTouch Scene Proxy in your scene before begin using the addon-on.

1- Right click in hierarchy or uses GameObject menu



This gameobject allows for the global transitions, and automatic adding local proxy on FSM using the add-on



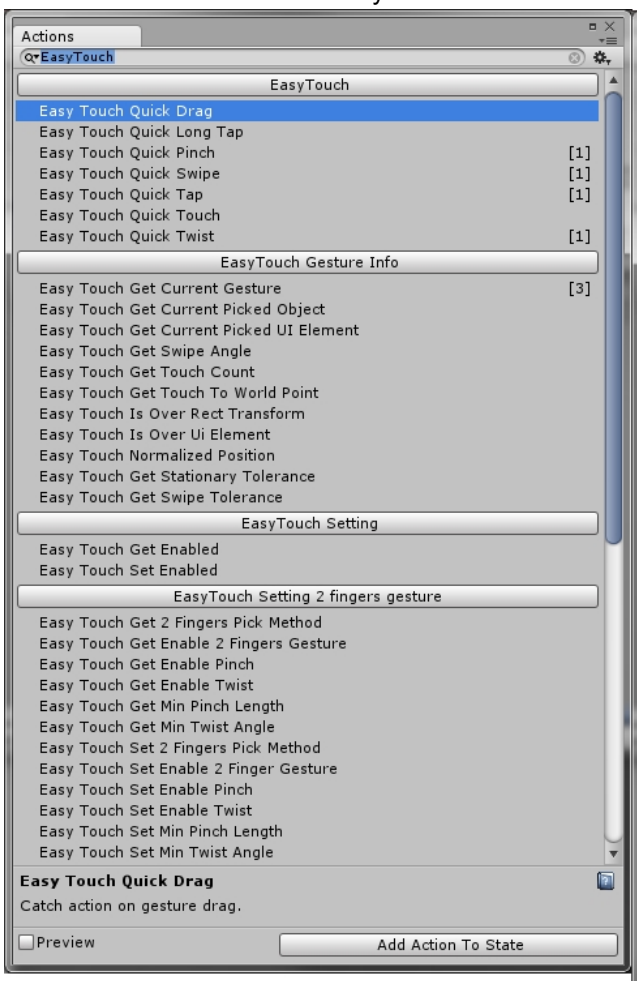
Actions

EasyTouch : Match component QuickGesture, except for Quick Drag (physically drag is not performed)

EasyTouch Gesture Info : Allows you to retrieve information about the last action performed, and

exposes the methods available in the gesture class

Other Chapter : Refer to the classic features of EasyTouch and its inspector, please refer to classical documention of EasyTouch

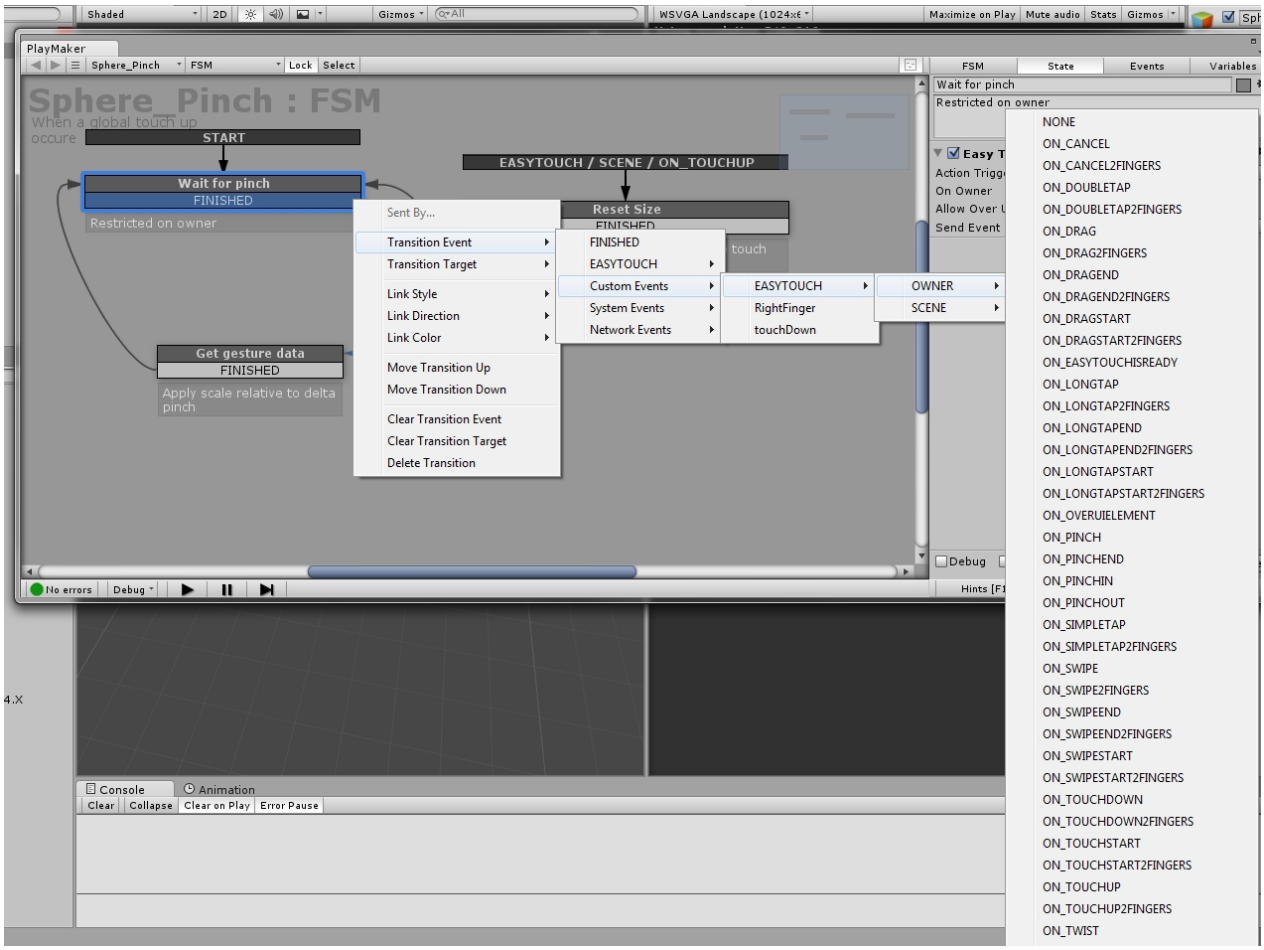


Transition

Transition

The transitions correspond to all events of EasyTouch (see classical documentation) are of two kinds, Owner & Scene.

Owner = The gesture must be perform over the owner of the FSM.
Scene = The gesture is received in all cases.


PlayMaker
Sphere_Pinch : FSM
When a global touch up occure
START
Wait for pinch
FINISHED
Restricted on owner
EASYTOUCH / SCENE / ON_TOUCHUP
Reset Size
FINISHED
Get gesture data
FINISHED
Apply scale relative to delta pinch
Sent By...
Transition Event
Transition Target
Link Style
Link Direction
Link Color
Move Transition Up
Move Transition Down
Clear Transition Event
Clear Transition Target
Delete Transition
FINISHED
EASYTOUCH
Custom Events
System Events
Network Events
EASYTOUCH
RightFinger
touchDown
OWNER
SCENE
NONE
ON_CANCEL
ON_CANCEL2FINGERS
ON_DOUBLETAP
ON_DOUBLETAP2FINGERS
ON_DRAG
ON_DRAG2FINGERS
ON_DRAGEND
ON_DRAGEND2FINGERS
ON_DRAGSTART
ON_DRAGSTART2FINGERS
ON_EASYTOUCHISREADY
ON_LONGTAP
ON_LONGTAP2FINGERS
ON_LONGTAPEND
ON_LONGTAPEND2FINGERS
ON_LONGTAPSTART
ON_LONGTAPSTART2FINGERS
ON_OVERUIELEMENT
ON_PINCH
ON_PINCHEND
ON_PINCHIN
ON_PINCHOUT
ON_SIMPLETAP
ON_SIMPLETAP2FINGERS
ON_SWIPE
ON_SWIPE2FINGERS
ON_SWIPEEND
ON_SWIPEEND2FINGERS
ON_SWIPESTART
ON_SWIPESTART2FINGERS
ON_TOUCHDOWN
ON_TOUCHDOWN2FINGERS
ON_TOUCHSTART
ON_TOUCHSTART2FINGERS
ON_TOUCHUP
ON_TOUCHUP2FINGERS
ON_TWIST