

EasyTouch 2.3



User documentation

Table of contents

Introduction
What is EasyTouch
What's new in 2.0
Quick Start
Quick Start (Javascript)
Quick Start with auto-select (Javascript)
Quick Start (C#)
Quick Start with auto-select (C#)
Examples scenes
Concept
Event system
Auto-Select
Inspector properties
Inspector
Classes
Events
EasyTouch Class
Gesture Class

Introduction

What is EasyTouch

EasyTouch allow you to quickly and easily develop actions based on a touchscreen or a mouse .

All major gestures are recognized such as touch, tap, double tap, twist, pinch.. and much more.

EasyTouch allows automatic detection of all gameobjects with collider on collision layers that you can set, To allow easy handling of the various elements that make up your scene

EasyTouch 2.0 is written in C# , it notifies you of an action on the touch screen or with mouse by events. The events are sent by delegate system for C # or internal messaging function of Unity (SendMessage) for Javascript developers.

Each event passes a parameter class, with all the informations about the current action (position, angle, picked object ...)

EasyTouch simulates the second finger when you press the ALT or CTRL key, an orange circle appears on the screen to the simulated position.

ALT => For twist or pinch gesture

CTRL => For swipe or drag gesture

What's new in 2.4 & 2.5

News

- Add static method IsRectUnderTouch : to get if a touch is in a rect.
- New inspector style for pro & free skin
- Add hierarchy icon to identify EasyTouch gameObject
- Remove string comparisons by enumeration, for better performance

Bugs fixed

- Fix 2 static methods that didn't properly reference the EasyTouch instance

What's new in 2.3

News

- Added support for the Unity Remote (tested on iPad & Nexus7)
Thank you to fulvio Massini for the support he has given us to implement this functionality

What's new in 2.1 & 2.2

News

- - Add new Static method : GetCurrentPickedObject(int fingerIndex) taht return the current gameobject under touch Look at CameController example.

Bugs fixed

- On_TouchStart & On_TouchTap events and are no longer sent after the end of a two-fingers gesture

What's new in 2.0

News

- C# migration
- Implementing delegate for sending messages. (Broadcast messages is retained with a parameter for javascript developers)
- Management of multiple layer for the auto selection
- Management of fake singleton, in case you have more than one EasyTouch per scene by error
- Add custom inspector
- Add Debug.LogError if no camera with flag MainCamera was found in the scene
- New documentation

EasyTouch class

- remove SetPickableLayer & GetPickableLayer static methods
- Add static method GetTouchCount : to get the number of touches.

Gesture class

- Add method (GetScreenToWorldPoint(Camera cam,float z) that return the world coordinate position for a camera and z position
- Add method (GetSwipeRrDragAngle()) that return the swipe or drag angle in degree

Bugs fixed

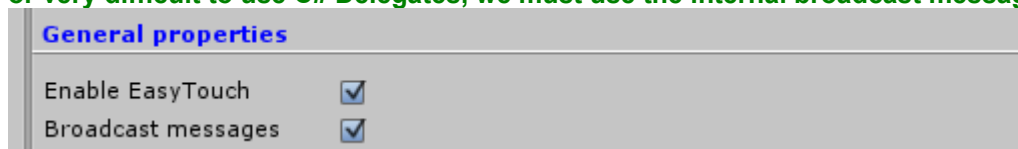
- On_DragEnd2Fingers and On_SwipeEnd2Fingers messages were sent to wrong during a drag or a swipe.
- On_Cancel2Fingers is now sent to the picked object (if auto-select)

Quick Start

Quick Start (Javascript)

- 1- Import EasyTouch Package.
- 2- Place the plugins directory in the root of your project (This action allows you to use C # classes in Javascript)
- 3- Create an empty gameObject, and name it EasyTouch.(You can choose another name)
- 4- Add the EasyTouch.cs script on the EasyTouch gameObject that you just created.
- 5- Select gameObject EasyTouch, and set to TRUE Broadcast messages in the inspector.

EasyTouch is set by default to send messages by C # events. In javascript it is impossible or very difficult to use C# Delegates, we must use the internal broadcast messages of Unity.



- 6- Create a new javascript

- 7- Add this fonction :

```
// Touch start event
function On_TouchStart( gesture:Gesture){

    Debug.Log( "Touch in " + gesture.position);
}
```

- 8- Add this script on the EasyTouch gameObject

In BroadCast messages mode, the events are sent to the gameObject with the EasyTouch script. (If no gameObject has been selected with the auto-select)

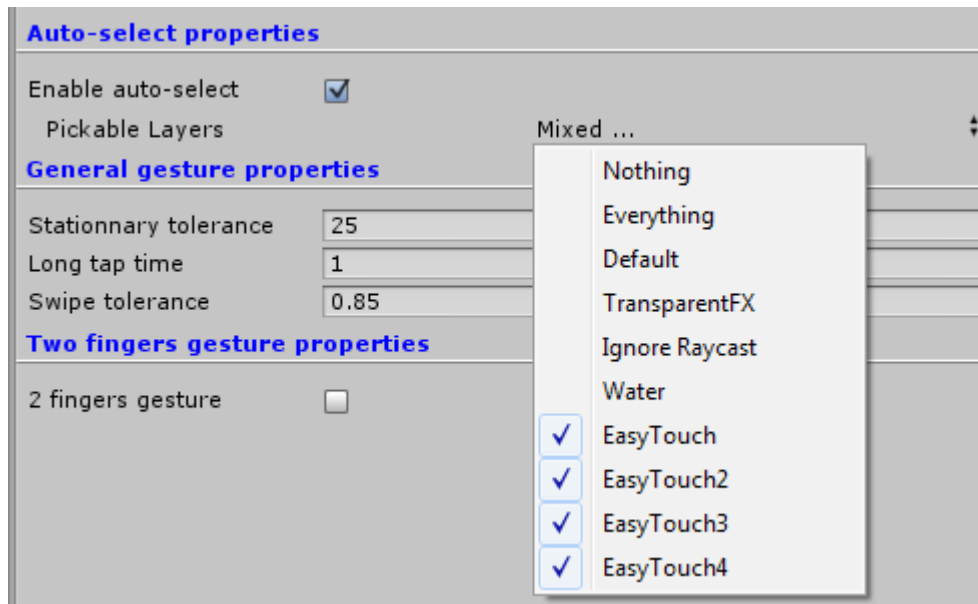
- 9- Run it in editor, and click on the screen

This example will do nothing if you run it on your mobile devise, it's just to show you how to use EasyTouch events with javascript.

Quick Start with auto-select (Javascript)

1- Do steps 1 to 5 of the Quick Start (Javascript)

2- Select the EasyTouch gameobject , and add a pickable layers in the Auto-select properties in the inspector



3- Create a sphere and assign it a simple diffuse material

4- Setting the auto-select on the sphere : Assign the same layer that you assign as parameter to the sphere.

5- Create a new javascript

6- Add this function :

```
function On_TouchStart( gesture:Gesture){  
    gameObject.renderer.material.color = Color( Random.Range(0.0,1.0),  
    Random.Range(0.0,1.0), Random.Range(0.0,1.0));  
}
```

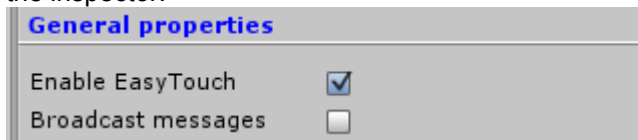
7- Add this new script to the **sphere**

In BroadCast messages mode, If EasyTouch auto-selects a gameobject, it sends the event to this object and not the holder of the EasyTouch script

8- Run it in editor mode or in your device. If you touch the sphere or click on it, it will change color.

Quick Start (C#)

- 1- Import EasyTouch Package.
- 2- Create an empty gameObject, and name it EasyTouch.(You can choose another name)
- 3- Add the EasyTouch.cs script on the EasyTouch gameObject that you just created.
- 4- Select the EasyTouch gameObject, and verifies that Broadcast messages is set to FALSE in the inspector.



- 5- Create a new C# script MyFirstTouch
- 6- Add these methods

```
// 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;
}

// Touch start event
public void On_TouchStart(Gesture gesture){
    Debug.Log( "Touch in " + gesture.position);
}
```

- 7- Create an empty gameObject, and name it Receiver.
- 8- Add MyFirstTouch script to the gameObject Receiver.
- 9- Run it in editor, and click on the screen

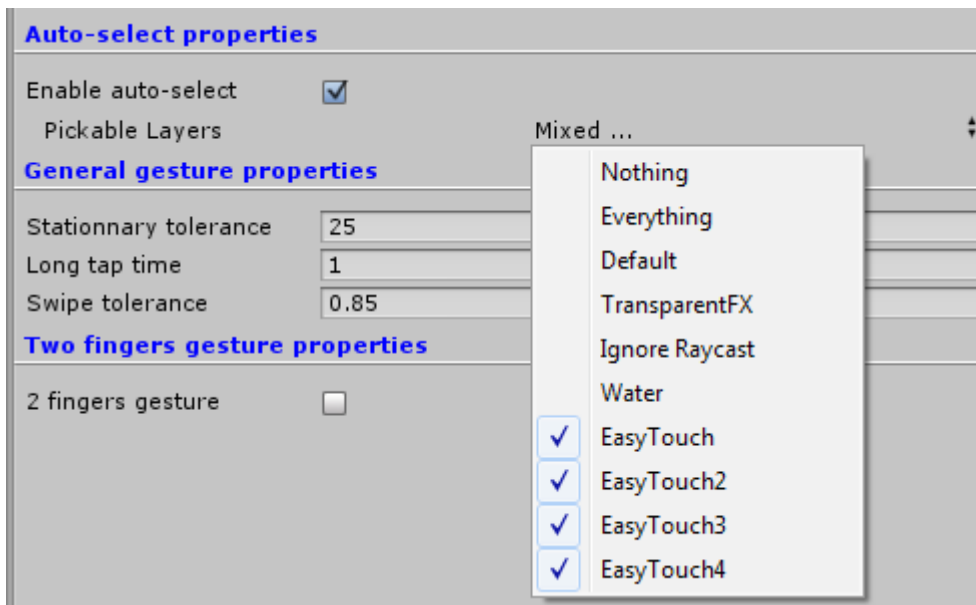
This example will do nothing if you run it on your mobile device, it's just to show you how to use EasyTouch events with C#.

With C# events all script that are subscribed to an event will receive the event

Quick Start with auto-select (C#)

1- Do steps 1 to 4 of the Quick Start (C#)

2- Select the EasyTouch gameobject , and add a pickable layers in the Auto-select properties in the inspector



3- Create a sphere and assign it a simple diffuse material

4- Setting the auto-select on the sphere : Assign the same layer that you assign as parameter to the sphere.

5- Create a new C#

6- Add these methods :

```
// 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){

    // Verification that the action on the object
    if (gesture.pickObject == 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));
}
```


7- Add this script to the sphere.

8- Run it in editor mode or in your device. If you touch the sphere or click on it, it will change color.

Examples scenes

EasyTouch comes with several C# examples with Event-Delegate notification (examples folder).

OneFinger and TwoFinger examples let you watch a simple implementation of each event.

Concepts

Event system

When you make an action with the touch screen or mouse, EasyTouch raise an event corresponding to the current action. Each event will have a parameter of type [Gesture](#).

You only have to write the code according to the gesture, as in a motor GUI events.

Events can be raised in two different ways (look at [general properties inspector](#))

- Event / Delegate system.
- Unity built-in Sending message system (SendMessage)

To help you make your choice :

	Unity built-in SendMessage	Event / Delegate system
Advantage	<ul style="list-style-type: none">• Events are sent to the object selected, simply add a script with a method corresponding to the event processing• Simplify implementation with respect to the event Delegate	<ul style="list-style-type: none">• Faster than Unity built-in SendMessage• Notify several objects at once
Disadvantage	<ul style="list-style-type: none">• Slower than Event-Delegate• Only two objects can be notified at the same time. <p>The holder of EasyTouch script if no object is selected, or the selected object.</p> <p>Another gameObject at the discretion of the developer via method SetOtherReceiverObject</p>	<ul style="list-style-type: none">• The event is sent to all objects that subscribe to it• More line of code is required (subscribe and unsubscribe event)• A test condition on the member PickedObject(Gesture class) must be carried out in the beginning of each treatment in the event• not compatible with javascript

Auto-Select

When you perform an action on the touch screen or mouse, EasyTouch can detect if the action is performed on a gameObject.

If gameObject is detected, it will be assign to the class member [pickedObject of the Gesture class](#) which will be sent with the [event](#). With the [broadcast message mode](#) (built in unity), the event itself will be sent to this object.

This setting is enabled by default ([Inspector properties](#)), you just indicate the layers that can be selected.

Warning:

With Event-Delegate mode, events are sent to all objects that have subscribed to the event, unlike the broadcast mode which sends the object itself.

Even if the script that processes an event is attached to a selectable object, you need to test the `pickedObject` member.

```
public void On_TouchStart(Gesture gesture){  
    // Verification  
    if (gesture.pickedObject == gameObject)  
        gameObject.renderer.material.color = Color.red;  
}
```

Inspector properties

EasyTouch has different parameters that you can customize with the inspector, the default settings work correctly on all mobile platforms.

All these properties can be setting with script, look at : [EasyTouch Class](#)

General properties

Enable EasyTouch ☒

Enable unity remote ☐

Broadcast messages ☐

Auto-select properties

Enable auto-select ☒

Pickable Layers EasyTouch

General gesture properties

Stationnary tolerance 25

Long tap time 1

Swipe tolerance 0.85

Two fingers gesture properties

2 fingers gesture ☒

Enable Pinch ☒

Min pinch length 0

Enable twist ☒

Min twist angle 1

[General properties](#)

Enable EasyTouch : Enables ou disables EasyTouch

Enable unity remote : Enables ou disables the support og Unity Remote

Broadcast messages : True = The events will be sent with then internal messaging function of Unity (SendMessage). Use this if you develop in Javascript (you can use in C# too, if you want)

False = The events will be sent with events/Delegate C# system, never use this mode with Javascript

[Auto-select properties](#)

Enable auto-select : Enables or disables the auto-select of gameobject under the touch.

Pickable Layers : To set up all layer that EasyTouch must test

[General gesture properties](#)

Stationnary tolerance : Distance below which a finger movemenst generate static touch, Act on this value if the tap are not detected correctly. The default value works well on a IPAD2

Long tap time : Represents the maximum time to generate an event tap, or the minimum time to generate a long tap event

Swipe tolerance : This value is used to detect the orientation of a gesture swipe or drag. It must be between 0 and 1

0 => Gesture imprecise

1 => Gesture very precise

Tow fingers gesture properties

2 Fingers gesture : Enables or disables the recognize of 2 fingers gesture

Enable Pinch : Enables or disables the recognize of pinch gesture

Min pinch lenght : The minimum distance that fingers must travel to detect a pinch gesture

Enable twist : Enables or disables the recognize of twist gesture.

Min twist angle : The minimum angle that fingers must travail to detect a twist gesture

Classes

Events

On_Cancel([Gesture gesture](#))

Occurs when The system cancelled tracking for the touch, as when (for example) the user puts the device to her face.

On_Cancel2Fingers([Gesture gesture](#))

Occurs when the touch count is no longer equal to 2 and different to 0, after the beginning of a two fingers gesture.

On_TouchStart([Gesture gesture](#))

Occurs when a finger touched the screen.

On_TouchDown([Gesture gesture](#))

Occurs as the touch is active.

On_TouchUp([Gesture gesture](#))

Occurs when a finger was lifted from the screen.

On_SimpleTap([Gesture gesture](#))

Occurs when a finger was lifted from the screen, and the time elapsed since the beginning of the touch is less than the time required for the detection of a long tap.

On_DoubleTap([Gesture gesture](#))

Occurs when the number of taps is equal to 2 in a short time.

On_LongTapStart([Gesture gesture](#))

Occurs when a finger is touching the screen, but hasn't moved since the time required for the detection of a long tap.

On_LongTap([Gesture gesture](#))

Occurs as the touch is active after a LongTapStart

On_LongTapEnd([Gesture gesture](#))

Occurs when a finger was lifted from the screen, and the time elapsed since the beginning of the touch is more than the time required for the detection of a long tap.

On_DragStart([Gesture gesture](#))

Occurs when a drag start. A drag is a swipe on a pickable object

On_Drag([Gesture gesture](#))

Occurs as the drag is active.

On_DragEnd([Gesture gesture](#))

Occurs when a finger that raise the drag event , is lifted from the screen.

On_SwipeStart([Gesture gesture](#))

Occurs when swipe start.

On_Swipe([Gesture gesture](#))

Occurs as the swipe is active.

On_SwipeEnd([Gesture gesture](#))

Occurs when a finger that raise the swipe event , is lifted from the screen.

[On_TouchStart2Fingers\(Gesture gesture\)](#)

Like On_TouchStart but for a 2 fingers gesture.

[On_TouchDown2Fingers\(Gesture gesture\)](#)

Like On_TouchDown but for a 2 fingers gesture.

[On_TouchUp2Fingers\(Gesture gesture\)](#)

Like On_TouchUp but for a 2 fingers gesture.

[On_SimpleTap2Fingers\(Gesture gesture\)](#)

Like On_SimpleTap but for a 2 fingers gesture.

[On_DoubleTap2Fingers\(Gesture gesture\)](#)

Like On_DoubleTap but for a 2 fingers gesture.

[On_LongTapStart2Fingers\(Gesture gesture\)](#)

Like On_LongTapStart but for a 2 fingers gesture.

[On_LongTap2Fingers\(Gesture gesture\)](#)

Like On_LongTap but for a 2 fingers gesture.

[On_LongTapEnd2Fingers\(Gesture gesture\)](#)

Like On_LongTapEnd but for a 2 fingers gesture.

[On_Twist\(Gesture gesture\)](#)

Occurs when a twist gesture start

[On_TwistEnd\(Gesture gesture\)](#)

Occurs as the twist gesture is active.

[On_PinchIn\(Gesture gesture\)](#)

Occurs as the twist in gesture is active.

[On_PinchOut\(Gesture gesture\)](#)

Occurs as the pinch out gesture is active.

[On_PinchEnd\(Gesture gesture\)](#)

Occurs when the 2 fingers that raise the pinch event , are lifted from the screen.

[On_DragStart2Fingers\(Gesture gesture\)](#)

Like On_DragStart but for a 2 fingers gesture.

[On_Drag2Fingers\(Gesture gesture\)](#)

Like On_Drag but for a 2 fingers gesture.

[On_DragEnd2Fingers\(Gesture gesture\)](#)

Like On_DragEnd2Fingers but for a 2 fingers gesture.

[On_SwipeStart2Fingers\(Gesture gesture\)](#)

Like On_SwipeStart but for a 2 fingers gesture.

[On_Swipe2Fingers\(Gesture gesture\)](#)

Like On_Swipe but for a 2 fingers gesture.

[On_SwipeEnd2Fingers\(Gesture gesture\)](#)

Like On_SwipeEnd but for a 2 fingers gesture.

EasyTouch Class

This is the main class, you need to add it to your main camera or on a empty gameobject in your scene.

Member Enumeration Documentation

enum SwipeType{ None, Left, Right, Up, Down, Other }

Represents the different directions for a swipe or drag gesture.

Static Method Documentation

static Camera GetCamera ()

Return the camera used by EasyTouch for the auto-selection. EasyTouch research at start the camera with the tag
MainCamera.

Returns

The camera

static bool GetEnable2FingersGesture ()

Return if 2 fingers gesture is enabled or disabled

Returns

true = enabled

false = disabled

static bool GetEnableAutoSelect ()

Return if auto select is enabled or disabled

Returns

true = enabled false = disables

static bool GetEnabled ()

Return if EasyTouch is enabled or disabled

Returns

True = Enabled False = Disabled

static bool GetEnablePinch ()

Return if 2 pinch gesture is enabled or disabled

Returns

true = enabled

false = disables

static bool GetEnableTwist ()

Return if 2 twist gesture is enabled or disabled

Returns

true = enabled

false = disabled

static float GetlongTapTime ()

Return the long the tap time.

Returns

the long tap time

static float GetMinPinchLength ()

Return the minimum length of the pinch.

Returns

the minimum length

static float GetMinTwistAngle ()

Gets the minimum twist angle.

Returns

the minimum twist angle

static GameObject GetOtherReceiverObject ()

Return the other event receiver.

Returns

GameObject

static float GetStationnaryTolerance ()

Return the stationnary tolerance.

Returns

the stationnary tolerance

static float GetSwipeTolerance ()

Return the swipe tolerance.

Returns

the swipe tolerance.

static int GetTouchCount ()

Return the current touches count.

Returns

int

static void GetCurrentPickedObject(int fingerIndex)

Gets the current picked object.

Parameters

fingerindex int

Returns

gameobject

static void SetEnable2FingersGesture (bool enable)

Enables or disables the recognize of 2 fingers gesture.

Parameters

enable true = enabled

false = disabled

static void SetEnableAutoSelect (bool enable)

Enables or disables auto select.

Parameters

enable true = enabled

false = disables

static void SetEnabled (bool enable)

Enables or disables Easy Touch.

Parameters

enable true = enabled

false = disabled

static void SetEnablePinch (bool enable)

Enables or disables the recognize of pinch gesture

Parameters

enable true = enabled

false = disables

static void SetEnableTwist (bool enable)

Enables or disables the recognize of twist gesture

Parameters

enable true = enabled

false = disables

static void SetLongTapTime (float time)

Set the long tap time in second

Parameters

time Float

static void SetMinPinchLength (float length)

Sets the minimum length of the pinch.

Parameters

length Float.

static void SetMinTwistAngle (float angle)

Sets the minimum twist angle.

Parameters

angle Float

static void SetOtherReceiverObject (GameObject receiver)

Sets the other receiver for EasyTouch event.

Parameters

receiver GameObject.

static void SetStationnaryTolerance (float tolerance)

Sets the stationnary tolerance.

Parameters

tolerance float Tolerance.

static void SetSwipeTolerance (float tolerance)

Sets the swipe tolerance.

Parameters

tolerance Float

static static bool IsRectUnderTouch(Rect rect)

Determines whether a touch is under a specified rect

Parameters

rec Rect

Returns

True if this a touch is under a specified rect, otherwise, false

Gesture Class

This is the class passed as parameter by EasyTouch events, that containing all informations about the touch that raise the event, or by the tow fingers gesture that raise the event.

Public Member Functions

Vector3 GetTouchToWorldPoint (float z)

Transforms touch position into world space, or the center position between the two touches for a two fingers gesture.

Returns

Vector3 : world position

Parameters

z The z position in world units from the camera

float GetSwipeOrDragAngle ()

Gets the swipe or drag angle. (calculate from swipe Vector)

Returns

Float : The swipe or drag angle.

Public Attributes

int fingerIndex

The index of the finger that raise the event (Starts at 0), or -1 for a two fingers gesture.

int touchCount

The touches count.

Vector2 startPosition

The start position of the current gesture, or the center position between the two touches for a two fingers gesture.

Vector2 position

The current position of the touch that raise the event, or the center position between the two touches for a two fingers gesture.

Vector2 deltaPosition

The position delta since last change.

float actionTime

Time since the beginning of the gesture.

float deltaTime

Amount of time passed since last change.

EasyTouch.SwipeType swipe

The siwpe or drag type (look at [SwipeType enumeration](#)).

float swipeLength

The length of the swipe.

Vector2 swipeVector

The swipe vector direction.

float deltaPinch

The pinch length delta since last change.

float twistAngle

The angle of the twist.

GameObject pickObject

The current picked gameObject under the touch that raise the event.

GameObject otherReceiver

Other receiver of the event.