EasyTouch组件：

Enable EasyTouch：启用或关闭EasyTouch功能，如果关闭，则所有事件绑定失效

Enable Unity Remote：是否允许与UnityRemote这个手机远程调试插件进行关联，记得在真正导出项目到手机的时候去掉这个勾勾

EasyTouch.GUI Compatibility

Enable Uniyt UI detection：启用或禁用对UGUI进行手势检测

Unity UI compatibility：

启用：如果粘键（同时触摸查过一个UGUI）则UGUI不会被检测到（手势事件不会被触发）

禁用：不管有没有粘键，都会触发手势事件，并且会将手势事件所在的类Gesture中的isOverUIElement这个属性设置为true

Auto update picked Unity UI：是否需要EasyTouch每一帧都去更新触摸检测到的所有UI数据

Enable NGUI Compatibility：是否启用对NGUI所属的层对象和NGUI摄像机的检测

NGUI Layers：哪个层是NGUI对象所在的层

Camera：哪个摄像机是NGUI摄像机

EasyTouch.Automatic selection:

Enable auto-select:是否对EasyTouch的自动检测规则进行设置

pickable layers 3D：对哪个层所在的物体进行检测（如果物体是预制体拉到场景中生成的时候，必须设置该项为true，不然easytouch识别不了）

Auto update picked gameobject：启用：EasyTouch每一帧都去更新触摸检测到的所有游戏物体的所属数据（例如触碰到什么，现在你的手指所在的屏幕坐标......，具体的可查看Gesture类）

禁用：有些游戏物体的数据则可能不会被实时得到，例如手指所在的屏幕坐标，通俗来说就是当你手指持续触摸屏幕的时候，Gesture中会有些数据不会实时更新，而这些数据只会在你离开停止触摸屏幕，然后在下一次触摸屏幕的那一刻只更新一次！

Enable 2D collider：是否让2D的且具有碰撞体的游戏物体也能被Easytouch检测到

pickable 2D layers：哪个层的2D游戏物体能够被检测到

EasyTouch.General gesture properties:

Priority to:你要的手势是点击的还是滑动的

Stationary tolerance:这是点击手势，其手势直径的容错设置，建议保持默认值

Long tap time:当点击手势保持多长时间（长按多少秒）才触发检测

Double tap time：双击的时候，以哪个时间为准，建议保持默认值

Swipe tolerance：在滑动手势发生时，至少滑动多长距离才算是滑动，建议保持默认值

Always sent swip event：在滑动过程中，是否需要不管滑动距离是否达到Swipe tolerance所需要的最少距离，都不断发送（触发）滑动事件(注意：建议选false，因为触发频率过高的话，可能会导致机子吃不消)

EasyTouch.Two fingers gesture properties:

2 Fingers gesture：是否支持双手指的手势

Pick method：以哪种方式为标准进行双手势识别，选Figure则手指必须要大面积的覆盖（触碰）到游戏物体，选Average则当你的手指同时碰到两个游戏物体的时候，计算位置平均值来选择你的手指所要选择的物体，个人建议选择Figure

Enable swipe & drag:是否启用滑动或拖动手势

Enable Pinch:是否启用缩放手势

Min pinch length:用于检测缩放手势的像素点的最小阀值距离

Enable twist:是否启用扭曲手势

Min twist angle：用于检测扭曲手势的捻度的最小阀值角

EasyTouch.Second finger simulation

Enable simulation:是否开启第二手指的模拟体

Texture：第二手指的模拟体用哪张图片显示

Twist & pinch key / Swipe key：按键盘的哪个键去激活第二手指的模拟体

类：

EasyTouch类：

枚举：

GesturePriority（手势优先级）：按下还是滑动

SwipeDirection （滑动屏幕的方向）：没有，左，右，上，下，上左，上右，下左，下右，其他

TwoFingerPickMethod（双手指检测方法）：是用两只手指头进行触摸检测还是用平均值进行触摸检测

事件：

单手指事件：

On\_Cancel ：当系统取消跟踪触摸的时候发生

On\_TouchStart：当手指刚触摸到屏幕时发生

On\_TouchDown：当触摸正式被激活的时候发生

On\_TouchUp ：当手指刚离开屏幕那一刻发生

On\_SimpleTap：也是当手指刚离开屏幕时发生

On\_DoubleTap：当手指连续按下两次的时候发生

On\_LongTapStart ：当手指按下那一刻发生，但不能有移动直到所设置的长按的所需时间到了

On\_LongTap ：当手指按在屏幕上达到长按所需的时间后则被认为长按有效，也就是此时发生，所以触发时机在On\_LongTapStart之后

On\_LongTapEnd ：当手指完成有效长按时，抬起的那一刻发生

On\_DragStart ：当开始拖动可选对象那一刻发生

On\_Drag ：当拖动正式被激活时发生

On\_DragEnd：当不再拖动可选物体，手指在离开屏幕那一刻发生

On\_SwipeStart：手指开始滑动屏幕的那一刻发生

On\_Swipe：手指滑动屏幕正式被激活的时候发生

On\_SwipeEnd：手指滑动完屏幕，离开屏幕的那一刻发生

双手指事件：

On\_TouchStart2Fingers：就像On\_TouchStart，但要用两个手指去触发

On\_TouchDown2Fingers ：就像On\_TouchDown，但要用两个手指去触发

On\_TouchUp2Fingers：就像On\_TouchUp，但要用两个手指去触发

On\_SimpleTap2Fingers：就像On\_SimpleTap，但要用两个手指去触发

On\_DoubleTap2Fingers：就像On\_DoubleTap，但要用两个手指去触发

On\_LongTapStart2Fingers：就像On\_LongTapStart，但要用两个手指去触发

On\_LongTap2Fingers：就像On\_LongTap，但要用两个手指去触发

On\_LongTapEnd2Fingers：就像On\_LongTapEnd，但要用两个手指去触发

On\_Twist：当扭动手势发生时触发

On\_TwistEnd：当扭动手势结束时触发

On\_Pinch：当缩放手势被正式激活时触发

On\_PinchIn：当缩放手势是缩小手势时触发

On\_PinchOut：当缩放手势是扩大手势时触发

On\_PinchEnd：当缩放手势结束并且离开屏幕的那一刻触发

On\_DragStart2Fingers ：就像On\_DragStart，但要用两个手指去触发

On\_Drag2Fingers：就像On\_Drag，但要用两个手指去触发

On\_DragEnd2Fingers：就像On\_DragEnd，但要用两个手指去触发

On\_SwipeStart2Fingers：就像On\_SwipeStart，但要用两个手指去触发

On\_Swipe2Fingers：就像On\_Swipe，但要用两个手指去触发

On\_SwipeEnd2Fingers：就像On\_SwipeEnd，但要用两个手指去触发

Unity UI：

On\_OverUIElement ：当手指触碰到UGUI游戏对象的时候触发，前提是EasyTocuh的Unity UI compatibillity被勾选

On\_UIElementUp：当手指离开UGUI游戏对象的那一刻触发，前提是EasyTocuh的Unity UI compatibillity被勾选

静态函数：

SetEnabled： 启用或禁用EasyTouch

GetEnabled: 启用EasyTouch就返回true

SetUICompatibily : 是否启用了UGUI兼容性（EasyTouch能否检测到UGUI）

GetUIComptability:启用UGUI兼容性就返回true

SetAutoUpdateUI:当你的手指在屏幕中滑动的时候，一碰到UI元素，是否把碰到的UI元素的信息进行自动更新

GetAutoUpdateUI:如果AutoUpdateUI启用了就返回true

SetEnableAutoSelect:是否启用自动选择

GetEnableAutoSelect:EnableAutoSelect启用的时候，返回true

SetAutoUpdatePickedObject:是否启用自动更新选择的可选游戏对象

GetAutoUpdatePickedObject:如果AutoUpdatePickedObject启用就返回true

Set3DPickableLayer:设置哪个层的3D游戏对象可以被选取

Get3DPickableLayer:返回可选取的3D游戏对象所在的层

AddCamera:添加一个深度摄像机（就是Camera游戏对象的ClearFlags为Depth Only，而场景自带的MainCamera的ClearFlags为Skybox，通常一个场景只有一个Skybox模式的主摄像机，负责将其拍摄范围内的物体显示到屏幕上，但是又可以有多个深度摄像机，使得就算不在主摄像机拍摄范围的物体但在深度摄像机拍摄范围的物体也能显示到屏幕上），使得被这深度摄像机所拍摄的物体也能受到EasyTouch的屏幕检测

RemoveCamera：删除一个摄像机

GetCamera：返回一个摄像机

SetEnable2DCollider：是否启用2D碰撞检测

GetEnable2DCollider：如果Enable2DCollider被启用就返回true

Set2DPickableLayer：设置哪个层的2D游戏对象能够进行选取

Get2DPickableLayer：返回可选取的2D游戏对象所在的层

SetGesturePriority：设置手势优先级

GetGesturePriority：得到手势优先级

SetStationaryTolerance：设置固定的容错率（因为手指很可能并不能完全覆盖在选取的游戏对象上，所以有容错率）

GetStationaryTolerance：得到固定的容错率

SetLongTapTime：设置长按所需的时间

GetlongTapTime：得到长按所需的时间

SetDoubleTapTime：设置双击间隔时间

GetDoubleTapTime：得到双击间隔时间

SetSwipeTolerance：设置滑动容错率（因为手指很可能并不能完全覆盖在选取的游戏对象上，所以有容错率）

GetSwipeTolerance：得到滑到容错率

SetEnable2FingersGesture：是否启用双手指

GetEnable2FingersGesture：如果Enable2FingersGesture启用就返回true

SetTwoFingerPickMethod：设置双手指选取物体的方式

GetTwoFingerPickMethod：返回双手指选取物体的方式

SetEnablePinch：是否启用缩放手势

GetEnablePinch：EnablePinch启用了，就返回true

SetMinPinchLength：设置最小缩放检测长度

GetMinPinchLength：返回最小缩放检测长度

SetEnableTwist：是否启用旋动手势

GetEnableTwist：EnableTwist启用了就返回true

SetMinTwistAngle：设置最小旋动检测角度

GetMinTwistAngle：得到最小旋动检测角度

GetSecondeFingerSimulation：是否启用第二只手指的仿真体

SetSecondFingerSimulation：如果第二手指的仿真体启用了，就返回true

Gesture类：

变量：

fingerIndex：触发手势事件的手指所对应的下标，如果是双手指触发则返回-1

touchCount ：被激活的触摸数量。用于管理多点触摸。

startPosition：触发触摸事件的时候，单手指的起始位置，或者多手指触发时的起始平均位置

position ：触发触摸事件的时候，单手指触发的当前位置，或者多手指触发时的当前平均位置

deltaPosition：因为输入相应一般都在Update函数里，所以自然是有位置和时间的变化量的，而deltaPosition就是最近一次触摸与本次触摸的位置变化量

actionTime：反应时间，就是说easytouch的同一个事件触发之间是有时间间隔的

deltaTime：因为输入相应一般都在Update函数里，所以自然是有位置和时间的变化量的，而deltaTime就是最近一次触摸与本次触摸的时间变化量

swipe：滑动或拖动的方向，这个值是通过计算当前手势的deltaPosition得到的，也就是手势的startPosition和手势结束时的当前position所得到的计算结果

swipeLength：手指在屏幕上滑动了多少像素，这个值是通过计算当手势结束时，该手势的startPosition和position得到的

swipeVector：在屏幕上滑动的矢量方向

deltaPinch：最近一次缩放与当前缩放的缩放变化量。此变化量对于On\_Pinchin和On\_PinchOut是正数。对于扩大手势是正数，对于缩小手势是负数

twistAngle：最新一次调用的旋动（扭动）手势的角度与当前角度的变化值，顺时针是负值，逆时针是正值

twoFingerDistance：双手指触发时，两只手指在屏幕上相距的像素距离

pickedObject：当触摸开始那一刻所碰到的可选取的游戏对象（如果不是可选取对象则返回none）又或者当auto update picked gameobject被设置为true时，自动更新所得到的可选取对象（也就是你贴着屏幕滑动，只要你碰到可选取游戏对象，那么这个pickedObject代表的就自动变更为那个游戏对象，当然这一切是基于auto update picked gameobject这个属性被设置为true的情况下）

pickedCamera：用手势去操纵的摄像机（例如操纵视野的放大缩小，镜头移动会用到）

isGuiCamera：用手势去选取操纵的摄像机是否GUI摄像机

isOverGUI：当前触摸发生的那一刹那，手指是否触碰到UGUI游戏对象，又或者在当Auto update picked Unity UI被勾选上设置为true的时候，实时检测手指是否在UGUI上。只有在UI Compatibility Mode设置发false禁用了，才会根据情况返回正确的值，不然则恒定返回false

pickedUIElement：当前触摸发生的那一刹那，手指是否触碰到并且选取到UGUI游戏对象，又或者在当Auto update picked Unity UI被勾选上设置为true的时候，实时检测手指是否在UGUI上。只有在UI Compatibility Mode设置为false禁用了，才会根据情况返回正确的值，不然则恒定返回null

方法：

GetTouchToWorldPoint(float z)：将当前触摸位置从当前屏幕坐标转换为距离摄像机z个单位处的世界坐标

GetTouchToWorldPoint(Vector3 position3D)：将当前触摸位置从当前屏幕坐标转换为世界坐标，这个position3D参数是目标在世界坐标系中的位置，正因为如此，可以用来计算摄像机与目标物体的距离长度

GetSwipeOrDragAngle：返回滑动或拖动的角度

NormalizedPosition：返回相对于屏幕的归一化位置

GetCurrentPickedObject：获取当前所选取的游戏对象，如果你把Auto update picked gameobject设置为true，则不需要用到这函数

IsOverUIElement：如果当前触摸发生在UGUI游戏对象上就返回true

IsOverRectTransform：如果当前触摸的是recttransform，则返回true

GetCurrentFirstPickedUIElement：得到当前被选中的UGUI游戏对象，如果Auto update picked Unity UI勾选设置为true了，则不需要用到此函数