

Touch Controller Kit Lite

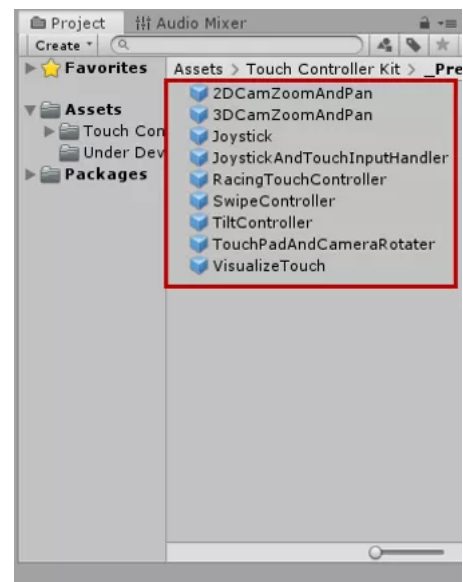
Table Of Contents:

1. [Overview](#)
2. [How to use the Asset](#)
3. [How to Implement Swipes and Taps](#)
4. [How to use Mobile Tilt Controller](#)
5. [How to Implement Joystick](#)
6. [How to Implement Zoom and Pan](#)
7. [How to Visualize Touches on Screen](#)
8. [How to Combine Several Controllers to create a more Robust Custom Controller](#)

Overview

Touch Controller Kit Lite has Five different types of input controllers and some other touch visualization tools. You can use a combination of these input controllers to achieve most types of controls for your games.

- Visualize Touch
 - Animate Touches with a beautiful animation.
- Swipe Controller
 - 4-directional Swipe.
 - Tap & Double Tap.
- Tilt Controller
 - Tilt Control for devices in all axis.
- Joystick
 - On-Screen virtual Joystick on left side.
 - Can customize look and settings.
- Zoom and Pan
 - Zoom in or out in 2D Environment.
 - Pan around the world.
 - Set bounds to zoom and pan to limit to a specific level.



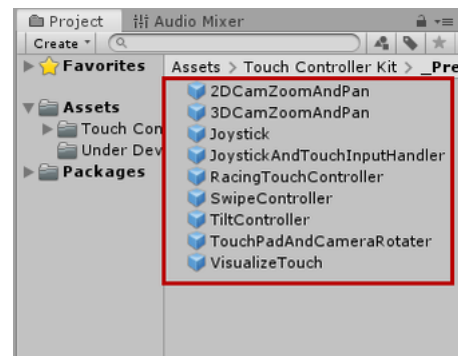
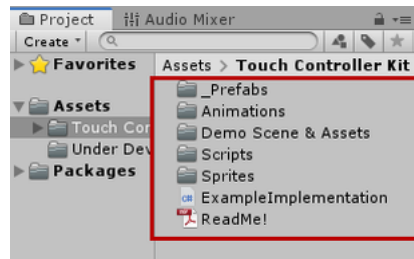
How to use this Asset

The prefabs folder inside this Touch Controller Kit assets' folder contains all the necessary prefabs needed for all types of inputs for phones. Choose the (Prefab) Input type you want to use in your

project and drag it inside your scene hierarchy. For example, if you want a zoom and pan functionality in a 2d game then you need to drag 2DCamZoomAndPan prefab into your scene. Each of the controller prefabs have some settings and variables in the inspector that may be tweaked and adjusted according to your needs. If you are unsure of what a variable does then you can hover over its name and a pop-up will appear explaining its function.

There is also an ExampleImplementation.cs script inside the asset which contains sample codes on using various types of controllers in your game. Lastly, if you are reading this documentation online then there is also an offline version of this added as a ReadMe! file inside the asset itself.

If you need help setting up specific asset check out the following sections.

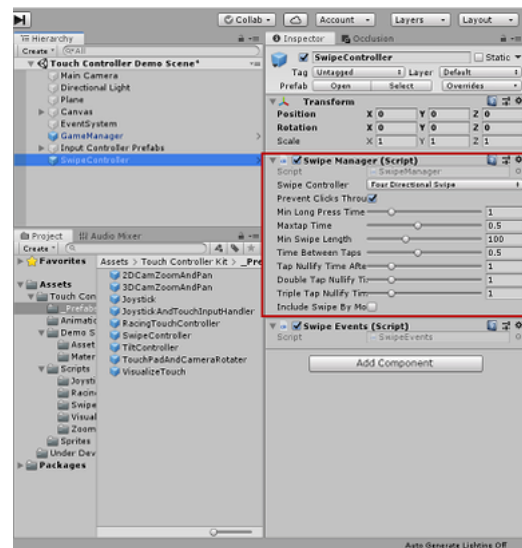


How to Implement Swipes And Taps

The Swipe Controller gives functionality for 4-directional swipe as well as several types of tap controls such as tap & double tap. Drag the Swipe Controller prefab from the prefabs folder into your scene. Adjust the values in the inspector. Most values are self-explanatory but you can hover the mouse over them to know more about it. You can switch between 4-directional and 8-directional swipe by changing the first value in the inspector.

Now, in the script you want to use the swipe or tap you need subscribe to some delegates. You should only subscribe to those specific delegates you will need in your game. You could also subscribe to all the delegates as given in the ExampleImplementation.cs script and only use the ones you want, leaving the other delegate event callbacks empty.

It is necessary to subscribe to these delegates in onEnable and unsubscribe in OnDisable as shown in the images below.



```
//Subscribe to delegate events in onenable
0 references
void OnEnable()
{
    //Swipe events
    SwipeEvents.OnSwipeUp += OnSwipeUp;
    SwipeEvents.OnSwipeDown += OnSwipeDown;
    SwipeEvents.OnSwipeLeft += OnSwipeLeft;
    SwipeEvents.OnSwipeRight += OnSwipeRight;

    //Tap events
    SwipeManager.OnSingleTap += OnSingleTap;
    SwipeManager.OnDoubleTap += OnDoubleTap;
}
```

```
//Unsubscribe to all events you have subscribe
0 references
void OnDisable()
{
    //!! Always unsubscribe to events if you have
    //Swipe Events
    SwipeEvents.OnSwipeUp -= OnSwipeUp;
    SwipeEvents.OnSwipeDown -= OnSwipeDown;
    SwipeEvents.OnSwipeLeft -= OnSwipeLeft;
    SwipeEvents.OnSwipeRight -= OnSwipeRight;

    //Tap Events
    SwipeManager.OnSingleTap -= OnSingleTap;
    SwipeManager.OnDoubleTap -= OnDoubleTap;
}
```

Now, Implement all the event callback methods as shown in the image. All your code (the thing that you want to do when the event gets called) corresponding to an specific event goes inside these respective callback methods. You can check out the ExampleImplementation.cs for more in-depth code.

```
//Swipe Event Callbacks. These functions will get called when the respective event occurs.
2 references
private void OnSwipeUp()
{
    Debug.Log("SwipeUp");
}

2 references
private void OnSwipeDown()
{
    Debug.Log("SwipeDown");
}

2 references
private void OnSwipeLeft()
{
    Debug.Log("SwipeLeft");
}
```

```
}
```

How to use Mobile Tilt Controller

This controller is used to implement tilt in your games in almost any direction. You must check which direction tilt you need carefully.

To use this controller, you need to drag the controller prefab from the assets' prefabs folder to your scene. You can adjust and tweak the value in the inspector to suit your needs. Hover over the variables to know more about them if it confuses you.

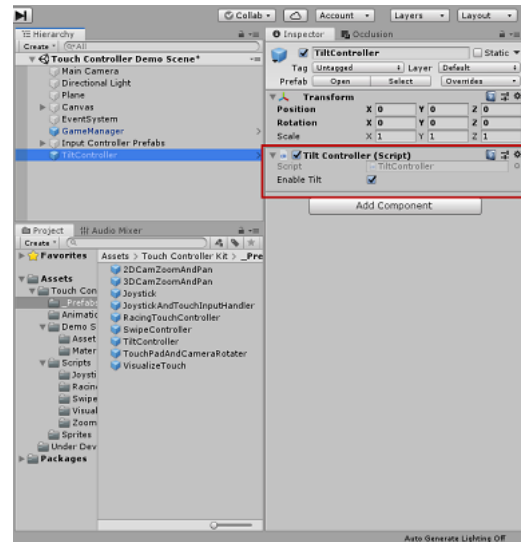
Now in your script, you need to create a Vector3 that will hold the tilt value and also create a reference to the TiltController as shown in the image.

```
0 references
public class ExampleImplementation : MonoBehaviour
{
    #region Unity Methods

    4 references
    private Vector2 movementInput; //this will hold the input value for joyst
    4 references
    private Vector3 tiltInput; //this will hold the tilt value of the device

    2 references
    [SerializeField] private InputHandler inputHandler; //this input handler
    1 reference
    [SerializeField] private TiltController tiltController;

    //Subscribe to delegate events in onenable
    0 references
}
```



Then, you can use the TiltController.GetTilt() function inside Update function and store the value inside a Vector3 and use it as you need. This is shown in the image. You should only use tilt directions that you need, mostly its x and y axis.

```
0 references
void Update()
{
    //We are getting the horizontal and vertical touch or joystick input
    movementInput.x = InputHandler.GetHorizontalInput();
    movementInput.y = InputHandler.GetVerticalInput();

    Debug.Log("Input Value: (" + movementInput.x + " , " + movementInput.y + ")"); //Logs the input value

    //Get the tilt vector
    tiltInput = TiltController.GetTilt();

    Debug.Log("Tilt Value: (" + tiltInput.x + " , " + tiltInput.y + " , " + tiltInput.z + ")"); //Logs the tilt value
}
```

How to implement Joystick

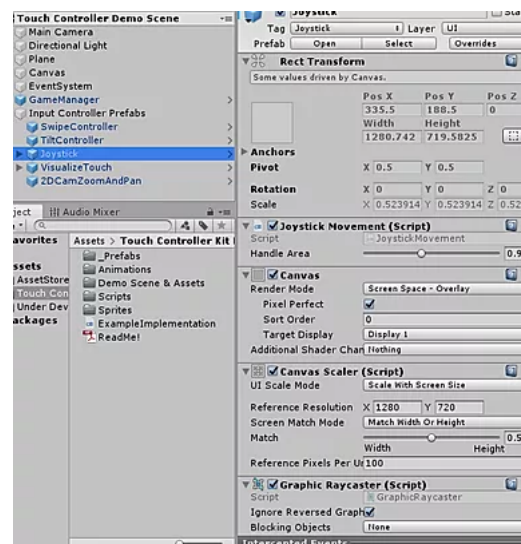
To use the Joystick, you need to drag the joystick prefab to your scene. Adjust and tweak the value in the inspector as per your need. Hover over the variables to know more about them if it confuses you.

Now, going towards the scripting side, you first need to create a Vector2 that will hold the input value and also create a reference to the Joystick.

Initialize the Joystick as shown in the below image.

```
private Vector2 movementInput; //t
private Vector3 tiltInput; //This

JoystickMovement joystick;
[SerializeField] private TiltContro
```



Now, use the Joystick.HorizontalInput() and Joystick.VerticalInput() to get horizontal and vertical input values and store this inside the Vector2. You can then use this input vector to drive your character according to its value.

```
//We are getting the horizontal and vertical to
movementInput.x = Joystick.HorizontalInput();
movementInput.y = Joystick.VerticalInput();
```

How to implement Zoom and Pan

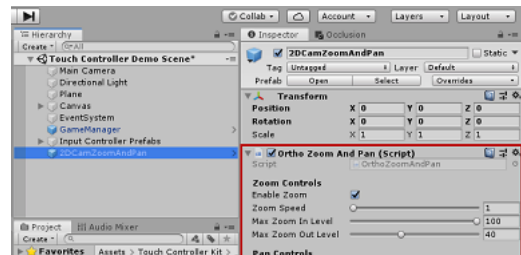
This controller will let the players zoom or pan in 2D. You can also set min and max zoom levels and you can also bound the pan between a min and max in each of the axis.

Implementing Zoom and Pan is simple. All you have to do is drag the respective prefab to your scene. For example, Drag the 2DCamZoomAndPan into your project. Set the zoom levels in the inspector and also set the pan bounds. The pan bounds is specified

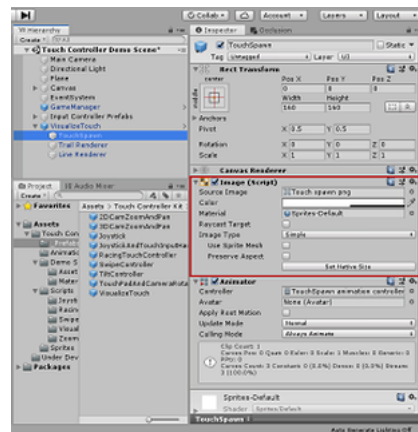
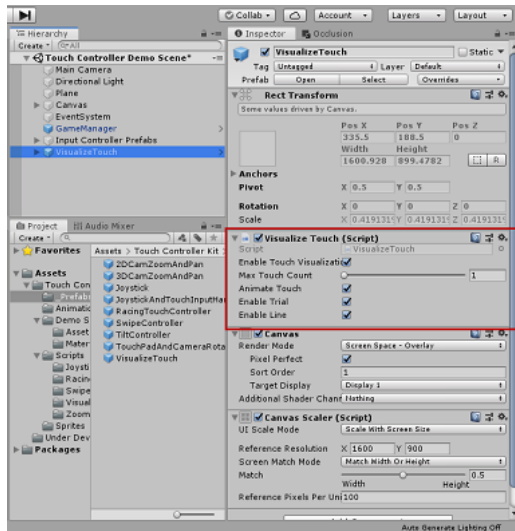
inspector and also set the pan bounds. The pan bounds is specified in respect to camera's position. For example, If your min and max x-

axis pan bounds are -30 and 30 then the camera's x-axis position cannot go below or above the min and max value while panning.

Due to the above approach you may initially get a "screen point out of view frustum" error if your camera's position is way more or less than zero since the camera's default pan bound values are set assuming that the camera is initially at (0,0,0) position. To fix this when you drag the prefab into your scene set the pan bounds according to your camera's current position.



How to Visualize Touches on Screen



The Visualize Touches prefab gives you the ability to animate touches with an image. To use it, you just drag the prefab into your scene, enable only the functionalities that you want.

The children of the Visualize Touch prefab contains the visualization components that will be drawn on touch you can edit them to make it look how you want. The touchspawn is the image that gets animated when a touch occurs.

How to Combine Several Controllers to create a more Robust Custom Controller

Every controller is provided in this asset separately. But you can use simultaneously to create custom controllers that your game needs. You can use the joystick and the tap simultaneously and like-wise you can make combination of two or more input controllers. Please note that not every controller would work with each other as they all use the same functionalities such as touches they may interfere with each other which could wrongly trigger an event. So, double check that everything is working correctly while using combination of controllers.

For example, if you are using joystick then you can use it with tap as it will not interfere with taps, however you cannot use it with touchPad as it will interfere with the taps. Similarly you can make a combination of several other and try out to see if it works in your case. We have prepared a list whose combination should work perfectly.

- Joystick & Visualization tools must work well with most others.
- Swipe and Racing touch controller must work without interference.
- Tilt must work with most others.
- Further you can use Double tap, triple tap and long tap with almost all other.
- For other types of usage, you can disable the interfering controller, enable the other one and when it is no longer needed, disable it and re-enable the former one.

If all these documentation didn't help you or you are encountering an error then do do contact us from the contact section or email us from asset store page. We will try to help you with the best we can. Don't forget you can also order a custom controller customized for your specific game for an extra charge. Just contact us in either case.