

Touch Input Component Manual

1. Introduction

Touch Input Component package contains a total of three components, they are the virtual joystick, the dynamic joystick and the virtual button. All of these components are signed for mobile platform, it's very easy to create and use for your game.



=> *virtual joystick*



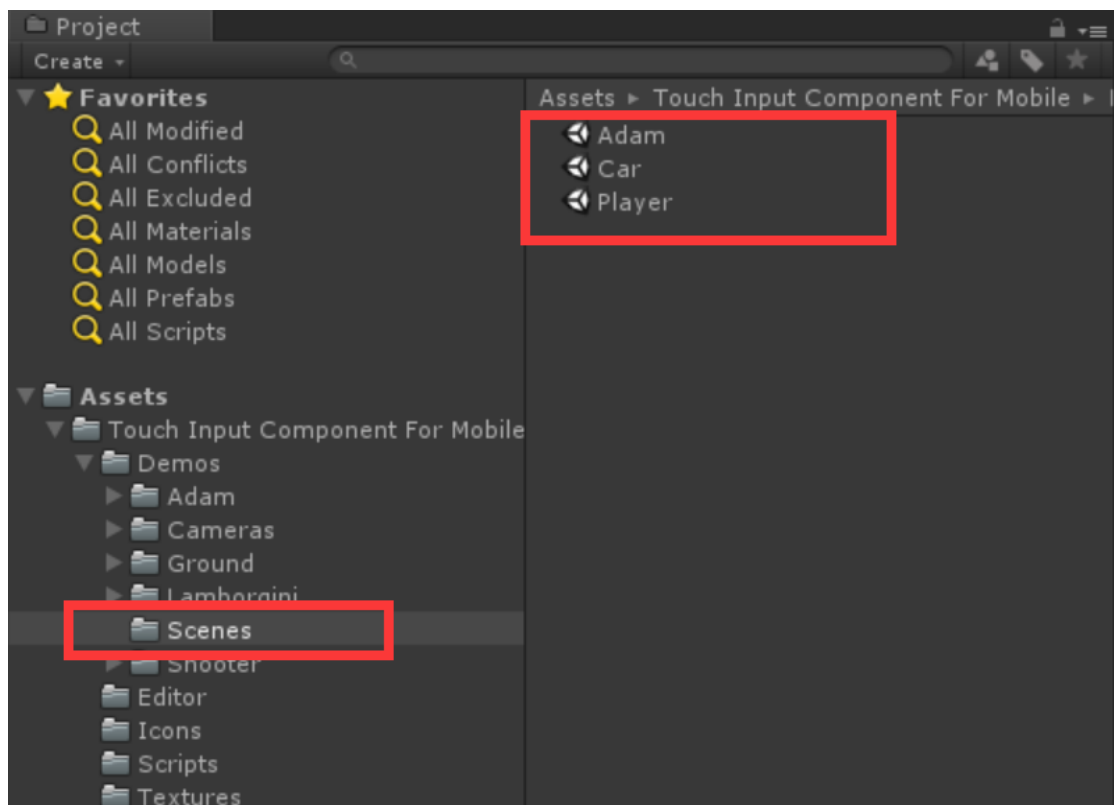
=> *dynamic joystick*



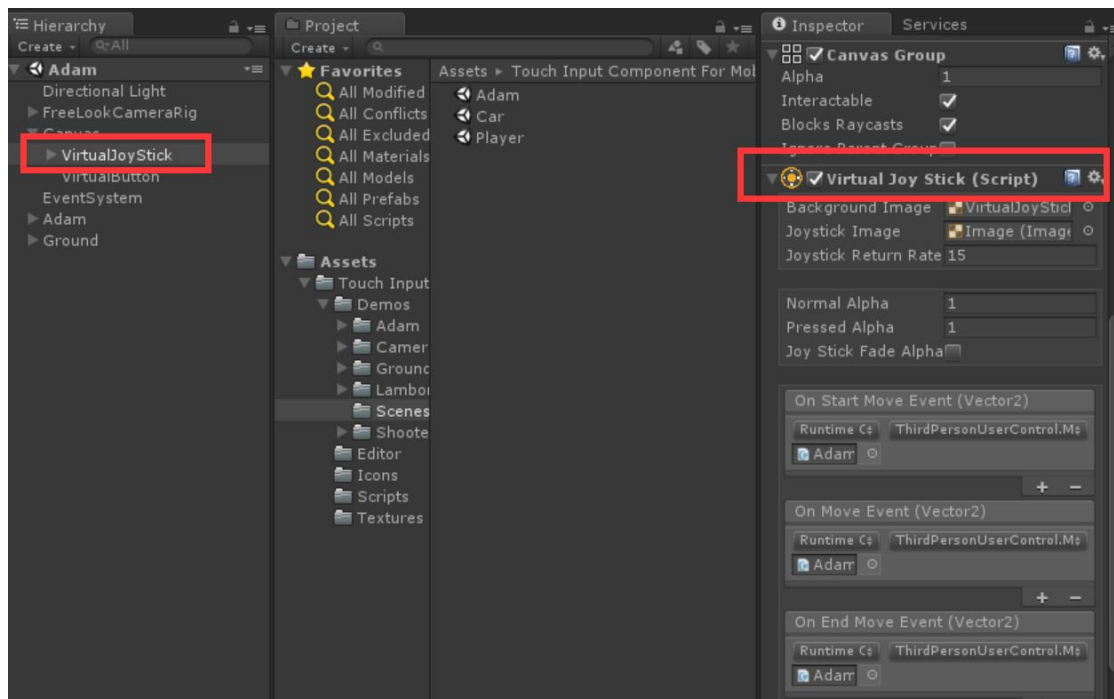
=> *virtual button*

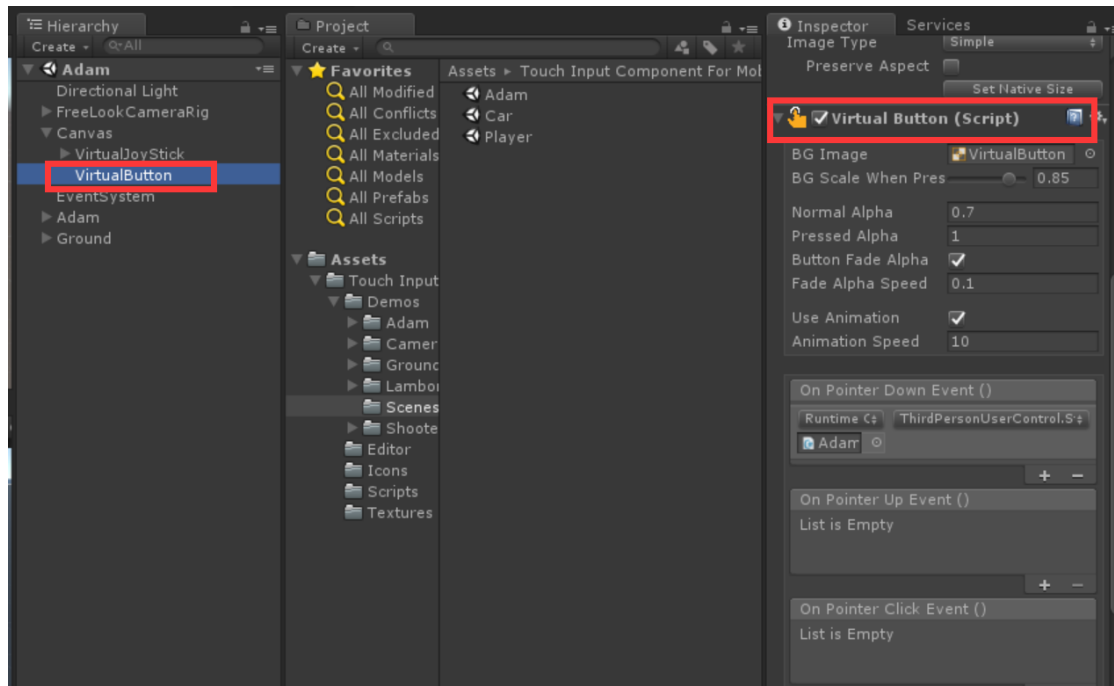
2. Run the demo scene

There are there demo scenes in this packages:

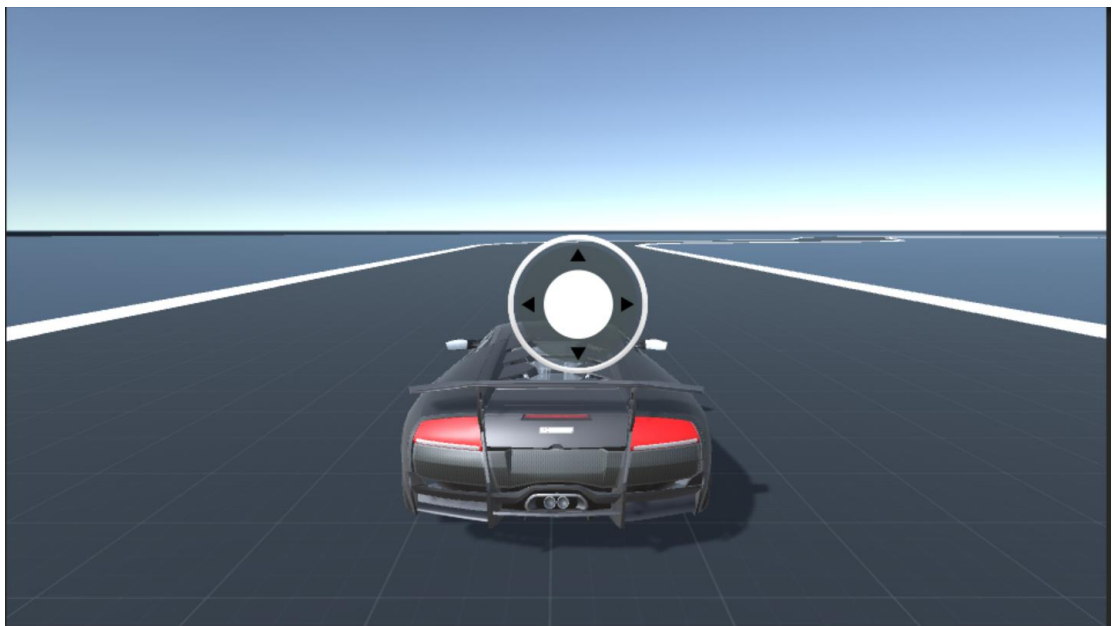


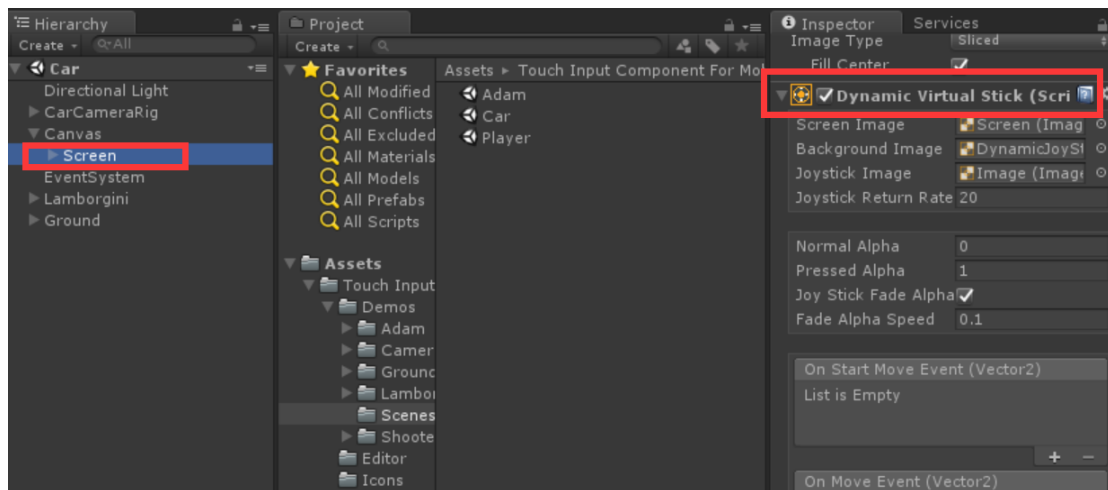
(1).In the *Adam* scene,you can move character by virtual joystick and let him jump by click the virtual button:





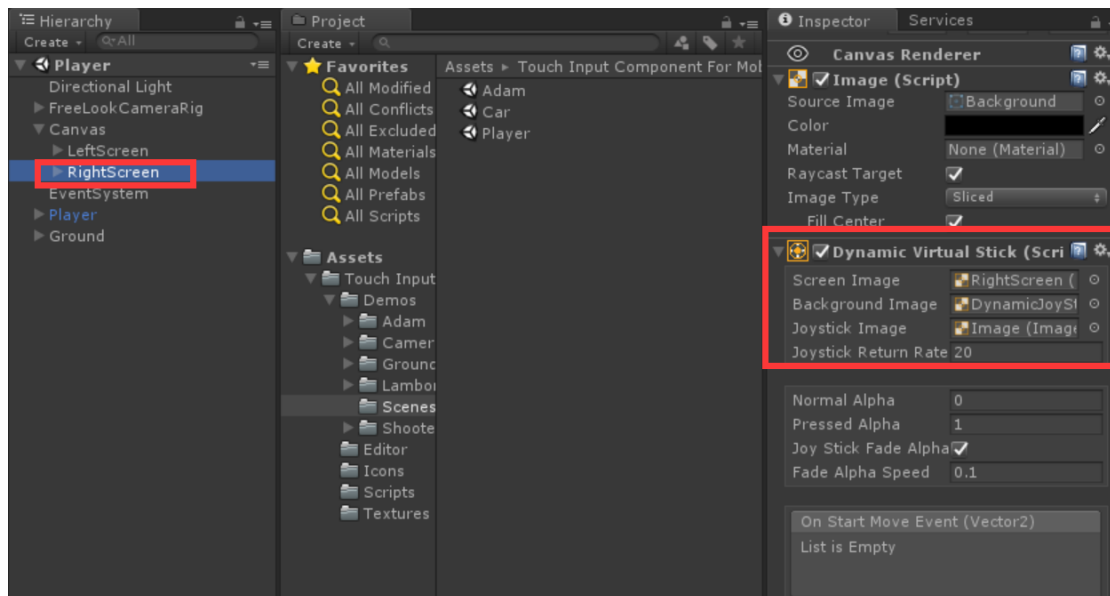
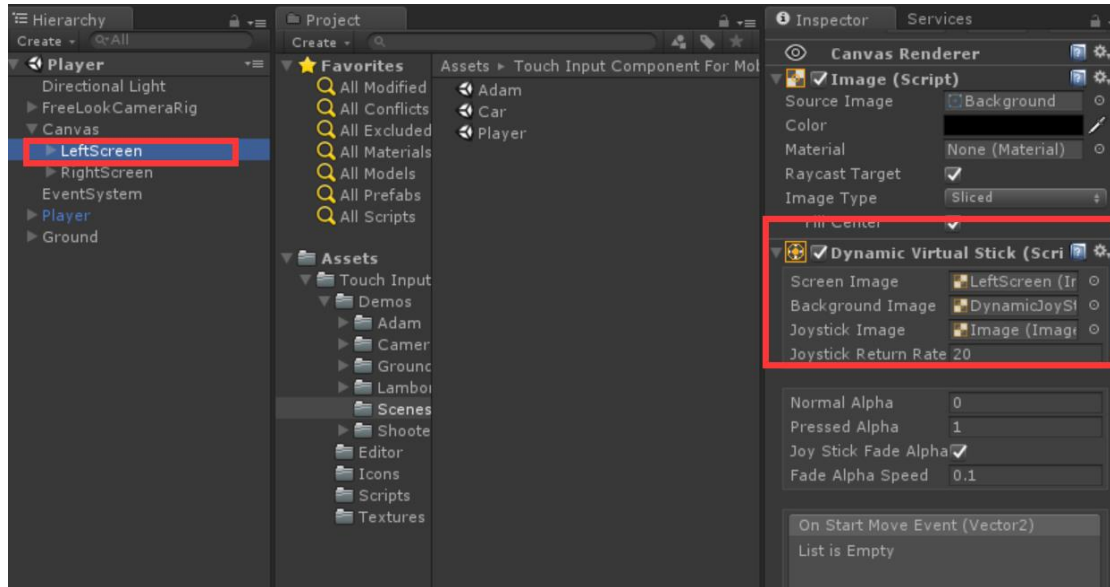
(2).In the *Car* scene,you can drive the car by dynamic joystick on screen,the dynamic joystick is different from the virtual stick, its position on the screen is not fixed and the position where it appears is where your finger touches.





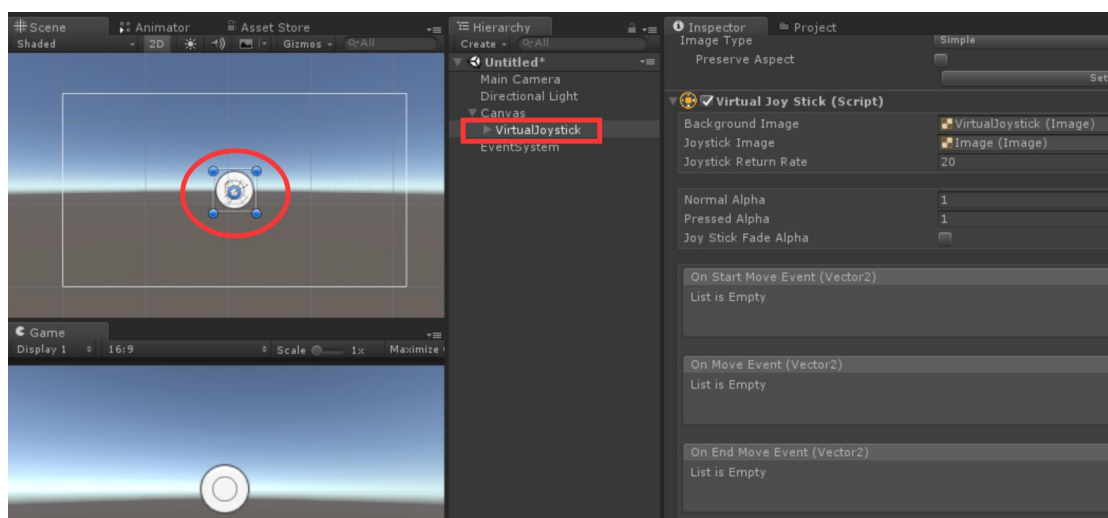
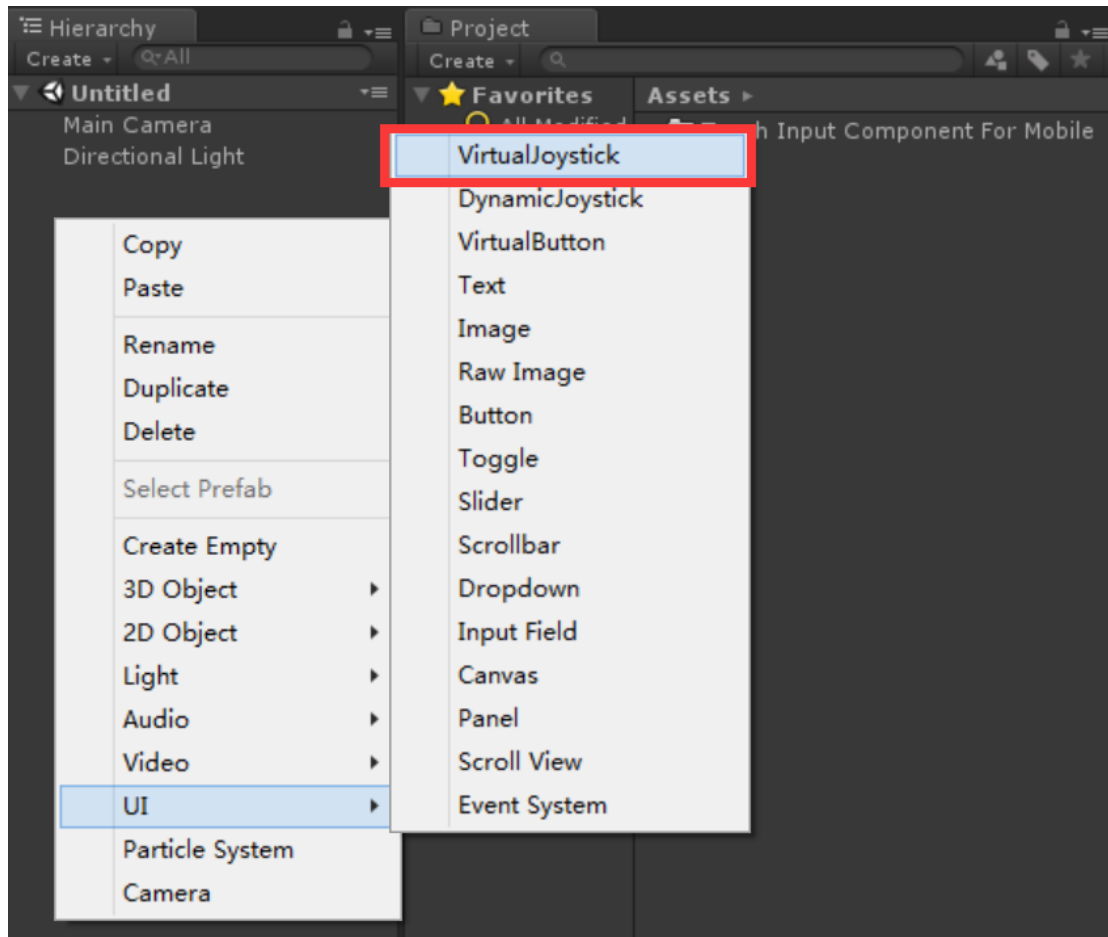
(3).In the *Player* scene, you can move character by dynamic joystick located on the left half of the screen,and control the direction of the character's shooting by dynamic joystick located on the right half of the screen.





3. Create virtual joystick for your game

(1).Create: Right click the empty area, select the virtual joystick,then a virtual joystick will be created on the screen.



(2).Virtual Joystick component Fields Description:

Background Image: The background image of virtual joystick.

Joystick Image: The joystick image.

Joystick Return Rate: The speed at which the joystick image returns to the center of the background image when release the joystick.

Normal Alpha: The transparency of joystick when it at normal state.

Pressed Alpha: The transparency of joystick when it at pressed state.

Joystick Fade Alpha: The transparency of the joystick changes gradually or immediately.

Fade Alpha Speed: The speed at which the joystick's transparency gradually changes.

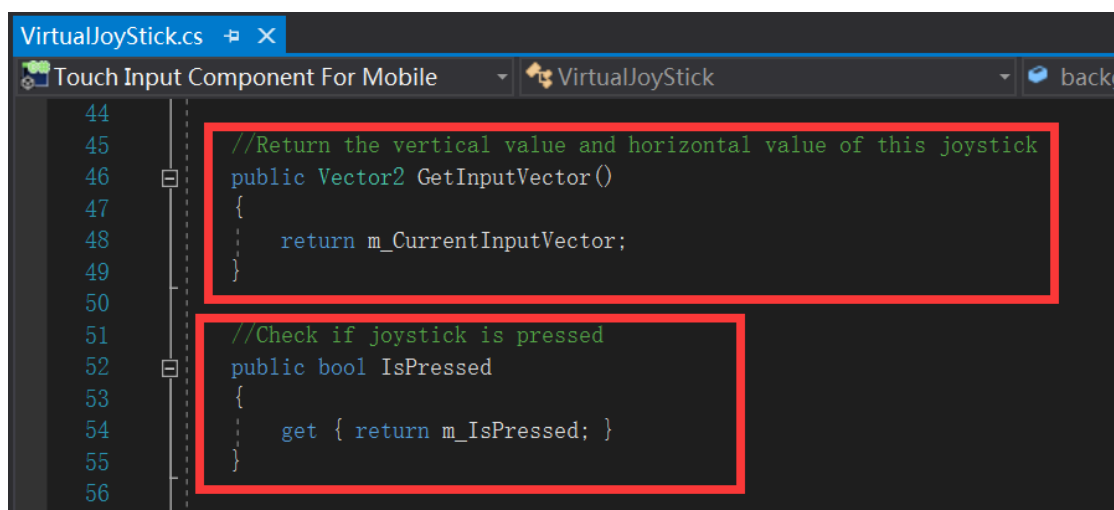
On Start Move Event(Vector2): The event when joystick is press down.

On Move Event(Vector2): The event when joystick is pressed.

On End Move Event(Vector2): The event when joystick is press up.

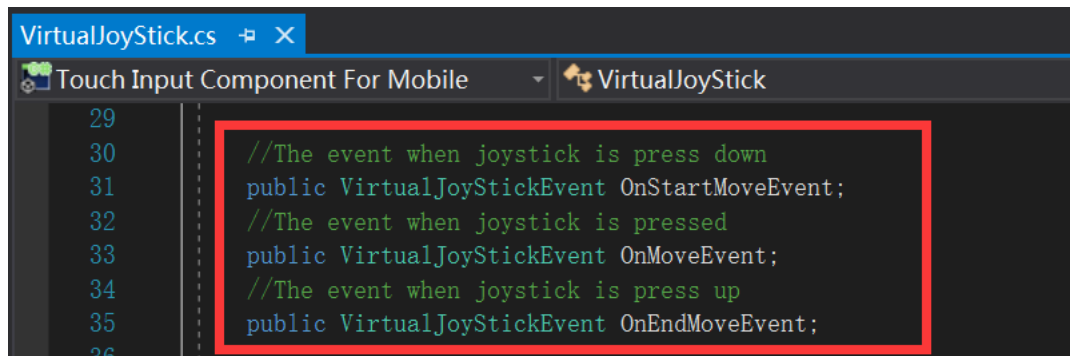
(3).Use:

In coding,you can get the input value of the virtual joystick by using the *GetInputVector()* function,and check if virtual joystick is pressed by using the *IsPressed* property.



```
44
45 //Return the vertical value and horizontal value of this joystick
46 public Vector2 GetInputVector()
47 {
48     return m_CurrentInputVector;
49 }
50
51 //Check if joystick is pressed
52 public bool IsPressed
53 {
54     get { return m_IsPressed; }
55 }
56
```

Or you can also regist the function to UnityEvent If you want to control character in more detail by using virtual joystick:

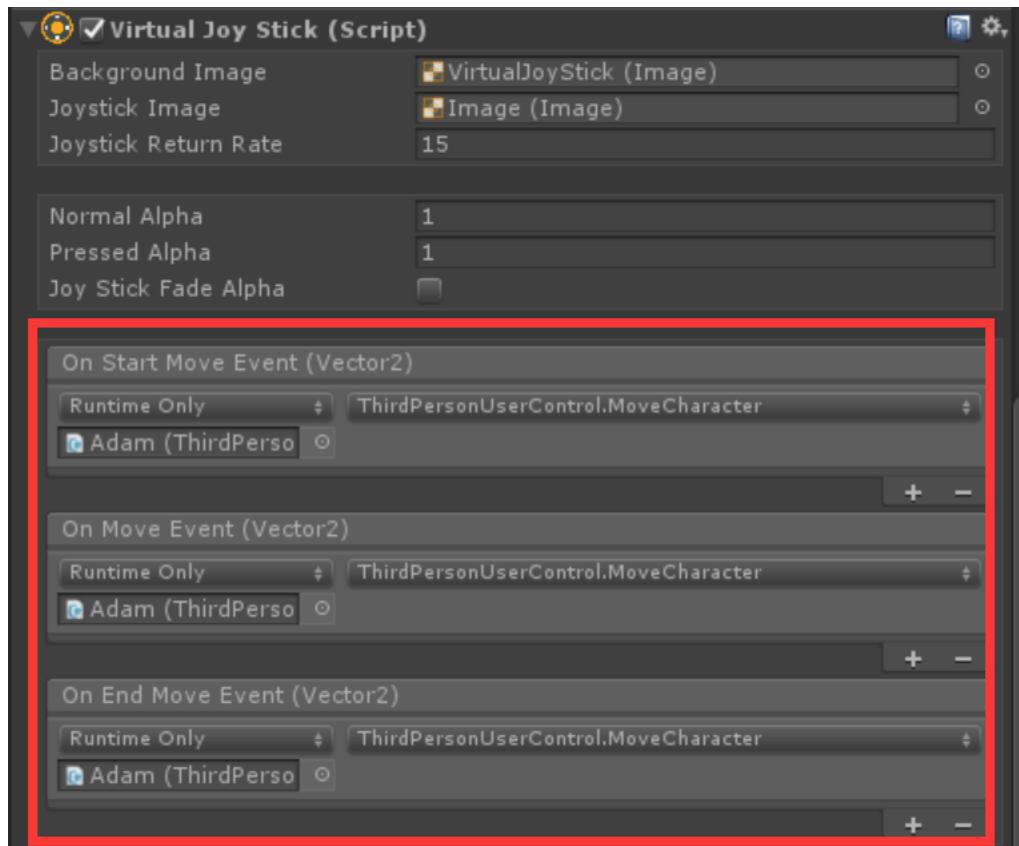


```
29
30 //The event when joystick is press down
31 public VirtualJoyStickEvent OnStartMoveEvent;
32 //The event when joystick is pressed
33 public VirtualJoyStickEvent OnMoveEvent;
34 //The event when joystick is press up
35 public VirtualJoyStickEvent OnEndMoveEvent;
36
```

For example,if you want to regist function to UnityEvent In codeing:

```
{  
  
    public VirtualJoyStick virtualJoyStick;  
  
    // Use this for initialization  
    void Start()  
    {  
        virtualJoyStick.OnStartMoveEvent.AddListener(FunctionA);  
        virtualJoyStick.OnMoveEvent.AddListener(FunctionB);  
        virtualJoyStick.OnEndMoveEvent.AddListener(FunctionC);  
    }  
  
    void FunctionA(Vector2 inputDir)  
    {  
        //...do something  
    }  
  
    void FunctionB(Vector2 inputDir)  
    {  
        //...do something  
    }  
  
    void FunctionC(Vector2 inputDir)  
    {  
        //...do something  
    }  
}
```

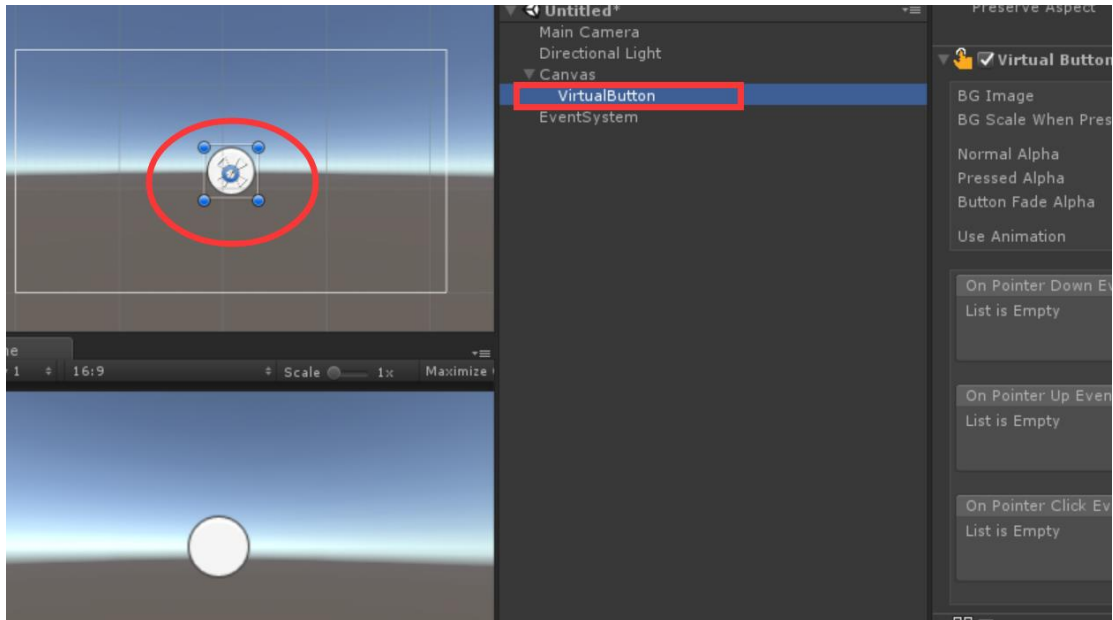

If you want to regist function to UnityEvent on virtual joystick inspector,Here is the example in the Adam scene:



For more detail about how to use the virtual joystick, please check the Adam demo scene.

4. Create virtual button for your game

(1).Create: Right click the empty area, select the virtual button, then a virtual button will be created on screen:



(2).Virtual Button component Fields Description:

BG Image: The background image of virtual button.

BG Scale When Press: The scale of the button changes gradually or changes immediately.

Normal Alpha: The transparency of button when it at normal state.

Pressed Alpha: The transparency of button when it at pressed state.

Button Fade Alpha: The transparency of the button changes gradually or immediately.

Use Animation: The scale of the button changes gradually or changes immediately.

Animation Speed: The speed at which the button's scale gradually changes.

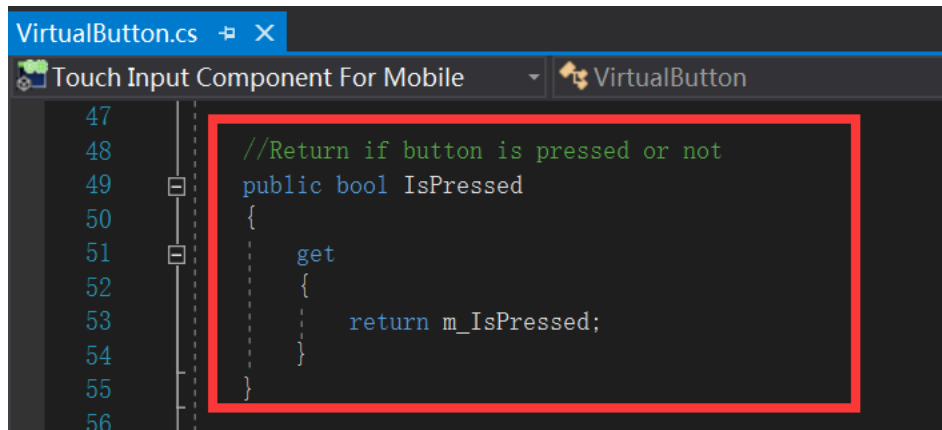
On Pointer Down Event: The event when button is press down.

On Pointer Up Event: The event when button is press up.

On Pointer Click Event: The event when button is clicked.

(3).Use:

In coding,you can check if virtual button is pressed by using the *IsPressed* property.

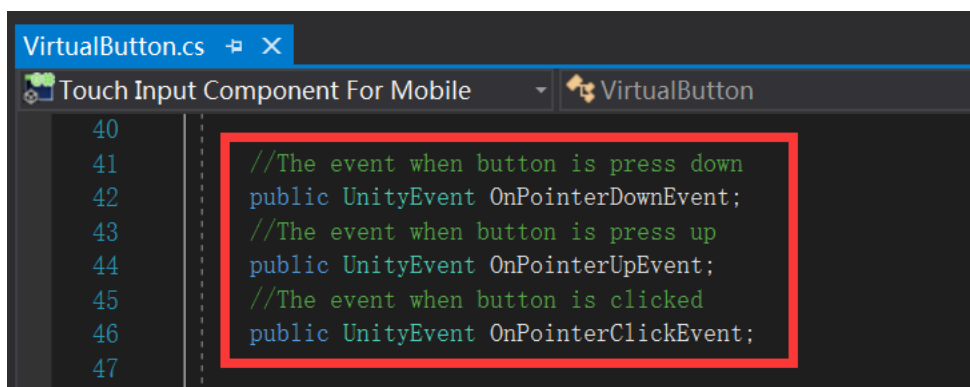


The screenshot shows a code editor window titled 'VirtualButton.cs'. The file is part of a project named 'Touch Input Component For Mobile' and is associated with the 'VirtualButton' class. The code is as follows:

```
47
48 //Return if button is pressed or not
49 public bool IsPressed
50 {
51     get
52     {
53         return m_IsPressed;
54     }
55 }
56
```

The code block for the `IsPressed` property is highlighted with a red rectangle.

Or you can also regist the function to UnityEvent If you want to control character in more detail by using virtual button:



The screenshot shows a code editor window titled 'VirtualButton.cs'. The file is part of a project named 'Touch Input Component For Mobile' and is associated with the 'VirtualButton' class. The code is as follows:

```
40
41 //The event when button is press down
42 public UnityEvent OnPointerDownEvent;
43 //The event when button is press up
44 public UnityEvent OnPointerUpEvent;
45 //The event when button is clicked
46 public UnityEvent OnPointerClickEvent;
47
48 //The event when button is pressed
```

The code block for the UnityEvent declarations is highlighted with a red rectangle.

For example, if you want to register function to UnityEvent In coding:

```
public VirtualButton virtualButton;

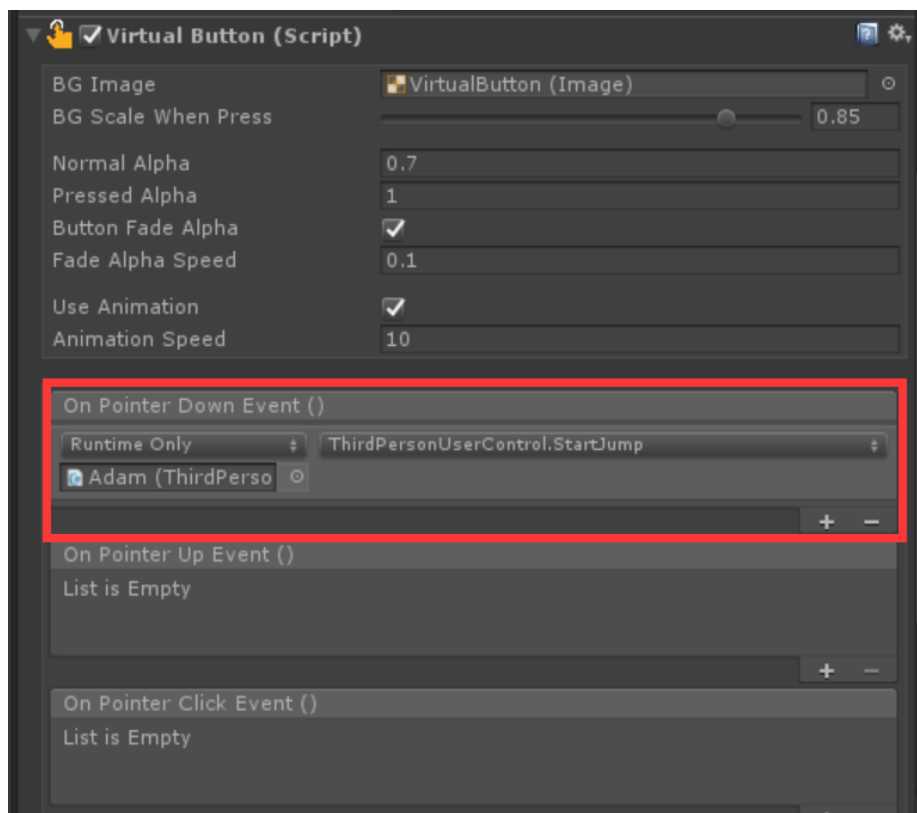
// Use this for initialization
void Start()
{
    virtualButton.OnPointerDownEvent.AddListener(FunctionA);
    virtualButton.OnPointerUpEvent.AddListener(FunctionB);
    virtualButton.OnPointerClickEvent.AddListener(FunctionC);
}

void FunctionA()
{
    //...do something
}

void FunctionB()
{
    //...do something
}

void FunctionC()
{
    //...do something
}
```

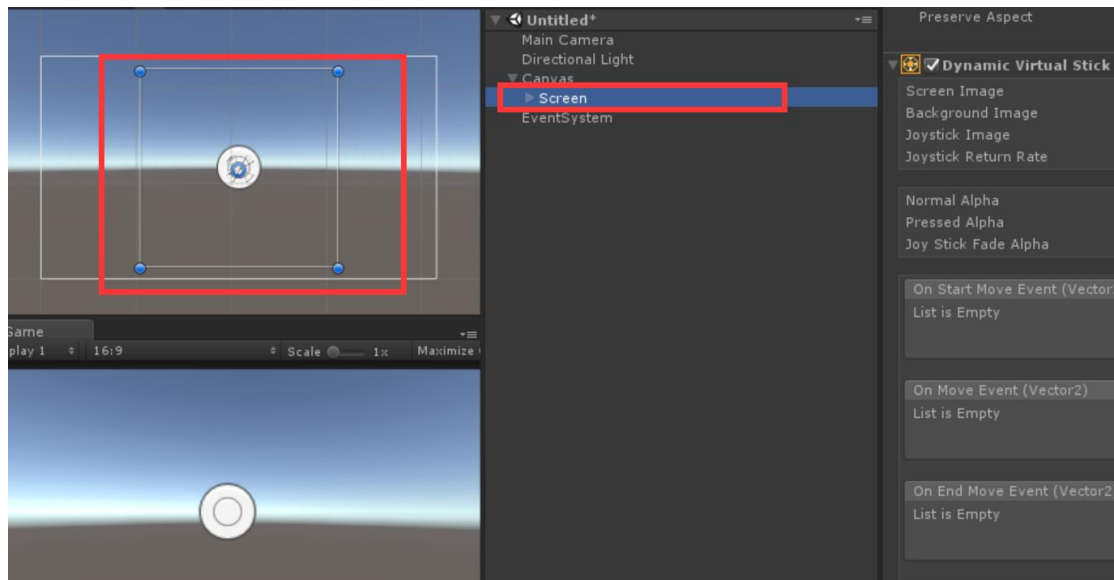
Or if you want to register function to UnityEvent on virtual button inspector, Here is the example in the Adam scene:



For more detail about how to use the virtual button, please check the Adam demo scene.

5. Create dynamic joystick for your game

(1).Create: Right click the empty area, select the dynamic joystick,then a dynamic joystick will be created on screen:



(2).Dynamic Joystick component Fields Description:

Screen Image: This image decide the area that the dynamic joystick can appear.

Background Image: The background image of virtual joystick.

Joystick Image: The joystick image.

Joystick Return Rate: The speed at which the joystick image returns to the center of the background image when release the joystick.

Normal Alpha: The transparency of joystick when it at normal state.

Pressed Alpha: The transparency of joystick when it at pressed state.

Joystick Fade Alpha: The transparency of the joystick changes gradually or immediately.

Fade Alpha Speed: The speed at which the joystick's transparency gradually changes.

On Start Move Event(Vector2): The event when joystick is press down.

On Move Event(Vector2): The event when joystick is pressed.

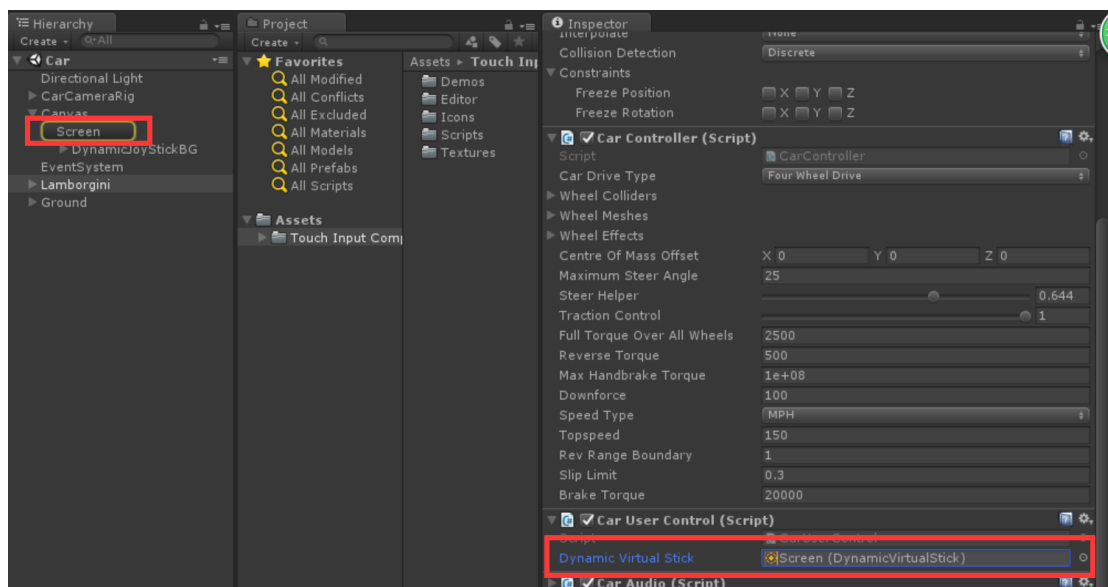
On End Move Event(Vector2): The event when joystick is press up.

(3).Use:

In coding,you can get the input value of the dynamic joystick by using the *GetInputVector()* function,and check if dynamic joystick is pressed by using the *IsPressed* property.

```
44
45 //Return the vertical value and horizontal value of this joys
46 public Vector2 GetInputVector()
47 {
48     return m_CurrentInputVector;
49 }
50
51 //Check if joystick is pressed
52 public bool IsPressed
53 {
54     get { return m_IsPressed; }
55 }
56
```

Here is the example in the Car scene:



```
CarUserController.cs ➤ X
Touch Input Component For Mobile CarUserController dynamicVirtualS
5
6 [RequireComponent(typeof(CarController))]
7 public class CarUserController : MonoBehaviour
8 {
9     public DynamicVirtualStick dynamicVirtualStick;
10    private CarController m_Car; // the car controller we want to use
11
12
13    private void Awake()
14    {
15        // get the car controller
16        m_Car = GetComponent<CarController>();
17    }
18
19
20    private void FixedUpdate()
21    {
22        // pass the input to the car!
23        Vector2 input = dynamicVirtualStick.GetInputVector();
24
25        m_Car.Move(input.x, input.y, input.y, 0f);
26    }
27
28 }
```

Or you can also register the function to UnityEvent If you want to control character in more detail by using dynamic joystick:

```
DynamicVirtualStick.cs ➤ X
Touch Input Component For Mobile DynamicVirtualStick
27 [System.Serializable]
28 public class DynamicJoyStickEvent : UnityEvent<Vector2> { }
29
30 //The event when joystick is press down
31 public DynamicJoyStickEvent OnStartMoveEvent;
32 //The event when joystick is pressed
33 public DynamicJoyStickEvent OnMoveEvent;
34 //The event when joystick is press up
35 public DynamicJoyStickEvent OnEndMoveEvent;
36
```

Here is the example:

```

public DynamicVirtualStick dynamicVirtualStick;

// Use this for initialization
void Start()
{
    dynamicVirtualStick.OnStartMoveEvent.AddListener(FunctionA);
    dynamicVirtualStick.OnMoveEvent.AddListener(FunctionB);
    dynamicVirtualStick.OnEndMoveEvent.AddListener(FunctionC);
}

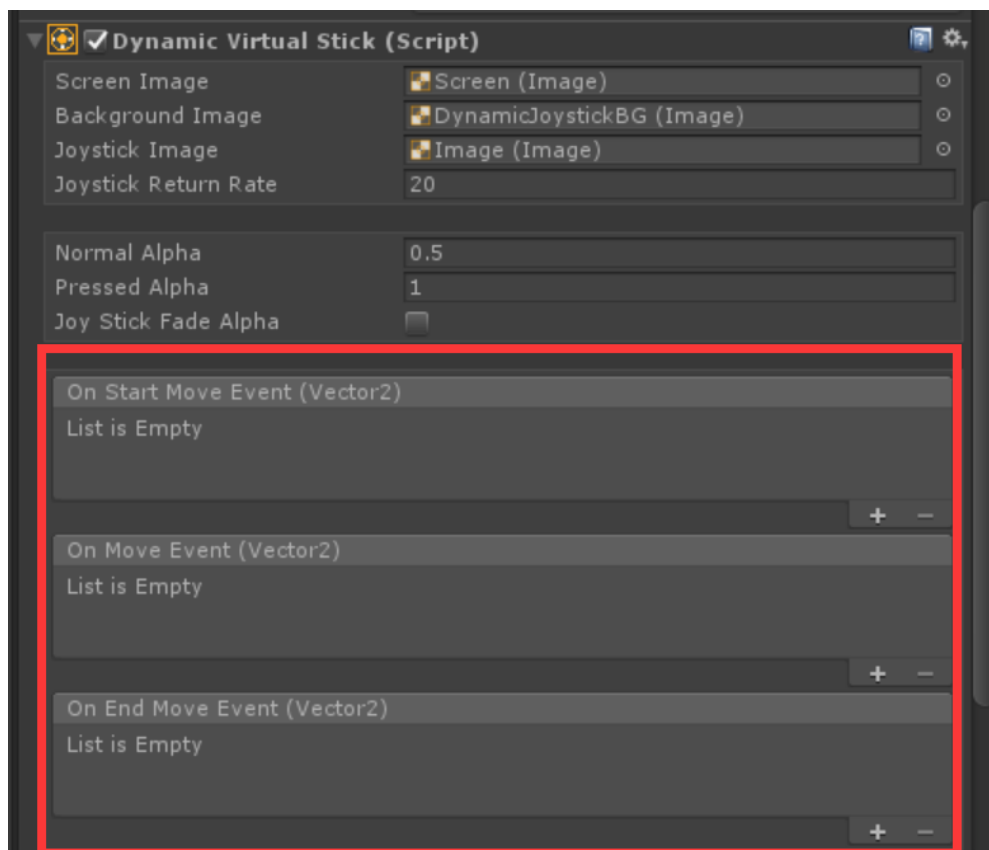
void FunctionA(Vector2 inputDir)
{
    //...do something
}

void FunctionB(Vector2 inputDir)
{
    //...do something
}

void FunctionC(Vector2 inputDir)
{
    //...do something
}

```

Or you can also regist function to UnityEvent on dynamic joystick inspector:



For more detail about how to use the dynamic joystick, please check the Car demo scene or the Player demo scene.

6. Contact

If you have any questions, please email me: swordmaster0080@gmail.com