



FIRST PERSON MOBILE CONTROLLER KIT
DOCUMENTATION

Contents

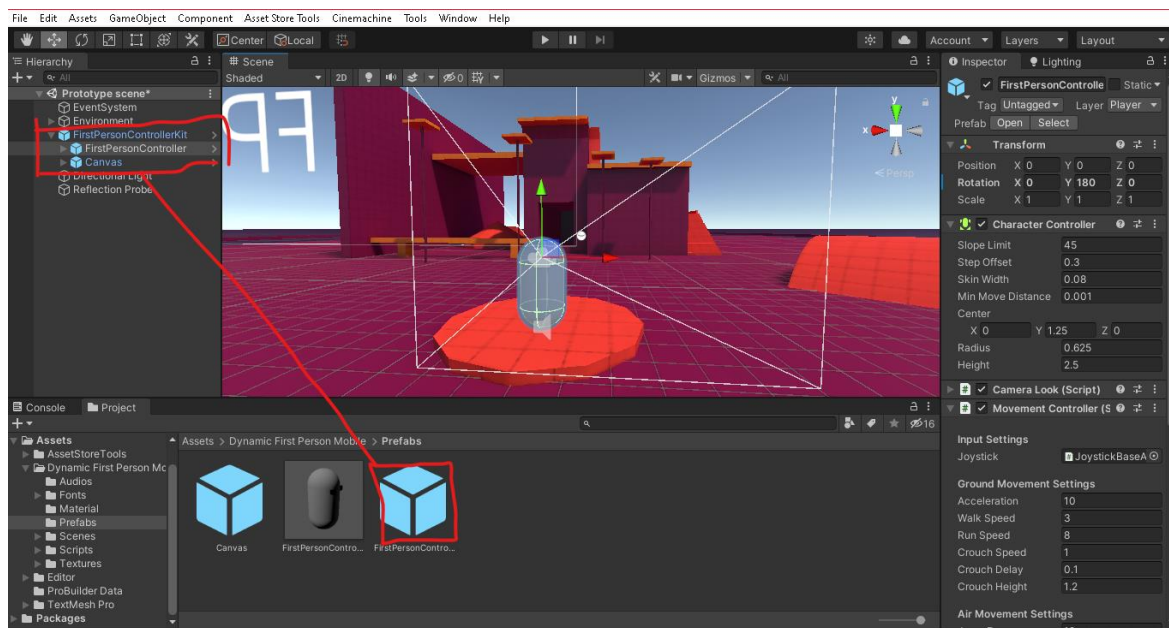
A. How to use	2
a. Basic Set up	2
b. Setting up the controller seperately	3
• Player (First Person Controller) prefab	3
• Canvas Prefab.....	3
c. Custom Button	4
• Single Tap.....	4
• Toggle.....	4
• Press And Release.....	4
d. Custom Joystick.....	5
• Fixed.....	5
• Fixed Floating.....	5
• Floating	5
• Dynamic	5
B. Adding more function or settings	6
a. Function	6
b. Settings	6
C. Performance	7

A. How to use

a. Basic Set up

Make sure your unity project is on the **android build platform**, and the scene has **event system**. The input uses the **old input manager**, if your project using the new input system, you need to set the input setting into both. To set it, find for **edit/project settings/Player/Other Settings/Active Input Handling***. In the project window, find for folder “**Mobile First Person/Prefab** ”, select “**FirstPersonControllerKit.prefab**”, drag and drop it into your scene.

(If your scene already has a camera, delete it, since you can only have one camera in a single scene, and it would interrupt the controller script. The project uses the old input manager because the new input system does not work with Unity Remote 5)



In your game window, you should have the button and joystick for mobile controller set up. Press run, to test the controller.

b. Setting up the controller seperately

The kit is made up by two individual prefab, **the player (First person controller)** and **the canvas**. The script is made as dynamically as possible to widen the possibilites of its uses. If you want to use them seperately, you have to do a little set up individually.

- **Player (First Person Controller) prefab**

The Player prefab comes with **non-mobile input** for testing the controller in editor, so it could work properly without any input from joystick and buttons.

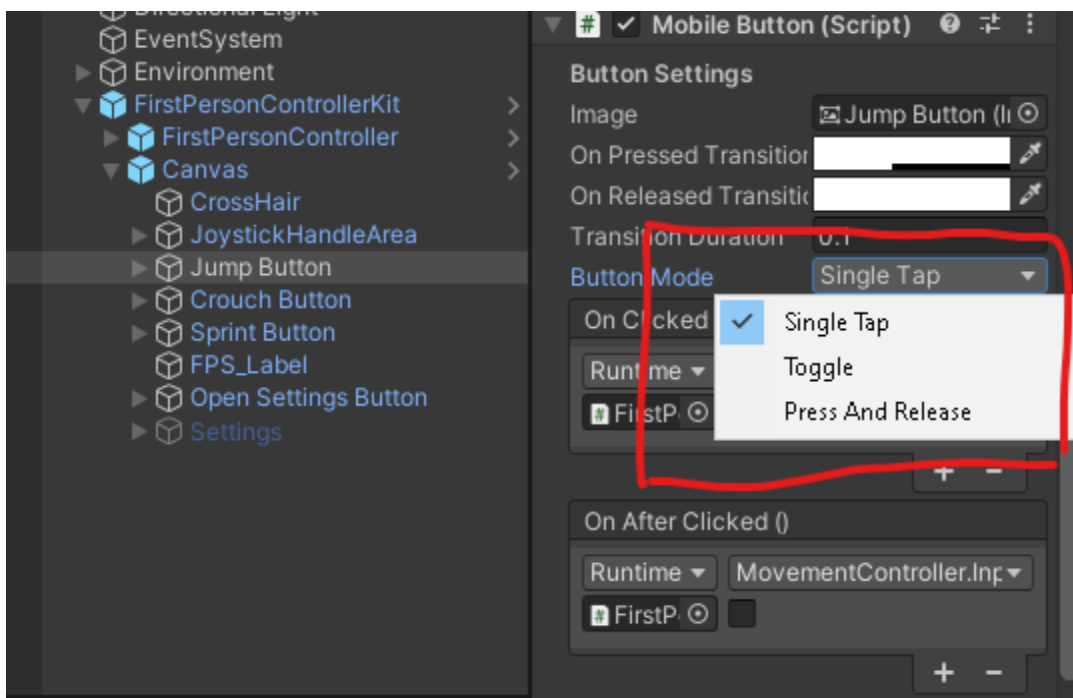


- **Canvas Prefab**

The canvas includes buttons, joystick, and settings, it is compatible with anything that takes input. The **joystick** give an ouput float from 0 to 1, **button** has 3 ways to give an output (read more on Custom Button part below), and **settings** which is dynamically modifiable.



c. Custom Button



- **Single Tap**

Single tap has 2 event actions, **OnClick()**, it trigger the event right the moment the button is pressed, and **onAfterClicked()**, that triggers after Time.deltaTime since the button is pressed. This is useful for bool variable like jump that needs to be true only for a moment.

- **Toggle**

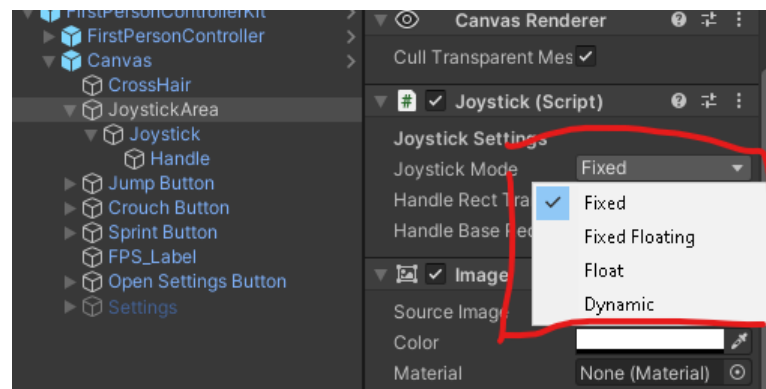
Toggle has 2 event actions, **OnToggleOn()**, and **OnToggleOff()**. Both could be triggered by a simple press on the button, and work just like a switch.

- **Press And Release**

Press and release has 2 event actions, **OnButtonPressed()**, which triggers the action the moment the button is pressed, and **OnButtonReleased()**, that triggers when the button is released.

d. Custom Joystick

The joystick returns 2 value that vary between -1.000 to 1.000. **To access the value**, make a field that reference the joystick, and access the “Vertical” to get the vertical value of the joystick, and “Horizontal” to get the horizontal value of the joystick. (*work exactly like `Input.GetAxis(“Horizontal”)`*)



- **Fixed**

The joystick origin is fixed in one place means the whole joystick wont move, only the handle can be moved.

- **Fixed Floating**

The joystick origin is fixed, but the function is the same as **floating joystick**.

- **Floating**

The joystick origin will be retransformed into the first touch input position whenever touch is detected, and will function like a normal **fixed joystick**. (Touch detection in this mode is limited to the joystick base rect transform area)

- **Dynamic**

The joystick origin will be retransformed into the first touch input position whenever touch is detected, and will move dynamically following the movement of the handle.

B. Adding more function or settings

a. Function

The important part of the script is already commented, make sure you have the basic of c# if you want to modify the script. **To set a new action**, for example eating action, copy and paste a button, make a new function or bool variable, and assign it to the new button “Mobile Button” script unity event.

```

    }
    Controller))]
Behaviour {
    // Sprint
    m_Sprint { get; set; } // Accessed through [Sprint button] in the scene
    // Jump
    m_Jump { get; set; } // Accessed through [Jump button] in the scene
    // Crouch
    m_Crouch { get; set; } // Accessed through [Crouch button] in the scene

    // Walk Speed
    m_WalkSpeed { private get { return m_WalkSpeed; } set { m_WalkSpeed = value; } } // Accessed through [Walk Speed] in the scene
    // Run Speed
    m_RunSpeed { private get { return m_RunSpeed; } set { m_RunSpeed = value; } } // Accessed through [Run Speed] in the scene
    // Crouch Speed
    m_CrouchSpeed { private get { return m_CrouchSpeed; } set { m_CrouchSpeed = value; } } // Accessed through [Crouch Speed] in the scene
    // Jump Force
    m_JumpForce { private get { return m_JumpForce; } set { m_JumpForce = value; } } // Accessed through [Jump Force] in the scene
    // Acceleration
    m_Acceleration { private get { return m_Acceleration; } set { m_Acceleration = value; } } // Accessed through [Acceleration] in the scene
    // Land Momentum
    m_LandMomentum { private get { return m_LandMomentum; } set { m_LandMomentum = value; } } // Accessed through [Land Momentum] in the scene

    // Joystick
    m_Joystick { // Available Joystick mobile in the scene

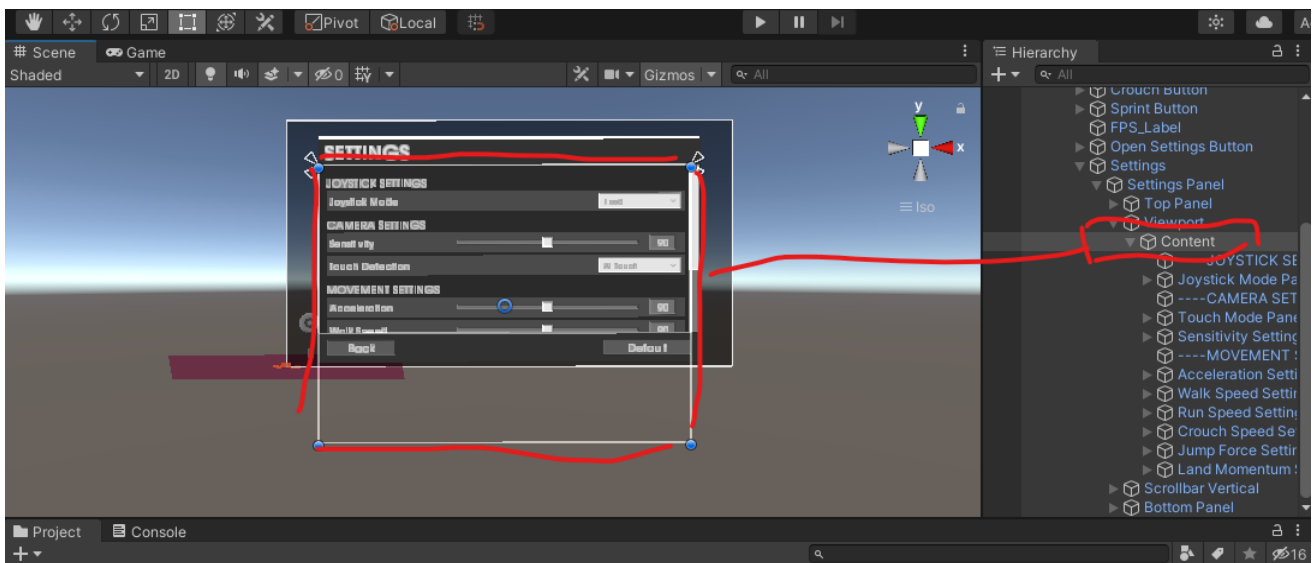
    // Settings
    m_Acceleration = 1.0f; // Acceleration
    m_WalkSpeed = 1.0f; // Walk Speed
    m_RunSpeed = 3.0f; // Run Speed
    m_CrouchSpeed = 0.5f; // Crouch Speed
    m_CrouchDelay = 0.5f; // Crouch transition time
    m_CrouchHeight = 1.0f; // Crouch target height

    m_JumpForce = 1.0f; // y-axis force for jumping
    m_Gravity = 10.0f; // Gravity force
    m_LandMomentum = 2.0f; // Movement momentum when landed
    m_FallLand = -10f; // Minimum falling velocity to be considered as landed
}

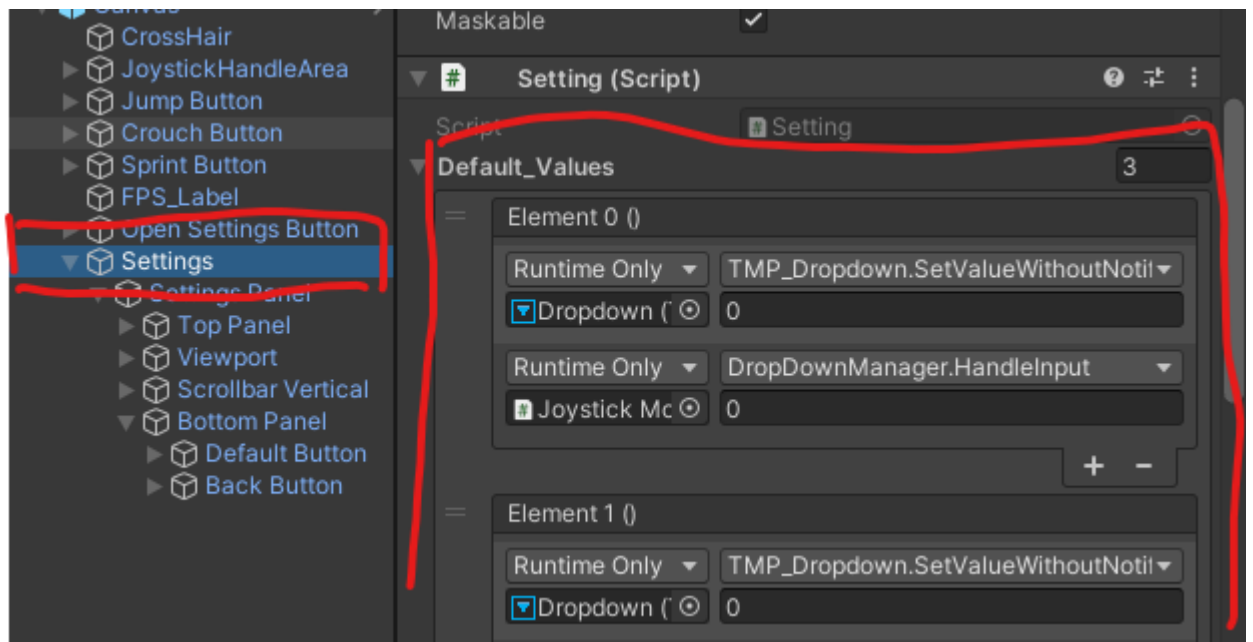
```

b. Settings

In the canvas prefab, the settings is ready-made and you could just simply **copy and paste** the element. In order to make sure the **view scroll** works properly, **make sure to expand the content rect transform** so the elements fit into the content area.

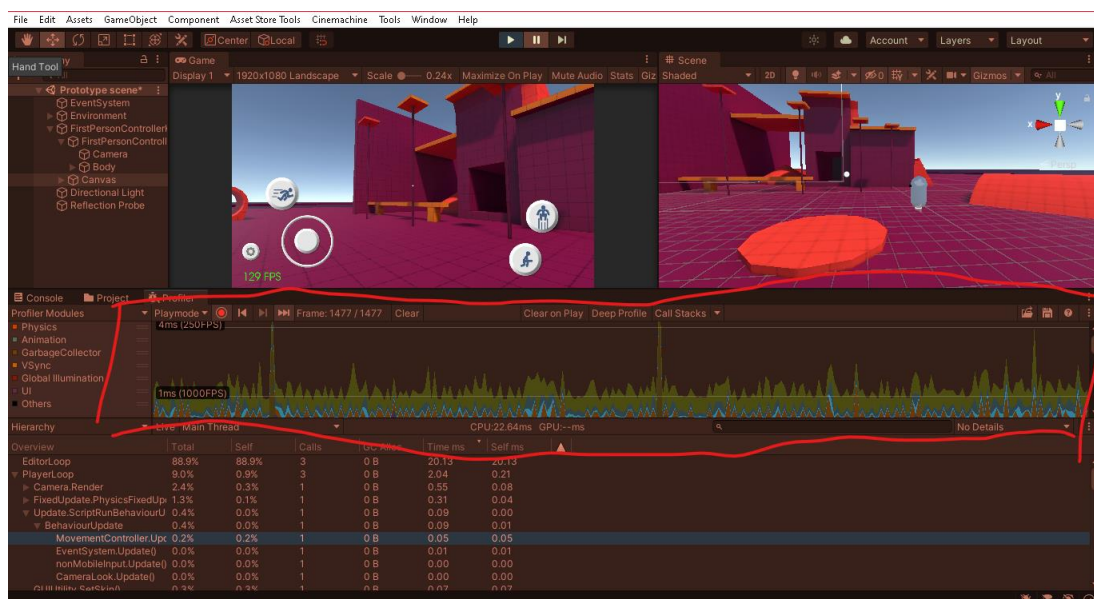


The settings only contain two type of setting, slider, and dropdown. For the default button, assign all the element default value into the unity event.



C. Performance

The script it self basically does not really takes that much of memory. Graphs below show the performance of its test. The light blue is the **movement controller script**, which by far takes the highest memory among the other script (The other script was only about 0.5%).



D. Problems

a. Pink Materials on sample scene

That might be because one of these reasons:

- Missing Probuilder packages

To fix this problem, just simply install probuilder onto your project

- Unsupported pipeline render shader

Similar to the first solution, simply install the shader material your render pipeline support in package manager probuilder sample, and replace all object material with the imported shader from probuilder sample. This package doesn't include probuilder package dependencies, so you need to set it by yourself.

The whole kit is still far from perfect, if you think there's some part that can be fixed or optimize, feel free to do all customization you want on this kit. And if you kind enough to tell me what to fix or optimize as a feedback, I would really appreciate it!

Contact: RkyLaurus@gmail.com