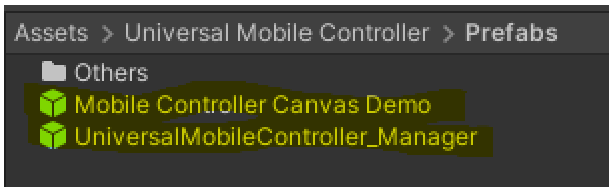
Universal Mobile Controller

Version: 1.0
Discord: Binary Wizards

Get Started:
These are all of the prefabs that you need to have in your scene for the asset to function:



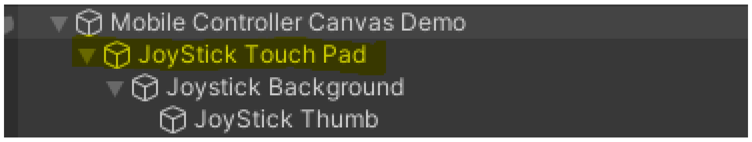
Assets > Universal Mobile Controller > Prefabs
Others
Mobile Controller Canvas Demo
UniversalMobileController_Manager

Drag these 2 prefabs in your scene.

1) Setting Up Joystick:

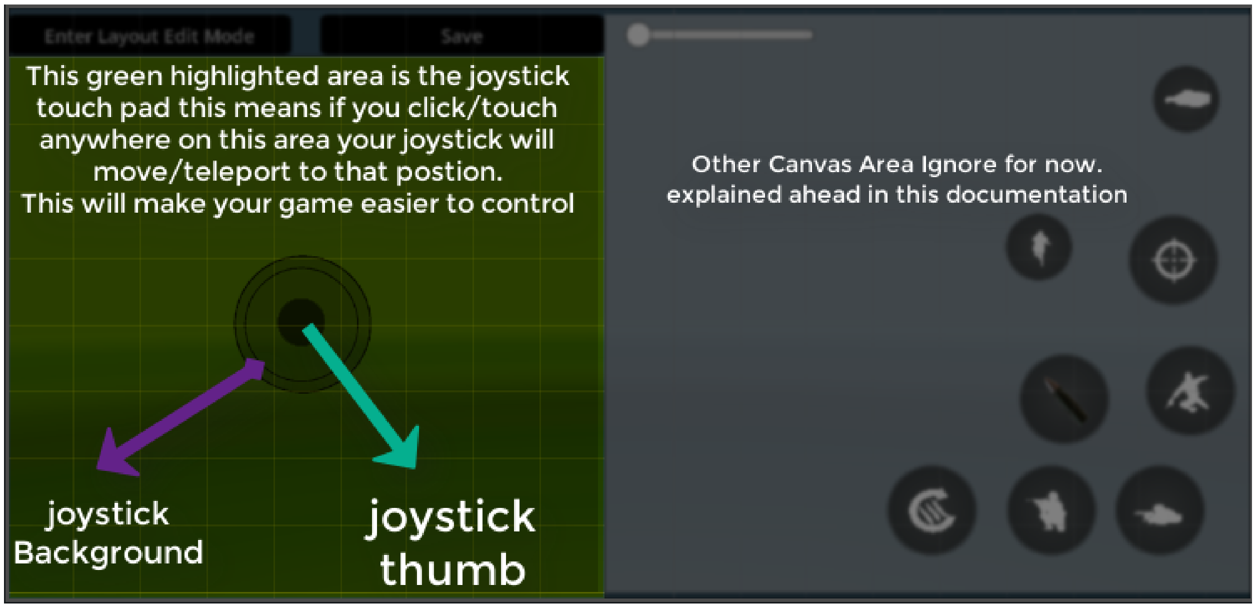
- Setting up joystick with your fps/tps controller

  Expand the ‘Mobile Controller Canvas Demo’ object, then ‘JoyStick Touchpad’, then ‘JoyStick Background ’.
  Shown in the image below



Mobile Controller Canvas Demo
JoyStick Touch Pad
Joystick Background
JoyStick Thumb

- The Joystick Touchpad Object is highlighted as green in the image below.
- The Joystick Background Object is represented by the purple arrow in the image below..
- The Joystick Thumb Object is represented by a cyan arrow in the image below.


Enter Layout Edit Mode
Save
This green highlighted area is the joystick touch pad this means if you click/touch anywhere on this area your joystick will move/teleport to that postion.
This will make your game easier to control
joystick Background
joystick thumb
Other Canvas Area Ignore for now. explained ahead in this documentation

Now if you want to move your character using this joystick. Open your player input script.
(Note: Here I am using a Input Script from HQ FPS Template)
This is how we take inputs on the PC.

```
// Movement.
Vector2 moveInput = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));
```

In order to use this joystick follow these steps.

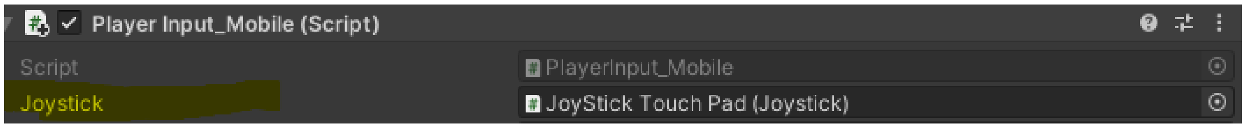
- **Import the namespace using UniversalMobileController**

```
using UniversalMobileController;
```

- **Now you need a reference to the joystick.Add this line in your script.**

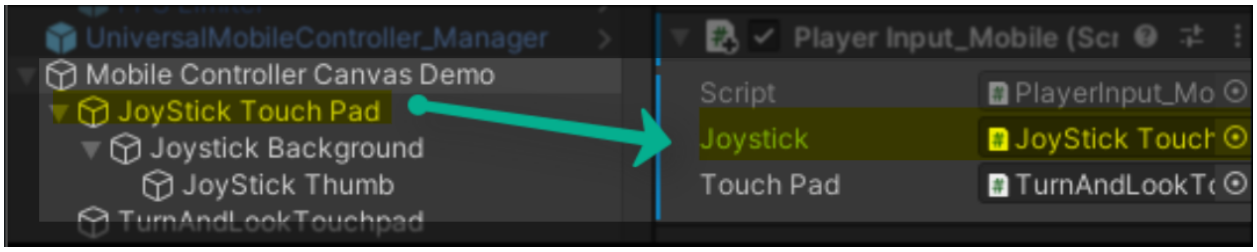
```
public Joystick joystick;
```

- **Save your script, go back to Unity. Now check your 'player input script' it should have a 'Joystick' field. Shown in the image below**


Player Input_Mobile (Script)

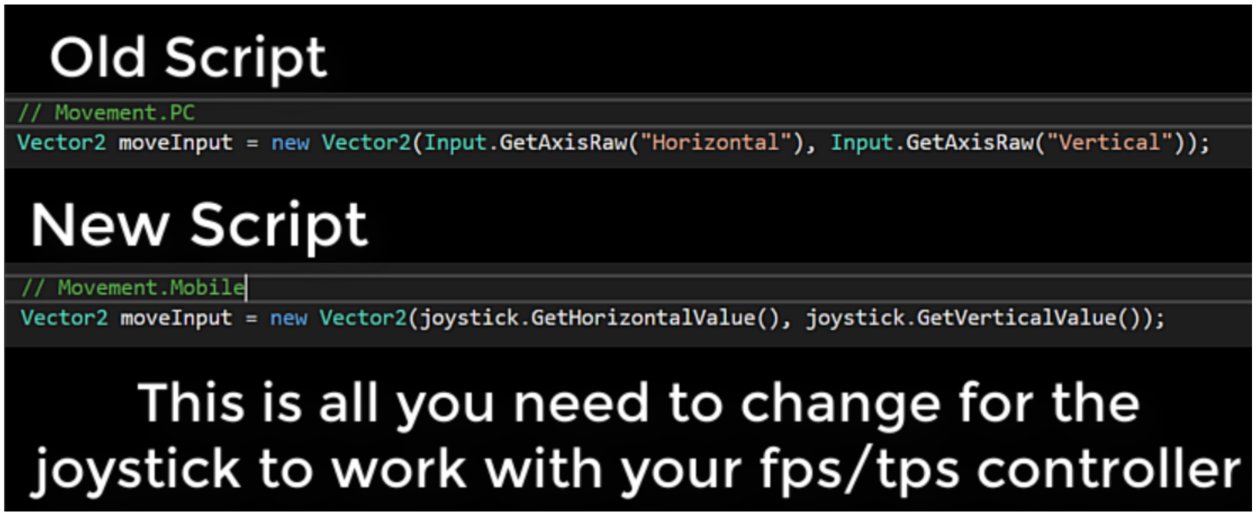
Script	PlayerInput_Mobile
Joystick	JoyStick Touch Pad (Joystick)

- **Drag and drop the ‘Joystick touchpad’ object in the ‘joystick’ field in your ‘player script’. Shown in the image below**



- **Now go back to your script where you were taking ‘Inputs’.**

- **Edit the script like this.**


Old Script

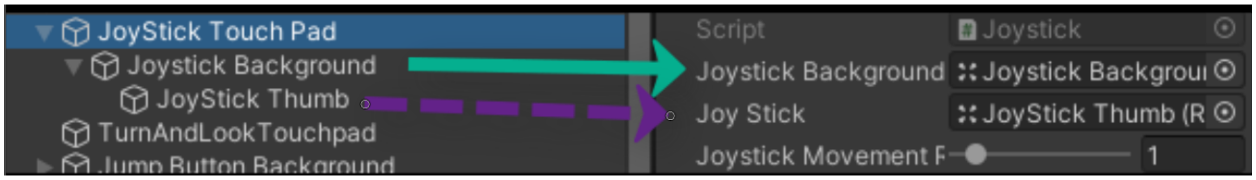
```
// Movement.PC
Vector2 moveInput = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));
```

New Script

```
// Movement.Mobile
Vector2 moveInput = new Vector2(joystick.GetHorizontalValue(), joystick.GetVerticalValue());
```

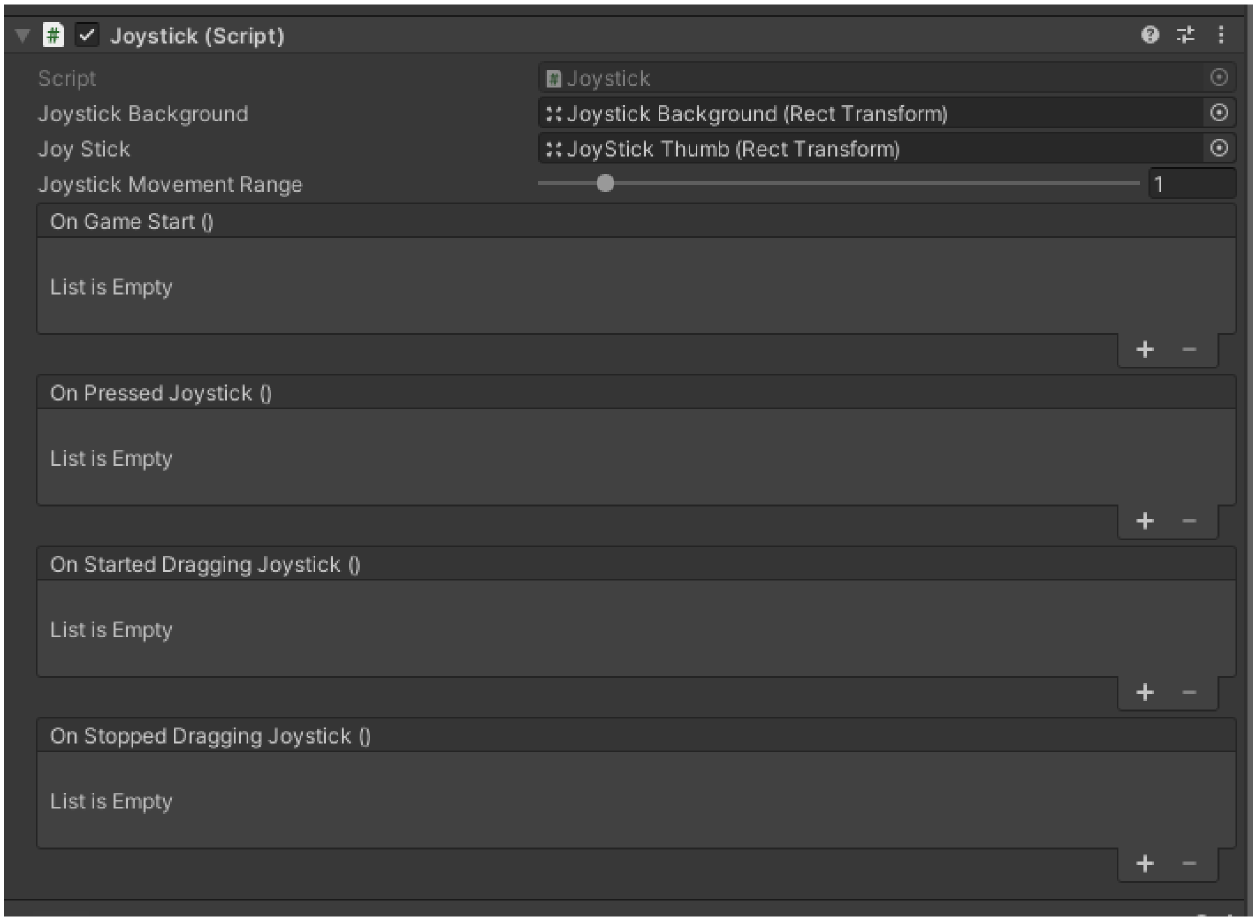
This is all you need to change for the joystick to work with your fps/tps controller

- **Select ‘JoyStick touchpad’ object.Make sure these fields are assigned like the image below.**

- 

Now you can use your joystick to control your player.

This is the inspector view of the ‘JoyStick touchpad’ object. Here are some events you can use if required. “If you are not sure what are the UnityEvents please watch tutorials on UnityEvents”


Joystick (Script)

Script	Joystick
Joystick Background	Joystick Background (Rect Transform)
Joy Stick	JoyStick Thumb (Rect Transform)
Joystick Movement Range	1

On Game Start ()
List is Empty

On Pressed Joystick ()
List is Empty

On Started Dragging Joystick ()
List is Empty

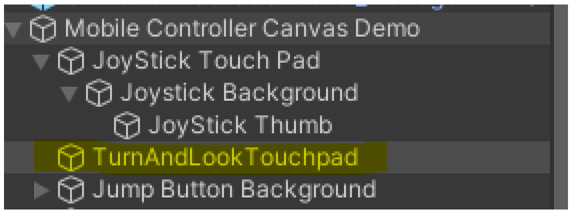
On Stopped Dragging Joystick ()
List is Empty

Now your joystick is ready, you can test it now.
Let's move on to LookTouchPad. On pc we call this mouseX and mouseY inputs.

2) Setting Up Touch Camera Movement:

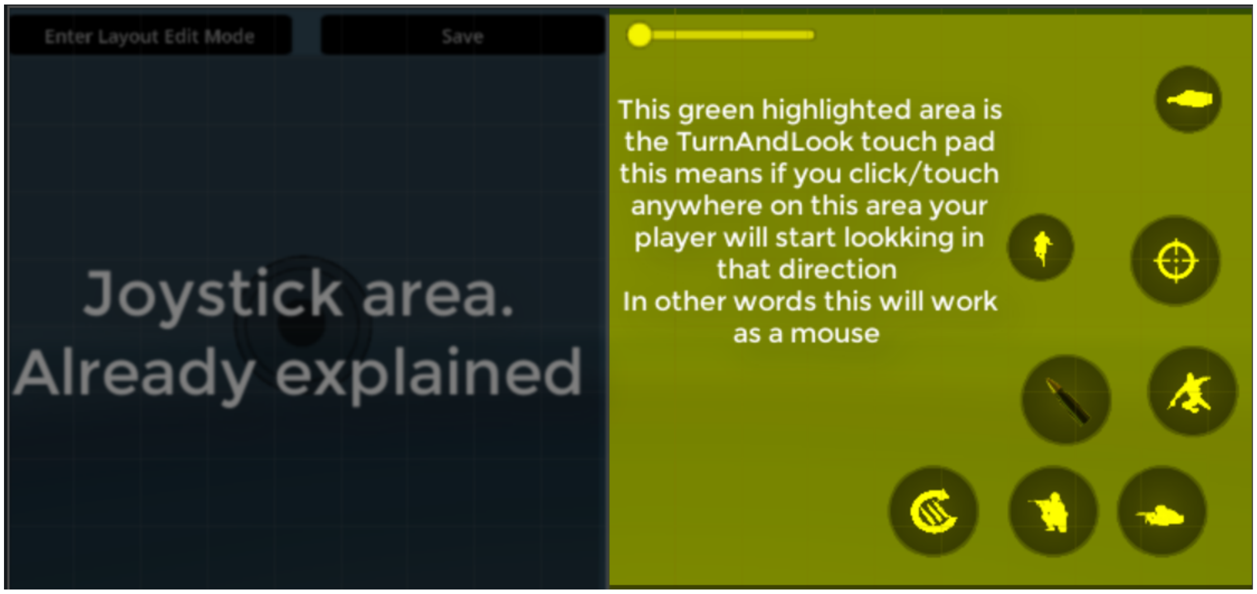
Setting up TurnAndLook TouchPad with your fps/tps controller

Expand the 'Mobile Controller Canvas Demo' object, then find 'TurnAndLookTouchPad' object. Shown in the image below



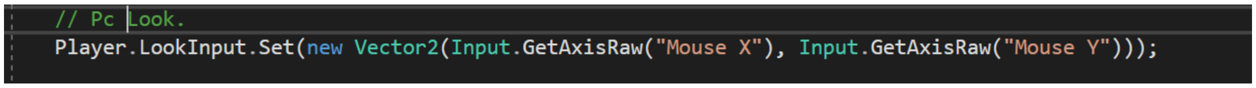
```
▼ Mobile Controller Canvas Demo
  ▼ JoyStick Touch Pad
    ▼ Joystick Background
        JoyStick Thumb
    TurnAndLookTouchpad
  ▶ Jump Button Background
```

The 'TurnAndLookTouchPad' Object is highlighted as green in the image below.



This green highlighted area is the TurnAndLook touch pad this means if you click/touch anywhere on this area your player will start lookking in that direction
In other words this will work as a mouse

Let's integrate this with our player.
(Note: Here I am using a Input Script from HQ FPS Template. Your script should be similar to this)
This is how we take mouse inputs on the PC.



```
    // Pc Look.
    Player.LookInput.Set(new Vector2(Input.GetAxisRaw("Mouse X"), Input.GetAxisRaw("Mouse Y")));
```

In order to use TurnAndLook TouchPad follow these steps.

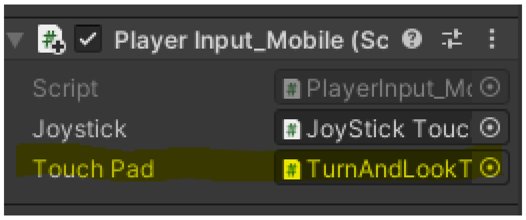
- **Import the namespace using UniversalMobileController (Note: you don't need to do this if you have already added it)**

```
using UniversalMobileController;
```

- **Now you need a reference to the 'TurnAndLookTouchPad' object .Add this line in your 'player' script.**

```
public TouchPad touchPad;
```

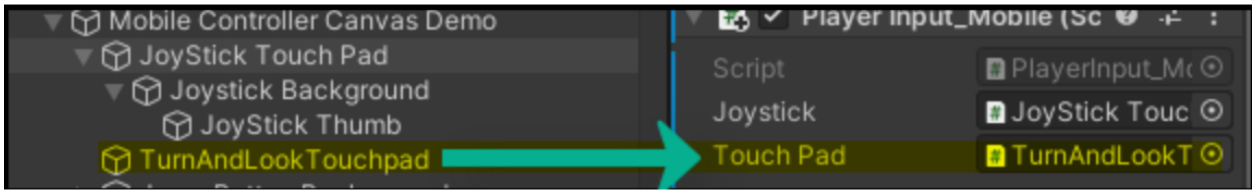
-
- **Save your script, go back to Unity. Now check your 'player script' it should have a 'TurnAndLookTouchPad' field. Shown in the image below**



Player Input_Mobile (Sc

Script	PlayerInput_Mo
Joystick	JoyStick Touc
Touch Pad	TurnAndLookT

- **Drag and drop the 'TurnAndLookTouchPad' in this field shown in the image below.**

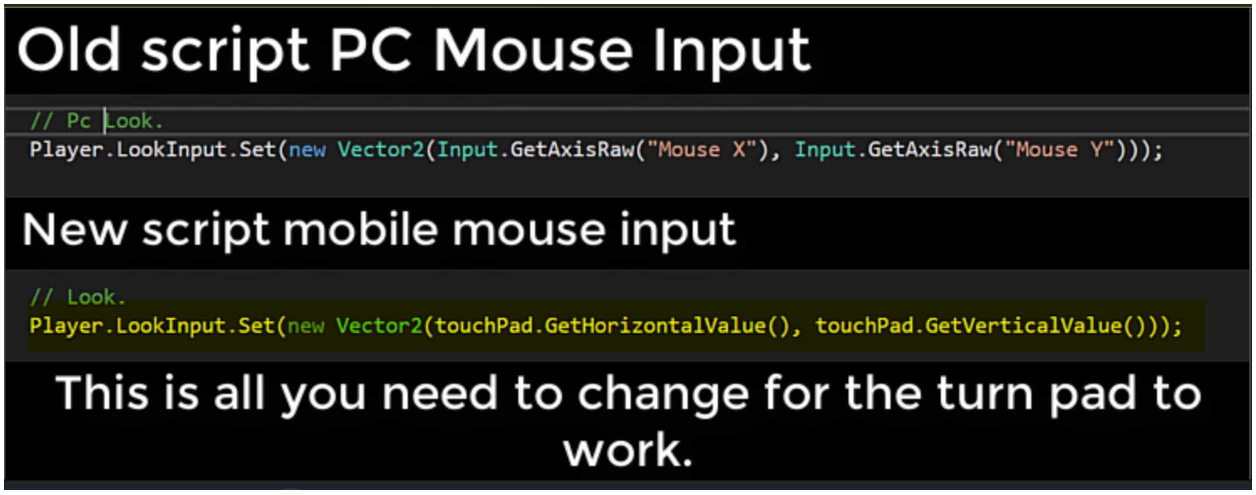


Player Input_Mobile (Sc

Script	PlayerInput_Mo
Joystick	JoyStick Touc
Touch Pad	TurnAndLookT

- **Now go back to your script where you were taking mouse 'Inputs'.**

- **Edit the script like this.**

- 

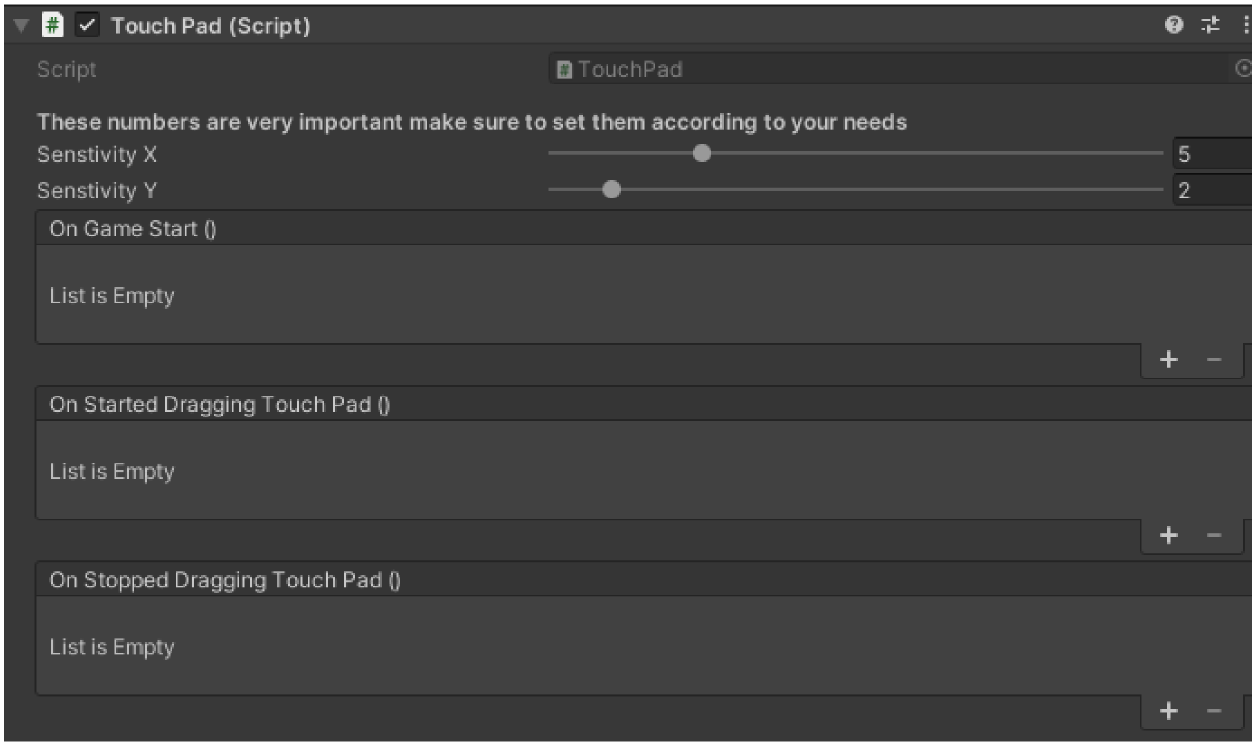
Old script PC Mouse Input

```
// Pc Look.
Player.LookInput.Set(new Vector2(Input.GetAxisRaw("Mouse X"), Input.GetAxisRaw("Mouse Y")));
```

New script mobile mouse input

```
// Look.
Player.LookInput.Set(new Vector2(touchPad.GetHorizontalValue(), touchPad.GetVerticalValue()));
```

This is all you need to change for the turn pad to work.

- **Now you can use the touchpad to rotate your player.**
- This is the inspector view of the 'TurnAndLookTouchPad' object. Here are some events you can use if required. "If you are not sure what are UnityEvents please watch tutorials on UnityEvents"
- **Note: These sensitivity values are very important. Make sure to adjust them according to your needs.**
- 

Touch Pad (Script)

Script: TouchPad

These numbers are very important make sure to set them according to your needs

Senstivity X: 5

Senstivity Y: 2

On Game Start ()

List is Empty

On Started Dragging Touch Pad ()

List is Empty

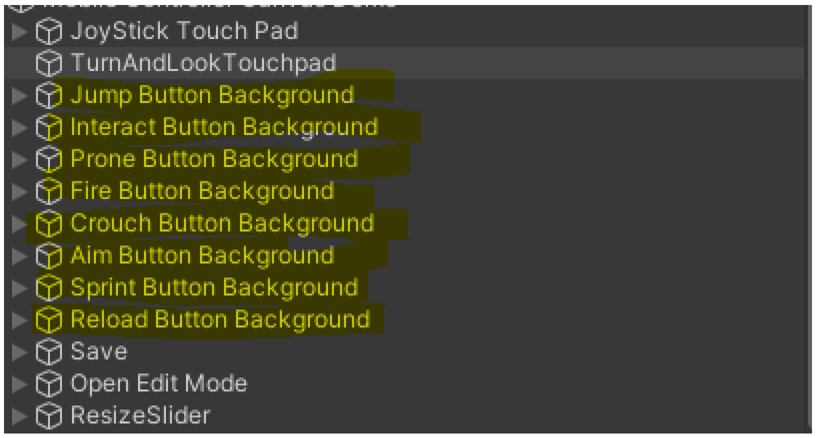
On Stopped Dragging Touch Pad ()

List is Empty

Now your 'TurnAndLookTouchPad' is ready, you can test it now.
Let's move on to Normal Buttons. Example: Jump,Crouch,Prone,Shoot,etc

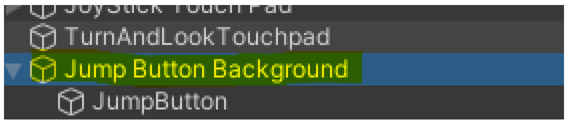
3) Setting up normal Buttons:Here is a list of default/pre-created buttons you can delete the buttons you don't need. Follow these steps so that you can use these buttons.

Note:Don't delete Save,OpenEditMode,ResizeSlider.


JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
ResizeSlider

- Let's set up the jump button.Click 'Jump Button Background'
- **Note: Backup your player script before doing these.**

- Note:Select the parent gameObject 'Jump Button Background' not the child object'JumpButton'. Shown in the image below.


TurnAndLookTouchpad
Jump Button Background
JumpButton

- These 4 components are mandatory for each normal button.Make sure you apply them.Don't apply them if they are already applied.

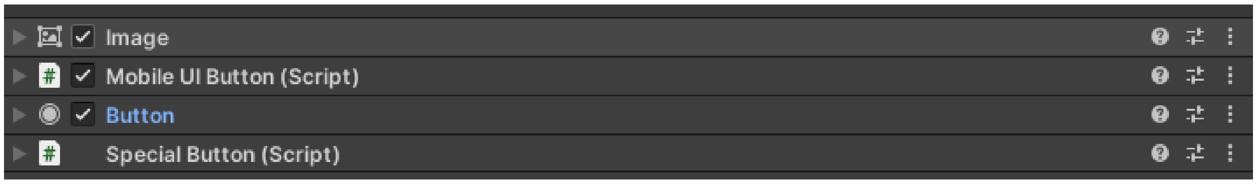
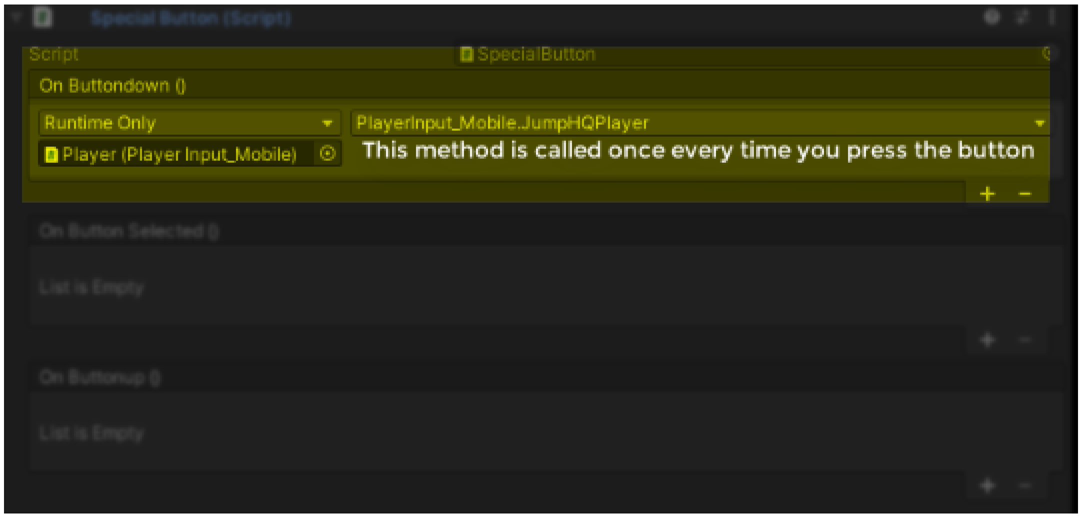


Image
Mobile UI Button (Script)
Button
Special Button (Script)

-
- Now expand the Special Button (Script)

- Here are definitions of all the 3 events from the Special Button script.



Special Button (Script)

Script: SpecialButton

On Buttondown ()

Runtime Only — PlayerInput_Mobile.JumpHQPlayer

Player (Player Input_Mobile)

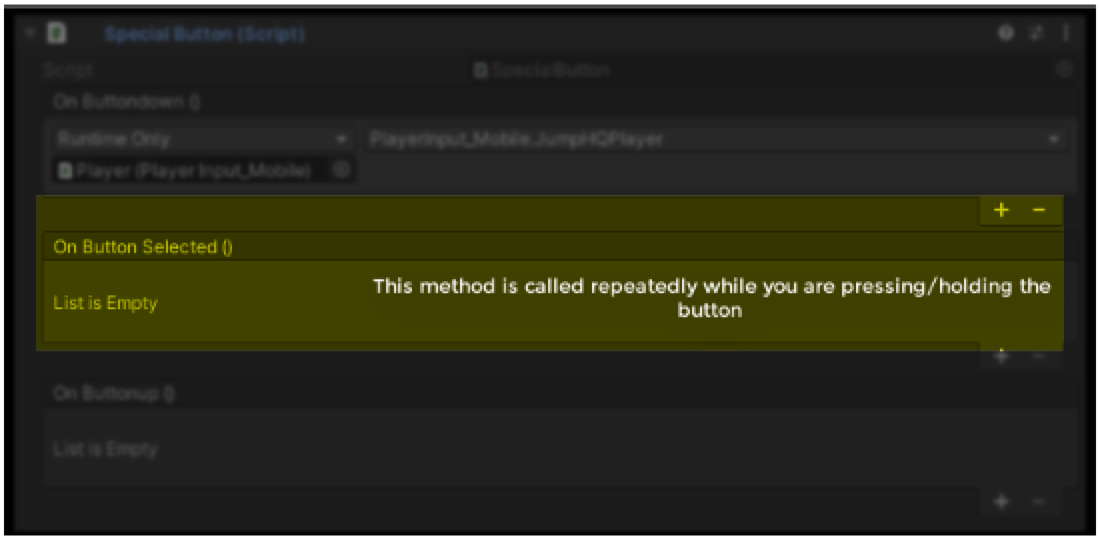
This method is called once every time you press the button

On Button Selected ()

List is Empty

On Buttonup ()

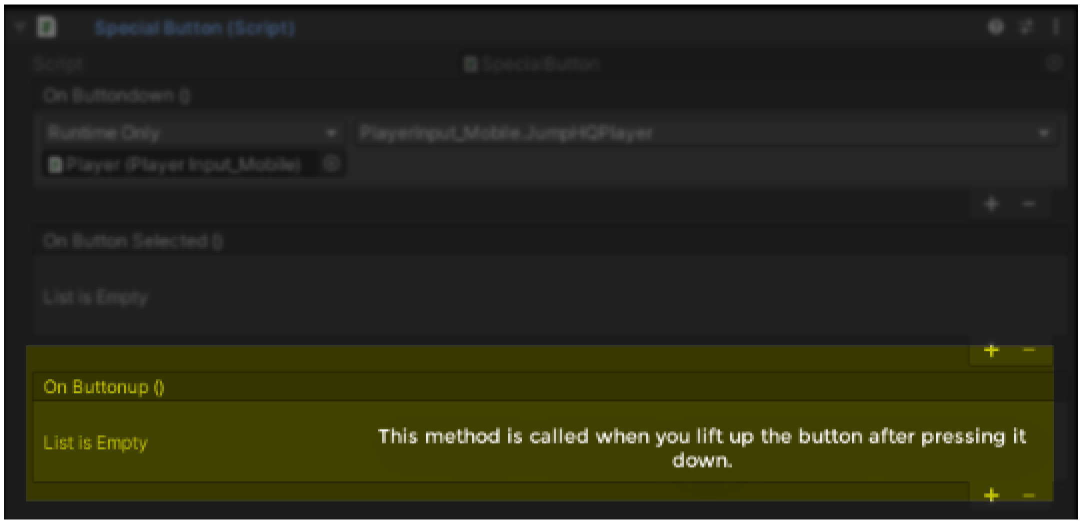
List is Empty



On Button Selected ()

List is Empty

This method is called repeatedly while you are pressing/holding the button

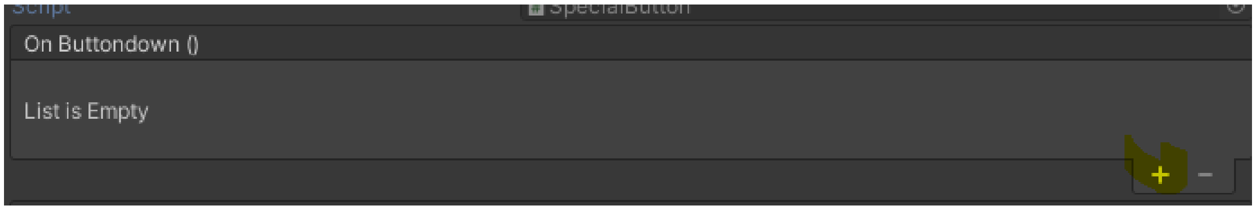


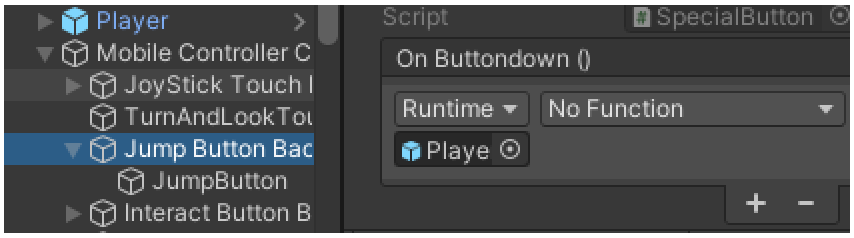
On Buttonup ()

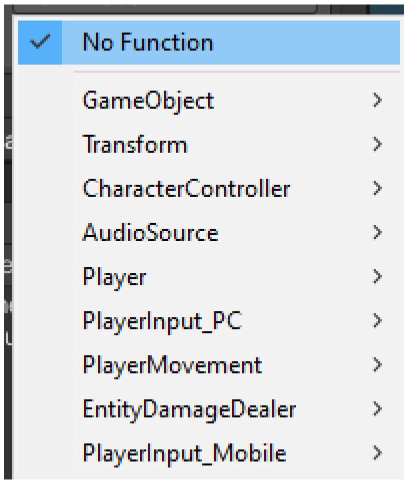
List is Empty

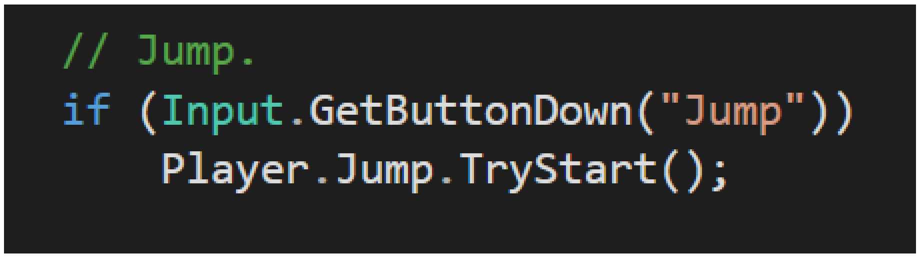
This method is called when you lift up the button after pressing it down.

- Now let's set up our jump button

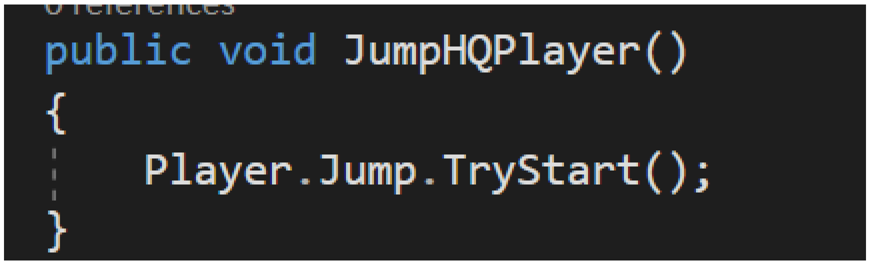
- Click the + icon on OnButtonDown()
- 

Now drag your player object in the object field
- 

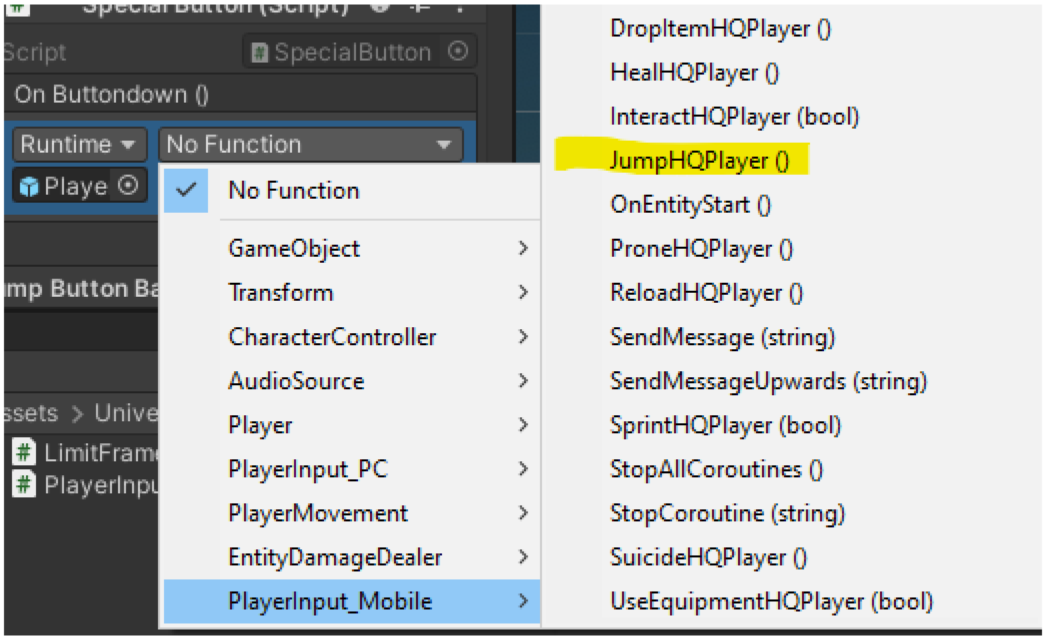
- Select No Function
- 

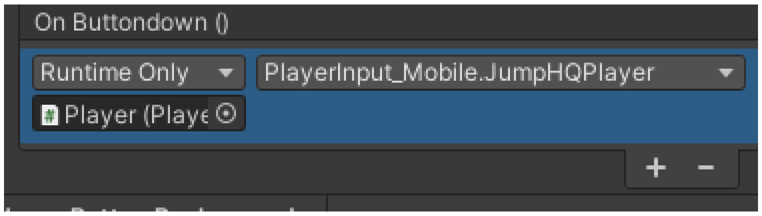
- You will see all the scripts and methods that you can call from this button
- Note: Your methods need to be public in order to show them here.
- 

```
// Jump.
if (Input.GetButtonDown("Jump"))
    Player.Jump.TryStart();
```
- The image above is the pc version of jump method from HQ FPS Template player script. You cannot call this method directly from the event window since it's not a public void.
- So make it a method like this.

- 

```
public void JumpHQPlayer()
{
    Player.Jump.TryStart();
}
```

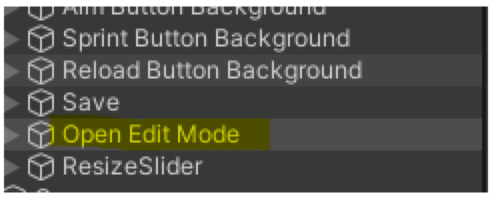
- Note:Now you don’t need to use Input.GetButtonDown because we will call it whenever the UI Button is pressed.
- Now go back to unity re select your noFunction button,select your script that has the jump method and then select the jump method
- 

- After you assign it.
- 

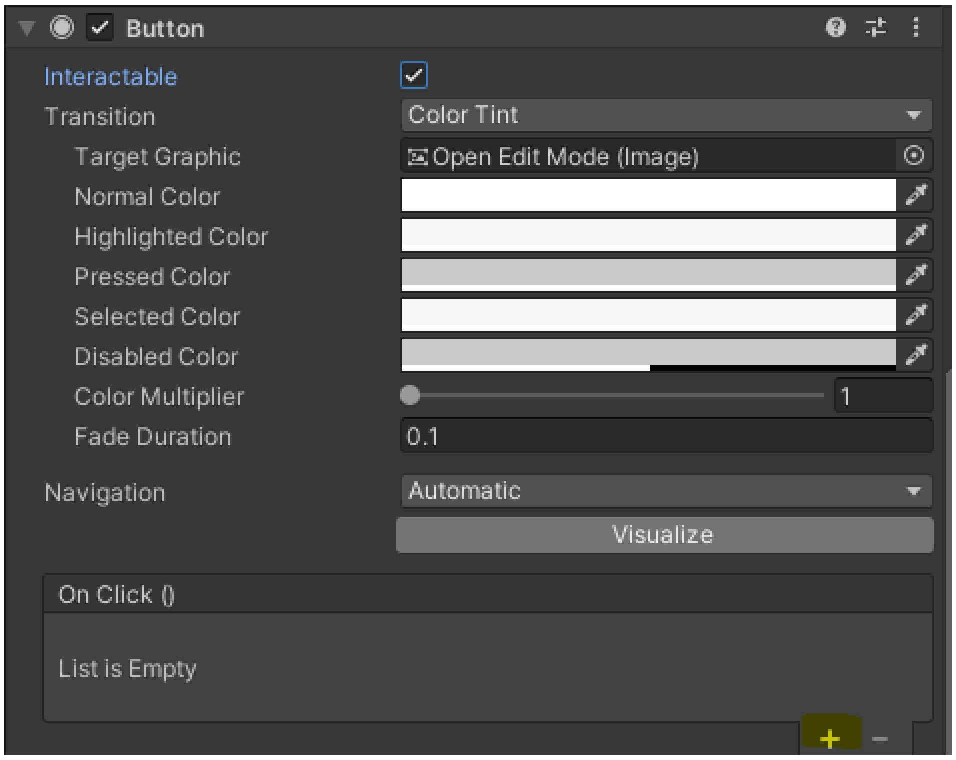
- This is how it would look.
- Now you have to do this will all the buttons remember to create a public void if you don’t have them.Also remember to remove all your Input.Get methods because you won’t need them.

4)Setting up Customization Mode:

- Find ‘Open Edit Mode; Button in your Mobile Controller Canvas Demo.

- 

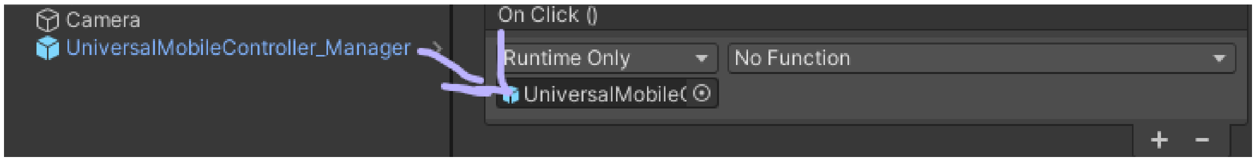
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
ResizeSlider

- In the inspector tab go to the button component of this game object.
- Press the + icon.
- 

Button

Interactable ✓
Transition: Color Tint
Target Graphic: Open Edit Mode (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic
Visualize

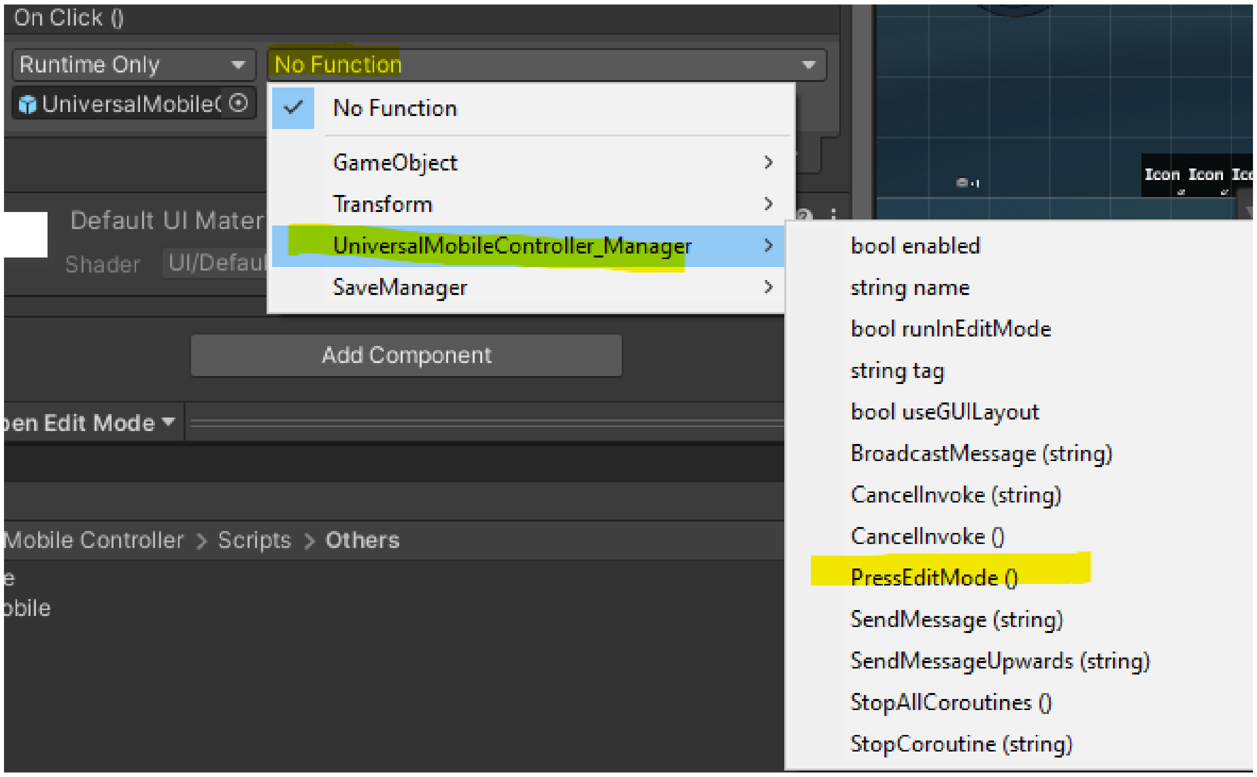
On Click ()
List is Empty

- Drag and drop the ‘UniversalMobileController_Manager’ in this object field.
- Select No Function
- 

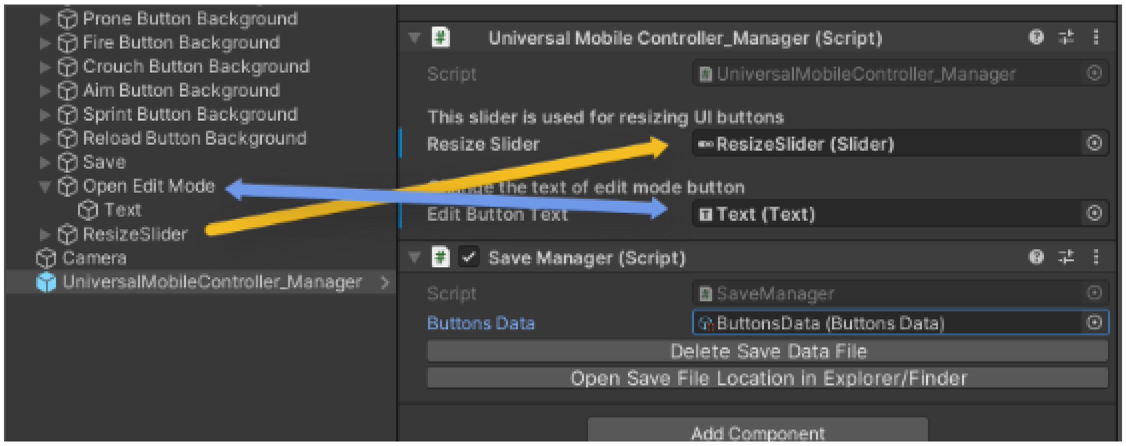
Camera
UniversalMobileController_Manager

On Click ()
Runtime Only | No Function
UniversalMobileC

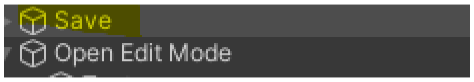
- Select ‘UniversalMobileController_Manager’ then assign the Press Edit Mode method. Shown in the image below.

- 

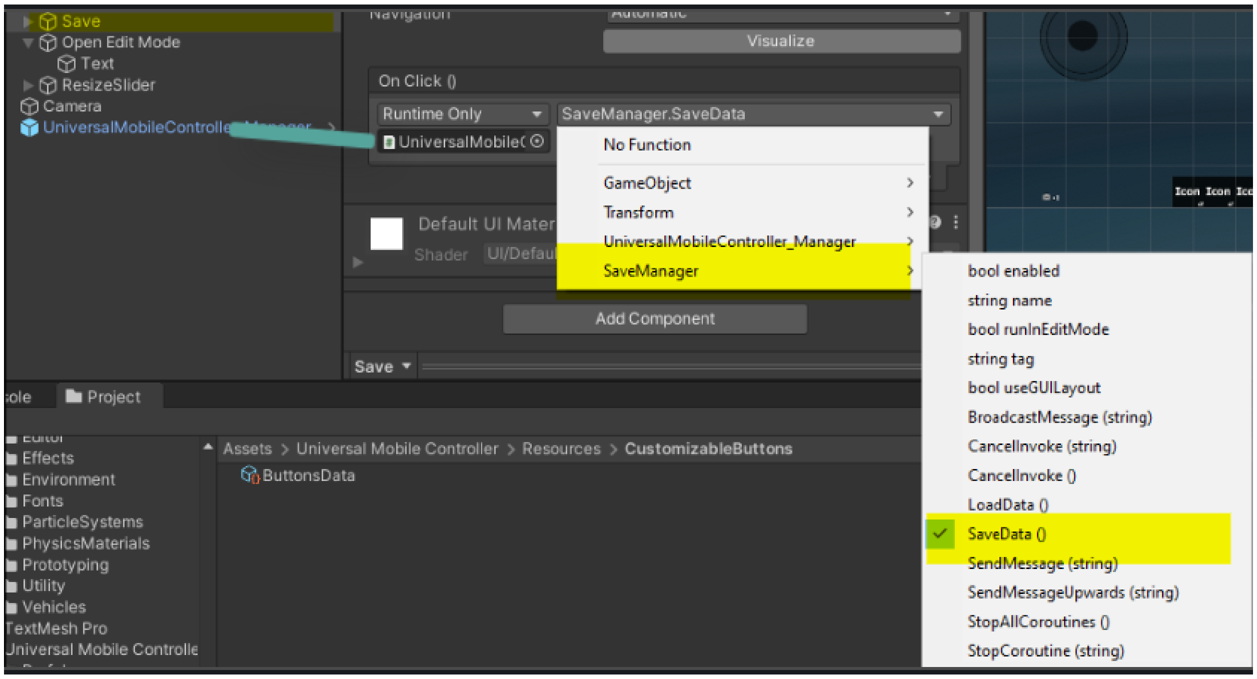
- Go to UniversalMobileController_Manager gameObject. Look in the inspector and make sure these fields are assigned.



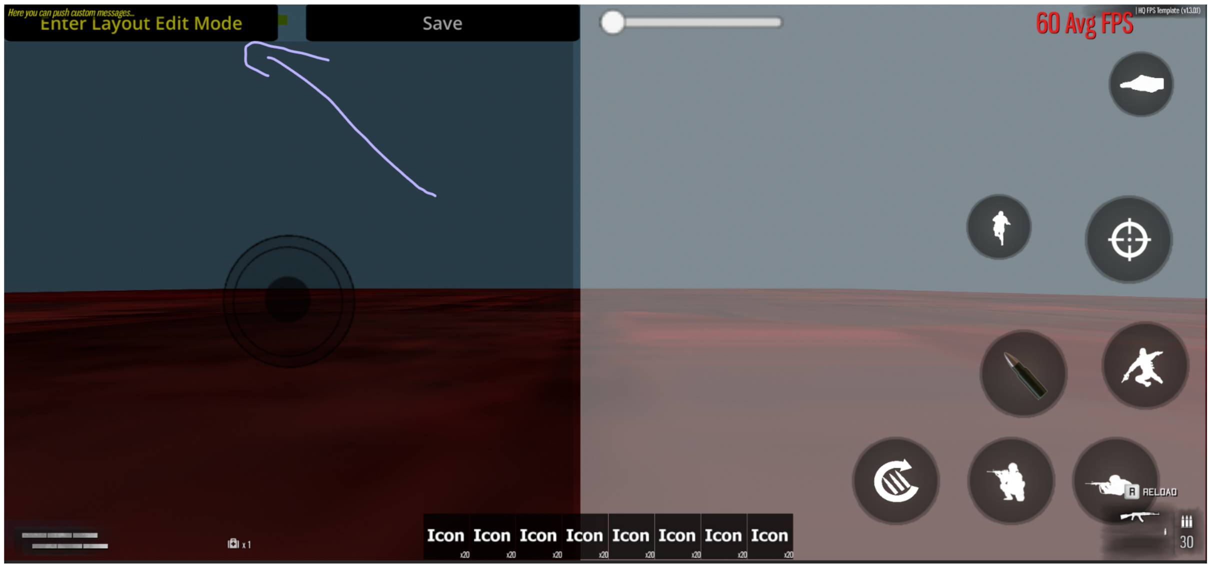
- Go To Save Button

- 

- In the inspector click the + icon on button component and drag the Universal mobile controller manager in the object field.Then select NoFunction and select SaveManager.SaveData method. Shown in the image below.

- 

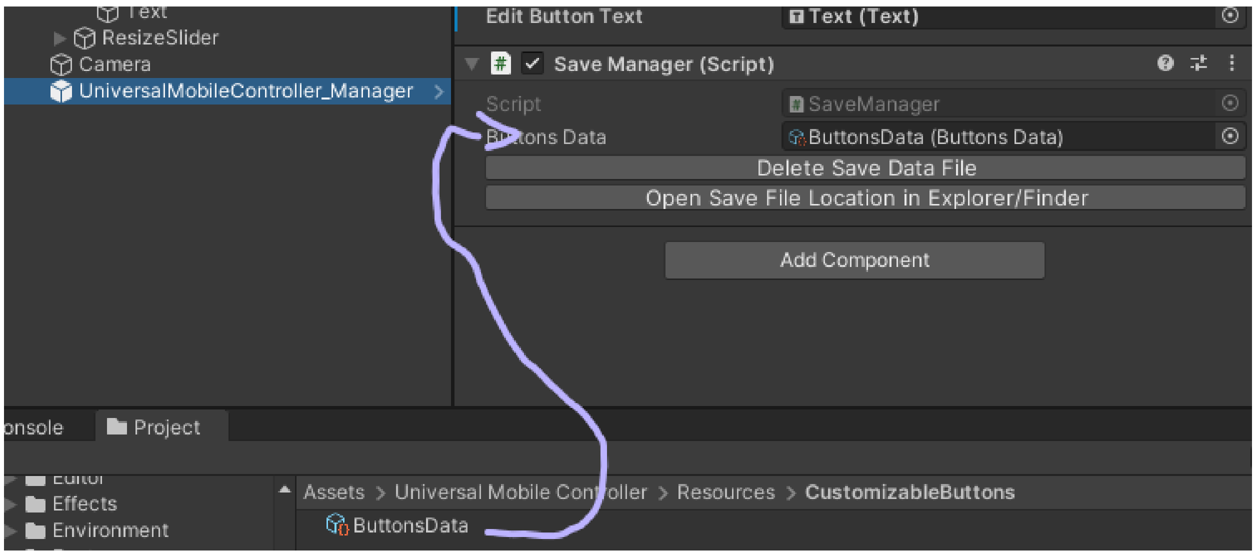
- Now you can customize your layout whenever you press the Enter Edit Layout Mode button in game view



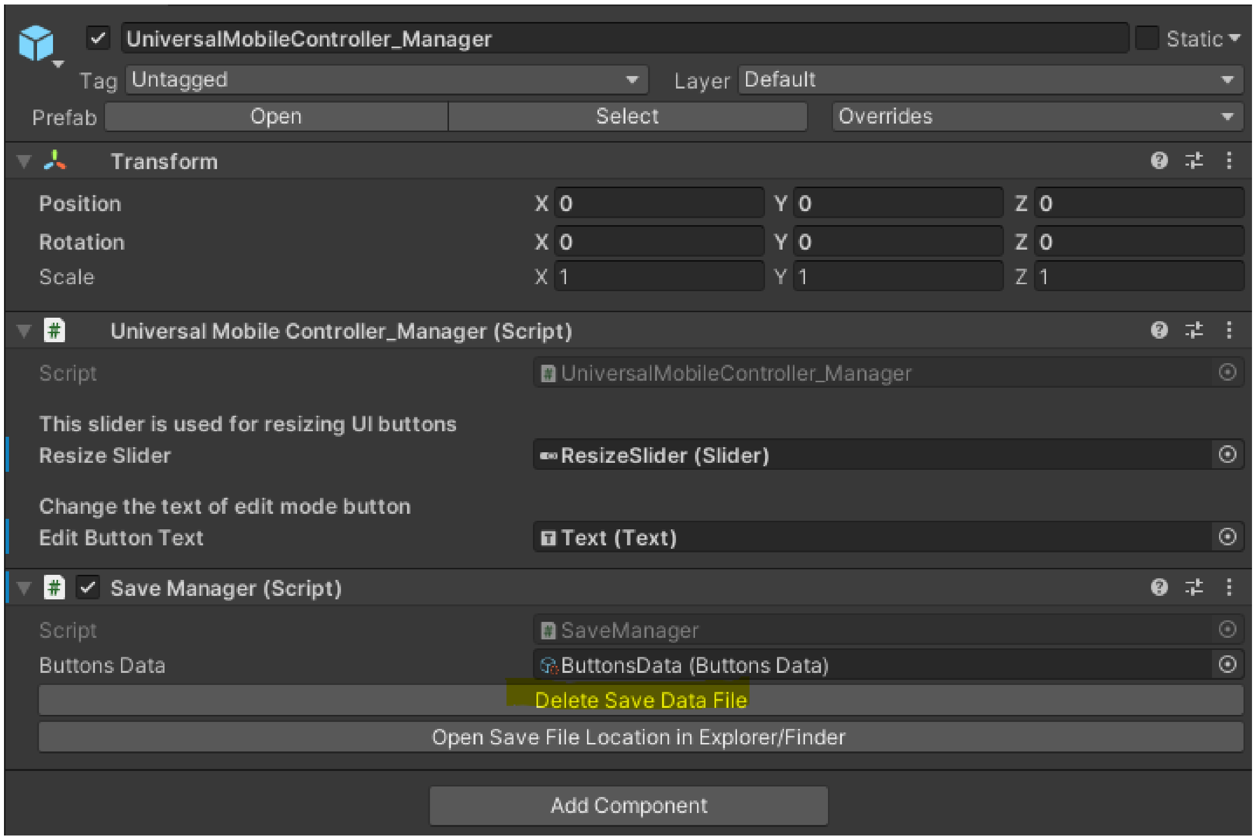
- For customizing just select Enter Edit Mode then select the button you want to customize and drag and place it anywhere you want.
- The slider on the top is for controlling the size of the buttons
- When you are done just press the save button and press the exit enter layout mode button.

5)Setting Up Save Manager:

- Go to UniversalMobileController_Manager GameObject and make sure this is assigned on the Save Manager component.

- 

6)Creating More Buttons/Deleting:

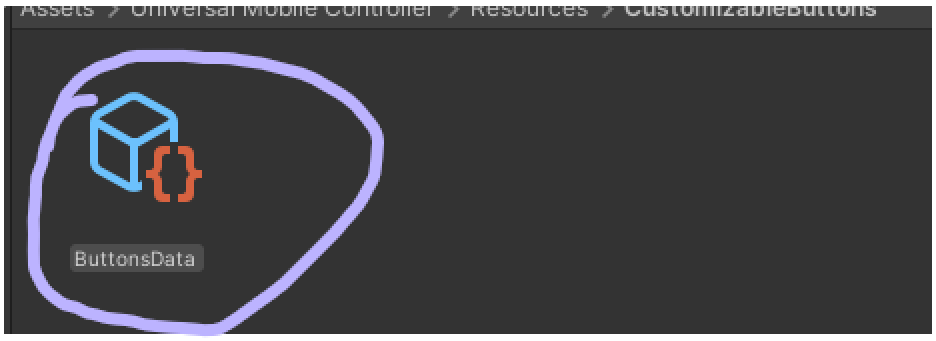
- First go to 'UniversalMobileController_Manager' object and press Delete Save data file
- 

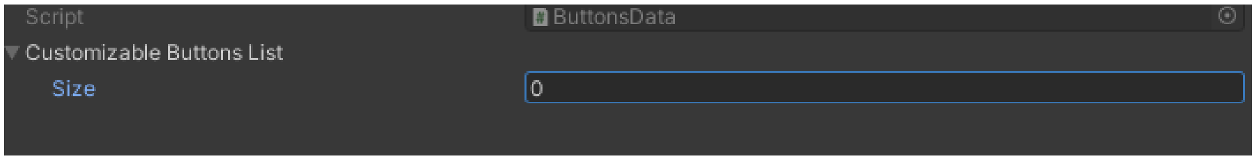
UniversalMobileController_Manager Static
Tag Untagged Layer Default
Prefab Open Select Overrides

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Universal Mobile Controller_Manager (Script)
Script UniversalMobileController_Manager
This slider is used for resizing UI buttons
Resize Slider ResizeSlider (Slider)
Change the text of edit mode button
Edit Button Text Text (Text)

Save Manager (Script)
Script SaveManager
Buttons Data ButtonsData (Buttons Data)

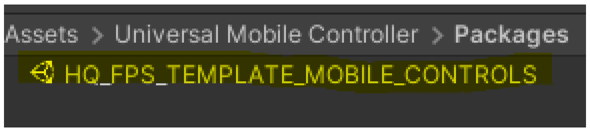
- Just duplicate any normal button like jump,shoot,prone or if you want to delete just delete the button.
- Rename the button
- Change the image sprite on the child object of the button
- Assign the proper methods on the parent gameObject.
- Now go to this object and set the value to 0
- 
Assets > Universal Mobile Controller > Resources > CustomizableButtons

- 
Script ButtonsData
Customizable Buttons List
Size 0
- Done.

Integrations: HQ FPS Template

Note: Only follow these steps if you are using HQ FPS TEMPLATE.

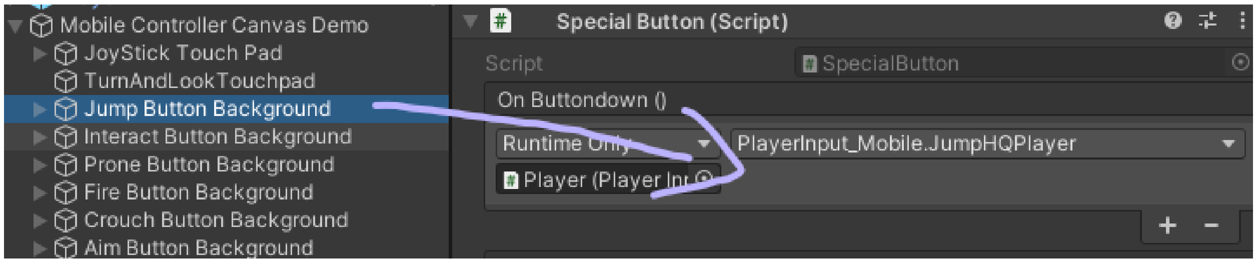
If you are using HQ Fps Template like me then all you need to do is import the HQ FPS Template Mobile Controls Package. From the Packages folder.
Shown in the image below.


Assets > Universal Mobile Controller > Packages
HQ_FPS_TEMPLATE_MOBILE_CONTROLS

Just import this package and attach the Player Mobile Input script to your player prefab make sure to remove the player input pc script from the player.

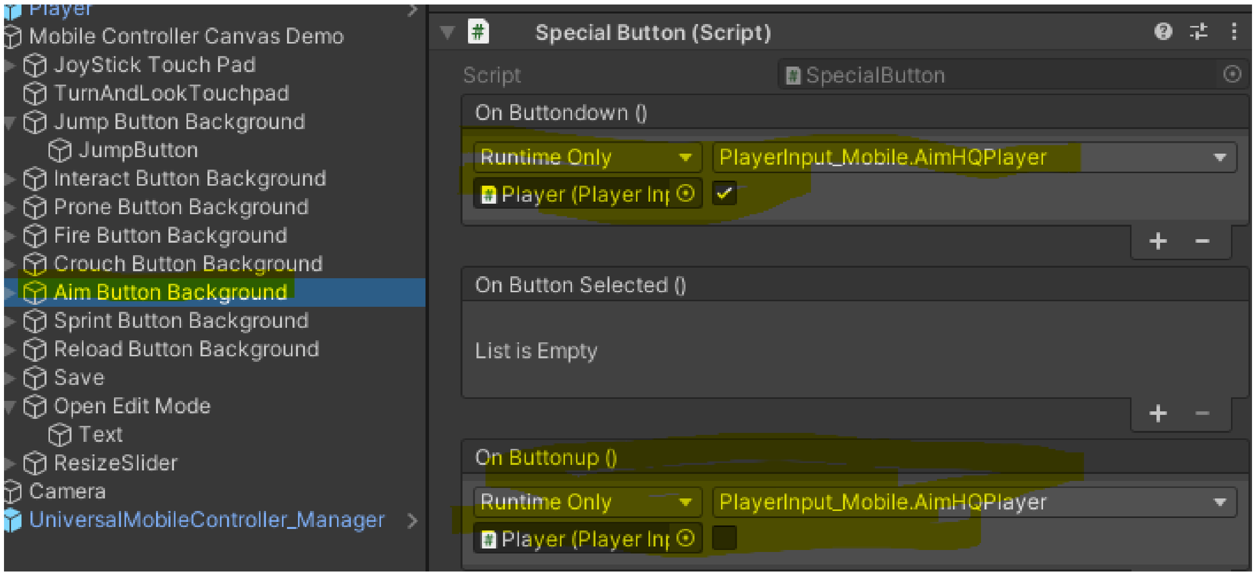
After assigning it to player prefab assign the object fields(Joystick, TouchPad) on the Player_Mobile_Input script(We did this above)

Now go to each normal button and just assign the methods like we did above.
Note: Assign the methods according to the button names
Here are list of all the buttons with methods you need to assign, according to button names:



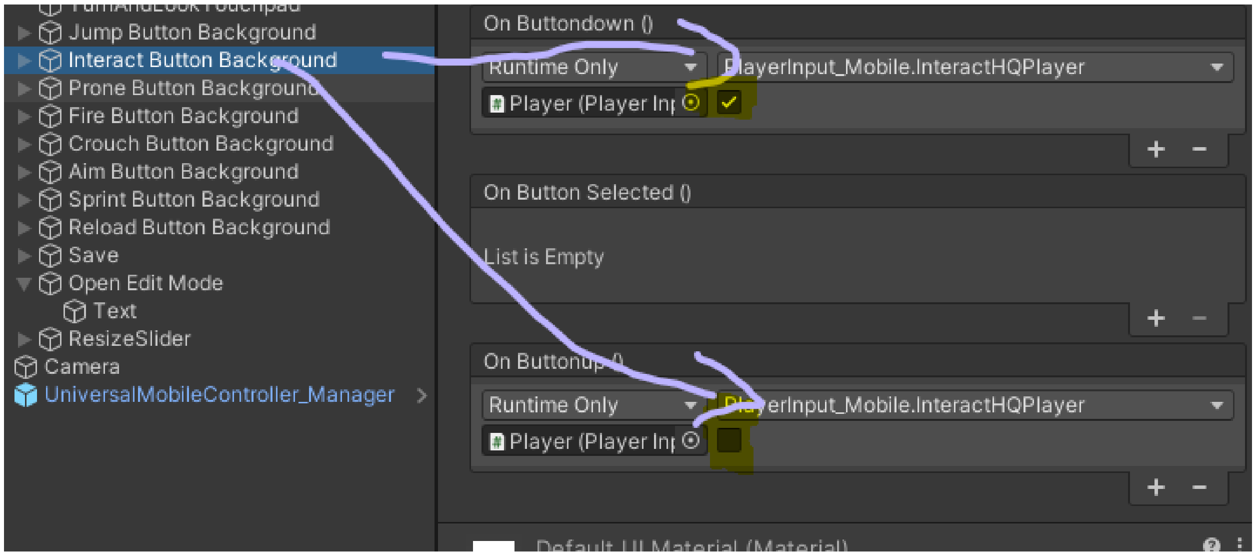
Mobile Controller Canvas Demo
JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background

Special Button (Script)
Script: SpecialButton
On Buttondown ()
Runtime Only — PlayerInput_Mobile.JumpHQPlayer
Player (Player Inp



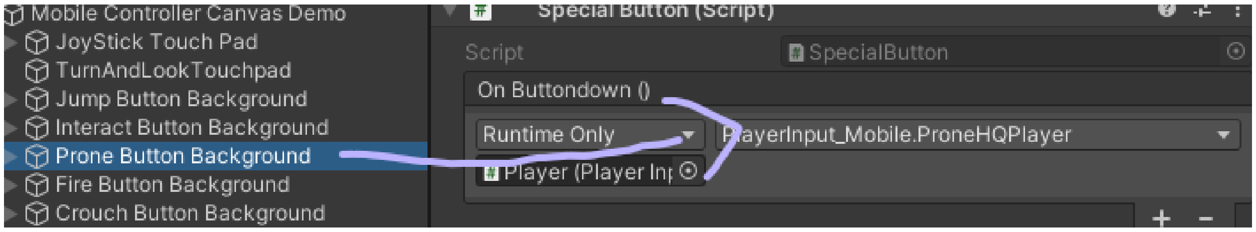
Player
Mobile Controller Canvas Demo
JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
JumpButton
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
Text
ResizeSlider
Camera
UniversalMobileController_Manager

Special Button (Script)
Script: SpecialButton
On Buttondown ()
Runtime Only — PlayerInput_Mobile.AimHQPlayer
Player (Player Inp ✓
On Button Selected ()
List is Empty
On Buttonup ()
Runtime Only — PlayerInput_Mobile.AimHQPlayer
Player (Player Inp ☐



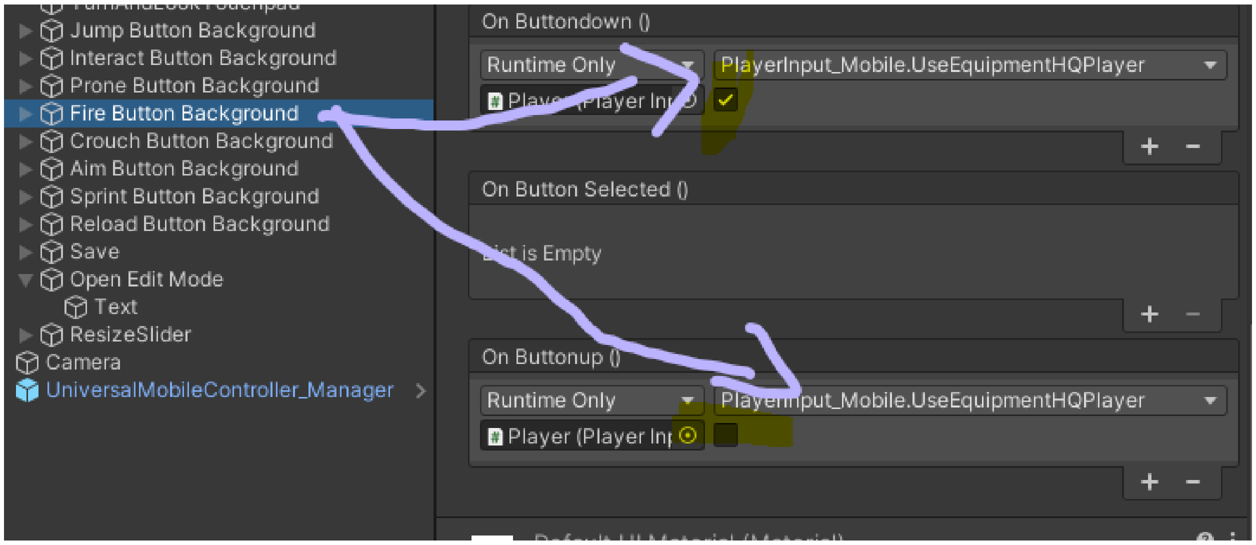
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
Text
ResizeSlider
Camera
UniversalMobileController_Manager

On Buttondown ()
Runtime Only — PlayerInput_Mobile.InteractHQPlayer
Player (Player Inp ✓
On Button Selected ()
List is Empty
On Buttonup ()
Runtime Only — PlayerInput_Mobile.InteractHQPlayer
Player (Player Inp ☐
Default UI Material (Material)



Mobile Controller Canvas Demo
JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background

Special Button (Script)
Script: SpecialButton
On Buttondown ()
Runtime Only — PlayerInput_Mobile.ProneHQPlayer
Player (Player Inp

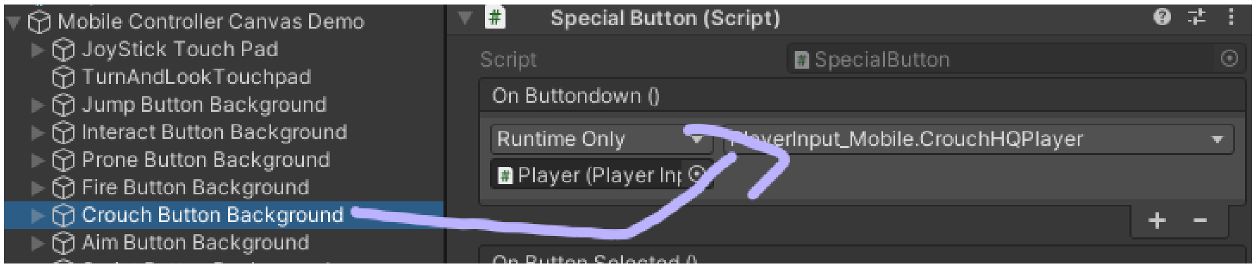


Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
Text
ResizeSlider
Camera
UniversalMobileController_Manager

On Buttondown ()
Runtime Only | PlayerInput_Mobile.UseEquipmentHQPlayer
Player (Player Inp | ✓

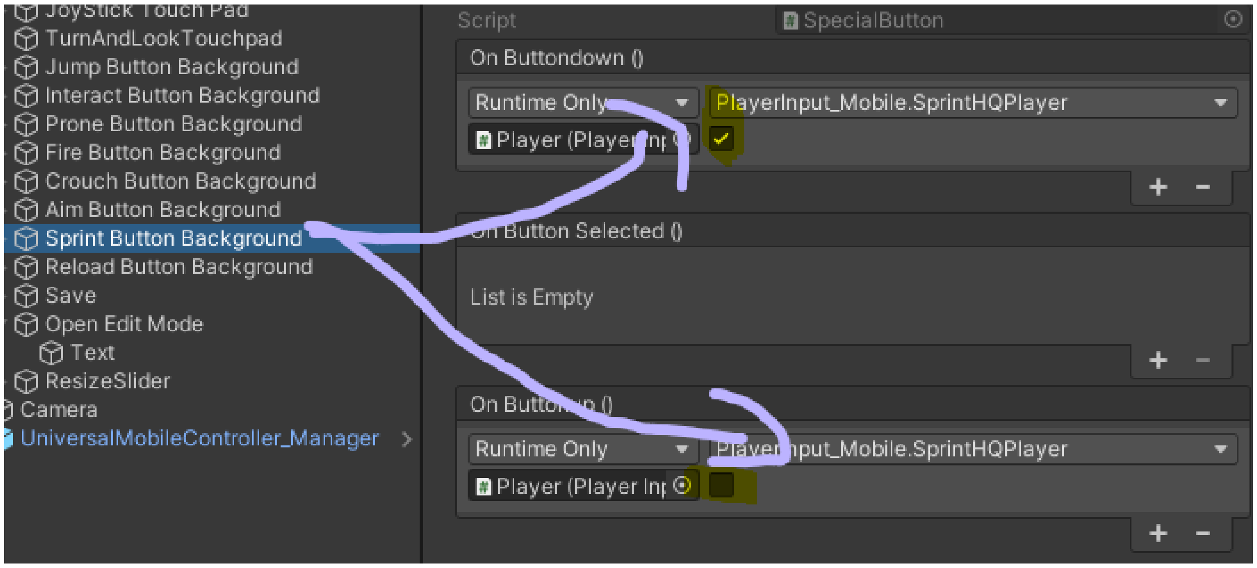
On Button Selected ()
List is Empty

On Buttonup ()
Runtime Only | PlayerInput_Mobile.UseEquipmentHQPlayer
Player (Player Inp | ☐



Mobile Controller Canvas Demo
JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background

Special Button (Script)
Script | SpecialButton
On Buttondown ()
Runtime Only | PlayerInput_Mobile.CrouchHQPlayer
Player (Player Inp
On Button Selected ()

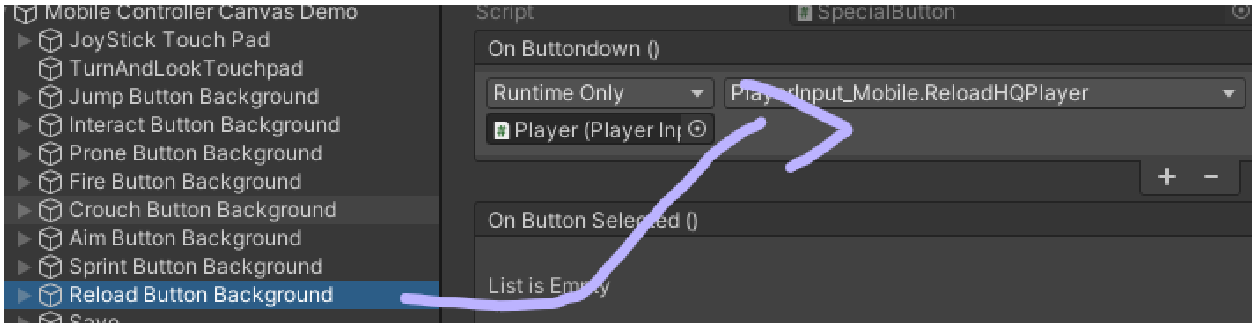


JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save
Open Edit Mode
Text
ResizeSlider
Camera
UniversalMobileController_Manager

Script | SpecialButton
On Buttondown ()
Runtime Only | PlayerInput_Mobile.SprintHQPlayer
Player (Player Inp | ✓

On Button Selected ()
List is Empty

On Buttonup ()
Runtime Only | PlayerInput_Mobile.SprintHQPlayer
Player (Player Inp | ☐



Mobile Controller Canvas Demo
JoyStick Touch Pad
TurnAndLookTouchpad
Jump Button Background
Interact Button Background
Prone Button Background
Fire Button Background
Crouch Button Background
Aim Button Background
Sprint Button Background
Reload Button Background
Save

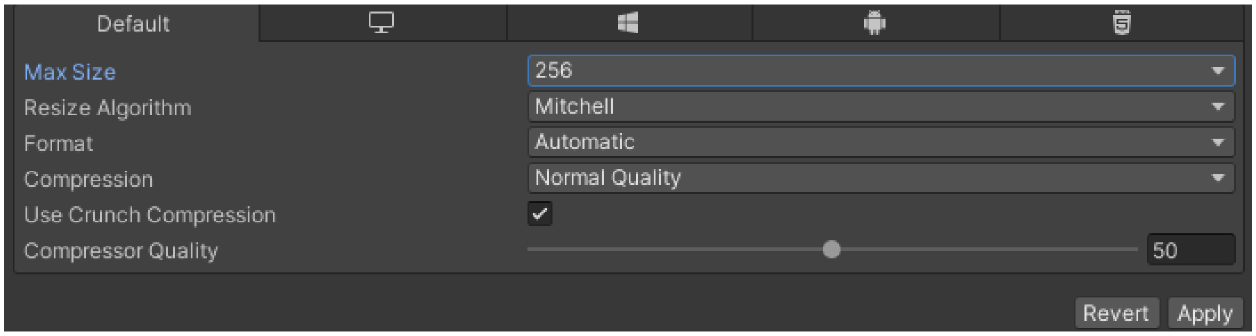
Script | SpecialButton
On Buttondown ()
Runtime Only | PlayerInput_Mobile.ReloadHQPlayer
Player (Player Inp

On Button Selected ()
List is Empty

Optimising HQ FPS Template For Mobile:

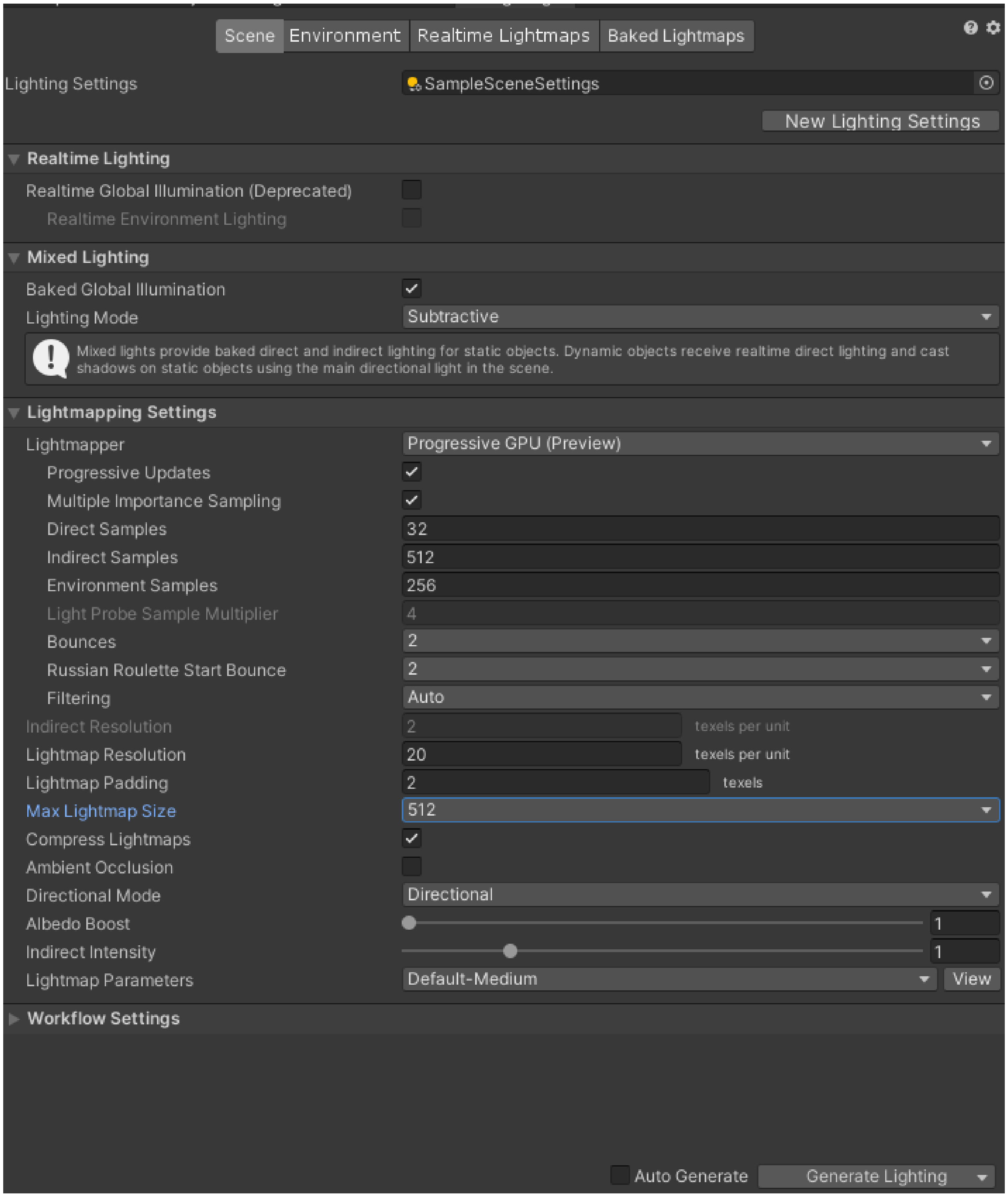
Note:Make sure to create a backup of your project before doing these steps (also you only need to do these steps only if you are using HQ FPS Template if you have your own controller asset these steps are 100% not necessary).

- **Go to GameManager.cs OnStart function and remove the Shader.WarmUpAllShaders();**
- **Select all the textures from hq fps template and set their size to 256 or 512.Also check the use crunch compression bool.**



Max Size: 256
Resize Algorithm: Mitchell
Format: Automatic
Compression: Normal Quality
Use Crunch Compression: ✓
Compressor Quality: 50

- **Delete Post Processing Manager from your scene or use an android optimised post processing profile.**
- **Delete all lighting effects like reflection probes,light groups**
- **Set Quality Settings to low**
- **Set Directional Light to Hard Shadows only and Strength to 0.7**
- **Set all the non moving objects like walls,trees,buildings to static.**
- **Go to the occlusion culling tab and bake it.**
- **Go to Lighting tab and generate lighting using these settings**



Scene | Environment | Realtime Lightmaps | Baked Lightmaps

Lighting Settings: SampleSceneSettings

New Lighting Settings

Realtime Lighting

Realtime Global Illumination (Deprecated): ☐

Realtime Environment Lighting: ☐

Mixed Lighting

Baked Global Illumination: ✓

Lighting Mode: Subtractive

Mixed lights provide baked direct and indirect lighting for static objects. Dynamic objects receive realtime direct lighting and cast shadows on static objects using the main directional light in the scene.

Lightmapping Settings

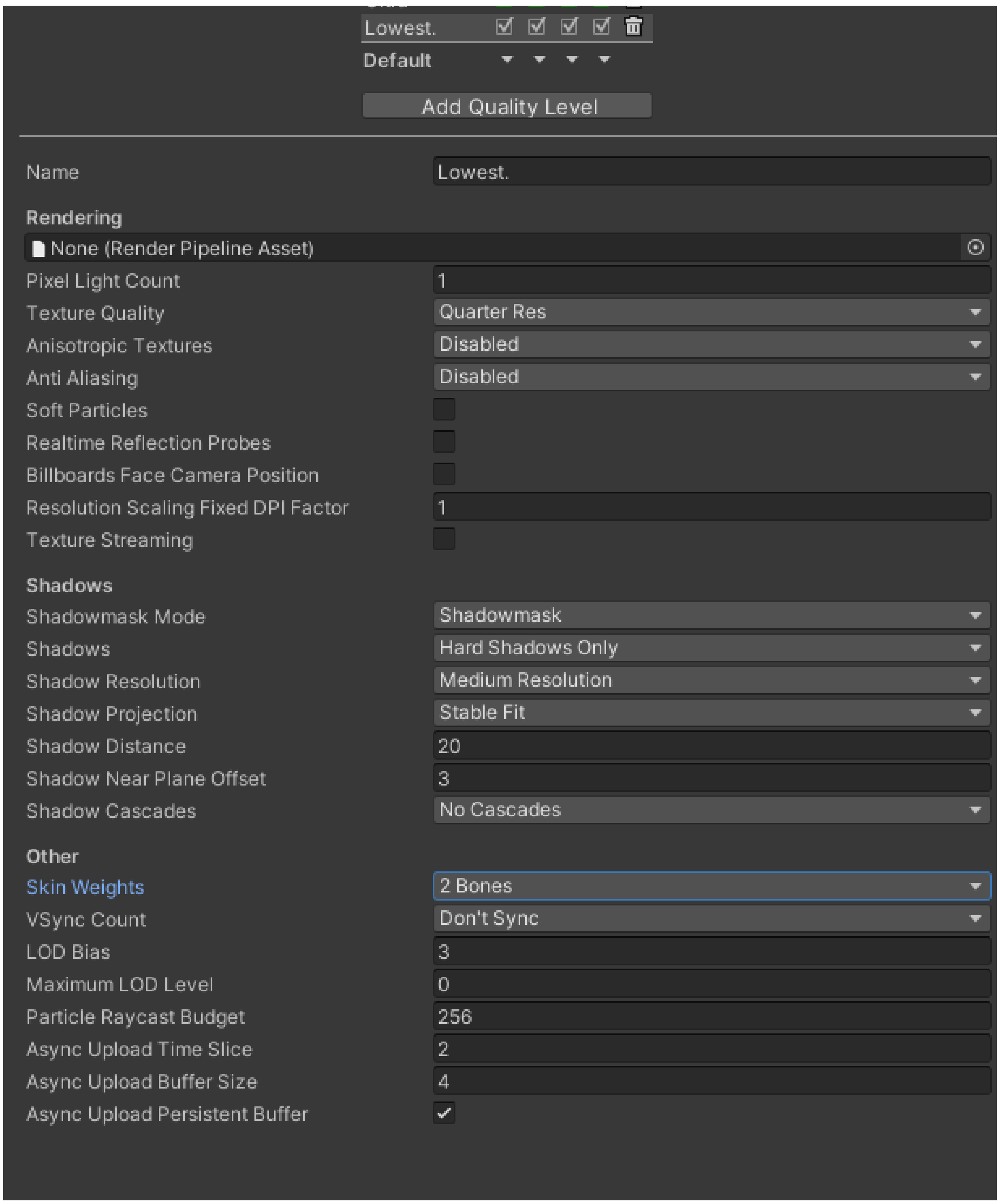
Lightmapper	Progressive GPU (Preview)
Progressive Updates	✓
Multiple Importance Sampling	✓
Direct Samples	32
Indirect Samples	512
Environment Samples	256
Light Probe Sample Multiplier	4
Bounces	2
Russian Roulette Start Bounce	2
Filtering	Auto
Indirect Resolution	2 texels per unit
Lightmap Resolution	20 texels per unit
Lightmap Padding	2 texels
Max Lightmap Size	512
Compress Lightmaps	✓
Ambient Occlusion	☐
Directional Mode	Directional
Albedo Boost	1
Indirect Intensity	1
Lightmap Parameters	Default-Medium (View)

Workflow Settings

☐ Auto Generate | Generate Lighting

- **Go to Quality settings create a new quality level and select these**

settings



Name	Lowest.
Rendering	
None (Render Pipeline Asset)	
Pixel Light Count	1
Texture Quality	Quarter Res
Anisotropic Textures	Disabled
Anti Aliasing	Disabled
Soft Particles	☐
Realtime Reflection Probes	☐
Billboards Face Camera Position	☐
Resolution Scaling Fixed DPI Factor	1
Texture Streaming	☐
Shadows	
Shadowmask Mode	Shadowmask
Shadows	Hard Shadows Only
Shadow Resolution	Medium Resolution
Shadow Projection	Stable Fit
Shadow Distance	20
Shadow Near Plane Offset	3
Shadow Cascades	No Cascades
Other	
Skin Weights	2 Bones
VSync Count	Don't Sync
LOD Bias	3
Maximum LOD Level	0
Particle Raycast Budget	256
Async Upload Time Slice	2
Async Upload Buffer Size	4
Async Upload Persistent Buffer	✓

- **Set the world camera far clip to 300-400 or even less if possible.**