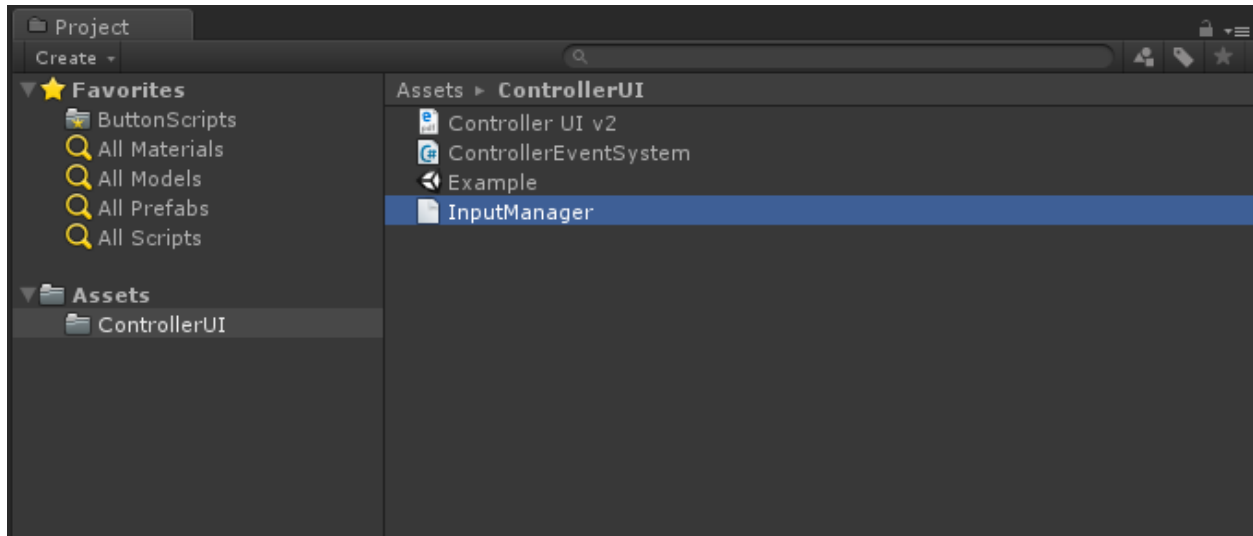# CONTROLLER UI *V2*

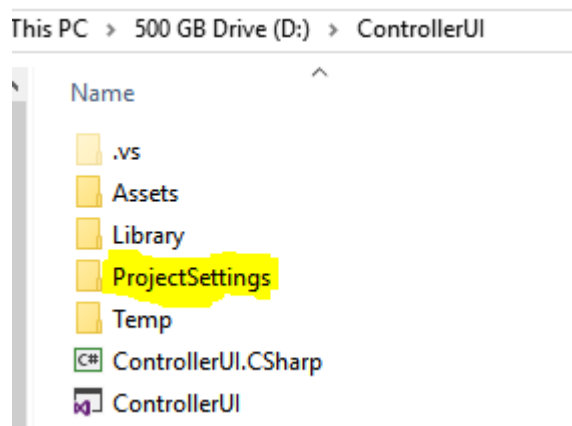*Option 1* Import Setup *** THIS WILL DELETE ALL CUSTOM INPUTS PRIOR TO ADDING**

Select the InputManager in the ControllerUI folder

Copy the file to the ProjectSettings folder in your project's main folder

_____

Option 2 Manual

Windows

Mac

DPadY	
Name	DPadY
Descriptive Nar	
Descriptive Neg	
Negative Buttor	
Positive Button	joystick button 5
Alt Negative Bu	
Alt Positive But	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	☐
Invert	☐
Type	Key or Mouse Button
Axis	X axis
Joy Num	Joystick 1

DPadY-	
Name	DPadY-
Descriptive Nar	
Descriptive Neg	
Negative Buttor	
Positive Button	joystick button 6
Alt Negative Bu	
Alt Positive But	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	☐
Invert	☐
Type	Key or Mouse Button
Axis	X axis
Joy Num	Joystick 1

DPadX	
Name	DPadX
Descriptive Nar	
Descriptive Neg	
Negative Buttor	
Positive Button	joystick button 7
Alt Negative Bu	
Alt Positive But	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	☐
Invert	☐
Type	Key or Mouse Button
Axis	X axis
Joy Num	Joystick 1

▶ DPadY
▶ DPadY-

DPadX-	
Name	DPadX-
Descriptive Nar	
Descriptive Neg	
Negative Buttor	
Positive Button	joystick button 8
Alt Negative Bu	
Alt Positive But	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	☐
Invert	☐
Type	Key or Mouse Button
Axis	X axis
Joy Num	Joystick 1

SETTING UP THE CONTROLLER SYSTEM

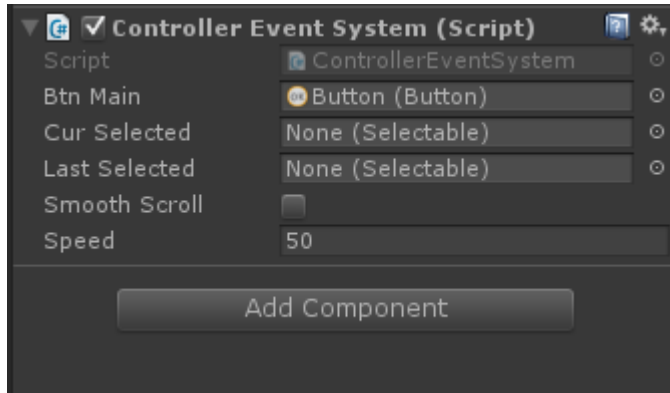*Step 1*

Drag the ControllerEventSystem.cs onto the canvas that you are using.
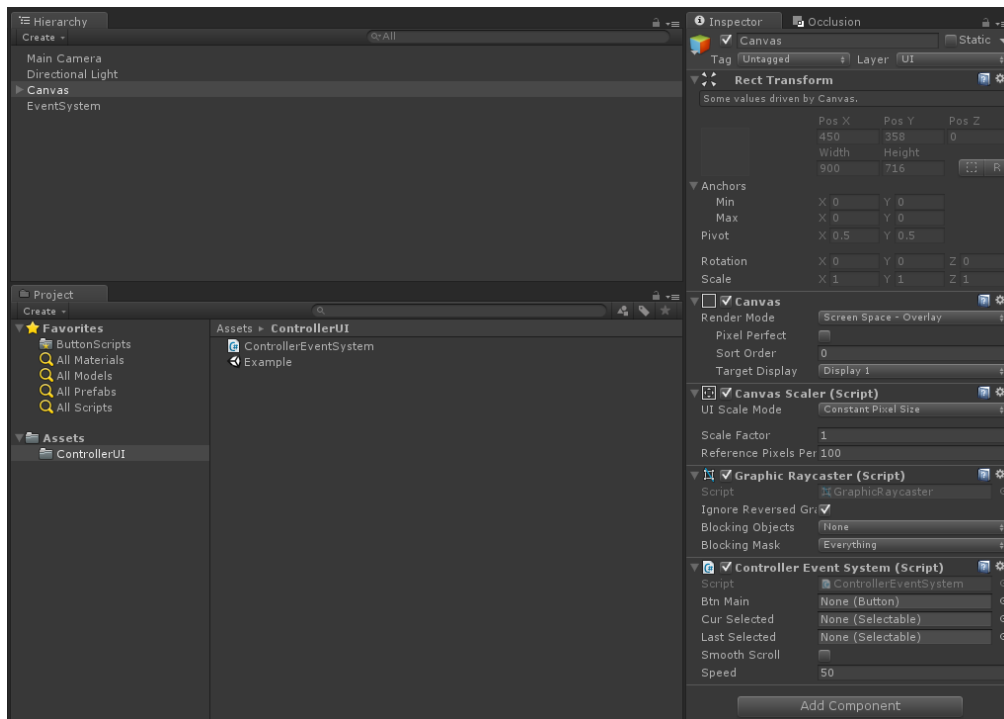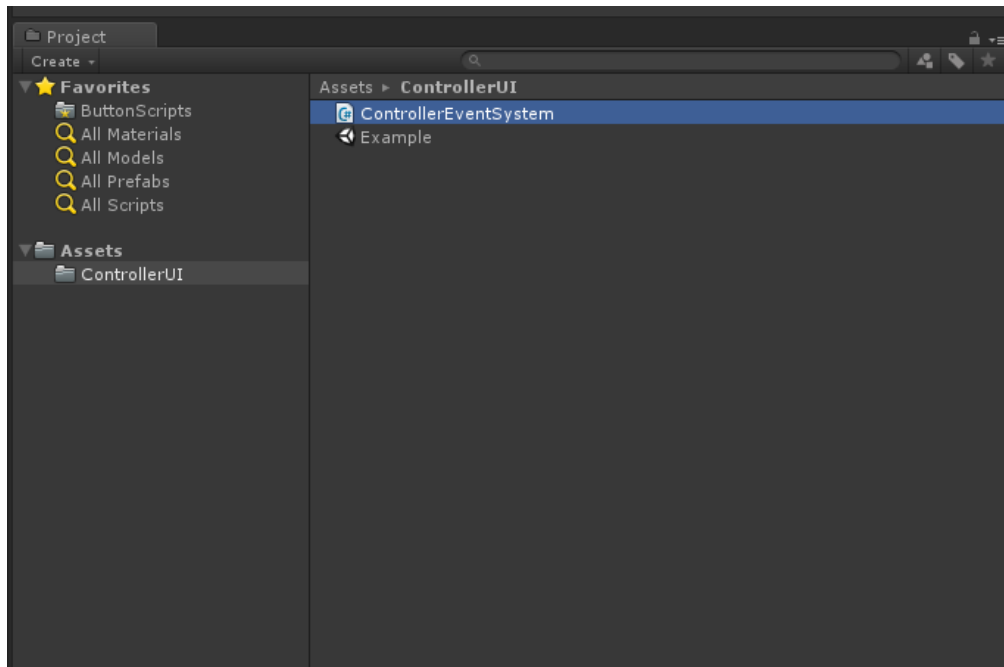
On Btn Main drag the object that starts your UI. For example, if I am creating a list my button main would be the first button in the list.
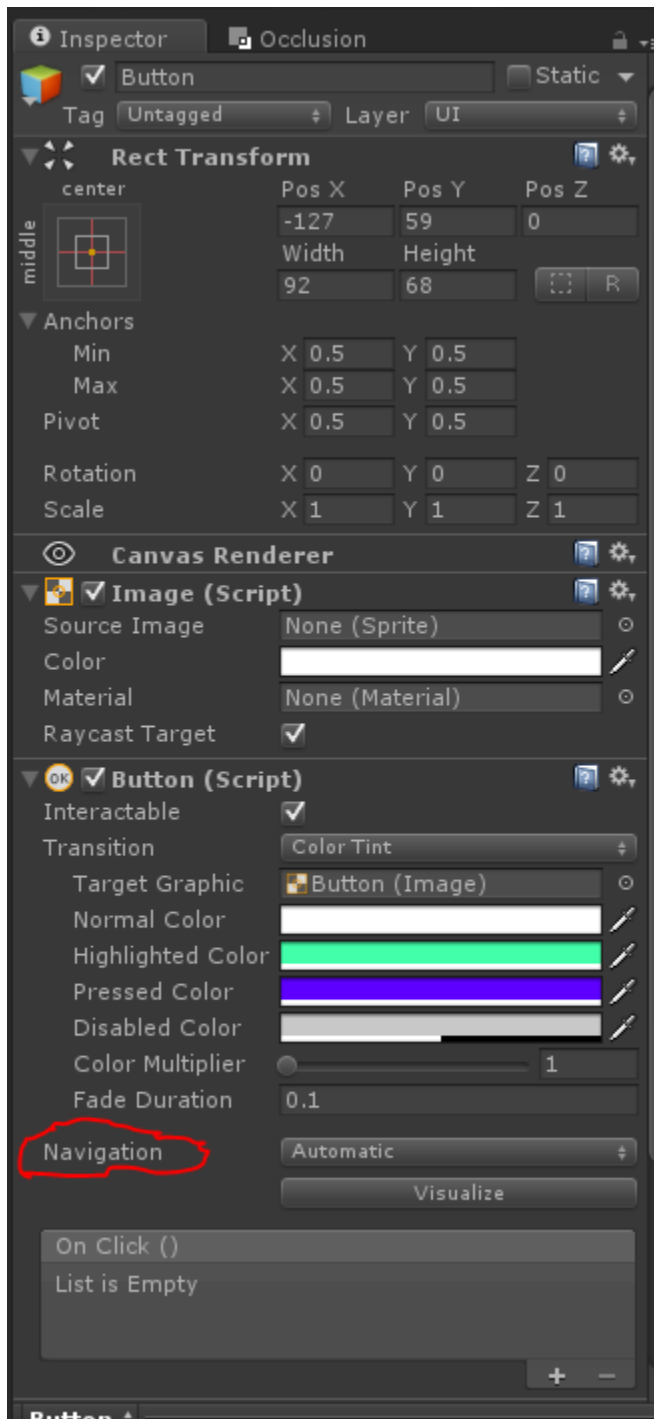

IF THIS STEP IS NOT DONE A RANDOM BUTTON WILL BE ASSIGNED

FIXING NAVIGATION ISSUES

The new system uses navigation events which can be altered inside of the editor. To assign more strict navigation select the UI object you want to alter:

Then in the inspector go to Navigation:

There you can choose how the object navigates to one another.

KNOWING THE CODE

All the inputs can be renamed in the update function:

```
// Sends the Navigation events for windows
    if (Input.GetAxis("DPadX") > 0 && canSelect)
    {
        StartCoroutine(CanMove());
        SelectRight();
```

You can subscribe events to the Click and Select actions. This can be helpful when adding sound to your buttons.

For example, in a new script you can write:

```
// Initializes
  void Start () {

        ControllerEventSystem.onClicked += voidName;
}

// Plays audio
Void voidName{

AudioSource audio = GetComponent<AudioSource>();
        audio.Play();
}
```

See more at: https://unity3d.com/learn/tutorials/modules/intermediate/scripting/events

Feel free to reach out to contact@babilinapps.com