Table of contents

Mouse As Controller

by a1creator

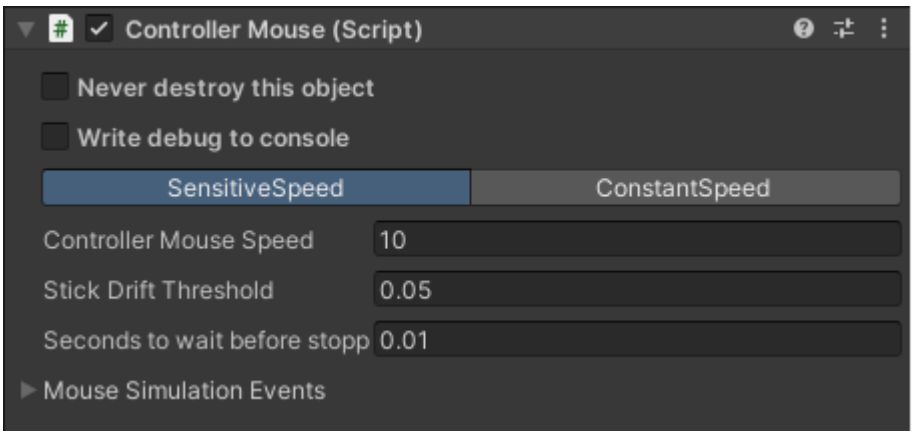
Documentation

Set-up

- Go into the folder [*a1creator* > *MouseAsController* > ***Prefabs***]
- Drag the EventSystem prefab into your scene

- Press play and try using the mouse with your controller.
  It should work.

Adjusting



This component is found on the EventSystem object.

• “Never destroy this object”

Set to true in your main scene if you intend on using this package in your whole project.
You can have the prefab in all scenes, allowing you to test each scene individually. The object with this setting set to true will always take priority and delete the other, when you switch scenes. I advise having it set to true in the first loaded scene.
Only set it to true on one instance.

• “Write debug to console”

If the prefab is not working as intended, try setting this to true and troubleshoot. If it’s not outputting anything when using the controller, the controller/connection is the problem.

• “SensitiveSpeed” VS. “ConstantSpeed”

Determines the movement behaviour of the cursor when using controller. Constant speed means that no matter how soft you press the stick, it will have a speed factor of 1f when moving. If set to sensitive speed, the speed factor will equal the stick’s distance from the center to the edge. D-Pad and keys as movement will use constant speed either way. *This only determines the behaviour of thumb sticks.*

• “Controller Mouse Speed”

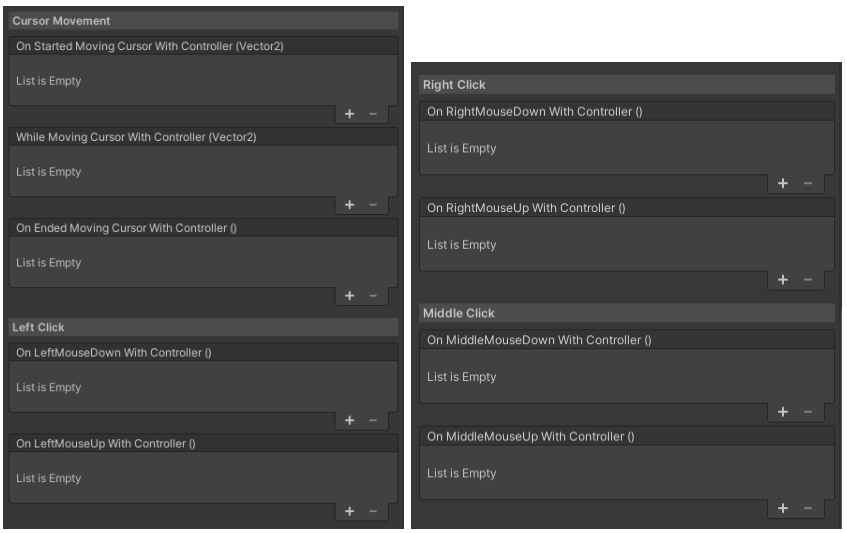
Determines the movement speed of the cursor when using controller.

• “Stick Drift Threshold”

Cursor will only start moving when the stick is moved above this threshold. The value is 0f at stick’s center and 1f at the edge.

• “Seconds to wait before stopping”

When you release the movement input, there is a very small delay before the cursor actually stops. You can change this value if you want to, but *I would advise leaving it at 0.01f.*



These events are shown when “Mouse Simulation Events” is folded out.

• Events

These events trigger when their respective action happens. “OnStartedMoving” and “WhileMoving” pass the thumb stick’s direction as a Vector2. These are not necessary to use. They simply exist, should you need them.

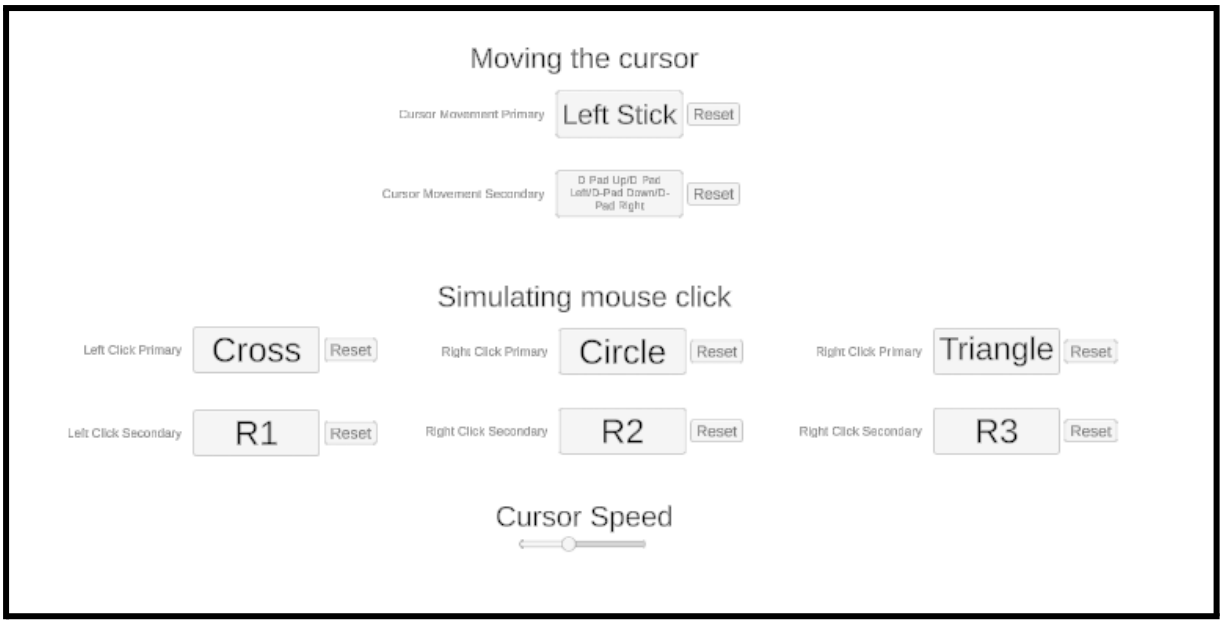
Rebinding

• TL;DR

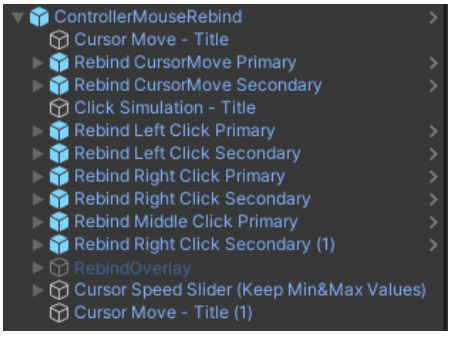
You don't need to know what any of these prefabs do. Just drag the 

prefab into your scene, delete the things you don't want and modify the visuals to match your aesthetic. It should work when you press play.

If you don't like the default bindings, change them in the input settings.

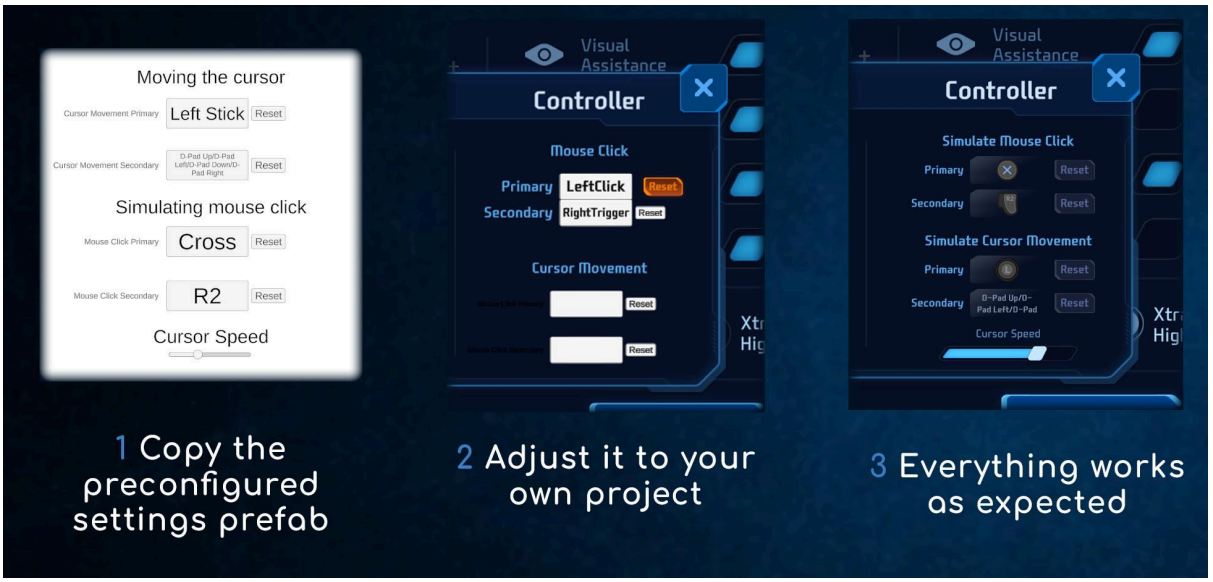


To add rebinding to your game, go to [*a1creator* > *MouseAsController* > ***Prefabs***] and drag "ControllerMouseRebind" into your scene's UI canvas.



Test if it works by pressing play and fiddling with the settings. Also end play session and start again to see if the rebinds saved. Adjust and change the UI elements to fit your game's aesthetic.

Here's an example from my own game:

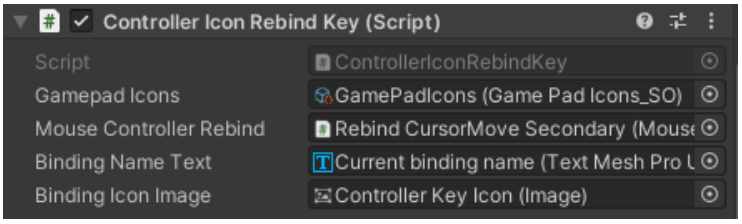


• Understanding the rebinding components

You really don't need to know this in most cases. Try using the prefabs before wasting time reading this.

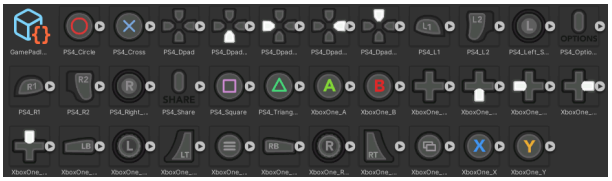
Note: Every time I say "inputfield" below, I mean a button with a text component which edits itself according to the selected binding.

The "Rebind Key" prefab contains everything needed to rebind one single key via UI. A title, an input field and a reset button.



All of the references in this component should already be correct when creating the gameobject with the prefab.

• “GamepadIcons”



Keep the reference. The only reason this is in a ScriptableObjects is to not accidentally lose the many references in it when adding/removing components. You can freely click on the ScriptableObject itself and replace the images with your own.

• “Mouse Controller Rebind”

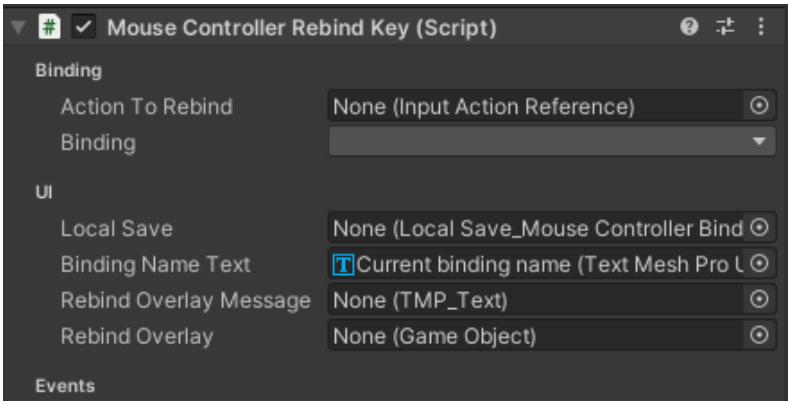
Keep the reference. It might aswell have been a case of a hidden RequireComponent-reference. Just leave it be. If it’s empty, drag the prefab into this reference field.

• “Binding Name Text”

Keep the reference. It is the text of the inputfield that is clicked when rebinding this key.

• “Binding Icon Image”

Keep the reference. It is the image to show instead of the binding name when an icon exists for the binding.



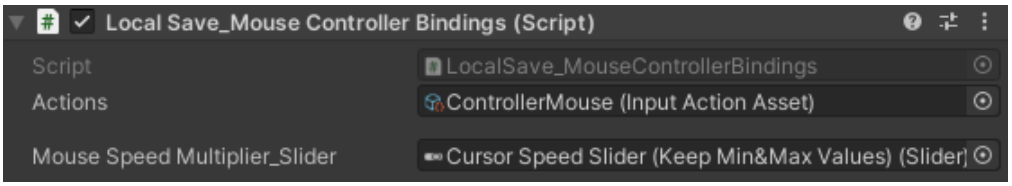
• “Action To Rebind”

The input action you want to rebind with this UI element.

• “Binding”

Choose which of the bindings for the action you want to rebind. For example, a primary and secondary rebind element will have the same “Action To Rebind” but different “Binding”s.

• “Local Save”



Each “MouseControllerRebindKey” must have one of these components as a reference to save the selected rebinds. If you are using the “ControllerMouseRebind” prefab, this reference should already be set and if not it’s placed on the root object. If you are applying the components yourself, just use one “LocalSave_MouseControllerBindings” component and reference the same one in all your “MouseControllerRebindKey”s.

• “Binding Name Text”

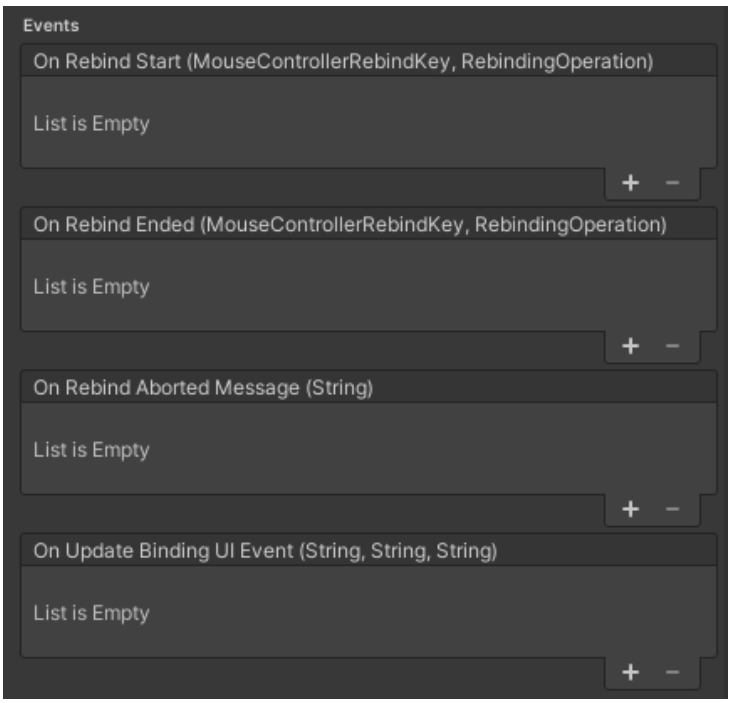
The name of the keybind in the inputfield. The reason there is a reference to this in two components is because one edits it and one disables/activates it.

• “Rebind Overlay Message”

The text element in your rebind overlay that will display a message to the user.

• “Rebind Overlay”

The UI element that covers the screen when the user is changing a keybind.



These events are shown when “Events” is folded out.

• Events

These events trigger when their respective action happens. These are not necessary to use. They simply exist, should you need them.

Event parameters:

On Rebind Start

- The “MouseControllerRebindKey” in use | The current RebindingOperation

On Rebind Ended

- The “MouseControllerRebindKey” in use | The current RebindingOperation

On Rebind Aborted Message

- Abort message

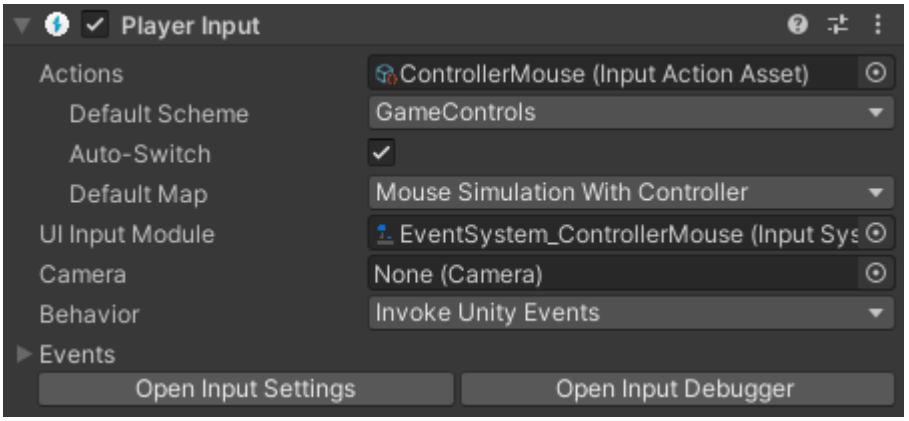
On Update Binding UI Event

- Name of binding | Device layout name | Binding path

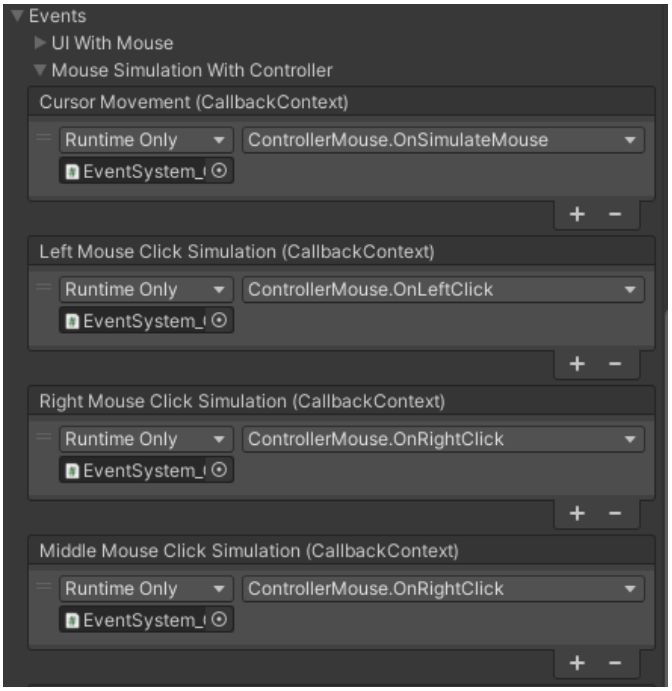
Disabling

All default bindings are enabled by default because disabling is simpler than activating. To disable, simply:

1: Go into the event system prefab and find the Player Input component.



2: Click to open “Events”.



3: To disable a click event, simply click the minus to remove the event listener. If you regret this, you can reset the prefab to undo.