Table of Contents

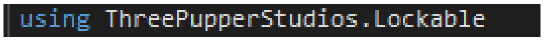
Lockable Properties

Lockable Properties is a simple Editor tool to help protect your work in the final stages. This simple tool can ease your worries about accidentally making changes to finished Properties.

Designed to be as simple as possible, Lockable Properties allows you to set specific variables to lock down from future edits. Simply select the variables you need to finalize in the inspector and lock them to prevent accidental edits in the future.

Installation Steps

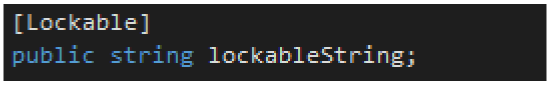
1. Simply include "ThreePupperStudios.Lockable" at the top of the scripts that you would like to have lockable properties.



```
using ThreePupperStudios.Lockable
```

Figure 1: Editor Tool in your script

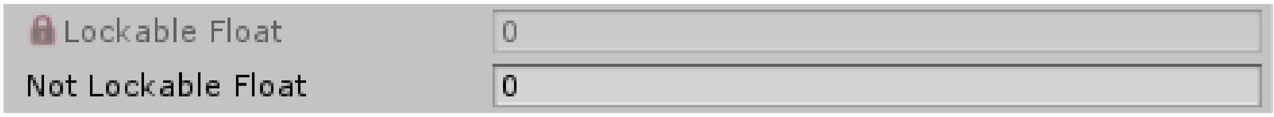
2. Add the [Lockable] Attribute to the top of your property in your script.



```
[Lockable]
public string lockableString;
```

Figure 2: Lockable Attribute in your script

Doing so will make your property appear disabled by default in the unity inspector:



🔒 Lockable Float	0
Not Lockable Float	0

Figure 3: Unity Inspector showing disabled property

Note: while in this state, you will not be able to edit the Property fields.

In the Unity Inspector, locked Properties will display a locked icon beside them.

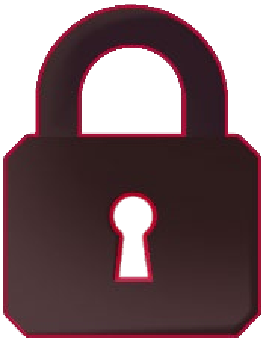


Figure 4: Locked Icon

Unlocking a Locked Property

If you locked a Property by accident, or need to edit the Property fields, you can use the Inspector to quickly Unlock your Property.

1. Right click on the name of the property in the inspector to open the Lock/Unlock menu.

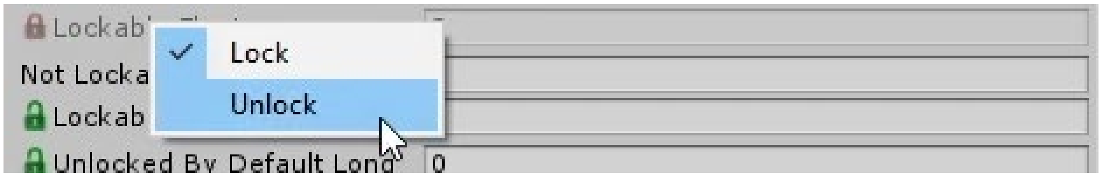
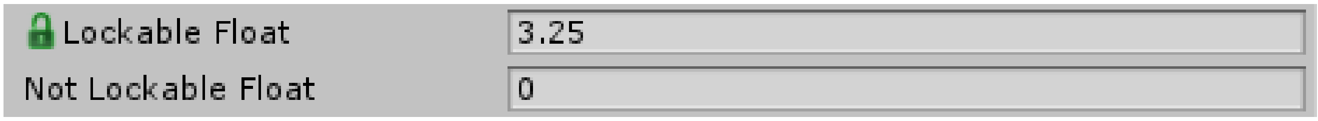


Figure 5: Lock/Unlock Menu in the Inspector

Selecting "Unlock" will cause the field to no longer be disabled and you will be able to edit it again.



Lockable Float 3.25

Not Lockable Float 0

Figure 6: Editable fields in the Inspector

With this tool enabled, Unlocked Properties will display an Unlocked Icon.

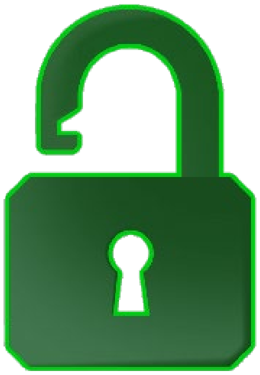


Figure 7: Unlocked Icon

To relock a Property when you're done making changes, go ahead and right click on the Property name again, select "Lock" and you're done!

Additional Bools

The Lockable Attribute offers two additional bools that can be included to add optional functionality, “locked” and “rememberSelection”. Both are set to true by default.

Locked

This bool allows you to set the default state of a locked property to locked, rather than unlocked. When this bool is included, all Properties will by default be locked. If changed to false for individual Properties, then the Property will not be initially locked.

1. To include and set the “locked” bool, add one of these options to the attribute:
    a. [Lockable(true)]
    b. [Lockable(locked: true)]
    c. [Lockable(locked = true)]

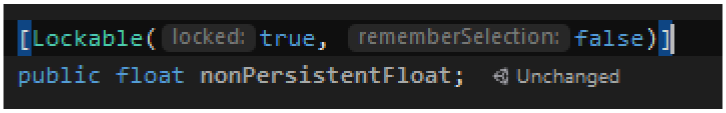
Remember Selection

This bool allows set states to persist beyond changing selections, code recomplilіng, or scene changes. If set to false, the locked state of the property in Unity will default to the value passed to the “locked” bool. If both “rememberSelection” and “locked” are set to false, whenever you change your selection, the field will unlock, but if “locked” is set to true and “rememberSelection” is set to false, it will automatically lock itself.

1. To include and set the “rememberSelection” bool, add one of these options to the attribute:
    a. [Lockable((true or false), false)]
    b. [Lockable(rememberSelection: true)]
    c. [Lockable(rememberSelection = true)]

“rememberSelection” is by default set to true. This means that the value you set will persist in almost every case excluding creating a prefab.

An example of the bools can be seen below. A full example is also available in the examples folder.



```
[Lockable( locked: true,  rememberSelection: false)]
public float nonPersistentFloat;  Unchanged
```

Figure 8: Additional Bools settings in your script