



## 5. Save

### Table of Content

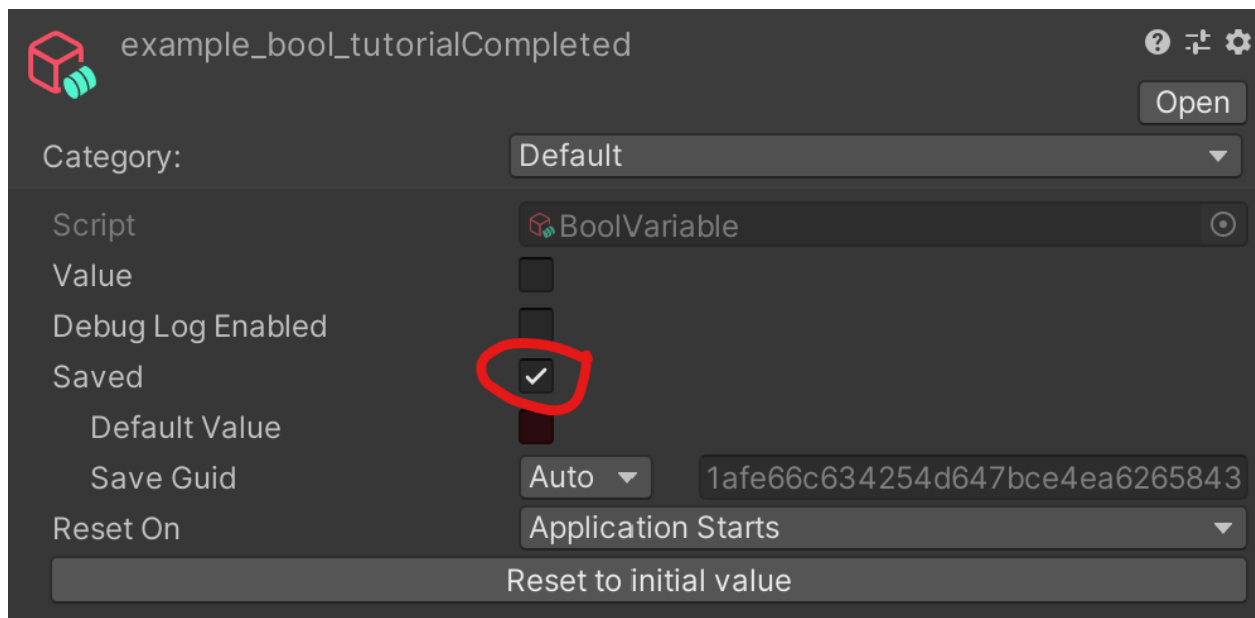
Table of Content.....	1
Single click save.....	2
Clearing the save.....	4
Exercise.....	4

## Single click save

Let's start by testing the scene. If you enter play mode, and press the button "**End Tutorial**", you will see that the **Tutorial** object will disable itself.

Now exit play mode and play again. You can see that the changes have persisted. It's because the **example\_bool\_tutorialCompleted** variable state has been saved to Player Prefs. To enable saving on your variable, select the variable and enable the property "**Saved**". All scriptable variables available in the package can be saved to Player Prefs **with a single click**. (Except the `GameObjectVariable`).

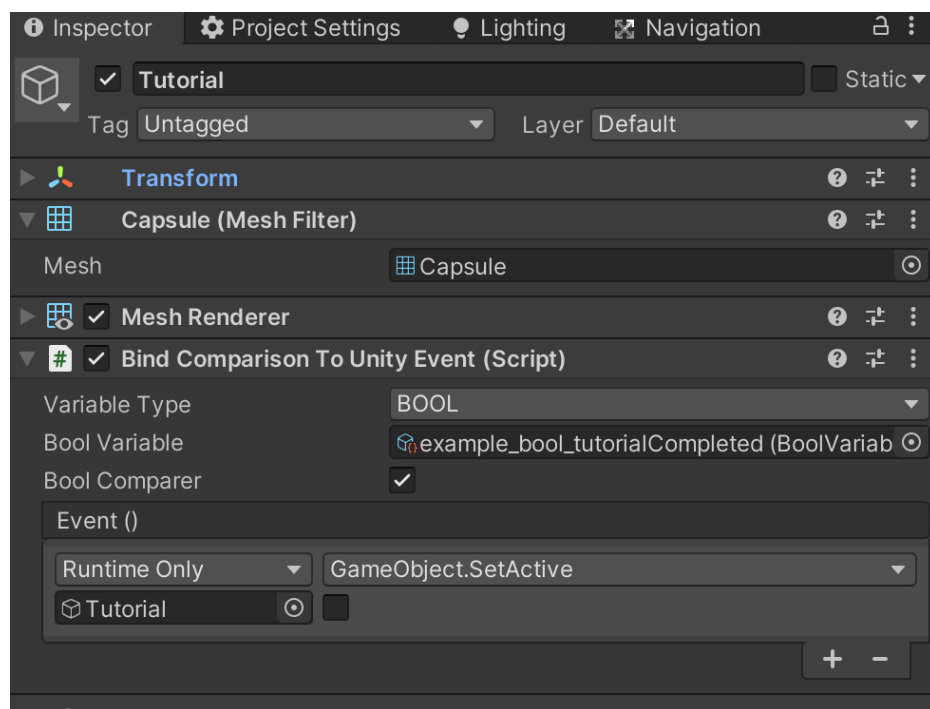
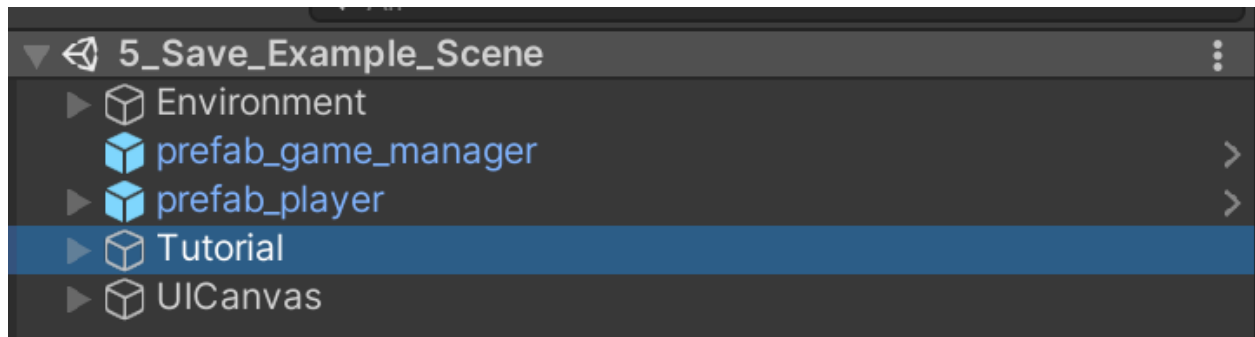
When loaded for the first time, the variable will be set to the **Default Value**.



The key used for Player Prefs is the Save Guid of the ScriptableVariable. This Guid is generated using the path of the asset. If Auto is selected, the Guid is generated automatically. If you change it to Manual, you can override the Guid.

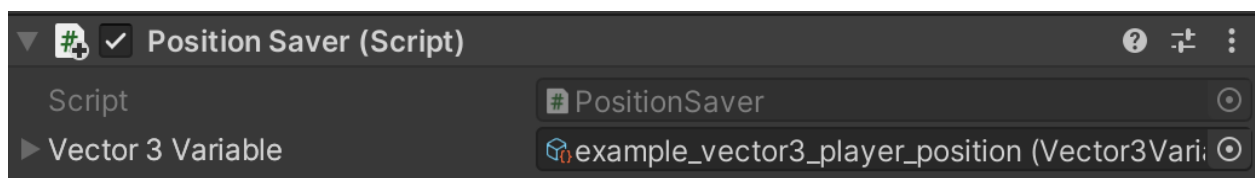
You can use the fact that a variable has been saved in combination with a Binding script. If you select the **Tutorial** in the scene, you can see that its

component `BindComparisonToUnityEvent` will disable the object if the condition is true.



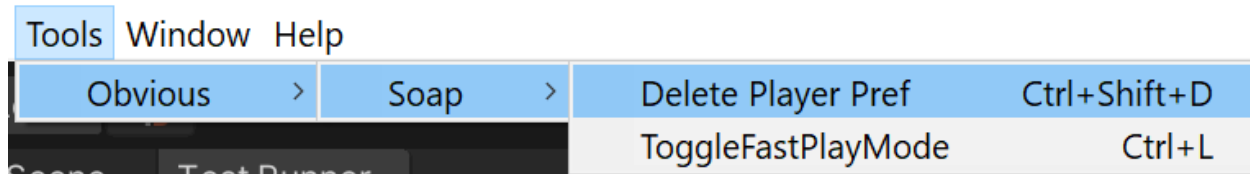
Saving is useful for things like level index, player score, coins, tutorial completed, etc.

The player position is also saved, check the **PositionSaver.cs** on the **prefab\_player** in the scene.



## Clearing the save

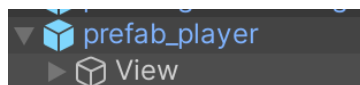
If you want to **clear the save** (deleting all Player Prefs), use the shortcut (CTRL+Shift+D) or the menu item:



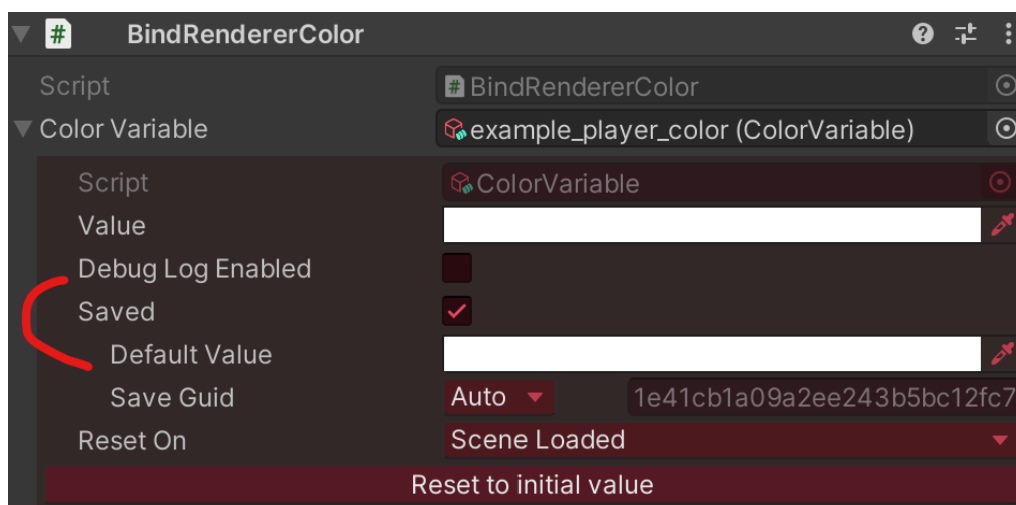
## Exercise

You can set a random color to the player by clicking on the “Random Color” button on the bottom of the screen at runtime. Currently, the color is not saved. **Can you find how to save the player color?**

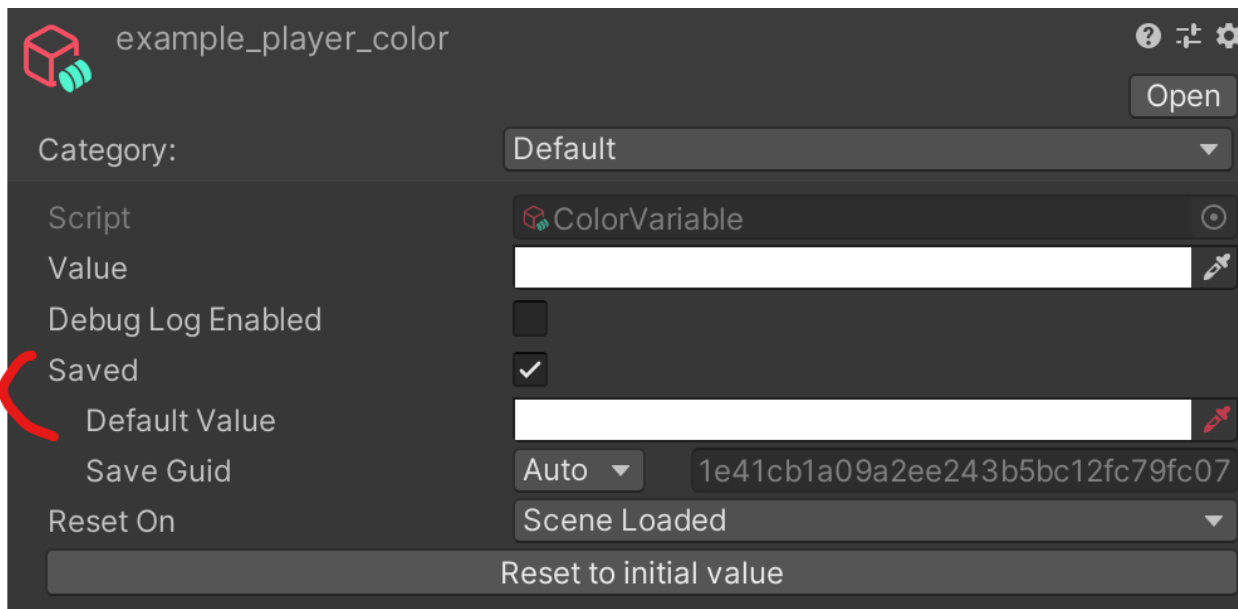
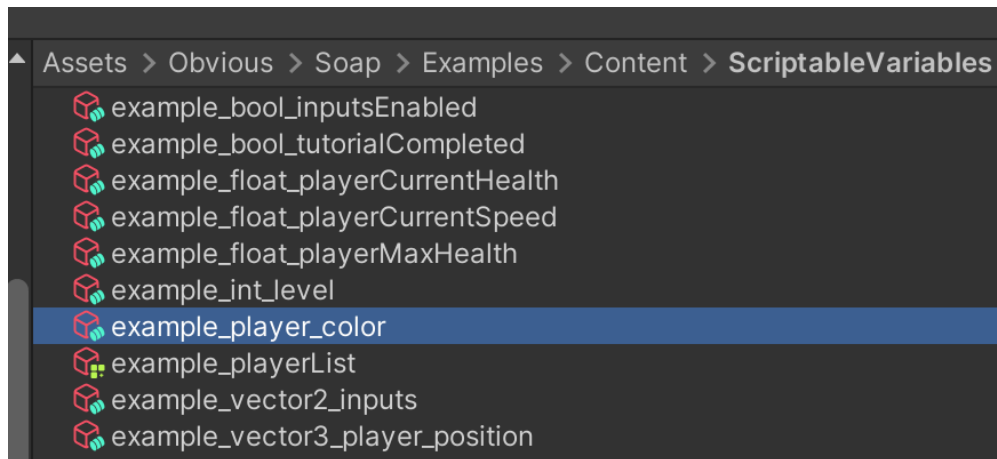
**Solution 1:** find the **BindRendererColor.cs** component on the prefab\_player ->View.



Then, expand the Scriptable Variable, enable the “Saved” property and set the default color to white.



**Solution 2:** find the scriptable variable **example\_player\_color** and enable the “Saved” property.



**Note:** when you create a new scriptable variable type, you can override the Save and Load method, and have custom save/load for them using Player Prefs 😊. Check the ColorVariable or Vector2Variable to have a concrete example.