

Save Is Easy - Readme.pdf

How to use it

In a GameObject of the scene add the component **"SavelsEasyManager"** and in this select the Prefabs that you want that can be loaded in the game by the asset.

Add to your GameObjects or Prefabs the component **"SavelsEasyComponent"** and in the list select the fields that you want to save and load.

Important

- The main class of the Asset is **"SavelsEasyAPI"**, Examples: `SaveIsEasyAPI.SaveAll()`, `SaveIsEasyAPI.LoadAll()`.
- You need to have one **"SavelsEasyManager"** in a GameObject in the scene you want to load or save.
- You can save the level in thread mode or not, this will change the performance of the game, in thread mode the script from save run in a separate thread and will not affect the FPS but will last longer, if it is not threaded mode this will significantly impact in the FPS but will be all more quickly.
- The asset supports the use of multiple scenes loaded at the same time but if you do not want to use this function, the API is designed to not have to specify the scene, but you can not have more than one scene loaded at the same time otherwise the asset is going to throw an error.

Warning

If you use the **"On Exit Save"** option, avoid use the unity function:

```
public static void Destroy(Object obj, float t);
```

with time, in an object that you want to be save. The consequence is that when you close the game this will remove your objects before it can be saved.

Multi-Scene support

This allows you to use the asset in situations where there are multiple scenes loaded at the same time and you can handle them independently, load or save them, without affecting the other scenes, this is useful for more complex saves.

Most API functions allow an argument where you can specify the desired scene that will execute the specified action. An example would be with the function: `LoadAll()` or how would it be to use specifying a scene `LoadAll(Scene scene)`.

Each scene needs to have its own **"SavelsEasyManager"**, and you have to put different FileName in each to avoid that the saving of one scene overwrites the saving of another.

Events

- In the **"SavelsEasyComponent"** you can choose the **"Call Start On Load"** option, this causes that when a game is loaded the asset has to call Start() of all the scripts of this GameObject, this is slow, if it is not used it is recommended to turn off .

- To use the events **OnLoad** and **OnSave** you have to implement the “**ISavelsEasyEvents**” interface in your code, The events are called regardless of whether the component is enabled or not.
 - **OnLoad**: It is called after all the new values are set to that script, it is necessary that at least one value is loaded in that script.
 - **OnSave**: it is called before the information is loaded into the fields and if there are fields to load in that script.
- In the “**SavelsEasyComponent**” you have the option of **On Load** that has a list with the following options:
 - **CallStartAfterLoading**: This will first load the `GameObject` information and then call the `Start()`.
 - **CallStartBeforeLoading**: This will cause the `Start()` to be called first and then the `GameObject` information will be loaded.
 - **CallStartBeforeAndAfterLoading**: This will cause the `Start()` to be called before and after the information is loaded.
 - **Off**: The asset is not going to call `Start()`, `Call Start()` is slow, if it is not used it is recommended to put on Off.

Attributes

You can use attributes in your fields to use some extra features.

- **AlwaysSaveAttribute**: To force the save of this variable, it can not be turned off from the editor, It will always be saved.
- **AvoidSavingAttribute**: If you want this field cannot be saved, it does not appear in the editor.
- **QuickAccessAttribute**: So the field is available with quick access.

QuickAccess

This allows you to save certain attributes separately from the game, so that they are available without having to load the entire file. It is used for the menu that lists the saved games.

You can save all kinds of information and this is available in the menu instantly as it is not required to load all the information in the file.

The types of data that Quick Access supports at the moment are: `bool`, `char`, `double`, `float`, `int`, `long`, `short`, `sbyte`, `uint`, `ulong`, `ushort`, `string`, `byte`, `byte[]`.

SavelsEasyStatistics

The asset comes with a component called “**SavelsEasyStatistics**” designed to store statistics of the game and using quick access that information is available in the menu. This component will save:

- The total time he spent playing at this save, in seconds.
- The amount of times the game was load.
- The time in which the game was created
- A screenshot when the game is saved.

File Viewer

Is a powerful tool that is found at "**Windows -> Save is easy -> File Viewer**" in the unity menu, that allows you to explore you saved files.

This can be useful to find problems, You can know if the problem is with the saved or with the load of your game. You can also use to optimize your saved, you can search for fields that are not used but are being saved and deactivated to gain speed and for the file to weigh less.

Adapters

Some Unity components are very complicated for SavelsEasy to save them and load them correctly, so the asset has adapters, it allows you to save more complicated components, such as animations, it is designed so that the user can create adapters according to their needs.

You can find the adapters in the folder: "**\Assets\Save is easy\Adapters**"

Contact

For any questions, problems or recommendations you can contact me by email in English or Spanish:
zedgeincorporation@gmail.com