UNITY SAVE LOAD MANAGER

Package name : Unity Save Load Manager
Version : 1.0
Author : Cachwir
Licence : GNU General Public License

Description

This package contains a basic manager able to handle your game's save and load mechanisms.

It includes the following :
- the manager (SaveLoadManager)
- a SceneSaveLoadManager which will handle your current scene save and load
- an OnStartCallbackCaller whose purpose is to make sure your loaded entities' data will be applied after their Start function has been called
- an AbstractSaveLoadManager which you will make the parent of your custom save load managers.
- an IGameLoadListener interface you will need to attach to each manager you have that needs to terminate its actions before the scene unloading (cinematics, dialogs, ...)
- an IPersistant interface you will need to attach to each object persisted through scenes (ie which calls DontDestroyOnLoad() at Start)
- a SaveLoadManagerHelper which contains a few methods you could use for other purposes (especially the ones that use the LocalIdInFile)

What this package can do :
- Save any gameObject you added manually to the hierarchy (as long as you created a custom SaveLoadManager for each type of them).

What this package cannot do :
- Save any gameObject created on the fly (such as shot arrows in the middle of their move).

BUT :
You can still save them if you make a reference to them on a gameObject you added manually to the hierarchy (such as the bow that fired them).

Installation

1. Add the SaveLoadManager to your hierarchy
2. Add the IGameLoadListener interface to each and every manager that needs to terminate its actions on scene unload
3. Add the IPersistent interface to each and every manager that calls DontDestroyOnLoad()
4. In Unity : Edit -> Project Settings -> Script Execution Order. Click the "+" to add the OnStartCallbackCaller to the list and give it one of the lowest numbers so its methods are called before other class' (I put it at -99)
4. Link the SaveLoadManager's Save and Load methods to buttons created to this effect
5. Create any custom SaveLoadManager to save whatever you want (your player's data, camera's data, quests, the position and actions of the current scene objects, etc)
6. Enjoy~

How to create a custom save load manager

0. Choose something you want to save and load. Let's say, your player's position and rotation (handled by your PlayerController class)
1. Create a new C# Script (say, PlayerSaveLoadManager)
2. Make it a child of the AbstractSaveLoadManager class
3. Add the following methods (replace PlayerController by your entity manager) :

```
protected override void Start()
{
	base.Start();
	entityManager = FindObjectOfType<PlayerController>();
}

public override ObjectData GetObjectData(MonoBehaviour entityManager)
{
	return new PlayerData((PlayerController) entityManager);
}

public override void RebuildFromData(ObjectData objectData)
{
	PlayerData playerData = (PlayerData) objectData;
	PlayerController playerController = (PlayerController) entityManager;

	// calls the delegated function after the scene was loaded
	CallAfterSceneLoaded(delegate ()
	{
		// call this to execute after the entity's Start function has been called (except for
IPersistent entities)
		PrepareDataForEntity(playerController, delegate ()
		{
			// playerController's Start has been executed, so it's time to update its data
			playerController.transform.position = new Vector3(
				playerData.position[0],
				playerData.position[1],
				playerData.position[2]
				);
			playerController.transform.rotation = new Quaternion(
				playerData.rotation[0],
				playerData.rotation[1],
				playerData.rotation[2],
				playerData.rotation[3]
				);

			// Your PlayerController has been updated !
		});
	});
}
```

```
[Serializable]
public class PlayerData : ObjectData
{
	public float[] position = new float[3];
	public float[] rotation = new float[4];

	public PlayerData(PlayerController playerController)
	{
		position[0] = playerController.transform.position.x;
		position[1] = playerController.transform.position.y;
		position[2] = playerController.transform.position.z;

		rotation[0] = playerController.transform.rotation.x;
		rotation[1] = playerController.transform.rotation.y;
		rotation[2] = playerController.transform.rotation.z;
		rotation[3] = playerController.transform.rotation.w;
	}
}
```

4. Locate the CustomManagers gameObject. It's a child of the SaveLoadManager. Create an empty gameObject as its child and add your newly created script to it.

5. In your newly added gameObject, change the File Name's value for anything you want. It needs to be unique so be careful. Here, you could choose : player.sav

5. Congratulations ! Your PlayerController's position and rotation are now correctly saved and loaded !

6. Info : The SaveLoadManager will check all children of the CustomManagers gameObject and treat them as save load managers. Keep in mind that your custom save load managers' code will be execute in the order you put them as Custom Managers' children.

FAQ

1. After loading, some of my persisted objects are disabled.
Answer : I'd guess you forgot to add the IPersistent interface to them

2. After loading, my SaveLoadManager is destroyed.
Answer : Pretty sure you put the SaveLoadManager as a child of a non persisted gameObject

3. I made a custom save load manager but when I save and load, my target entity's data are not loaded as if nothing changed.
Answer : Don't forget to add your custom save load manager to the list of the CustomManagers' children. If you've already done that, make sure you put the code that updates your entity's data as a delegate of the PrepareDataForEntity method, and to the CallAfterSceneLoaded method (read the example up there to see how it's done).

4. Can I use the SaveLoadManager to make a better version of it and then distribute it ?
Answer : Sure. I'd appreciate if you contacted me about this once you've done it, though (send me a message on twitter : @Cachwir).

5. Can I use the SaveLoadManager on professional projects ?
Answer : Sure. It's meant to be used so go ahead.

6. Are you single ?
Answer : That's a secret.

7. My question is not there, how can I contact you ?
Answer : You can contact me on twitter. Just send a message to @Cachwir.