Thank you for purchasing a copy of Persistent Data!

I can be contacted any time at solclovser@gmail.com and on my Discord server.

Please leave a review to help other potential users :-)

Table Of Contents

Setup Instructions

1) Drag and drop ‘Persistent Data’’ prefab to your starting scene. It is located under Sol Clovser > Persistent Data > Prefabs
2) Add your first module in the inspector.

Usage

You can check out the DemoTest script in the Demo > Scripts folder or on the Demo Test object in the hierarchy for line by line usage.

Start by adding the following line to the start of your script to start using it.

```
using SolClovser.PersistentData;
```

⏩ How to load a save?

If load on awake is enabled, the latest save file with matching save file name will be loaded.

Alternatively to load a save by code we use;

```
PersistentData.Instance.Load();
```

The method takes in one `Save` object.

You can get this `Save` object by using;

```
PersistentData.Instance.GetLatestSaveInfo();
```

So, to load the latest save with the matching save file name, the code becomes;

```
PersistentData.Instance.Load(PersistentData.Instance.GetLatestSaveInfo());
```

⏩ How to change save file name?

You can change the save file name from inspector and by accessing

```
PersistentData.Instance.SaveFileName
```

⏩ How to get all saves?

```
SavesInformation savesInfo =
PersistentData.Instance.GetAllSavesInfo();
```

This method returns a list with all the saves. All members is type of `Save`

`Save ` includes

```
string saveName
public string saveDate;
```

And a method that returns human readable time in `DateTime` format.

```
DateTime GetHumanReadableTime()
```

⏩ How to save?

It is as easy as calling;

```
PersistentData.Instance.Save();
```

⏩ How to delete a save?

It is as easy as calling;

```
PersistentData.Instance.DeleteSave(theSave);
```

You need to pass a save to this method to delete it from the disk.

⏩ How to access the modules?

First start by creating a local variable of the same type as the module you want to access. In this example we are going to access to `DemoModuleOne`

```
private DemoModuleOne _demoModuleOne;
```

Then in the start of our script;

```
_demoModuleOne =
PersistentData.Instance.GetModule(typeof(DemoModuleOne)) as
DemoModuleOne;
```

Now we can access our properties and fields by typing;

`_demoModuleOne.ValueOne`

⏩ How to load only one module but not the other?

We can use

```
PersistentData.Instance.LoadAndOverwriteModuleFromDisk()
```

method for this task.

It takes in 3 parameters;

`string saveName`
The save name this module was saved under.

`string saveDate`
The save date we know from the `Save` object

`PersistentDataModule moduleToOverwrite`
And the module itself we want new data from disk to applied to

As an example, this is a method that only loads the single module we want to;

```
public void LoadOnlyModule1()
{
    PersistentData.Instance.LoadAndOverwriteModuleFromDisk(
      theSave.saveName,
      theSave.saveDate,
PersistentData.Instance.GetModule(typeof(DemoModuleOne))
    );
}
```