Scriptable Object Data Browser

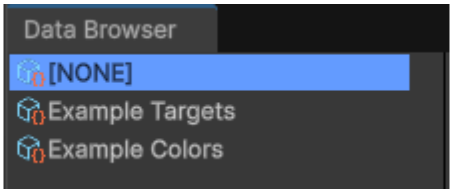
Overview

The Scriptable Object Data Browser allows you to organize and edit the properties of all your Scriptable Objects in one place, similar to how one might use a spreadsheet to edit static game data more traditionally. It also allows you to follow references to other Scriptable Objects referenced by an object that's being edited, so you can edit multiple relevant data objects all at once.

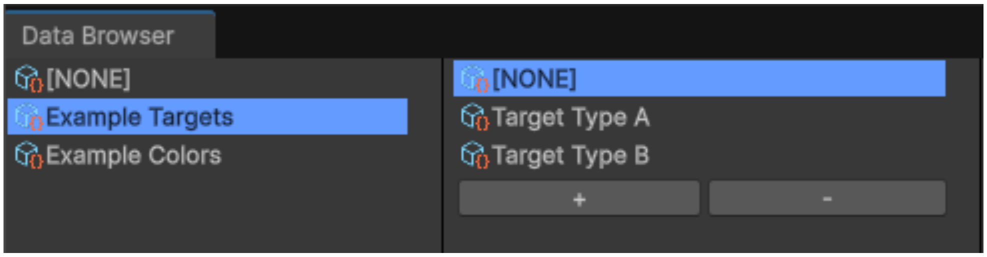
I've personally found it's sped up interacting with static game data quite a bit, and allows me to stay more focused on what I'm working on rather than clicking around all over my project trying to find where some data is located. Hope you enjoy! See my contact information at the bottom of this file if you have any questions!

Usage

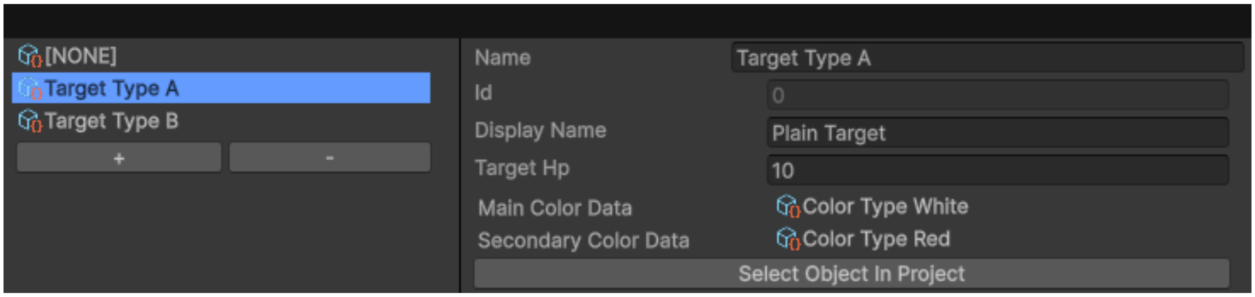
- The Data Browser can be accessed either via `Tools->Data Browser`, or by pressing `Alt+D`
- The list of tracked Scriptable Object types are listed in the left-most column:



- By clicking on a type, you will see a list of all objects of that type that exist in the project appear in a new tab to the right:

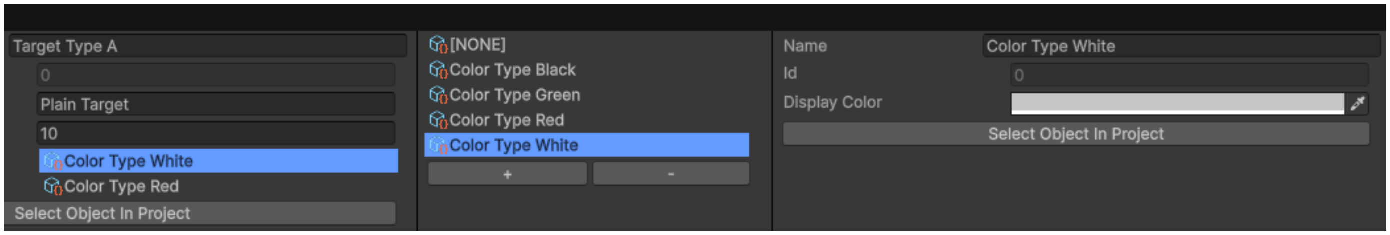


- Click on one of the objects to see its properties displayed similar to in Unity's Inspector, in a new tab:



Name	Target Type A
Id	0
Display Name	Plain Target
Target Hp	10
Main Color Data	Color Type White
Secondary Color Data	Color Type Red

- Here you can edit the properties of the object similar to how you would in the inspector. However, the main difference is that references to OTHER scriptable objects that are tracked by the browser will appear in a unique manner. By clicking on these references you will open a new tab in the browser, allowing you to inspect the fields of a referenced data object simultaneously with the original object:



Target Type A
0
Plain Target
10
Color Type White
Color Type Red
Select Object In Project

[NONE]
Color Type Black
Color Type Green
Color Type Red
Color Type White
+
-

Name: Color Type White
Id: 0
Display Color
Select Object In Project

- If you select another object within this list of objects of the type referenced by the selected field, you will change which object is referenced by that field, so keep that in mind. If you're going to be editing several objects of a given type, it is better to return to the first tab of the Data Browser and select that type there.
- You can always press `Select Object In Project` to highlight the Scriptable Object in your project.

Setup

Out of the box the Data Browser will show a set of example data. To get it to show your Scriptable Objects you'll want to follow these Setup steps:

1. Create your own class that inherits from `DataBrowserSupportedTypesContainer` (see `ExampleSupportedTypesContainer.cs` for an example - you can likely copy the contents of this file and just change its name).
   - You'll also want to change the `fileName` and `menuName` in the `CreateAssetMenu` attribute at the top of the file, so you can add an instance of this class to your project.
2. Within the method `GetSupportedDataBrowserTypes`, add the types you want to be displayed in the Data Browser to `supportedTypes`. The keys in the dictionary should be what you want the category for that data type to be called in the Browser.
   - You could get a bit more fancy here and do something like search a directory in your project for the types to add, if you wish. Some example code is commented out in the example file that could be used to do this.
3. Back in your project window, right click in the folder you'd like your Supported Types Container object to be located, and go to `Create->[menuName]->[fileName]` using the values you set in your Supported Types Container file.
   - Note that the object you created has an `Active` property that can be unchecked. This can be used to turn off and on what sets of Scriptable Objects are used by the Data Browser. For instance, if you'd like to hide the example data in your Data Browser (which you probably do) search your project for `ExampleSupportedTypesContainer` and uncheck its `Active` box.
4. Upon pressing 'Alt+D', you will now see it populated with your Scriptable Objects!

    - Keep in mind that you will have to add at least 1 object of a type you want tracked to your project before you will be able to add additional ones from within the Data Browser (because it needs an existing object to determine where to place new ones).

5. To speed up the performance of the Data Browser when using it to add or remove objects, edit `DEFAULT_DATAOBJECT_FOLDER` inside `StaticDataHelpers` to more specifically point to the parent folder under which all your Scriptable Objects that you want tracked by the Browser can be found.

Optional Additional Tools

- It's not required for using the Data Browser, but this package also includes a system I use to create a database of my Scriptable Objects that allows me to reference them by a generated integer ID, and do things like iterate over all objects of a given type. If you'd like to use this system, the remaining files within the `Example` folder demonstrate how!

Known Quirks

- Make sure to press 'Enter' or 'Return' after changing the `Name` of a Scriptable Object within the Browser to make sure the value is saved before clicking away to another object.

Dependencies

- Many of my tools are dependent on the NaughtyAttributes Unity Asset, so you will likely have to import it into your project: https://assetstore.unity.com/packages/tools/utilities/naughtyattributes-129996

Contact

- My support email: contact@fletchersgames.com
- My support website: http://www.fletchersgames.com/