Editor Native Database

Version 1.0

Introduction

Editor Native Database allows you to create and manage your database directly in Unity editor. It stores every database entry as a scriptable object rather than using a single file like traditional databases do. There are several reasons for this. There is a lot of ways to use scriptable objects in a project (to learn more about it, you can watch talks about scriptable objects from Unite). You can easily find, reference and use them. At runtime you don’t need any scripts to get your database entries. You don’t need enums or strings to identify them. You are much less likely to lose your entire database due to hard drive error or misediting. You can import any scriptable object into your database as a new entry. You can easily copy entries from one database to another.

Requirements

This asset’s scripting runtime version is .NET 4.6. If your project uses older one, open Unity Editor Player settings (Edit -> Project Settings, then select the Player category), and in the Configuration section set scripting runtime version to .NET 4.x.

Database Editor Window

The database editor window is divided into 3 subwindows. The left one shows you the database container tree. The middle one - entries from currently selected container. The right subwindow shows you either UI elements for editing your database if you have a container selected, or entry’s content if one is selected.

Options

On the top of the window you can see options panel with 3 foldouts. “Database options” foldout has following options:

“Current database” : currently open database.

"Show Containers As Single Column" : when checked containers will be shown not as tree but as a single column. Every new container will be created in database root and those not in database root will be hidden.

“Left window width” : controls width of the left window. Hidden when "Show Containers As Single Column" option is checked.

“Right window width” : controls width of the right window.

"Select Container with Parents" : when checked, selected container will have its parents selected as well. This is only a visual aid and does not affect how database works. Hidden when "Show Containers As Single Column" option is checked.

"Hide Search Text Field" : hides search text field in the lower left corner.

“Container Button Options” foldout has following options:

"Container Name Always Fits Button" : when checked every container button will be wide enough to fit the name.

"Container Button Width" : controls width of container buttons.

"Container Button Spacing" : controls spacing of container buttons.

“Font size” : controls container name font size.

"Normal Font Style" : container name font style when it’s not selected.

"Selected Font Style" : container name font style when it’s selected.

“Normal Button Color” : container button color when it’s not selected.

“Selected Button Color” : container button color when it’s selected.

“Normal Font Color” : container name font color when it’s not selected.

“Selected Font Color” : container name font color when it’s selected.

“Entry Button Options” foldout has following options:

“Font size” : controls entry name font size.

"Normal Font Style" : entry name font style when it’s not selected.

"Selected Font Style" : entry name font style when it’s selected.

“Normal Button Color” : entry button color when it’s not selected.

“Selected Button Color” : entry button color when it’s selected.

“Normal Font Color” : entry name font color when it’s not selected.

“Selected Font Color” : entry name font color when it’s selected.

The Workflow

Creating New Database

To create a new database right click in your project window in folder where you want to store it and select “Create -> Scriptable Object -> Database”. Database containers and entries are stored in a folder that has the same name as the database. If you want to move a database from one folder to another, you have to move database asset file together with that folder.

To open (or close if it’s open) the database editor window, in the top-level toolbar click “Tools -> Database -> Open database window”. You can also use “shift + control + d” hotkey to do it.

Creating New Container

Containers are scriptable objects that serve the same purpose as folders in a file system. In database tree a container can hold either containers or entries but not both.

When you have database root or a container selected, in the right window there will be “new name” text field and “create new container” button. Every container and entry name in database is unique. You won’t be able to use a name more than once. Checking if a name is already used is not case sensitive, so if you have a container or an entry named “Player”, you won’t be able to create another one using name “player”. If you create a new container when you have another selected, new container will be a child of the selected one and will be shown directly below its parent in the database tree. A container can have any number of children.

Creating New Entry

Before adding new entry type to your database, you have to create a new script that uses “ENDB” namespace and contains class that inherits from “DatabaseEntry”. You can use “GenericItemExample.sc” as a template.

To create a new entry, select a container, in the right window type in a new name, select an entry type and click “Add new entry” button. A container can hold entries only of one type. First entry in container determines the type. Other entries will be of that type or inherit from it.

Importing Entries

You can import any scriptable object into your database as long as it its class inherits from “DatabaseEntry”. Use “Import single entry” button in the right window to import an entry into selected container, and “Import every entry of type” button to import every entry of a certain type.

Finding Entry by Name

Use the text field in lower left corner to find an entry by its name. Current container and its children will be searched, if you have one selected, if not (or if “Show containers as single column” option is checked), entire database will be searched. When you start typing in the search text field, entry names that match the typed name will be shown in the middle window. To filter search results by type, select one from “Filter by type” dropdown menu. To stop searching and clear contents of the middle window, select a container.

Moving Entries From One Container to Another

To move entries from one container to another select a source container, click “Move entries from this container” button in the right window, select destination container and click “Move entries from “*source container name*” into this container” button. Entry type from destination and source containers must match If destination container already has entries. If you want to copy entries from one database to another, you should use import function.

Miscellaneous

Preview Icon

When viewing game items in database you usually want to see what they look like. To show a preview icon for a texture or a sprite, you can use “DatabaseAssetPreview” attribute. This attribute has two integer parameters. First one is preview icon width. Second one is preview icon height. Open “GenericItemExample.sc” to see how to use it. To create preview icon for a 3D model you can use “Preview Generator” asset from the Asset Store.

Customizing Entry Inspector

If you want to customize how an entry type is shown in database, you can use “DatabaseEditorWindowCustom” script. That file will not change in future updates.

Avoiding Recompiling Scripts From This Asset

To avoid recompiling scripts from this asset every time you edit other scripts in your project, you can move this asset to “Plugins” folder that should be created in “Assets” folder.

Examples

In “Examples” folder you will find an event system based on scriptable objects. Such an event system makes project more modular, which makes writing and debugging code easier. When you select an event entry in database, the right window shows you prefabs and game objects in active scene that raise and listen to the event. There also will be button to raise selected event if editor is in play mode. If you delete this folder, you will have to fix compilation errors in “DatabaseEditorWindowCustom.sc” and “DatabaseEditorWindow.cs”.

Contact Information

If you have any questions you can use the forum thread or send an email to qqqinbox@gmail.com.