

SHierarchy

Version 1.3.2

Introduction

SHierarchy and **sFavorite** are powerful Unity development tools designed to optimize productivity and streamline organization. By providing features like automatic component icons, a Quick Inspector, error tracking, and asset management, these tools enhance Unity development, making it faster, more intuitive, and less error-prone.

Key Features of SHierarchy:

Automatic Component Icons: Instantly assign intuitive, recognizable icons to GameObjects in your Hierarchy window. This helps Unity developers easily identify and organize components, speeding up scene navigation and management.

Runtime Save Component: Save and preserve component values during runtime, allowing you to retain changes made in play mode for debugging or testing adjustments. This feature ensures you don't lose any modifications when you stop the game.

Pin & Pin Tracker: Pin critical objects in your scene for quick access. The Pin Tracker window lets you view and manage all pinned objects, optimizing testing and debugging workflows.

Customizable Inspector Attributes:

[Label] Attribute: Add informative labels on top of serialized fields to provide extra context, improving the clarity of your code's functionality.

[Button] Attribute: Easily create clickable buttons in the Inspector to execute functions directly from the editor, saving valuable development time.

[ReadOnly] Attribute: Make specific fields or properties visible but non-editable, ensuring essential values can't be accidentally modified during development.

[EnableIf] / [DisableIf] Attributes: Control field visibility and interaction dynamically based on specific conditions, offering a more customized Inspector experience.

[ShowIf] / [HideIf] Attributes: Manage the display of fields based on conditions, improving UI clarity and reducing unnecessary clutter.

[Separator] Attribute: Organize your Inspector view with separator lines that enhance component grouping for better readability.

[MinMaxSlider] Attribute: Create adjustable sliders for Vector2 fields, providing an easy way to manipulate min/max values directly in the Inspector.

Quick Inspector:

Each component icon in the SHierarchy tool enables quick access to a focused Inspector view, speeding up editing and adjustments without the need to navigate through multiple layers.

Custom Icons:

Personalize your Unity workspace with custom icons for components and scripts, improving scene navigation and asset identification.

Error and Warning Tracking:

Efficiently track and resolve common issues with a robust error tracking system that identifies:

Negative Colliders: Colliders with negative dimensions that could affect physics.

Missing Scripts: GameObjects lacking required scripts.

LOD Issues: Missing renderers in Level of Detail groups.

Duplicate Event Systems/Audio Listeners: Alerts for multiple instances causing conflicts.

Missing References: Flags scripts with missing serialized references.

Streamline Asset Management with **sFavorite** for Unity

sFavorite is the perfect complement to SHierarchy, offering a simple and efficient way to manage your assets in Unity. It helps you stay organized by grouping your most-used assets, scripts, and prefabs, providing fast access and improving your workflow.

Key Features of sFavorite:

Organized Asset Management: Categorize and group assets, scripts, and prefabs for quick access. This feature keeps your project organized and minimizes search time for frequently used files.

Efficient File Access: Quickly drag and drop files into the sFavorite window for streamlined asset management. The search function allows you to find specific assets within seconds, while category collapse helps keep the interface clean and uncluttered.

Streamlined Workflow: Interact directly with files in the sFavorite window—open assets with a simple double click or view their file paths with a single click, significantly reducing the time spent navigating the Project window.

Unlock the Full Potential of Your Unity Projects with SHierarchy and sFavorite

Together, SHierarchy and sFavorite provide an all-in-one toolkit for Unity developers. These tools simplify scene management, error resolution, and asset organization, enabling a more efficient and productive development process. Whether you're working on a game, application, or tool, these Unity plugins are designed to maximize your development time and improve workflow efficiency.

How to Use SHierarchy

Once *SHierarchy* is imported into your Unity project, component icons will automatically appear next to GameObjects in the Hierarchy window. Clicking an icon allows you to quickly access the associated component's Inspector, significantly speeding up development and debugging.

Component Warnings and Errors

SHierarchy displays warning and error icons for common component issues, such as:

- **Negative Colliders:** Automatically highlights colliders with negative dimensions.
- **Missing Scripts:** Identifies GameObjects with missing *MonoBehaviour* scripts.
- **Missing Renderers on LOD:** Alerts you when a Level of Detail (LOD) group lacks renderers.
- **Duplicate Event Systems or Audio Listeners:** Warns you about multiple instances of event systems or audio listeners.
- **Missing References:** Highlights serialized fields in scripts that have missing references.
- Clicking any warning/error icon opens a detailed description, helping you quickly understand and resolve the issue. If a child GameObject has issues, the warning icon will also appear on the parent, making it easier to identify and address problems.

Custom Icons

You can assign custom icons to components that lack native Unity icons. To do this, use the *SHierarchy Manager*:

Navigate to ***SHierarchy* > *SHierarchy Manager***.

In the panel, enter the component name (without spaces).

Choose an icon sprite (drop custom sprites into the "Non Registered Icons" folder).

Click "Assign Icon."

Now, your custom components will display recognizable icons in the Hierarchy window, improving project organization.

Custom Options

You can customize more the Hierarchy

Navigate to ***SHierarchy* > *SHierarchy Manager***.

In the panel, setup the various options for better hierarchy workflow.

How to Use sFavorite

sFavorite is designed to simplify your workflow by providing a dedicated window where you can quickly access your most frequently used files. Here's how to make the most of it:

1. Open the sFavorite Window

Navigate to the Window menu and select sFavorite to open the window.

2. Create and Manage Categories

If you don't have any categories yet, create a new one with a single click.

Easily drag and drop files into the right-hand panel to add them to a category.

Want a more compact view? Collapse categories by clicking the arrow button at the top. This helps keep the window tidy and focuses on what's important.

3. Find Files Quickly

Use the search bar to locate specific elements within a category. This is especially helpful when your sFavorite list grows large.

Need to start fresh? Click Clear to reset the entire category with a single action.

4. Interact with Your sFavorite

Single Click: Opens the file path in the Inspector for quick reference.

Double Click: Instantly opens the file.

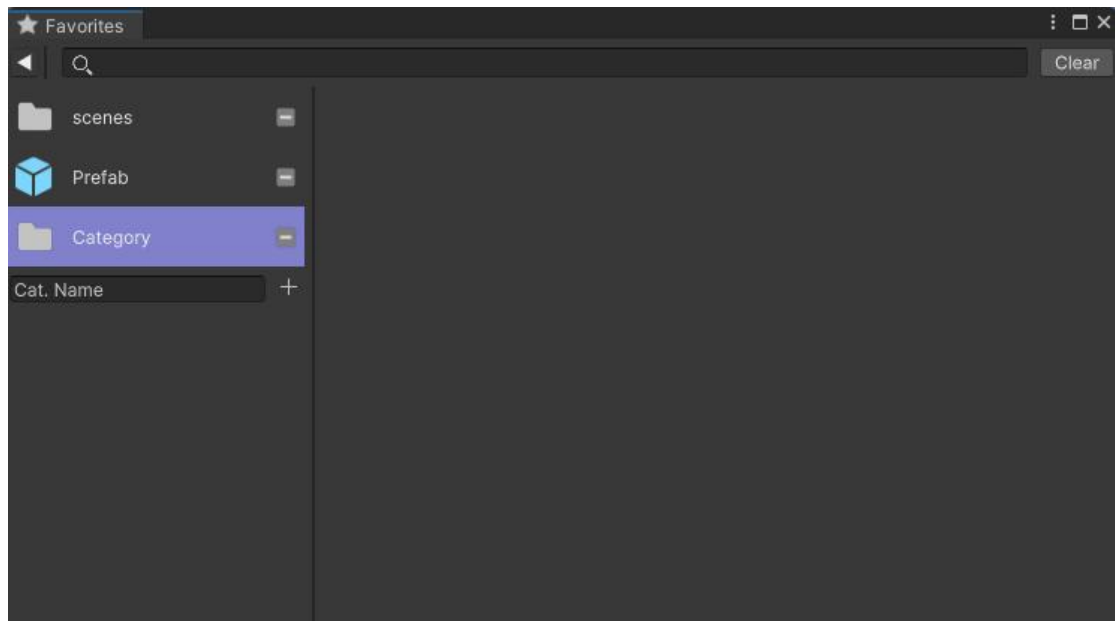
Remove Files: If you want to remove an item, simply click the button next to the file's name to take it off your list.

Why Use sFavorite?

Organize Your Workflow: Keep your most-used files, assets, and scripts neatly categorized.

Save Time: Access critical files in just a couple of clicks, eliminating the need to search through long project folders.

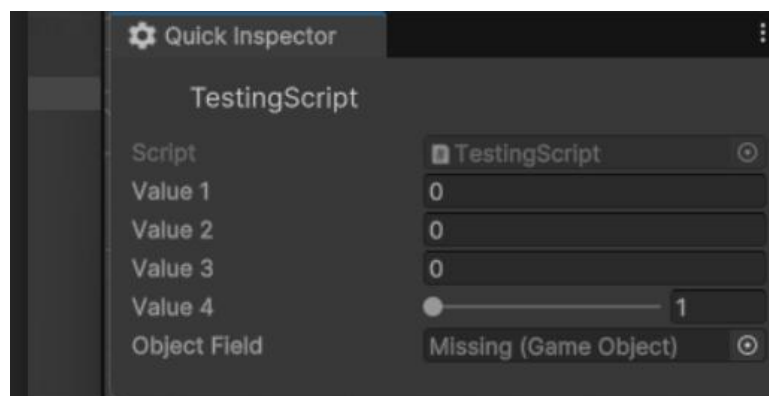
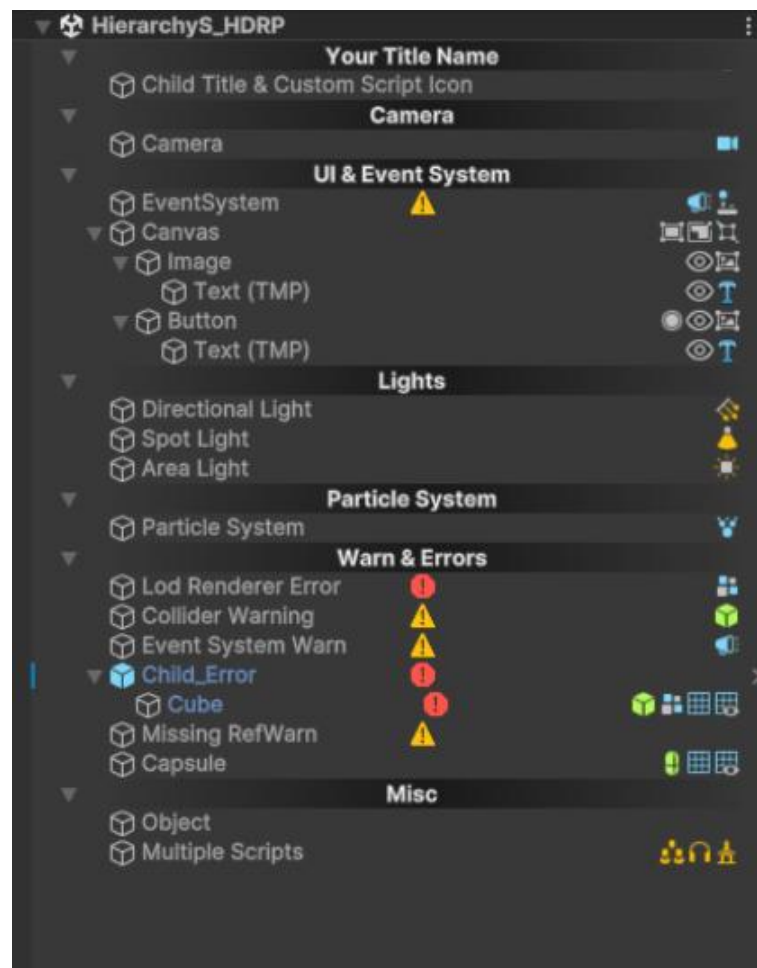
Personalize: Create categories that make sense for your workflow, whether it's by file type, project phase, or personal preference.



Features

1. Automatic Hierarchy Icons

SHierarchy automatically assigns icons to components in the Hierarchy window. Each icon serves as a button, providing quick access to the associated component's Inspector for easy value editing and management.

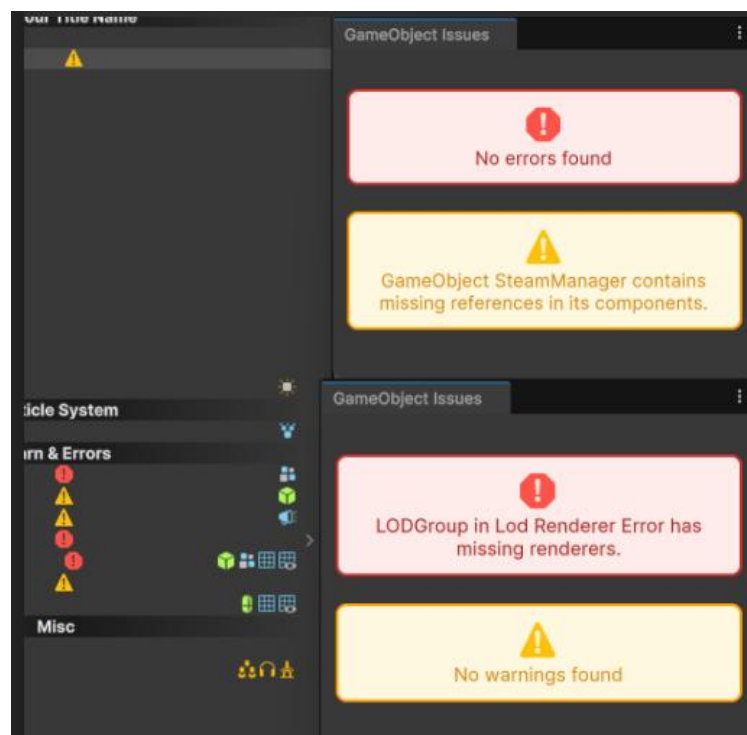


2. Warnings for Component Issues

SHierarchy identifies and marks common component-related issues with a warning or error icon:

- *Negative Colliders*
- *Missing Scripts*
- *Missing Renderers on LOD*
- *Duplicate Event Systems or Audio Listeners*
- *Missing References*

If an issue exists in a child GameObject, the icon will also appear on the parent for easier identification. Clicking the icon opens a description box to help you resolve the issue efficiently.



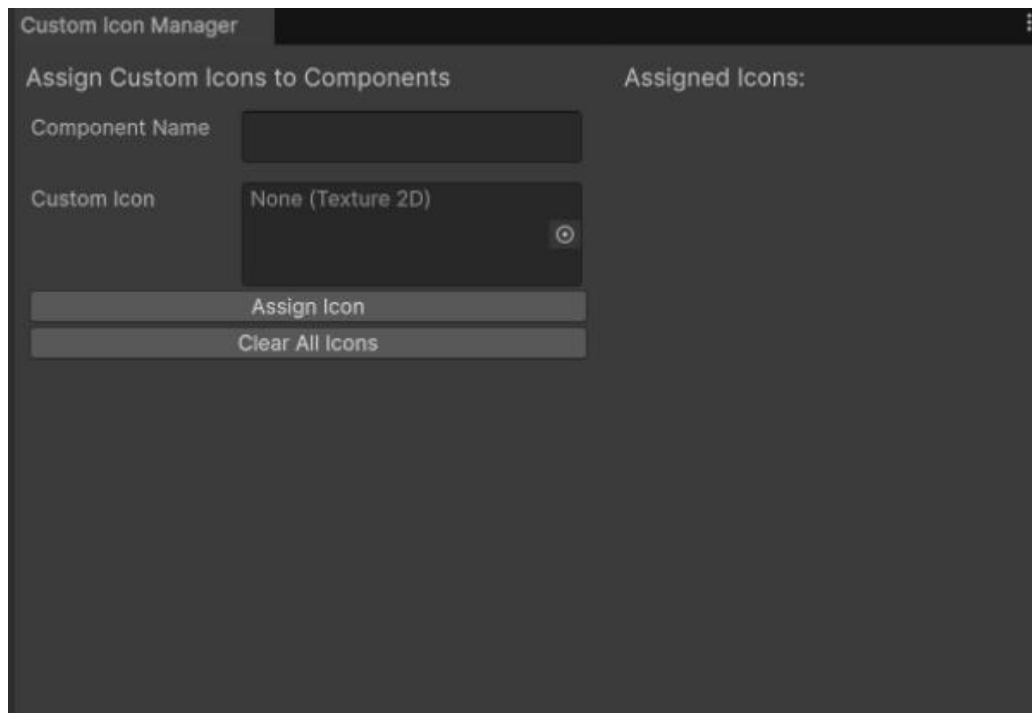
3. Custom Icons for Components

SHierarchy allows you to assign custom icons to components that do not have native Unity icons. This feature is particularly useful for custom scripts or third-party packages:

Open the Custom Icon Manager via **Tools > Custom Icon Manager**.

Enter the component name and assign an icon sprite.

Click "**Assign Icon**" to apply the custom icon in the Hierarchy window.



4. Title System

The Title System in SHierarchy helps you organize and visually structure your Hierarchy window, especially in complex scenes with many GameObjects. It allows you to create distinct, easy-to-identify sections within the Hierarchy by adding custom titles.

How to Use the Title System

Create a Title

In the Hierarchy window, create a new empty GameObject by right-clicking and selecting Create Empty.

Rename this GameObject using the format: `== (Your Title Name)`. Replace (Your Title Name) with the text you want to display as a title.

Customize Your Title

The title will appear prominently in the Hierarchy window, making it easy to organize and group related GameObjects.

You can create multiple titles to segment your Hierarchy into logical sections, such as *"Enemies," "Environment," "UI Elements,"* etc.

Visual Organization

Titles are a great way to enhance readability and navigation within your scene. They provide a clear visual separation between different parts of your scene, making it easier to locate and manage GameObjects.

Manage Titles

Drag and drop GameObjects under the appropriate title to group them accordingly. Titles act as headers, helping you to keep related items together and improving your overall project organization.

Benefits of the Title System

Enhanced Organization: Clearly define sections in your Hierarchy to better manage large and complex scenes.

Improved Navigation: Quickly find and access related GameObjects by navigating through well-labeled sections.

Streamlined Workflow: Reduce clutter and improve your development efficiency with organized visual groupings.

With the Title System, your Hierarchy window becomes a powerful organizational tool, helping you maintain clarity and control over your Unity project.



Attributes : How to Use Them

Use these attributes to improve your inspector layout, make your editor tools more user-friendly, and boost productivity.

[Label]

Adds a custom label above a variable in the Inspector.

Usage:

```
[Label("The object will do something!\nEnter your text here!")]
```

```
public string myText;
```

[Button]

Adds a clickable button in the Inspector to run a method—great for testing or editor utilities.

Usage:

```
[Button]
```

```
public void MyFunction() { }
```

```
[Button("Custom Button Name")]
```

```
public void MyOtherFunction() { }
```

[ReadOnly]

Shows a variable in the Inspector, but makes it non-editable.

Usage:

```
[ReadOnly]
```

```
public int currentHealth;
```

[HideIf] / [ShowIf]

Hides or shows variables based on a condition (usually a boolean).

Usage:

```
public bool isHiding = false;
```

```
[HideIf("isHiding")]
```

```
public int enableInt = 0;
```

```
[HideIf("isHiding")]
```

```
public int enableFloat = 0;
```

```
[ShowIf("isHiding")]
```

```
public int DisableInt = 0;
```

```
[ShowIf("isHiding")]
```

```
public int DisableFloat = 0;
```

[EnableIf] / [DisableIf]

Enables or disables editing of a field based on a condition.

Usage:

```
public bool IsEnabled = false;
```

```
[EnableIf("IsEnabled")]
```

```
public int enabledvalue = 0;
```

```
[DisableIf("IsEnabled")]
```

```
public int disableValue = 0;
```

[MinMaxSlider]

A slider for Vector2 values with min and max limits. Easier than setting values manually.

```
[MinMaxSlider(0f, 1f)]
```

```
public Vector2 range = new Vector2(0.2f, 0.8f);
```

[Separator]

Adds a visual line in the Inspector to organize sections.

Usage:

```
[Separator]
```

```
[Separator(2)]
```

```
[Separator(2, "Optional Title")]
```

Pin System:

Overview

The Pin System in SHierarchy allows developers to "pin" frequently accessed GameObjects. This is especially useful when working in large scenes where important objects can become difficult to locate.

How to Pin a GameObject

Click in the Pin on left side of the GameObject.

The GameObject will now appear with a special pinned color.

How to Unpin a GameObject

Click in the Pin on left side of the GameObject.

Select "Unpin" to remove it from the pinned list.

RunTime Save Component:

Overview:

The Runtime Save Component allows you to capture and persist changes made to GameObjects and their components while the Unity editor is in Play Mode. This is a powerful tool for fine-tuning object positions, values, or settings during runtime and then saving them directly back to the scene without manual re-entry.

Key Benefits

Avoid losing runtime tweaks after exiting Play Mode.

Speed up iteration when testing changes to object transforms, UI, AI parameters, etc.

Works seamlessly with commonly used components (Transform, Rigidbody, Light, etc.).

How to Use Runtime Save

Enter Play Mode in Unity.

Make any desired changes to a GameObject in the Inspector (e.g., change position, adjust light intensity, tweak Rigidbody values).

Once satisfied:

Click on the **Floppy Disk** that is presented on the component Header.

Exit Play Mode.

The changes will now persist in the scene, as if you had made them in Edit Mode.

Contacts for support

Discord (Faster): <https://discord.gg/VJ7PtAUv22>

Email: support@shadowprofile.pt