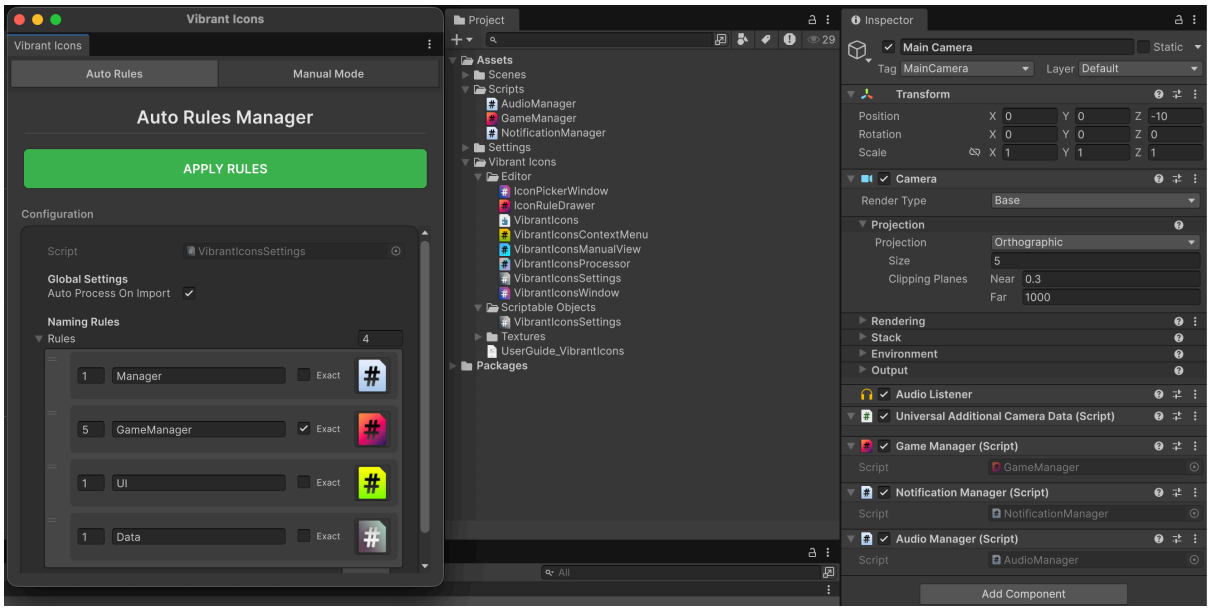
Hey there! 👋 I'm Paul, a 27-year-old developer passionate about creating tools that make life easier. My background is in Applied Computer Science, and I love optimizing workflows in Unity.

Overview

Vibrant Icons is a Unity Editor extension that helps you organize your project visually. Instead of staring at a sea of identical C# script icons, you can assign distinct colors and symbols to your scripts.

What makes this version special is the **Auto Rules** system - you define the naming convention (e.g., every script ending in "Manager" gets a blue icon), and the tool handles the rest automatically.



Compatibility

- ➤ **Unity Versions:** 2021.2.0f1 and newer (fully compatible with Unity 6).
- ➤ **Platform:** Windows, macOS, Linux.

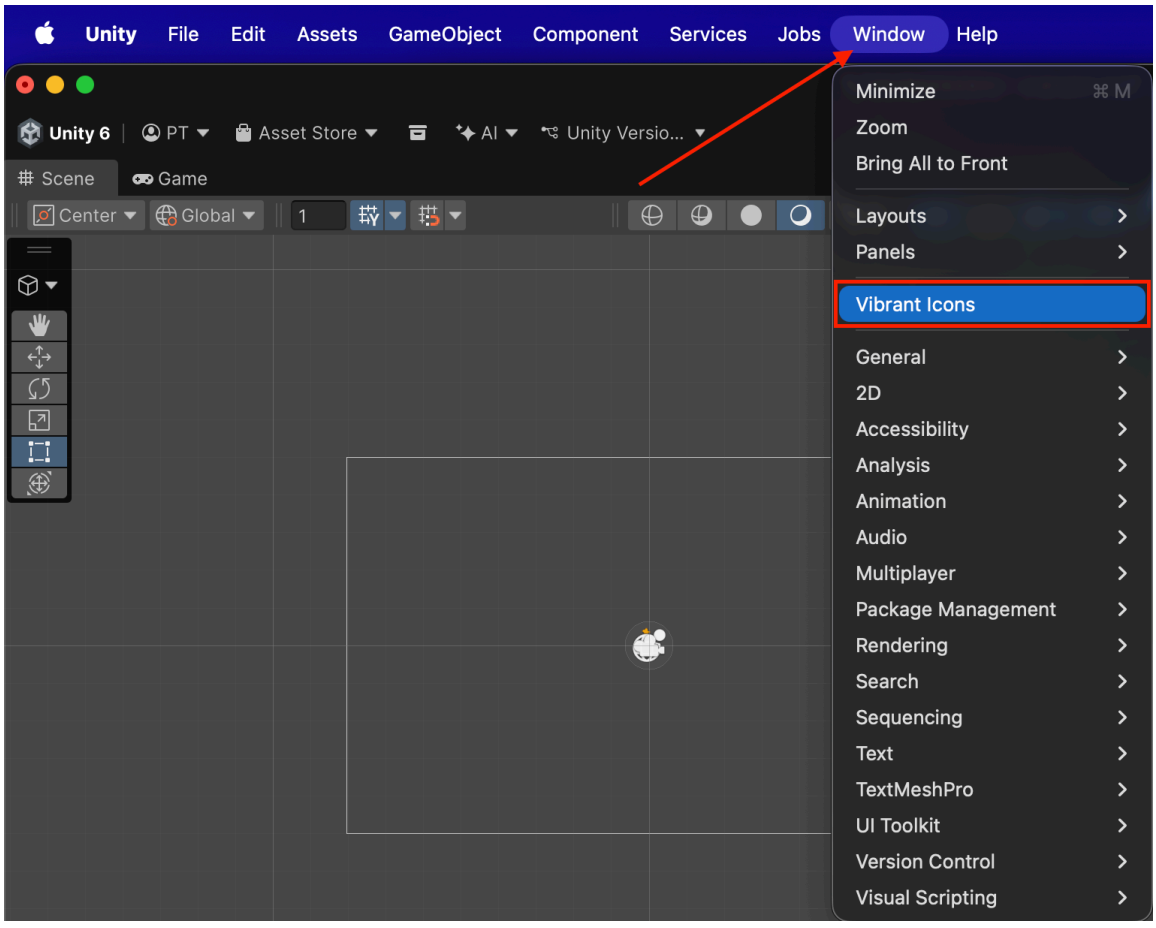
Features

- ✅ **Auto Rules Manager:** Set global rules based on script names (keywords). The tool automatically applies icons whenever you create or import a script.
- ✅ **Context Menu:** Right-click any script in the Project view to quickly change its icon without opening the main window.
- ✅ **Manual Override:** Need a specific icon for just one file? You can manually assign icons via Drag & Drop or by selecting file from the project.
- ✅ **Priority System:** Define which rules are more important (e.g., a "GameManager" can have a different icon than a standard "Manager").

User Guide

Getting Started

1. Import the package into your project.
2. Open the main window by navigating to: **Window > Vibrant Icons**.



3. If this is your first run, click **"Create Settings File"**. This will generate a configuration file in the Scriptable Objects folder within the package.

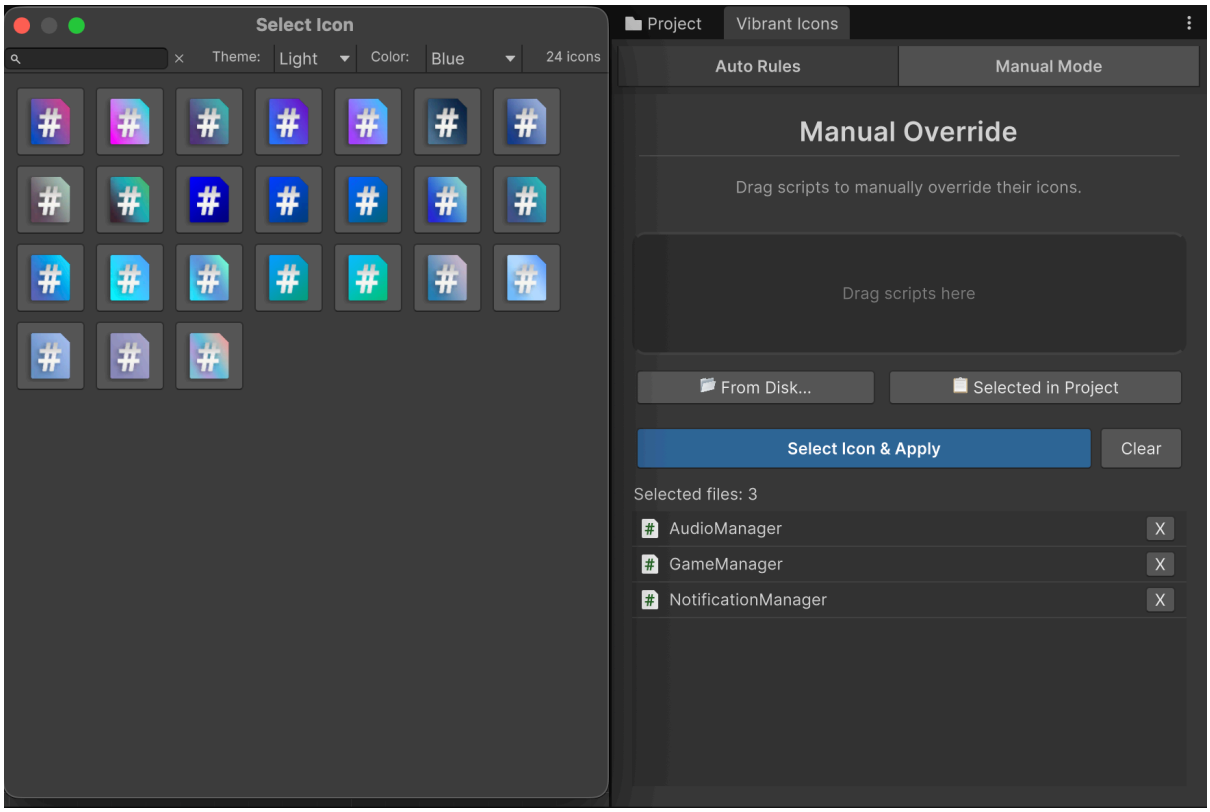
Auto Rules (Recommended Workflow)

This is the most efficient way to use the tool. You set the rules once, and Vibrant Icons keeps your project organized forever.

1. Go to the **Auto Rules** tab.
2. Click **"+"** to add a new rule.
3. **Keyword:** Type the text to look for in the script name (e.g., Manager, Controller, Data).
4. **Icon:** Click the button to select an icon from the library.
5. **Priority:** If a script matches multiple rules, the one with the higher priority number wins.
6. **Exact Match:** Check this if the filename must be exactly the same as the keyword (e.g., "Player" vs “PlayerController").
7. Click **APPLY RULES** to scan your entire project and apply these icons immediately.

Manual Mode

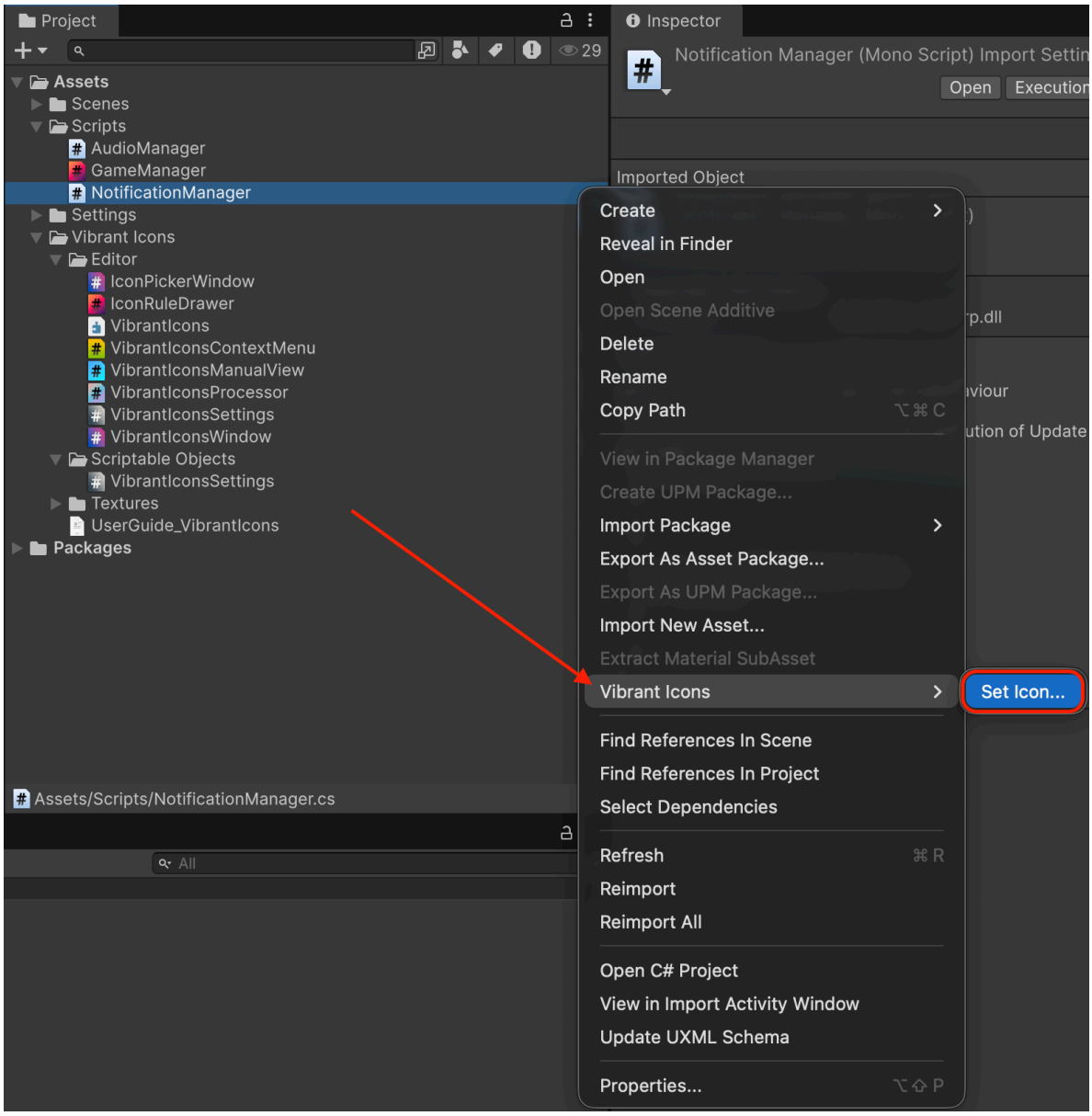
Use this for specific scripts that don't fit your general naming conventions.



1. Switch to the **Manual Mode** tab.
2. Add scripts to the list using one of three methods:
    a. **Drag & Drop** scripts into the drop zone.
    b. Click "**Selected in Project**" to add whatever you currently have selected in the Project window.
    c. Click "**From Disk...**" to browse for a specific file.
3. Click "Select Icon & Apply"

Context Menu (Quick Action)

You don't need to open the window for quick changes.

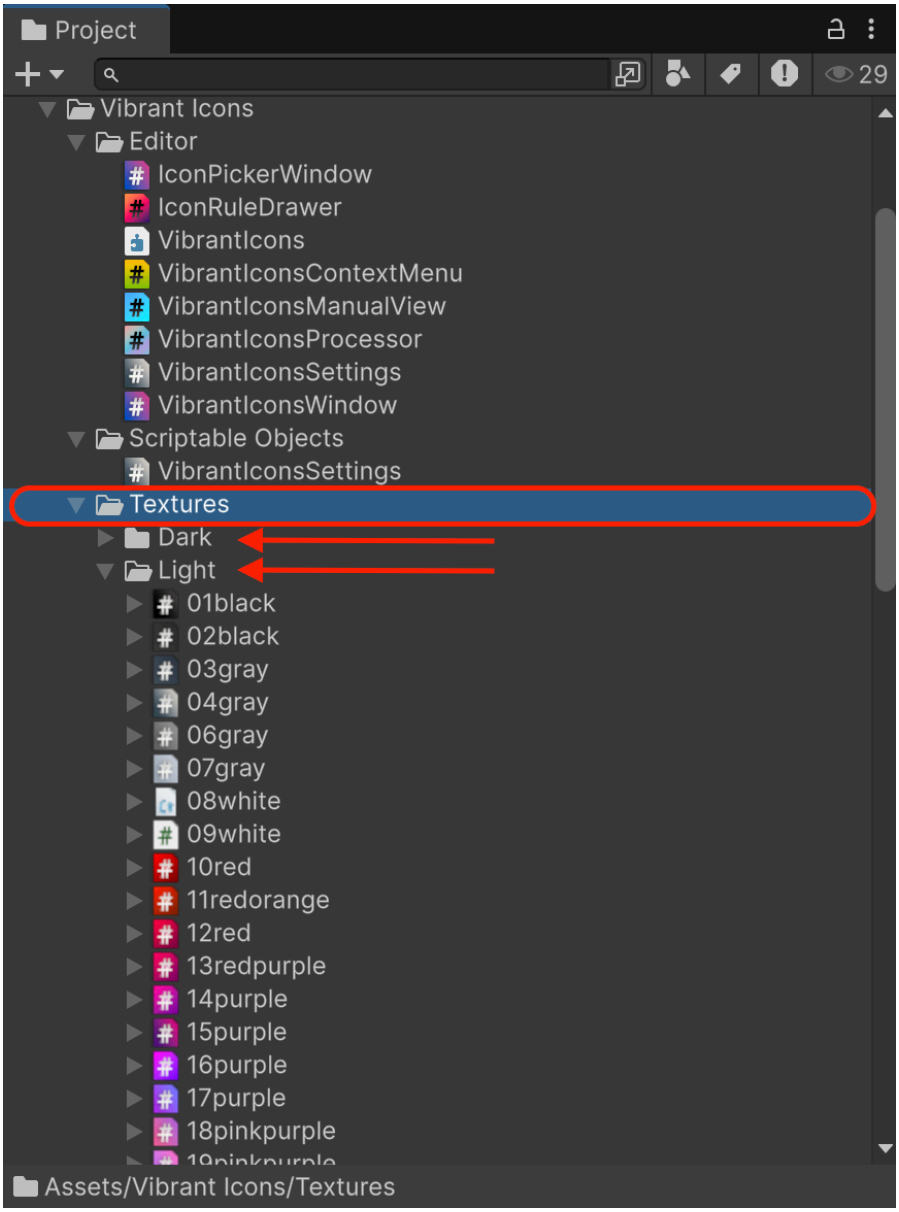


1. **Right-click** on any C# script (or multiple scripts) in the Project View.
2. Select Vibrant Icons > Set Icon....
3. Pick your icon from the popup.

Customization

You can easily add your own icons to the library.

1. Navigate to the Vibrant Icons/Textures folder.
2. You will see Dark and Light subfolders.
    a. Place icons for the **Dark Theme** in the Dark folder.
    b. Place icons for the **Light Theme** in the Light folder.



3. **Important:** Ensure the icons have the same filename in both folders so the automatic theme switching works correctly.

Support

If you have any questions, spot a bug, or just want to high-five, feel free to reach out at: paweltrojanski@gmail.com.

I'm here to help! 😇