Logger — Lightweight & Flexible Debugging Tool for Unity

A simple, elegant, and powerful logging tool to keep your development clean, clear, and under control.

Features

- Easy-to-use static logging API

- Color-coded log categories with emoji/icons

- Editor interface for category filtering

- Enable/disable logs globally or per category

- Lightweight and dependency-free

- Works in Editor and Build

Installation

1. Copy the `LoggerLogic/` folder into your Unity project.

2. Open the configuration asset:

   `Assets/LoggerLogic/Config/LoggerConfig.asset`

3. Adjust logging categories, colors, icons, and settings through the Inspector.

Configuration

Customize logging via the `LoggerConfig` asset: - **Enable All Logs** — Toggle all categories on/off - **Editor Only** — Logs will appear only in the Unity Editor - **Category Configuration** — Set up custom categories with colors and emojis

You can also modify the logger at runtime:

```
Logger.Enabled = true;
```

Usage

```
CustomLogger.Log("This is a system log", CustomLogger.LogCategory.System);
CustomLogger.LogWarning("UI warning", CustomLogger.LogCategory.UI);
CustomLogger.LogError("Gameplay error", CustomLogger.LogCategory.Gameplay);
```

Example Output

```
 [System] This is a system log
LoggerExample.Awake:13
```

Project Structure

```
LoggerLogic/
  Config/
      LoggerConfig.asset
  Editor/
  Example/
  Scenes/
  CustomLogger.cs
  LogCategory.cs
```

Integration

- No external dependencies
- Compatible with Odin Inspector, Zenject, UniTask, etc.
- Useful for in-editor tools or runtime debugging
- Designed to be fast and flexible for any Unity project

License

MIT License — free to use, modify, and distribute. See LICENSE for full details.

Contact

Questions, ideas, or bugs? Feel free to reach out!
GitHub: https://github.com/AndriiSviatenko/Logger-Lightweight-Flexible-Debugging-Tool-for-Unity **Email**: sviatenkoandrii@gmail.com