

Surventry - Essential Inventory Logic

- Overview (Docs are under construction)
 - Introduction

Overview (Docs are under construction)

Introduction

▣▣ What is it?

Surventory is the core of a fully modular inventory framework that's currently in development.

At its heart, Surventory is built entirely around **interfaces** — there's no fixed class hierarchy, no required base classes.

▣▣ Why is it different?

A lot of inventory systems try to be everything from the start — bloated with features, overloaded with assumptions.

But every game is different, and that's why **Surventory focuses on the absolute essentials**.

Instead of giving you a complex, rigid system, it delivers only the **core features you actually need** to build your own inventory system — clean, extensible, and easy to understand.

▣▣ Modularity is Key

Interfaces are the key to **true modularity, independence, and flexibility** — allowing you to build your own systems and plugins on top of it, without having to strip away things you don't need.

I like to think of good game architecture as a **brick wall**:

Every plugin or feature is a **brick** — simple, solid, self-contained.

The game code is the **mortar** — holding it all together and giving it meaning.

Consider Surventory as one of those bricks — just another brick in the wall or something to build more advanced systems on top.

☐☐ What's in the box?

☐☐ Core Inventory Functions

- Add item(s)
- Remove item(s)
- Remove a specific amount from a stack
- Split stacks
- Merge stacks
- List all items
- Remove all items

☐☐ It also takes care of:

- Network replication
 - Messaging (fully customizable/exchangeable message broker)
 - Configurable slot management (finite or infinite inventory)
-

☐☐ Why Is That Enough?

To make sure unnecessary stuff doesn't get in your way.

Weight systems, UI layers, categorization, drag & drop, or Action RPG style inventory grids — all of that is highly game-specific.

Should weight replace slots? Or combine with them? Should items be sorted by type, tags, or rarity? These aren't core questions — they're design choices.

Surventry doesn't make that choice for you.

It gives you the foundation: reliable item logic, stack handling, messaging, and replication — and leaves the rest open for you to define.

☐☐ Do I Have to Build the Rest Myself?

Most likely? Yes. But that's the reason we're here, right?

Hopefully we're not alone in this — keep reading and see where this could go. ☐☐

☐☐ My Vision

I would **love** to see people build features on top of Surventory — and I absolutely plan to create more "bricks" myself, covering things like weight limits, UI tools, or equipment systems.

By supporting this project, you're helping bring a truly modular inventory system and its extensions to life — and the best part is: you can test the full-featured core plugin today.

☐☐ Try the Core – For Free

Keep supporting for more.

<https://www.patreon.com/suricoon>

Even the **Free Tier** gives you access to the demo build, so there's nothing stopping you from giving it a spin. I've prepared a **demo version** that contains the full core functionality:

- ☐ All features
- ☐ Fully usable in Blueprints
- ⚠ The demo displays a small message window when launching the game or editor — this also appears in shipping builds
- ☐☐ Source code is not included

This version is intended for **evaluation and prototyping** — not for commercial or production use.

☐☐ How to Install the Surventory Demo Plugin (Engine-Wide)

To use the Surventory demo, the plugin must be installed **in your Unreal Engine installation, not inside a specific project.**

This makes the plugin available across all projects in your engine — just like a Marketplace plugin.

Installation Steps:

1. Download the suitable attachment of the patreon post
2. Close Unreal Engine if it's currently running
3. Locate your Unreal Engine installation directory

4. Navigate to: `Engine → Plugins → Marketplace`
(If the `Marketplace` folder doesn't exist, create it)
5. Extract the Surventory demo archive
6. Copy the entire `Surventory` plugin folder into the `Marketplace` directory
7. Restart Unreal Engine and open any project
8. Go to **Edit → Plugins**, search for **Surventory**, and make sure it is enabled

After these steps, the plugin is ready to use in all of your UE5 projects (for demo purposes).

📖 Further Reading

Looking to get the most out of Surventory?

Here are a few ways to stay informed and dive deeper into the system:

- **Read the official documentation** for technical insights, usage examples, and plugin architecture:
📄 <https://docs.suricoon.com>
- **Join the community on Discord** to ask questions, get early previews, and stay up to date with new releases:
📄 <https://discord.gg/5Bgj4mwzz9>

Make sure to **check back regularly** — updates, new features, and additional plugin modules are planned.

This is just the beginning.