

Quests

Quests are the engine that drive all narrative and timed events on the station — from long-form Patron arcs to small incidents like shipment delays or a visiting celebrity. They are **structured as staged timelines**, but instead of rigid “missions,” they are flexible story controllers that can touch many different parts of the station’s life.

How Quests Work

- **State Machine Core** - Every quest is made up of a set of numbered **stages**. Each stage represents a specific point in the quest’s progress — an active situation, a waiting period, or a completed outcome.
- **Script-Per-Stage Logic** - Each stage can have its own script that runs when the quest enters that stage. These scripts can:
 - Trigger dialogue
 - Cause environmental changes
 - Update the Story Ledger
 - Prompt other systems to react (delayed ship arrivals, new bulletin posts, special customer visits, changes in available goods)
- **Flexible Flow** - Stages do not need to happen in strict order; they can branch, skip, or loop depending on story needs or player actions.

Scope of a Quest

A quest can be:

- A **Patron Story Arc** that progresses across many visits
- A **Station-Wide Event** like a festival, docking delay, or power outage
- A **Small Favor or Request** such as sourcing a special item
- A **Recurring Situation** that can happen multiple times in different forms (e.g., supply shortages)
- A **World-Driven Unlock** - expanding what’s available to the player, such as:
 - Adding a new decorative item, fixture, or product into the shop’s **storage** as a reward
 - Unlocking a new product, style, or furnishing in the **ordering system**

What Quests Affect

Quests never “hard-wire” themselves into other systems — instead:

- The **Quest System** only tracks current stage and conditions
- The **Quest Stage Script** is what reaches out to other systems, asking them to change state or display something new
- Other systems (Shop Terminal, Space Station Systems, Customers, PDA) simply respond in their own way to the requested change
- This keeps each system self-contained, but lets quests weave them together

Example Flow

1. **Quest:** *Docking Delay Mystery*
Stage 10: “Announcement Made”
2. The Quest System moves to Stage 10 after necessary conditions are met (time of day, certain Patron visit, rumor overheard)
3. **Stage 10 Script** runs:
 - Tells Station Announcements to broadcast a delay notice
 - Updates the Docking Registry with new arrival time
 - Leaves a note in the Story Ledger
 - **Adds “Ore Sample” to shop storage** as a gift from a prospector, or enables ordering it from suppliers going forward
4. Station Systems and shop life change naturally as a result — quieter foot traffic, certain customers commenting — without the Quest System directly running them.

For Artists and Narrative

- A “quest” might require new **visual moments**: lighting changes, props appearing/disappearing, special NPC outfits, or environmental cues that signal an event is happening
- Treat quests as **story containers** — each stage is a place where you can build a vignette, create ambience, or drop story hints that players can notice in passing
- Unlocks that expand what the player can place or order should be framed as **natural rewards**: gifts left in storage, new stock arriving, or contacts offering fresh catalogue entries

Design Goals

- Keep quests **decoupled** from core mechanics so they can be reused across multiple projects
- Allow **script flexibility** at each stage to tie in different systems without forcing changes to those systems

- Make it easy for artists and narrative to **attach atmosphere and detail** to specific moments in a quest
 - Treat “upgrades” as **world-driven availability expansions**, not abstract menus or skill trees
 - Ensure the **Story Ledger** remains the player’s in-world quest log, showing progress, clues, and outcomes without breaking the immersive style
-

Revision #5

Created 2025-08-14 08:43:43 UTC by Mike

Updated 2025-08-14 09:20:14 UTC by Mike