Building Multi-Turn Agentic Workflows with MCP

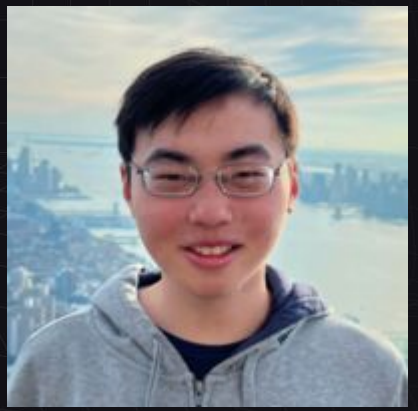
Lessons from Avatar Generation at Roblox

CHAPTER

Presenters

Rohan Gangaraju
Senior MLE

Economy ML



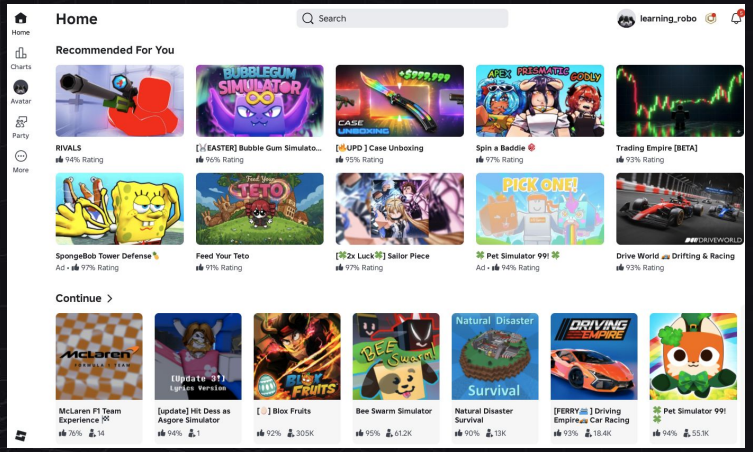
Jason Ding
Software Engineer

Economy Marketplace

ROBLOX

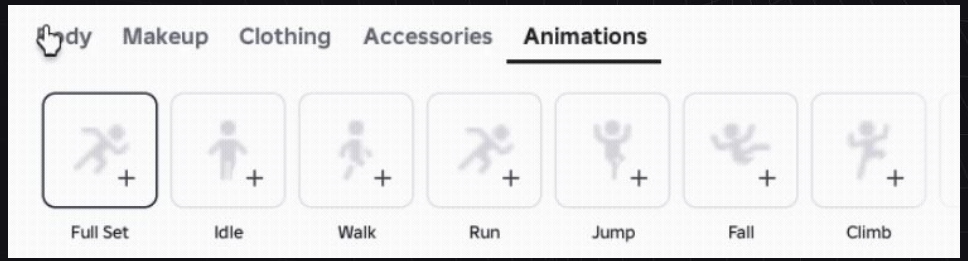
What is Roblox?

- Immersive global platform
- Users create, play and share experiences
- Powered by user-generated content (UGC)

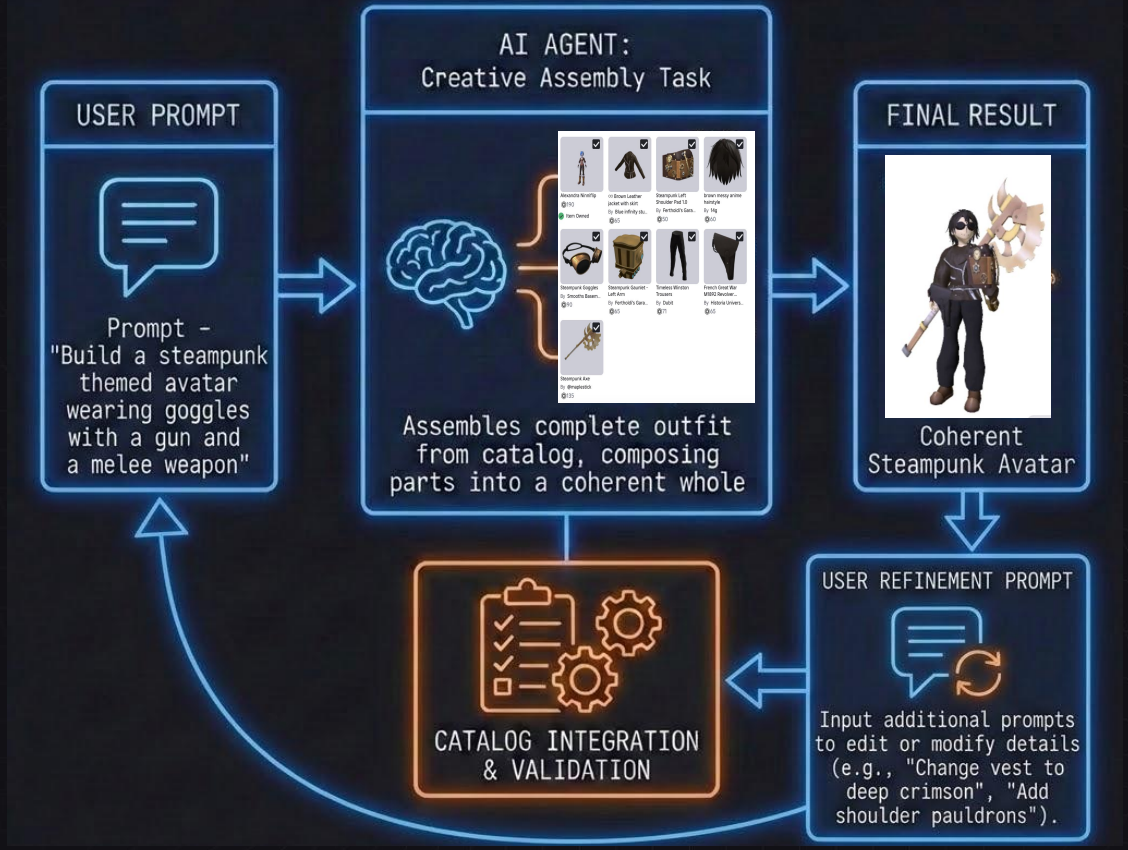


Avatars in Roblox

- Digital identity and self-expression
- Large catalog of assets in our UGC marketplace
- Highly compositional, layering items across body categories



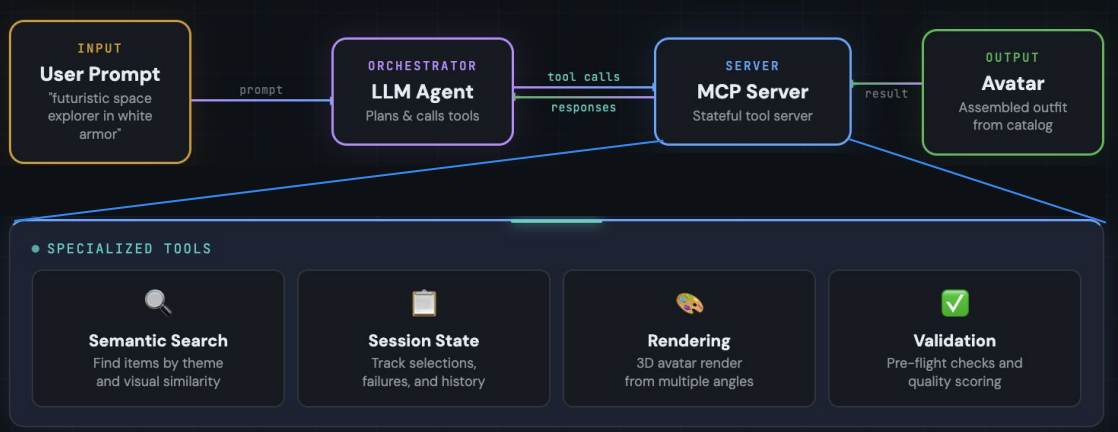
The Goal: Agentic Avatar Generation



What we built

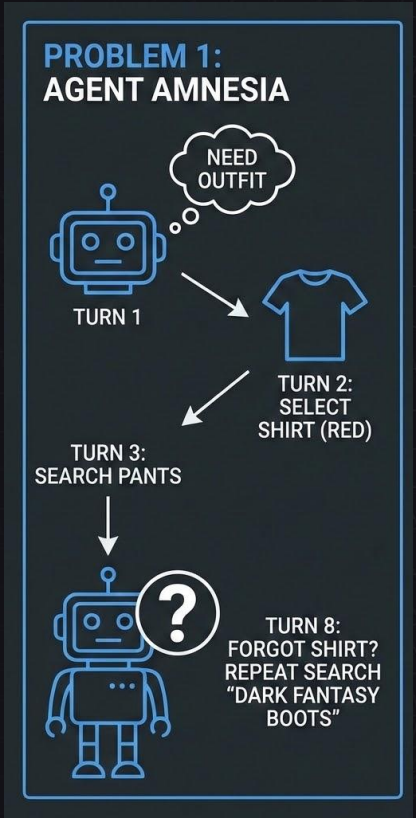
Specialized MCP Server with Tools for

- Analyzing prompts and themes
- Searching Items in our catalog
- Fetching Item visuals and other metadata
- 3D Rendering avatars
- Validating results

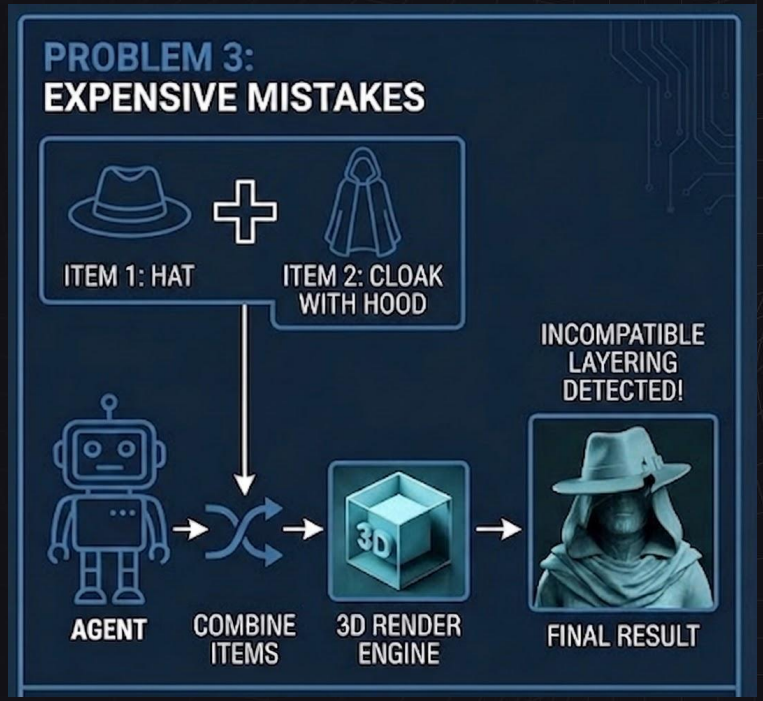


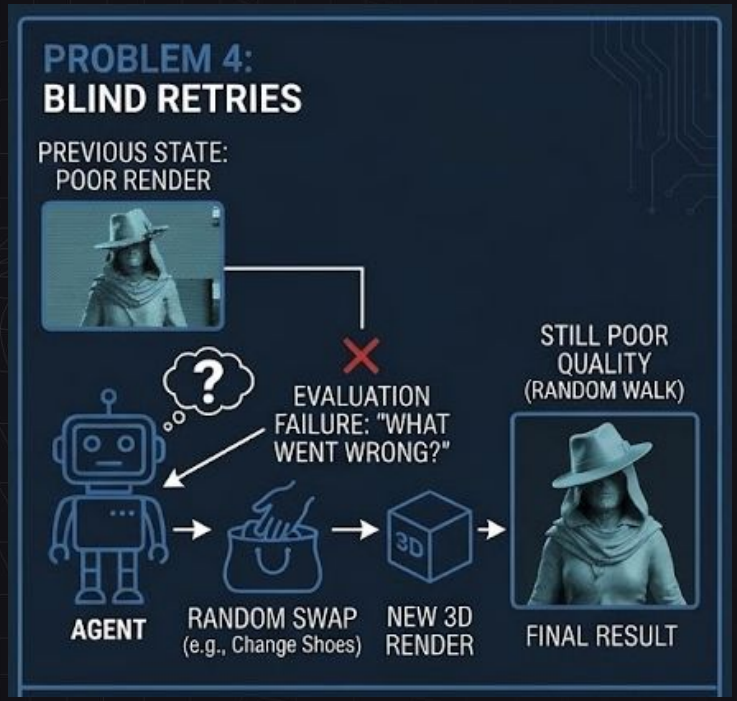
Demo

The Naive Agent: What didn't go so well



The Naive Agent: Where It Got Expensive





Four Patterns That made the Agent better

Problem	Pattern
Amnesia	Session Memory
The Round-Trip Trap	Composite Tools
Expensive Mistakes	Pre-flight validation
Blind Retries	Validate-and-Retry Loops

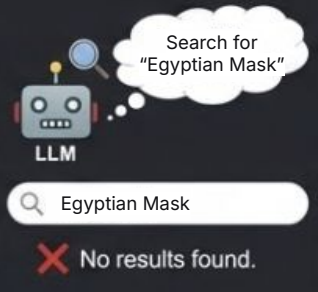
Session Memory

Issue: Agent Amnesia

Turn 1: Add Pants

...

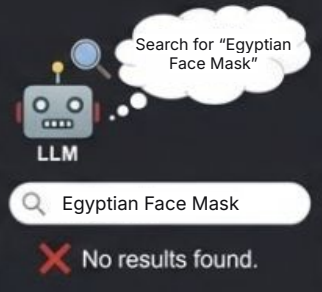
Turn 3: Failed Search



Turn 8: Amnesia / Duplicate Clothing

...

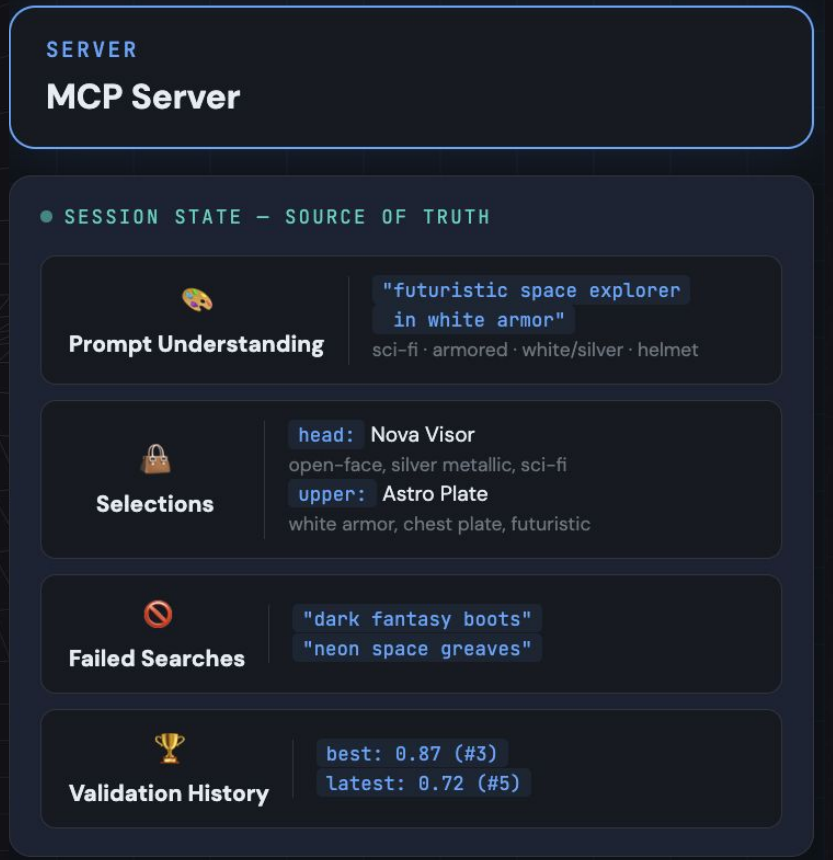
Turn 11: The Search Continues



Session Memory: Curing Amnesia

The server remembers so the agent doesn't have to

- Server-side state tracks
  - Processed user input
  - Selections in each step
  - Failed attempts
  - Validation history



Session Memory: What This Unlocks

Prevent Wearing Duplicate Clothing Types

Prevent Same Past Search

AGENT TIMELINE

Turn 1 — Add Pants
Agent selects `"Royal Egyptian Pants"`

SERVER
`Recorded: lower → "Royal Egyptian Pants"`

Turn 3 — Search "Egyptian Mask"
No results found

SERVER
`Dead-end logged: "Egyptian Mask"`

Turn 8 — Add Pants (again)
Agent forgot it already picked pants

SERVER
`lower body already has "Royal Egyptian Pants"`
`Keeping existing selection. Move on to head?`

Turn 11 — Search "Egyptian Face Mask"
Agent tries a similar query

SERVER
`Similar to failed: "Egyptian Mask"`
`Try: "anubis headpiece" or "jackal helm"`

Turn 12 — Search "Anubis Headpiece"
Follows server suggestion — finds results

SERVER
`Recorded: head → "Anubis Jackal Helm"`

Turn 14 — "Create the Outfit"
Agent calls create — no IDs needed

SERVER
`Auto-assembled from session state`
`head + upper + lower + feet → rendered avatar`

SESSION STATE
Server-Side Memory

SELECTIONS
`head:` "Sun God Crown" — turn 12
`lower:` "Royal Egyptian Pants" — turn 1

FAILED SEARCHES
1. "Egyptian Mask" — turn 3
2. "Egyptian Face Mask" — turn 11

PROMPT UNDERSTANDING
`theme:` "Ancient Egyptian royalty"
`style:` ornate, gold, regal

Composite Tools

Issue: Too many atomic tools

Upper Body

`search_items("angelic upper")` → `[ id_1136, id_7147, id_7637, ... ]`

`fetch_image(id_1136)` →  `fetch_image(id_7147)` →  `fetch_image(id_7637)` →  . . .

1 + N
1 search, then fetch each image one-by-one

Back

`search_items("angelic cape")` → `[ id_1352, id_7638, id_1105, ... ]`

`fetch_image(id_1352)` →  `fetch_image(id_7638)` →  `fetch_image(id_1105)` →  . . .

1 + N
Same pattern repeats for every body part

Wings

`search_items("angelic wings")` → `[ id_1617, id_1925, id_9396, ... ]`

`fetch_image(id_1617)` →  `fetch_image(id_1925)` →  `fetch_image(id_9396)` →  . . .

1 + N
Agent tracks raw IDs across all categories

. . . *Head, Lower Body, Feet, Hair … same pattern for every category*

1 + N
And again...

"Wait, which ID was the white cape? Was id_7638 the cape or the wings? Let me fetch it again..."

5 categories × (1 + N) = 20–30+ round-trips

Each round-trip is an LLM inference. Every extra call is a chance for the agent to **lose its thread**, **duplicate a fetch**, or **forget what it picked**.

Composite Tools: Escaping the Round-Trip Trap

If the agent always does X after Y, merge them

- Combine logically paired operations into single tools
- Search + fetch images → one call with visual context inline
- Search across all body categories → one call instead of many

Composite Tools: From 12 Calls to 3

❌ ATOMIC TOOLS

Each operation = separate LLM inference

```
1  → search_items("angelic halo")
2  ← [ id_1, id_2, id_3 ]
3  → fetch_image(id_1)
4  → fetch_image(id_2)
5  → fetch_image(id_3)
6  → search_items("angelic upper")
7  ← [ id_4, id_5, id_6 ]
8  → fetch_image(id_4)
9  → fetch_image(id_5)
   • • •
12 → search_items("angelic wings")
13 ← [ id_7, id_8, id_9 ]
   • • •
```

Total round-trips	12 - 18+
LLM inferences	12 - 18+
Context window	Bloated with IDs

✅ COMPOSITE TOOLS

Merged operations = richer context, fewer calls

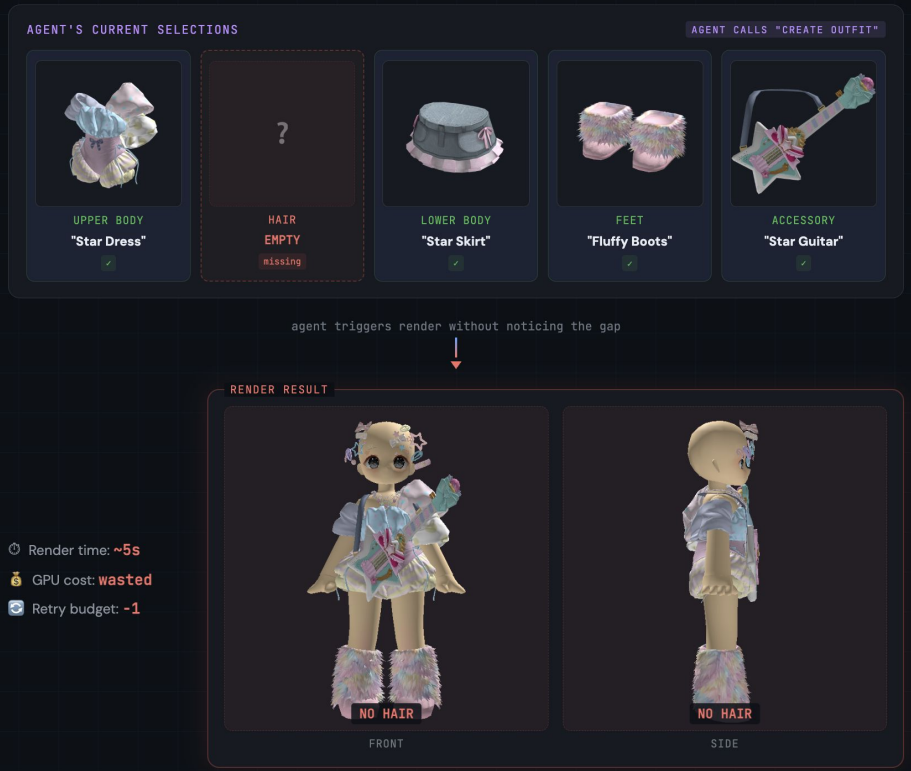
```
1 → search_all_with_images
2 ← rich response (below)
3 → select_items + create_outfit
  • • •
```

SINGLE RESPONSE INCLUDES IMAGES INLINE

Total round-trips	3
LLM inferences	3
Context window	Rich & visual

Pre-flight Validation

Pre-flight Validation: Stopping Expensive Mistakes



Pre-flight Validation: Stopping Expensive Mistakes

Cheap rules that run before expensive rendering

- Encode domain knowledge as lightweight constraint rules
- Check compatibility at item selection time, warnings returned inline
- "Hood + cloak = face hidden" caught in microseconds, not after a render

Pre-flight Validation: Two Checkpoints

- At individual item selection
  - Individual item checked against current selections
- Before full avatar creation
  - Global check across all selected items
  - Cross-category conflicts
    - (hood in "Head" + cloak in "Upper Body")
  - Missing essentials
    - "the pants check" - Missing essential items

Pre-flight Validation

Pre-flight Validation: Catching Problems Early

CHECKPOINT 1 — AT ITEM SELECTION

1

AGENT **Select "Dark Cloak" for Upper Body**

✓ Accepted. Recorded: upper → "Dark Cloak"

2

AGENT **Select "Mystic Hood" for Head**

⚠ Conflict: "hood" + "cloak" = face likely hidden
Suggestion: choose open headwear (crown, tiara, circlet)

3

AGENT **Select "Shadow Crown" for Head instead**

✓ No conflicts with "Dark Cloak". Accepted.

CHECKPOINT 2 — BEFORE AVATAR CREATION

4

AGENT **Calls "Create Outfit"**

Server runs global check across all selected slots...

`head: "Star Crown"` `upper: "Star Dress"` `hair: EMPTY` `feet: "Fluffy Boots"`

⚠ Missing essential: no hair selected
"Are you sure you want to render without hair?"
Suggestion: search for hair items matching theme

No hair = not ready to render

All slots filled = ready

5

AGENT **Adds "Pastel Candy Twintails" for Hair**

✓ hair → "Pastel Candy Twintails". All slots filled.

6

SERVER **Global Check Passes — Creating Outfit**

`head: "Star Crown"` `upper: "Star Dress"` `lower: "Royal Egyptian Wraps"` `feet: "Fluffy Boots"`

✓ No cross-category conflicts
✓ All essential slots filled
✓ Rendering avatar...

ROBLOX

Validation with Feedback

Validation with Feedback: Ending Blind Retries

RENDER #1

AGENT ASSUMES (NO FEEDBACK ON WHAT'S WRONG)

"Score is low... maybe it's the top? I'll try swapping it."

`swap Upper Body` → randomly picks a different item

ONLY FEEDBACK

score: 3.2
pass: false

~~no diagnosis~~
~~no fix suggestion~~

| *re-render (~5s + GPU cost)*

RENDER #2

AGENT ASSUMES (STILL NO FEEDBACK)

"Score went up a bit... maybe more accessories will help?"

`swap Head + Feet` → guesses more pieces might help

ONLY FEEDBACK

score: 4.1
pass: false

~~no diagnosis~~
~~no fix suggestion~~

▲ lucky

| *re-render (~5s + GPU cost)*

RENDER #3

AGENT ASSUMES (NO IDEA WHY SCORE DROPPED)

"Maybe layering more pieces will look cooler?"

`add layers + accessories` → piles on conflicting items, makes it worse

ONLY FEEDBACK

score: 2.8
pass: false

~~no diagnosis~~
~~no fix suggestion~~

▼ worse!

| *budget exhausted — returns last attempt, not best*

Blind retries — no feedback, no convergence

The agent knows something is wrong but not **what** or **why**. Each retry is a guess. Quality bounces randomly. Worse — without best-of-N tracking, the final result may be **worse** than attempt #2.

Validation with Feedback: Ending Blind Retries

"Try again" isn't feedback.

"The face is hidden by the hood, search for open headwear" is.

- Render the avatar → evaluate from multiple angles
- Structured scoring with weighted criteria
- Domain knowledge encoded in what matters most

Validation with Feedback: Guided Fixes

- Analyze what failed → suggest specific fixes
- Not "try again", provide ready-to-use search queries
- "Remove the hood → search for 'space explorer open face headwear'"
- Targeted suggestions → convergent improvement

Validation with Feedback: The Complete Loop

- Render → Score → Fix → Retry (up to N attempts)
- Best-of-N tracking
  - keep the best, not the last
- Retry budget
  - prevents infinite loops

Validation with Feedback: A Loop in Practice

VALIDATE-AND-RETRY LOOP

Render Avatar
Capture from multiple angles: front, side, back

Score with Weighted Criteria
VLM evaluates each angle against domain criteria

Face visibility		0.2	40%
Theme match		0.85	25%
Coherence		0.60	20%
Coverage		0.90	15%

Pass / Fail Gate
Critical failure (face hidden) overrides overall score

❌ Fail → diagnose & fix

✓ Pass → return best

Diagnose & Suggest Fix
Problem: Face hidden by "Mystic Hood"
Fix: Remove hood → search `"open face headwear egyptian"`

Agent Swaps Item
Follows guided suggestion, replaces problematic item

↺ retry from render

BEST-OF-N TRACKER

Attempt #1 0.42 fail

Style mismatch: maid dress + pink stockings, wrong theme

Fix: swap upper → "dark streetwear top"

Attempt #2 0.71 fail

Over-accessorized: spiked armor + chains clutter the look

Fix: remove shoulder armor → simplify

Attempt #3 0.87 ★ best

Clean dark streetwear. Cohesive silhouette, accessories complement.

Attempt #4 0.79 ↓ worse

Over-layered: ribbon shirt under jacket breaks coherence

Retry Budget

4 / 5 used — returning attempt #3 (best)

Recap: From Naive Agent to Reliable Workflow

Problem	Pattern	General Application
Amnesia	Session Memory	Content Curation, document assembly
The Round-Trip Trap	Composite Tools	Fetch + transform, query + enrich
Expensive Mistakes	Pre-flight validation	Schema validation, dependency checks
Blind Retries	Validation with Feedback	Quality checks + targeted fixes

Thank you.

ROBLOX