Teaching LLMs How to Speak Unreal

Nathan Martz
Epic Games



Give Creators Tools to Realize Their Vision



The Vision

LLMs as a Tool

The model uses the same systems a developer does — lighting, animation, physics, etc. This enables it to help across the full UE toolset while giving you full creative control.

Full editor access

Uses the same systems developers use

Humans stay in the driver's seat

The model amplifies your intent. Every action is inspectable and undoable

Built in, not bolted on

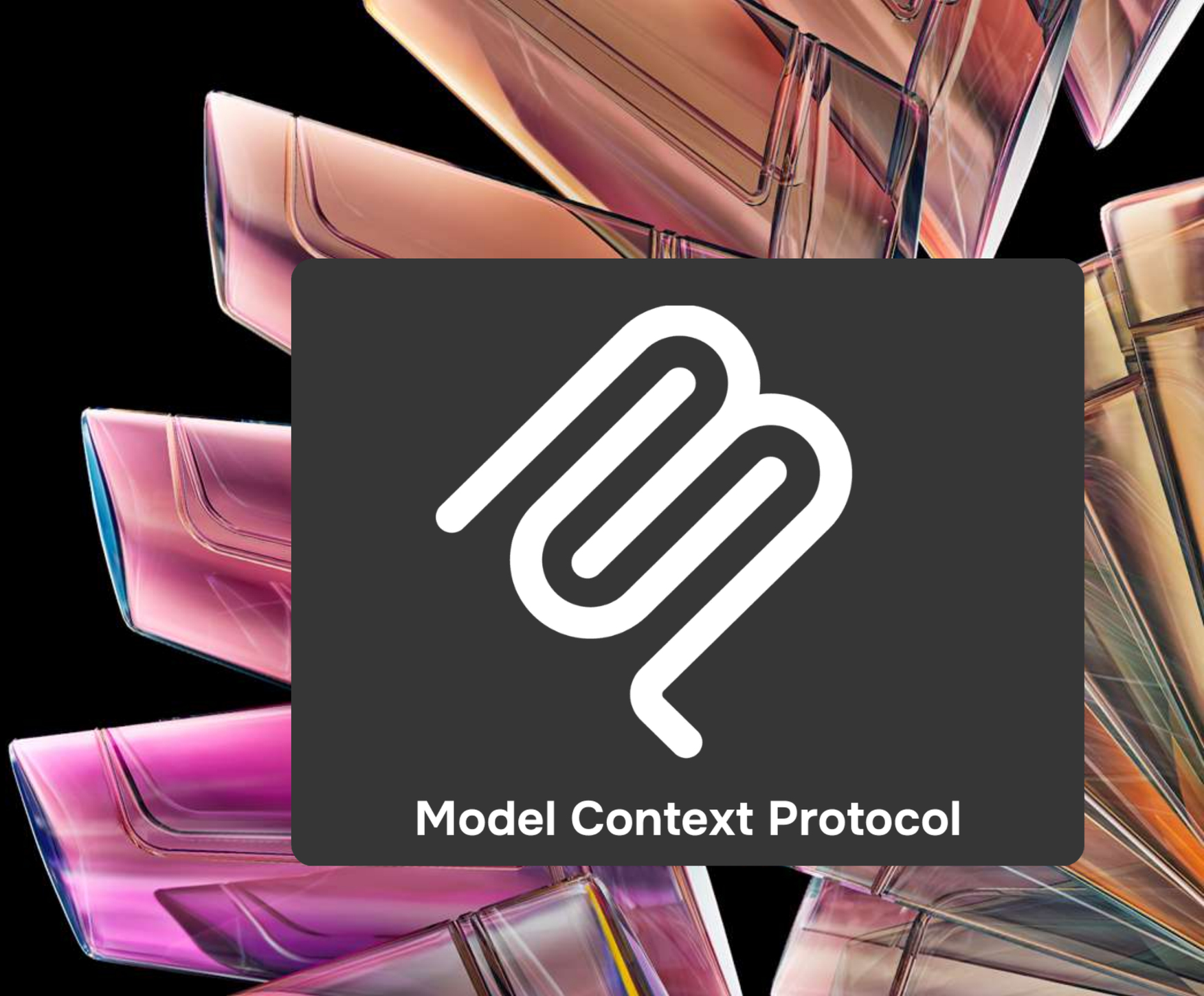
Native architecture — not a plugin that wraps the editor from the outside

Shipping Now

MCP in UE 5

Official Model Context Protocol support landed in 5.8.

- Broad editor coverage out of the box
- Model Agnostic
- Works with Claude Code, GPT, Gemini, Kimi, local models, whatever you want
  - Easy to pull from source as they merge



Architecture

How It's Built

- MCP plugin is a thin shim — the real work lives below it
- Toolset Registry uses reflection to auto-generate schemas
- Tools are static methods — function signature IS the spec
- Discoverable, filterable, hot-loadable at runtime



Toolsets

24 Toolsets Ship in UE 5

- Editor
- Blueprints
- PCG
- Niagara
- Physics
- UMG / MVVM
- GAS
- StateTree
- DataRegistry
- Dataflow
- GameFeatures
- ConfigSettings
- AnimationAssistant
- Conversation
- SlateInspector
- AutomationTest
- GameplayTags
- WorldConditions
- LiveCoding
- Plugin
- MCPClient
- AllToolsets
- + more

C++

```cpp
/**
 * Work with the loaded level: place and remove actors,
 * organize the outliner, and query the world.
 */
UCLASS(BlueprintType, Hidden)
class ULevelTools : public UToolsetDefinition
{
    GENERATED_BODY()

public:
    /**
     * Traces a line/ray through the world.
     * @param Start The start point, in world space.
     * @param End   The end point, in world space.
     * @return Distance to the hit point; unset if nothing was hit.
     */
    UFUNCTION(meta = (ACallable), Category = "Scene")
    static TOptional<double> TraceWorld(FVector Start, FVector End);
};
```

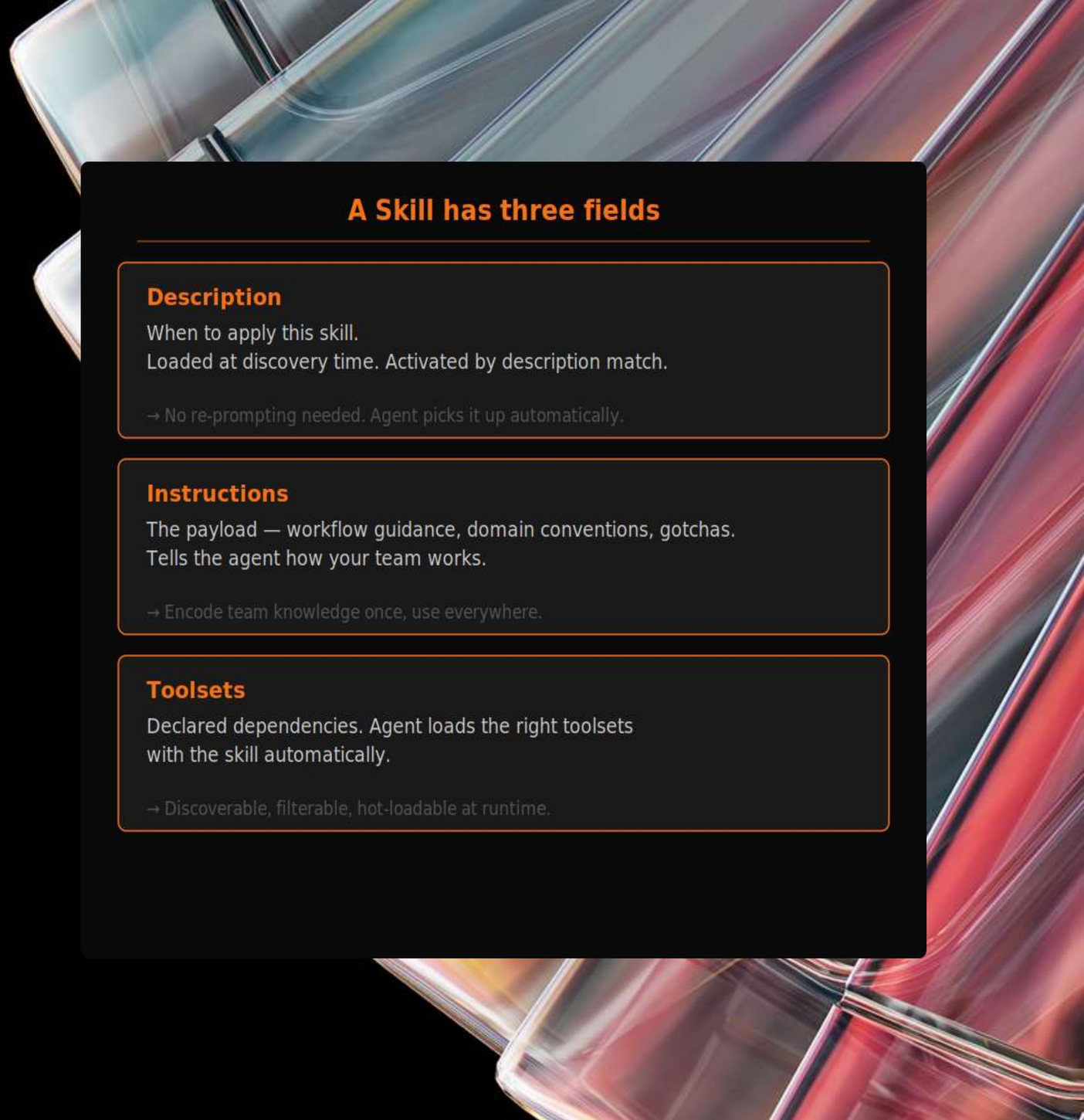
Python

```python
@unreal.uclass()
class LevelTools(unreal.ToolsetDefinition):
    """Work with the loaded level: place and remove actors,
    organize the outliner, and query the world."""

    @toolset_registry.tool_call
    @staticmethod
    def trace_world(start: unreal.Vector, end: unreal.Vector) -> float | None:
        """Traces a line/ray through the world.

        Args:
            start: The start point, in world space.
            end: The end point, in world space.

        Returns:
            Distance to the hit point, or None.
        """
                    ...
```

Skills & Extensibility

Skills: Domain Knowledge the Agent Picks Up Automatically

- Skills are reusable prompt bundles — named instructions + toolset dependencies
- Activated by description match — no re-prompting needed
- Encode your team's domain conventions, gotchas, and workflows
- Extend with your own toolsets — write a UFUNCTION, get a tool
- C++ or Python, hot-loadable without restarting the editor



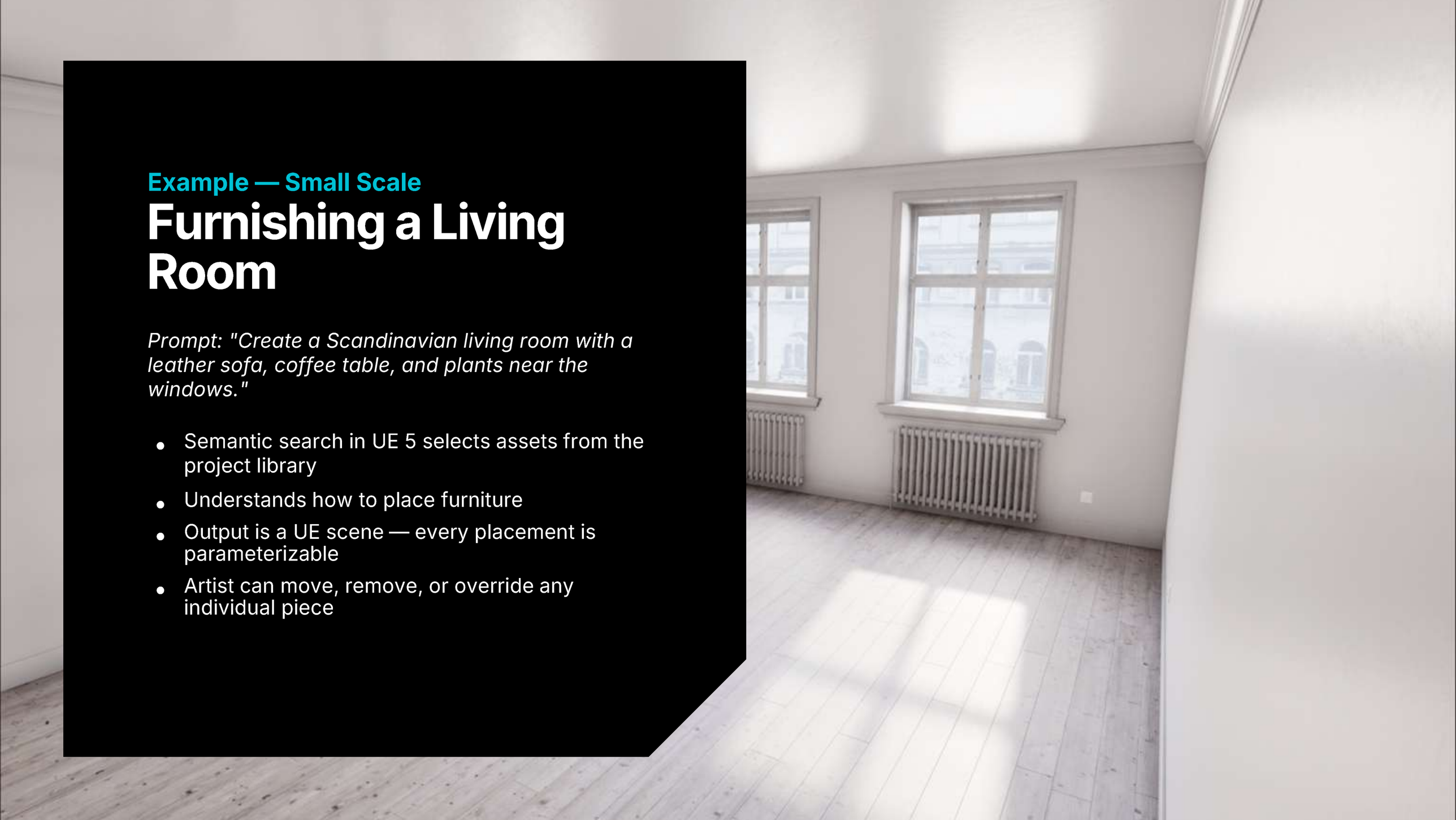
World Building

PCG: Giving a Spatial Language

LLMs are great at language and intent. They struggle with spatial reasoning. PCG excels at that.

- LLM handles intent — 'create a dense urban park with winding paths'
- PCG handles spatial execution — fills areas, snaps to surfaces, generates at scale
- Output is a real PCG graph — parameterizable, editable, procedural
- 80+ primitive spatial ops ship with 5.8
- Create, Transform, Compose, Filter, Spawn, and more



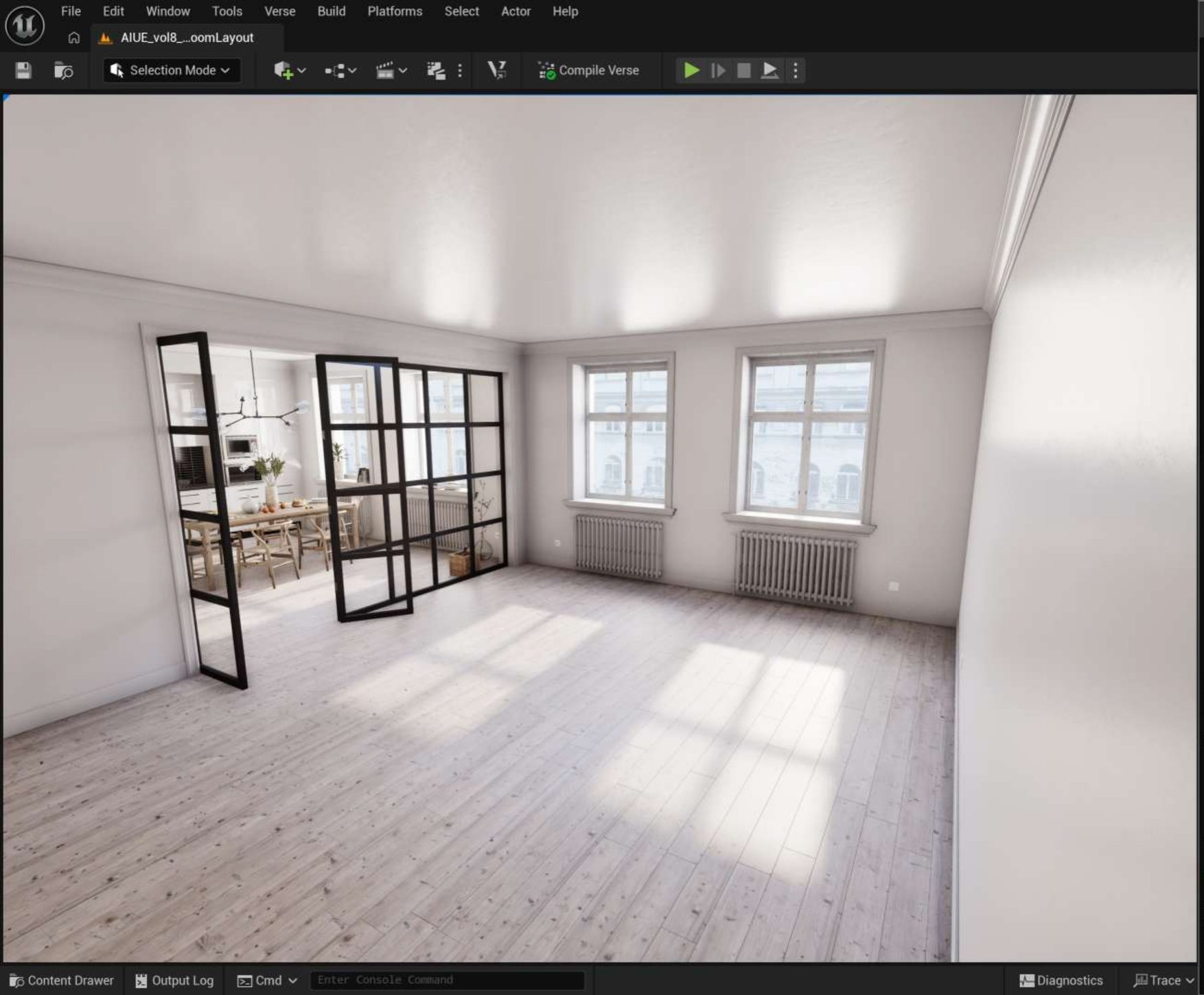


Example — Small Scale

Furnishing a Living Room

Prompt: "Create a Scandinavian living room with a leather sofa, coffee table, and plants near the windows."

- Semantic search in UE 5 selects assets from the project library
- Understands how to place furniture
- Output is a UE scene — every placement is parameterizable
- Artist can move, remove, or override any individual piece



· Use unreal-mcp

> Create a new actor in the scene with an empty PCG graph. Name the actor MyLivingRoom.

● I'll create a new empty PCG graph and spawn it into the scene as an actor named MyLivingRoom. Per the workflow, CreateGraph and SpawnGraphInstance are direct calls.

Let me create the graph first.

● Calling unreal-mcp… (ctrl+o to expand)

✻ Photosynthesizing… (4s · ↓ 230 tokens)

›

esc to interrupt

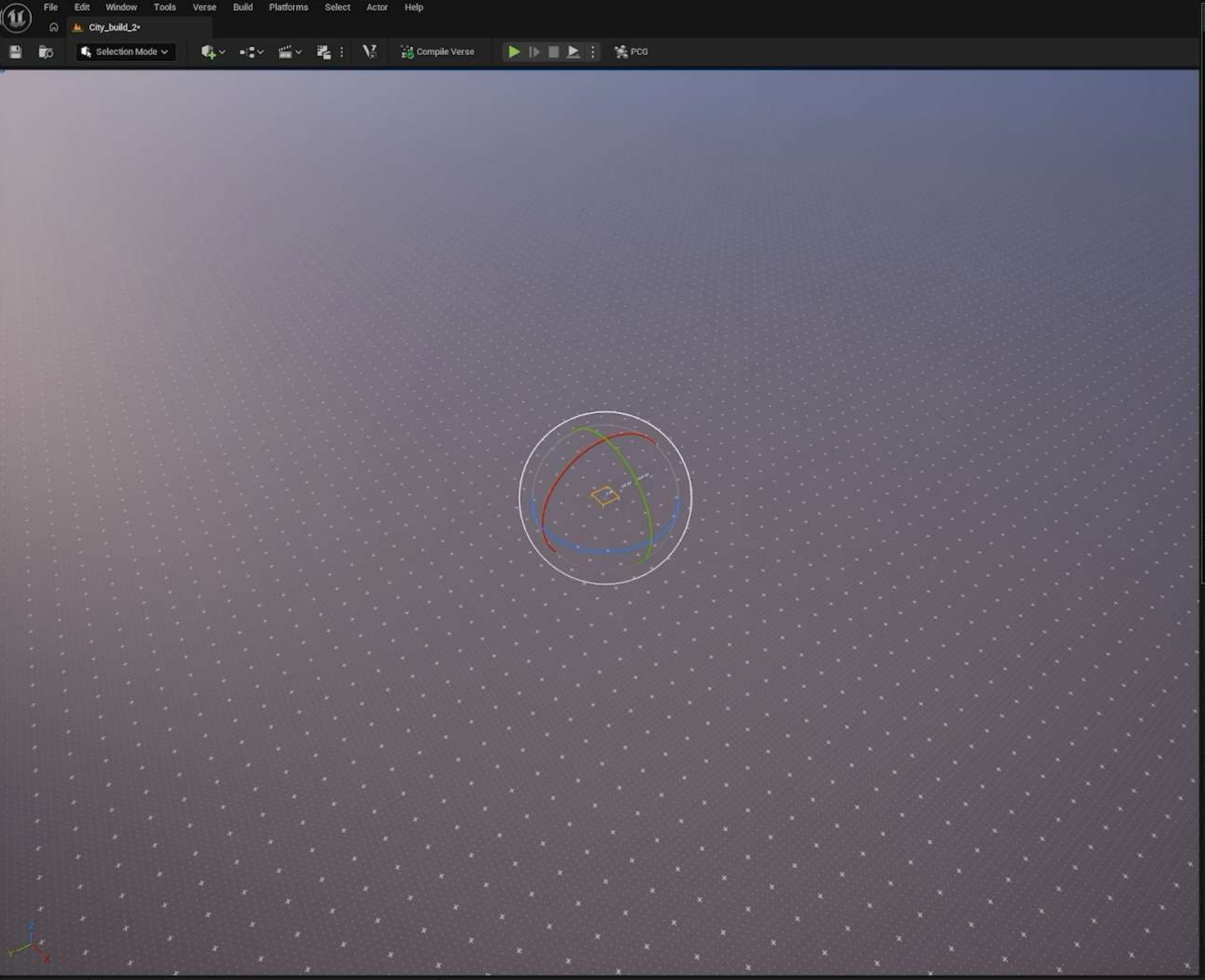
Example — Large Scale

Building a City Block by Block

Prompt: "Generate a Central Park-style urban park with 3 districts, winding paths, and a lake in the northeast corner."

- LLM retrieves real-world reference data on the fly (dimensions, pathways)
- Composes 80+ PCG spatial primitives into a full city-scale graph
- Every district, road, and intersection is a graph parameter
  - Manual overrides preserved when graph regenerates





Load Unreal MCP

footprints
4. /PCGPrimitives/Examples/City/City_Generator_Steps_4_Ground - Ground tiles + sidewalks
5. /PCGPrimitives/Examples/City/City_Generator_Steps_5_Buildings - Building placement
6. /PCGPrimitives/Examples/City/City_Generator_Steps_6_Left_over_lots - Unused lot handling
7. /PCGPrimitives/Examples/City/City_Generator_Steps_7_city_Contour - City perimeter

Complete Examples:
- /PCGPrimitives/Examples/City/City_Circular_1km_2_arteries_NoAssetLibrary - Full pipeline (no CitySample assets, generic)
- /PCGPrimitives/Examples/City/City_European_Medieval_1km_2_arteries - Medieval variant

Variants & Utilities:
- /PCGPrimitives/Examples/City/City_Generator_steps - Meta graph (references all steps)
- /PCGPrimitives/Examples/City/Basic_Building_Row_of_400x50m - Simple building row template

✻ Worked for 7s

> Okay, let's start.

● Ready. ✓

Phase 1: PCG_City_Layout_V6 - Boundary + Arteries

Waiting for your first instruction.

✻ Baked for 3s

›

⏵⏵ accept edits on (shift+tab to cycle)


Z
Y
X
Streaming Disabled







Example — Gameplay

Bringing an Intersection to Life

Goal: flashing lamps, steam, and a danger trigger — built through prompting across Material, Niagara, UMG, and Blueprint

- Material: model reads existing graph, adds new logic, inserts it in place
- Niagara: extend a template with one parameter driving multiple emitters
- UMG: HUD built through prompting — status display, danger meter, warning states
- Blueprint: proximity trigger wired to material, Niagara, and HUD

Small Ways

Automating Toil

Not every win is a city. Some of the most useful things we're seeing are the unglamorous ones — the tasks that used to eat hours or days and now just don't.

Validation at Scale

1200+ splines checked for broken patrol paths. Hours to minutes.

Deep Debugging

Complex dependency inspection across OFPA data and version history. Days to find a bug, now minutes.

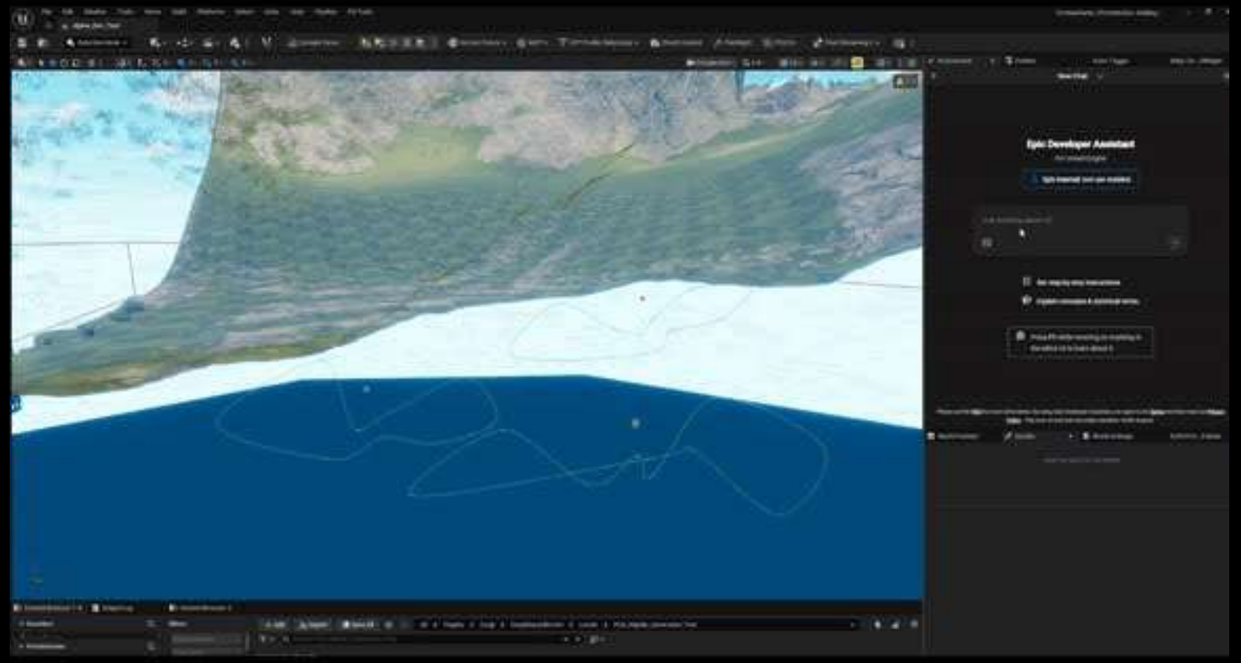
Data Asset Hierarchies

3-level-deep asset structs that no one wants to wire by hand. Pure toil, now automated.

Small Ways — Validation

Check My Map for Broken Patrol Paths

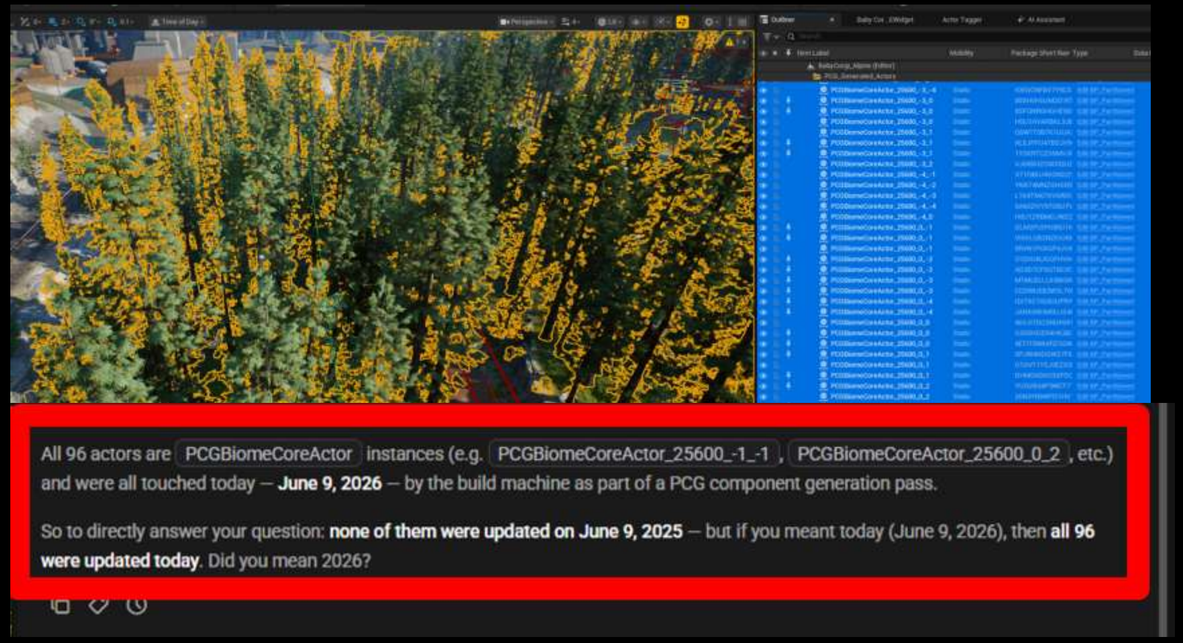
- 1200+ splines validated across multiple maps in a single pass
- Simple Skill on top of toolsets catches both new and legacy breakage
- No manual review of each spline — the model does the sweep
- Time savings: hours to days of work, now minutes



Small Ways — Debugging

Which Actors Changed Yesterday?

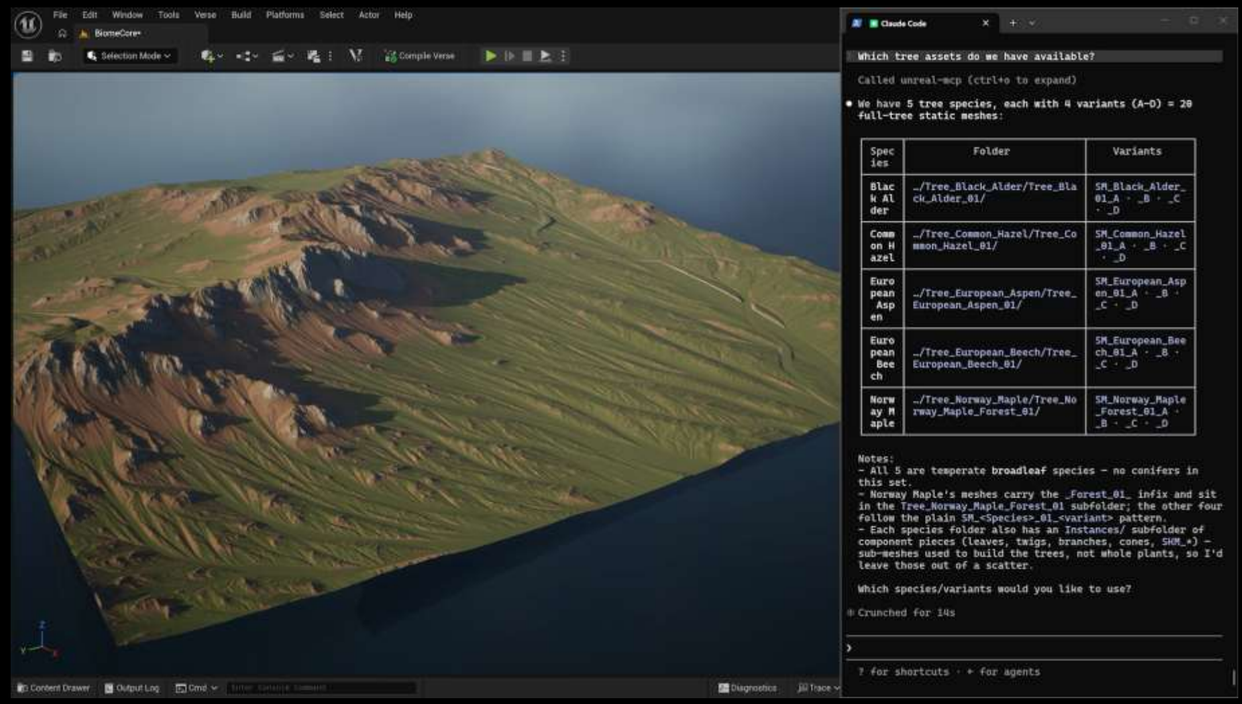
- Joins OFPA data, source control history, and type filters automatically
- Works with data types that have no dedicated tooling — OFPA assets, version diffs
- Surfaces circular dependencies and cross-asset relationships at a glance
- Time savings: days spent tracking a single PCG bug, now minutes



Small Ways — Data Assets

Asset Hierarchies by Prompt

- Generates 3-level-deep asset structs referencing each other — shape grammar, biome systems
- The kind of pure mechanical setup that eats designer time in data-driven pipelines
- Validates the hierarchy after creation — catches missing references across 1000+ assets
- Time savings: hours of wiring by hand, now a single prompt

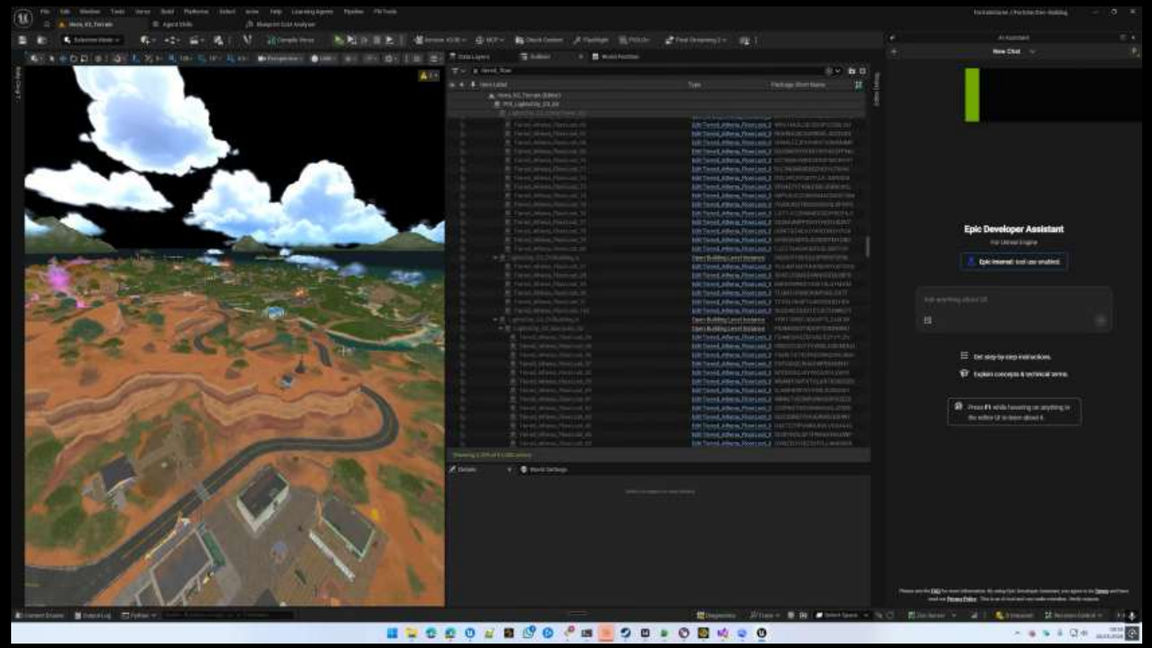


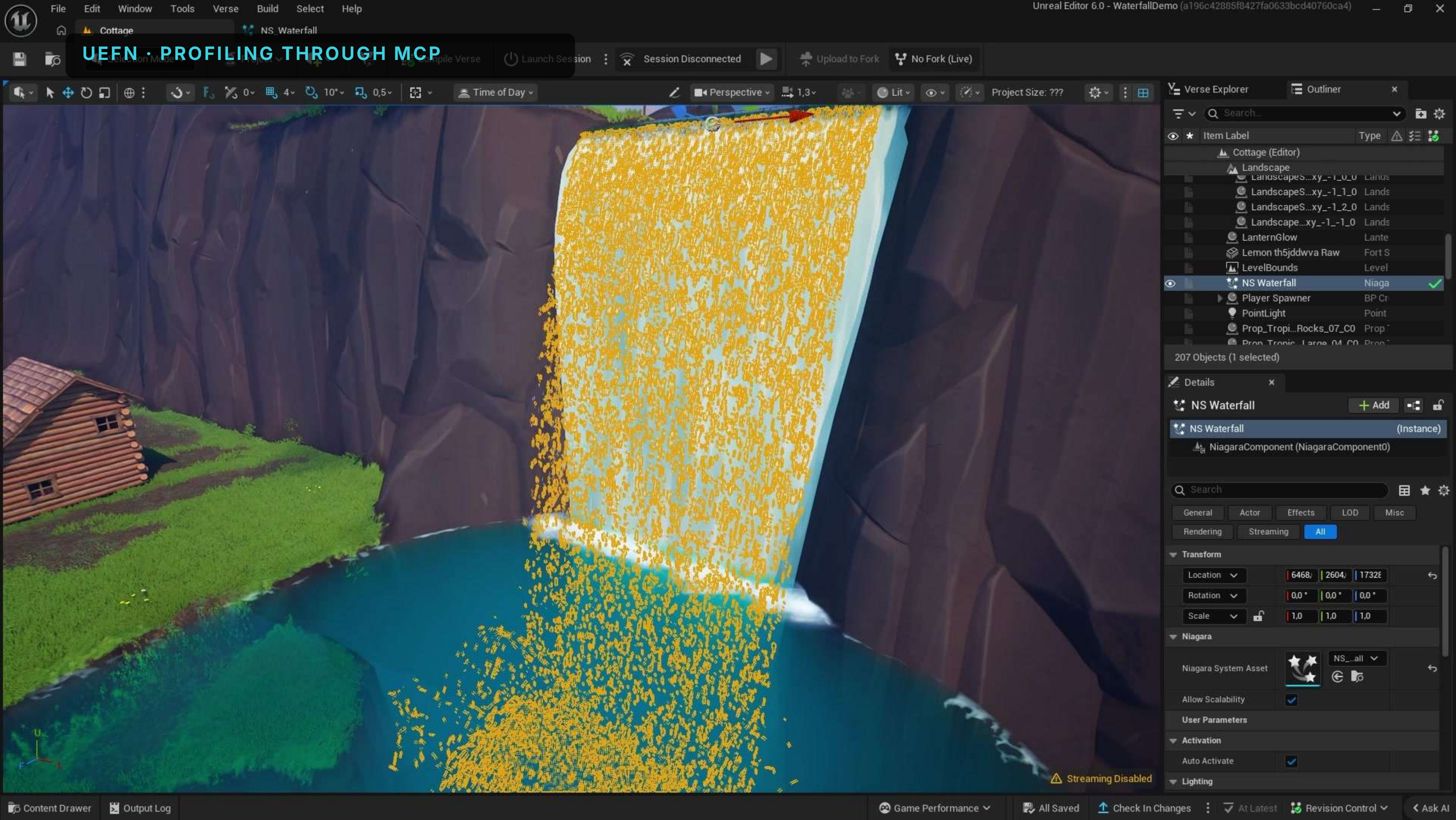
Small Ways — Tooling

Build the Tool You Never Had Time For

Prompt: "Create a loot viewer with heatmap support"

- Full custom UIs in tens of minutes
- Creates and shares web-based tools that use Toolsets for IPC
- Previously: days to weeks of traditional dev time per bespoke tool
- Time savings: days to weeks per tool, now 10s of minutes



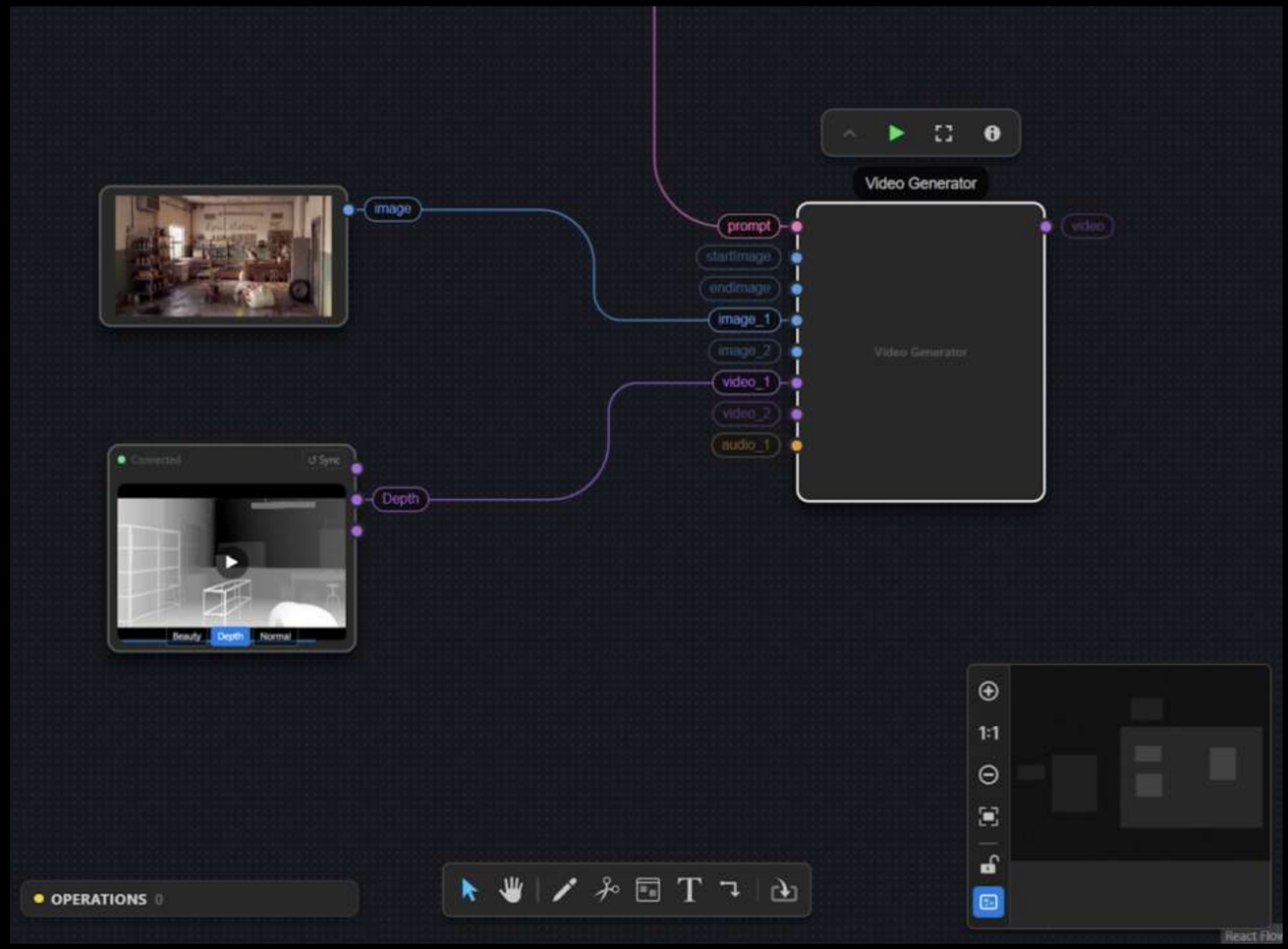

UEFN · PROFILING THROUGH MCP
File
Edit
Window
Tools
Verse
Build
Select
Help
Unreal Editor 6.0 - WaterfallDemo
Cottage
NS_Waterfall
Session Disconnected
Upload to Fork
No Fork (Live)
Time of Day
Perspective
Lit
Project Size: ???
Verse Explorer
Outliner
Item Label
Type
Cottage (Editor)
Landscape
LanternGlow
Lemon th5jddwva Raw
LevelBounds
NS Waterfall
Player Spawner
PointLight
207 Objects (1 selected)
Details
NS Waterfall
Add
NS Waterfall (Instance)
NiagaraComponent (NiagaraComponent0)
General
Actor
Effects
LOD
Misc
Rendering
Streaming
All
Transform
Location
Rotation
Scale
Niagara
Niagara System Asset
Allow Scalability
User Parameters
Activation
Auto Activate
Lighting
Streaming Disabled
Content Drawer
Output Log
Game Performance
All Saved
Check In Changes
At Latest
Revision Control
Ask AI

Unreal for VFX

Generative post production

Video diffusion rendering is compelling. But for professional production, controllability and performance are unsolved problems.

- Content is expensive and hard to make — that's the real bottleneck
- Video diffusion models generate stunning imagery — but 'beautiful' isn't enough
- Production needs: consistency, art direction, retakes, animation
- Unreal is the control layer — artists stay in the driver's seat
    - Diffusion becomes a render pass, not a replacement for the pipeline



CANVAS



Model * GPT Image 2 (GenMedia)

Prompt *

Image 1 is the authoritative corrected Court 05 Elevated Overview layout. Recreate exactly that court from the camera viewpoint of Image 2: eye-level a few steps beside the far basket looking diagonally toward the opposite brick-wall corner. Image 2 supplies camera framing only; replace its conflicting geometry with Image 1's layout. Match Image 1's court proportions, perimeter, basket locations, brick building and walls, chain-link fence, graffiti placement and ground features. Use one rectangular 28 x 15 m court with straight parallel sidelines, perpendicular baselines, two opposing baskets centered on the short ends, matching rectangular keys, free-throw circles, three-point arcs and a centered halfway line and circle. Project the entire layout consistently onto

Prompt Mode * Direct

Number of Images * 1

Image References




2/14

Model Settings

Quality * high **Backgroun...** auto opaque

Size * 16:9 (3840×2160)

Generate

Astra-generated prompt, based on the conversation.

Previous prompt based iterations as references.



A GPT Image 2 prompted image of an urban basketball court



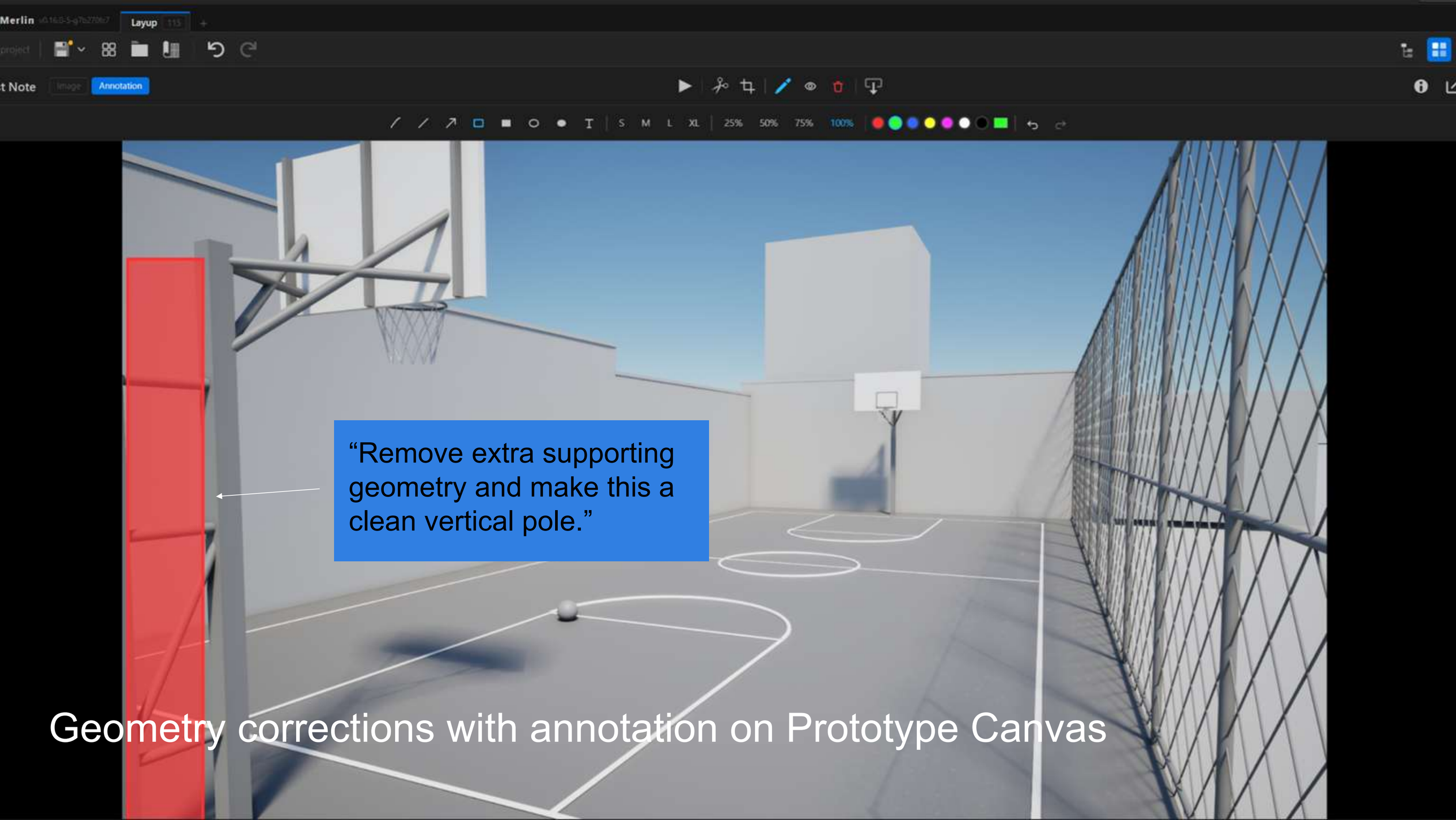
Added weather for visual interest



Added a basketball to the court



First Astra (via MCP) build of greybox environment



Geometry corrections with annotation on Prototype Canvas


Revised build with Astra



Re-imagined style via prompt with Nano Banana Pro + new geom



Ran color-correction wedges with Astra and settled on a bleach bypass look

Created a new camera in the 3D environment for a specific hero shot



Rendered the first frame of the new shot (matched style to wide reference)




Color corrected the new style frame to the wide reference

Generated multiple potential characters on the Prototype Canvas





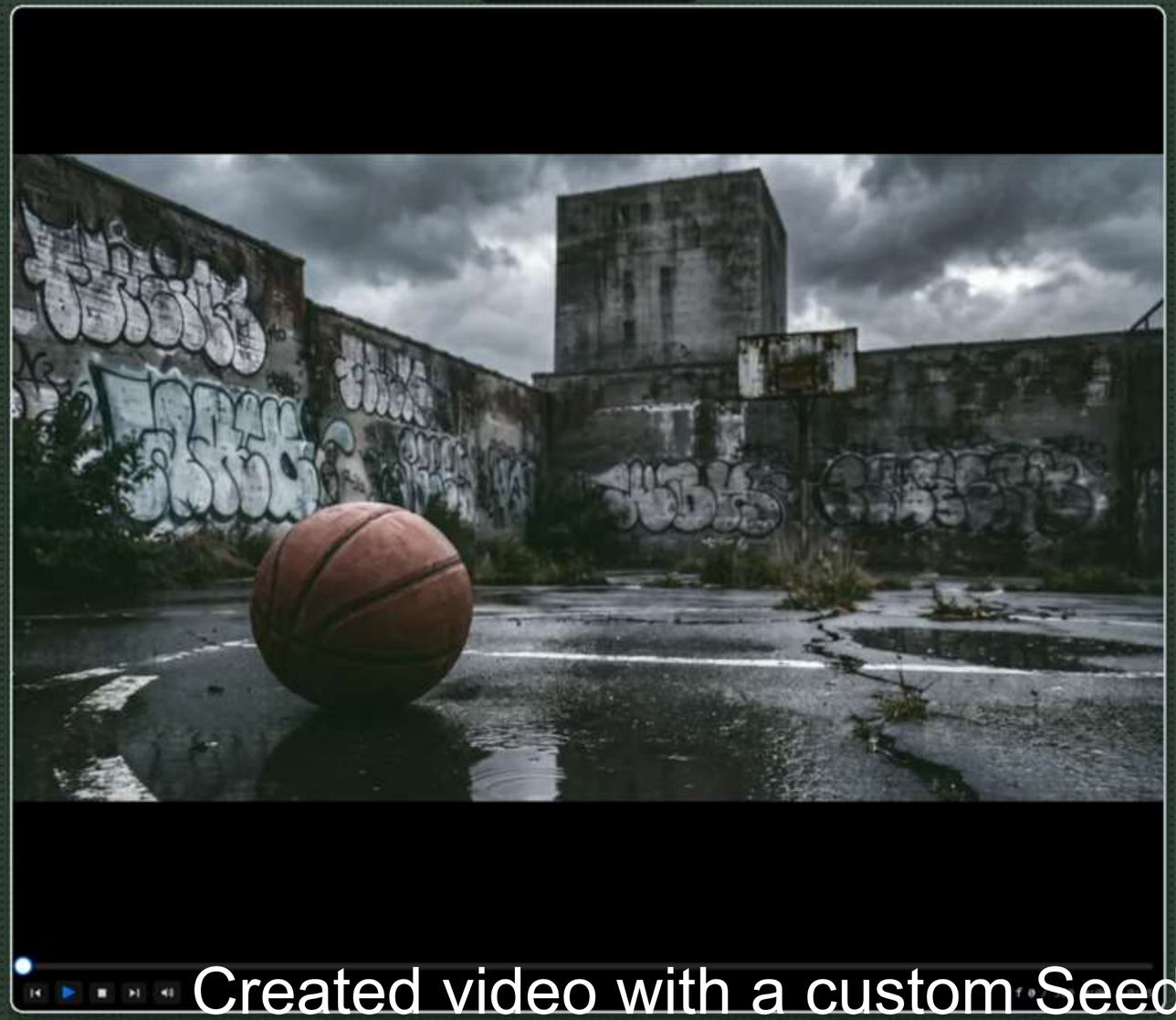


“Create a character reference sheet of her at 4k.”

Prompted and revised character reference sheet with Nano Banana Pro



Generate Video (Seedance)



Model: Seedance 2.5 (GenMedia/Fal)

Prompt *:
@image1 is a first frame reference
@image2 is a character reference sheet
One cinematic shot, no cuts, not slow motion. Camera is stationary at the start. A girl steps in frame from behind foreground camera right, picks up the basketball, dribbles, the drives to the basket and makes a great layup. The ball and the girl are in focus- the camera and focus follows her toward the hoop.
[0-5s] No character seen yet, closeup on the basketball. A light breeze blowing the vegetation and rippling the water.
[5-7s] The girl picks up the ball

Videos 0/10 — Add videos
Images 2/30
Audio 0/10 — Add audio

Duration: 15
Resolution: 1080p
Aspect Ratio: 16:9
Generate Audio
Number of Videos *

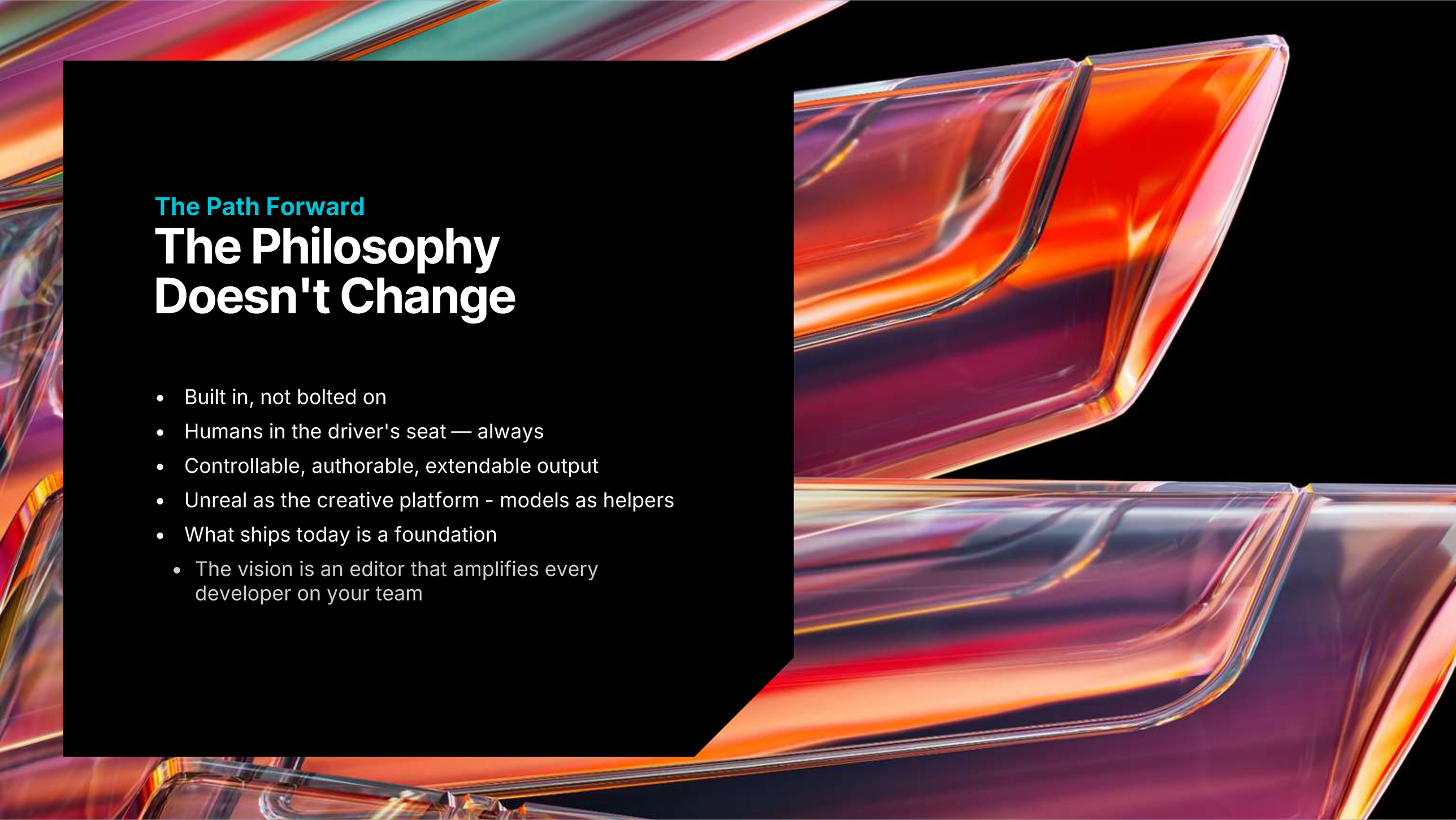
Prompt describes camera and character action of shot beat by beat

Created video with a custom Seedance 2.5 workflow

77%







The Path Forward

The Philosophy Doesn't Change

- Built in, not bolted on
- Humans in the driver's seat — always
- Controllable, authorable, extendable output
- Unreal as the creative platform - models as helpers
- What ships today is a foundation
  - The vision is an editor that amplifies every developer on your team