



# Modernizing Games With NVIDIA RTX Remix and Generative AI

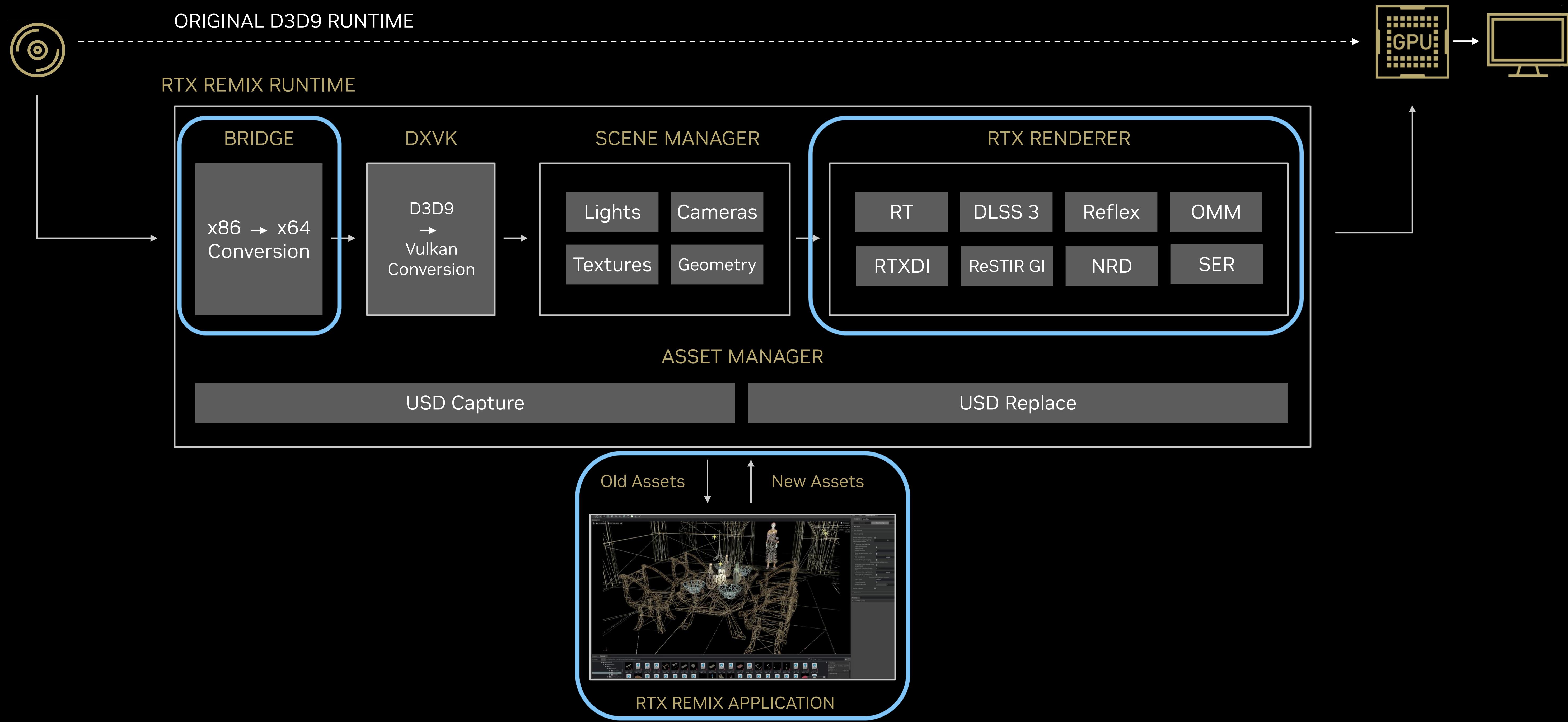
Damien Bataille, Pierre-Olivier Trottier | GTC 2024



# Agenda

- Introduction
- RTX Remix Overview Pierre-Olivier
- RTX Remix Toolkit Damien
- Generative AI Tools Pierre-Olivier
- Questions

# RTX Remix Overview



**Learn More:**

[https://docs.omniverse.nvidia.com/kit/docs/rtx\\_remix/latest/docs/remix-overview.html](https://docs.omniverse.nvidia.com/kit/docs/rtx_remix/latest/docs/remix-overview.html)

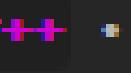


# RTX Remix Toolkit

# Making Content Malleable

The challenges of modding games today

- Each game has its own way to extract and ingest assets
  - Official modding tools are rarely available to the public
  - Gaming communities must invent numerous modding tools to handle each aspect of a game's graphics
  - Many games either lack modding tools or have complex workflows where modders juggle many tools/file formats
- Example, tools that can be used to mod Portal:

Third Party Source SDK tools	
Map editors	 Hammer++ ·  QuArK ·  Radiant ·  Sledge
Map compilers	Batch Compile Tool · Batch Compiler · Compile Pal · CSTBSP · CSTRAD · CSTVIS · Hazard Team Compile Tool · VBSP_h4xed · Zoner's Half-Life Tools → Vluzacn's Half-Life Tools
Map converters	BSP2MAP → WinBSPC · BSPSource · MapFool · VMEX
Map tools	AutoBSPpackingTool · BSP Viewer → Crafty · Entspy · Map Analyst · Material Enumerator · Map Obfuscator · Pakrat · Source Compile Analyzer · TAR · Vgroup · VIDE · WinBSPZip
Model compilers	GUIStudioMDL · Source Model Compiler
Model converters	 Crowbar · FireSoft Half-Life MDL · FireSoft MS3D to SMD · MDLDecompiler · Propper · Studiocompiler · VMF to SMD
Model tools	3Dcoat · 3DS Max ·  Blender · Cinema 4D · fragMOTION · GMax · kHED · Lightwave · Maya · MilkShape 3D · Modo · SourceOps · VSIF2VCD · Wings3D · XSI Mod Tool · Zbrush
Displacement tools	DispGen · Twister · Worldmachine
Particle converters	Sparc
Texture converters	360g · FixVTF · TGAtoDUDV · VTF Creator · VTFVer
Texture tools	GIMP · HL2Tex · Sign Maker · Sky Writer · Source model skin editor · Terragen · VMT Editor · VTF Explorer · VTF Shell Extensions · VTFCmd · VTFEdit · VTFTool
Sound tools	L4D2 Sound Mod Creator
VPK tools	GCFscape · Gibbed's VPK Extractor
Libraries	AVIKit · HLLib · VectorLib · VTFLib
Plugins	3DSMax tools · 3DSMax VTF · Blender Source Tools · Blender VertexLitGeneric · MESA · Notepad++ VDF languages · Photoshop VTF · SourceIO · SourceMod · Wall Worm Model Tools
Other	BEE2 · CtxConverter · Custom Compiler GUI · DuctTape · SSDK Windows Gadget · SteamCMDui ·  TeamSpen's Hammer Addons · VGUI Localization · VirtualDub · XBLAH's Modding Tool

# One Tool for All

## RTX Remix Toolkit

- **We came up with an idea of creating “one tool for all”**
  - Modders will use the same tool for a variety of games
  - Based on Omniverse Kit
- **Omniverse Kit**
  - It works with Pixar USD, a format that we will use to remaster games.
  - Because of USD, data can be exchanged easily between DCCs



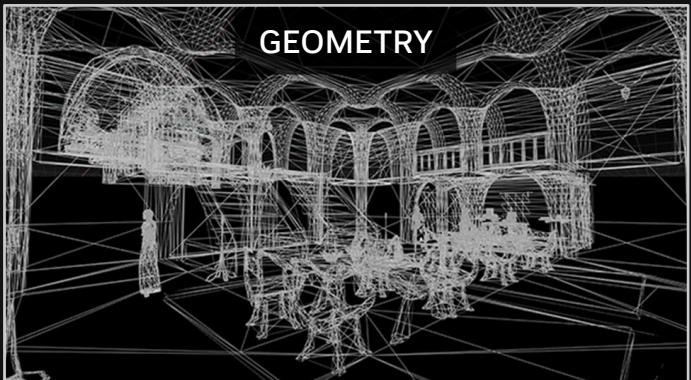
# Workflow

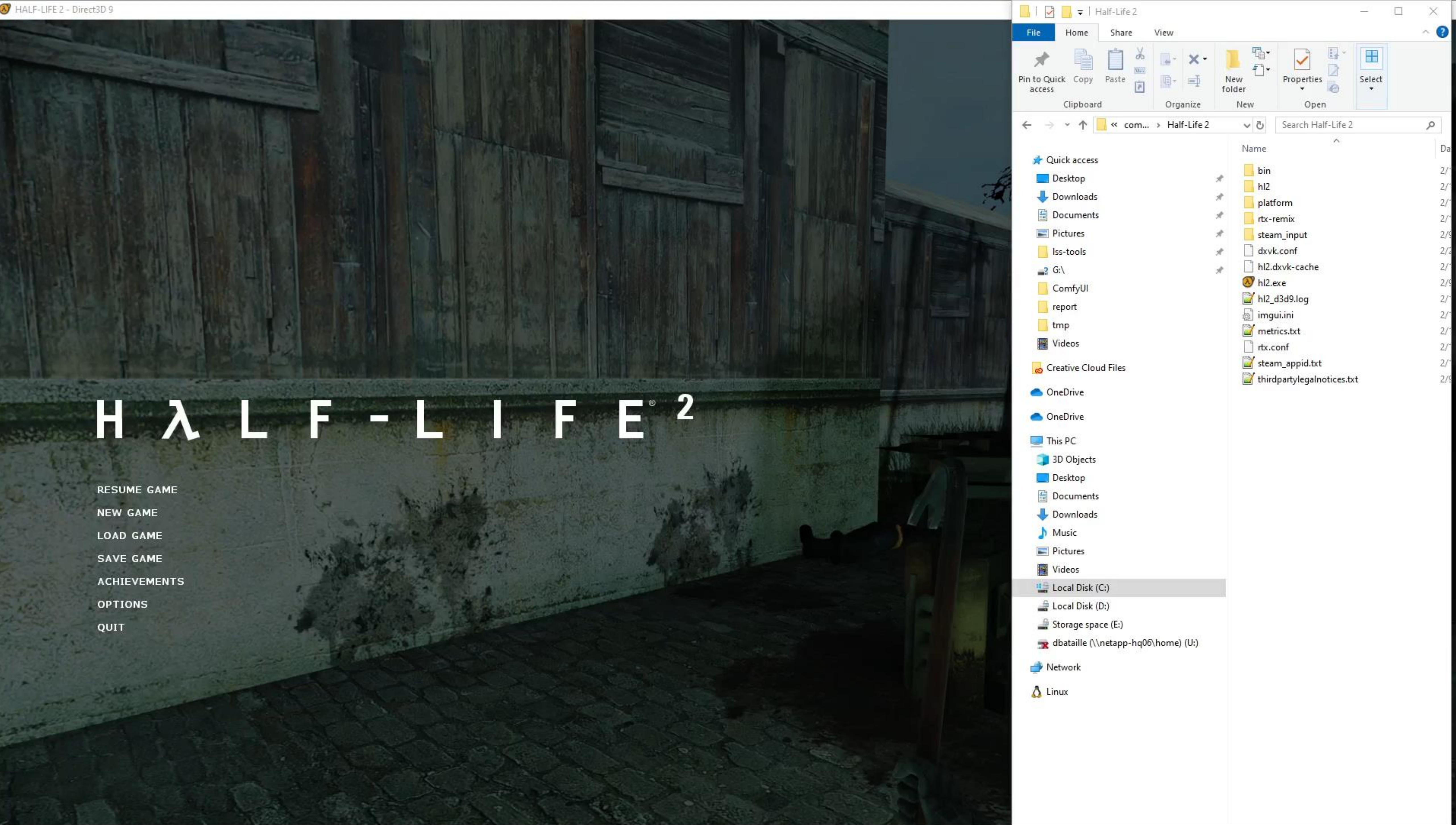
## Capture / Ingest / Replace

- Capture scene from a game, including textures, cameras, geometry and lights in OpenUSD format
- Import USD capture into RTX Remix Application
- Ingest user made asset(s) to convert them to USD
- Replace old asset(s) with new ones
- Leveraging the USD layering system: the base USD layer (capture) is replaced at runtime by the game



Game

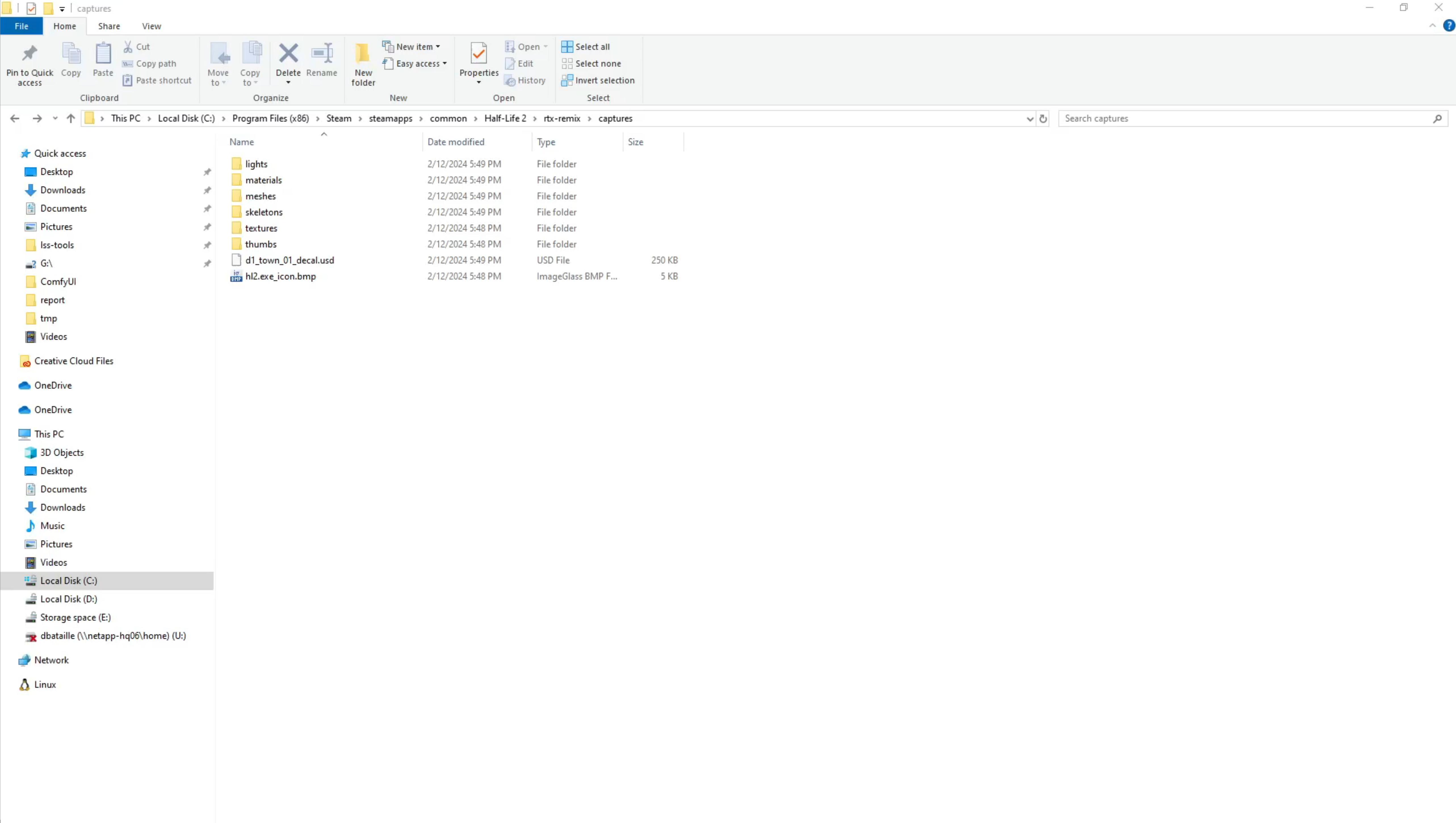




In-Game Capture

# RTX Remix Toolkit

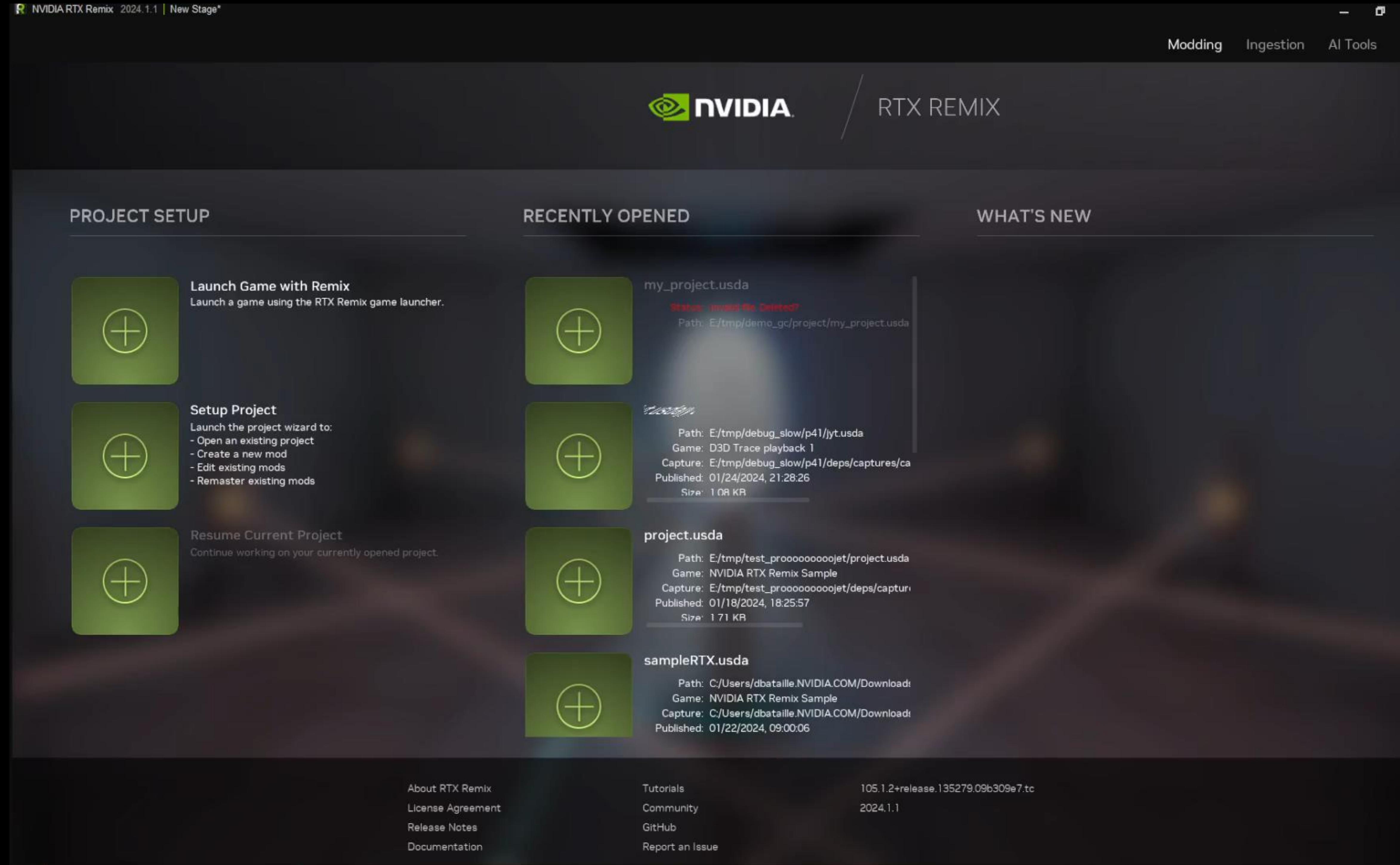
Omniverse Kit application to remaster games.



## Capture Files

# RTX Remix Toolkit

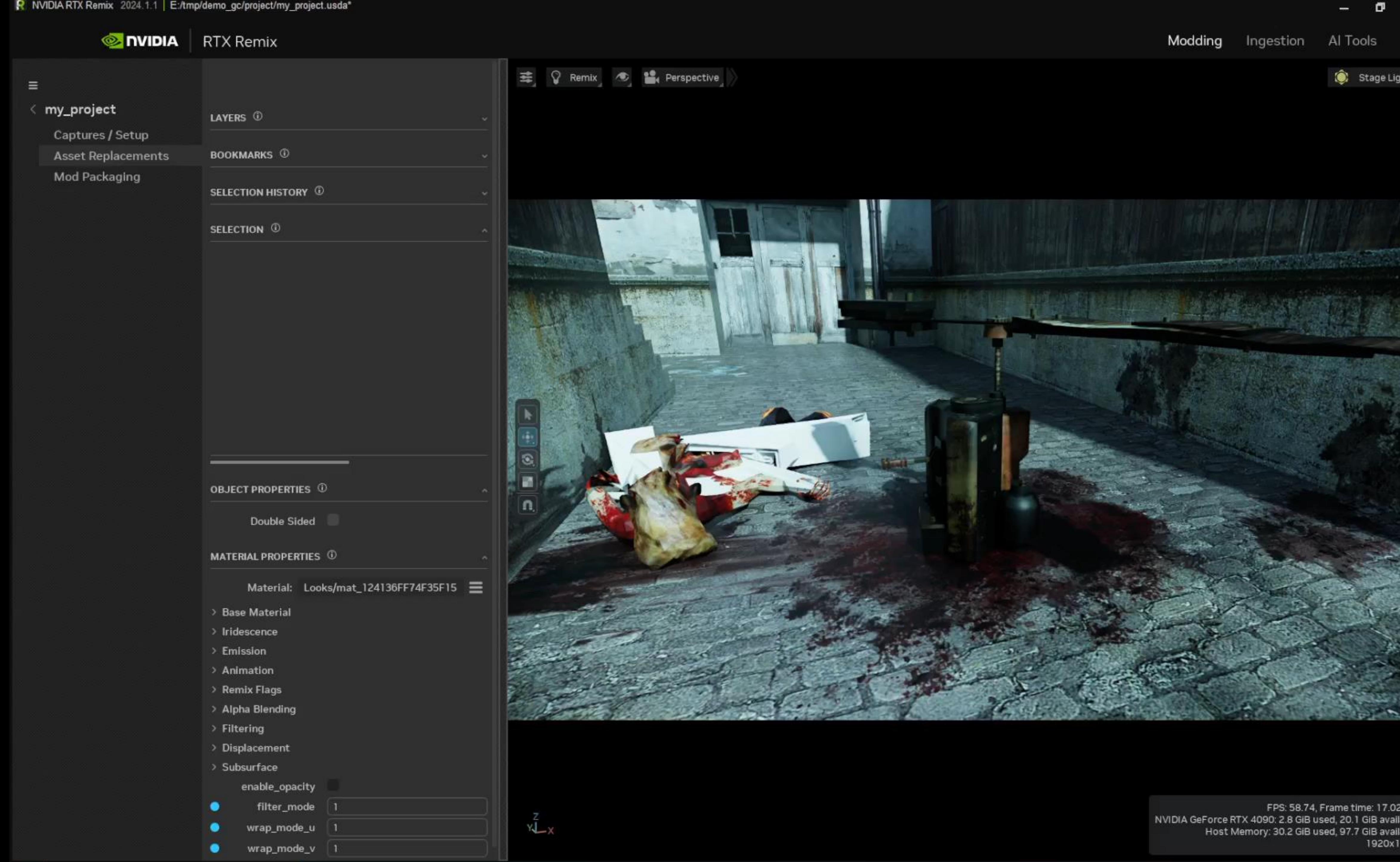
Omniverse Kit application to remaster games.



## Project Creation

# RTX Remix Toolkit

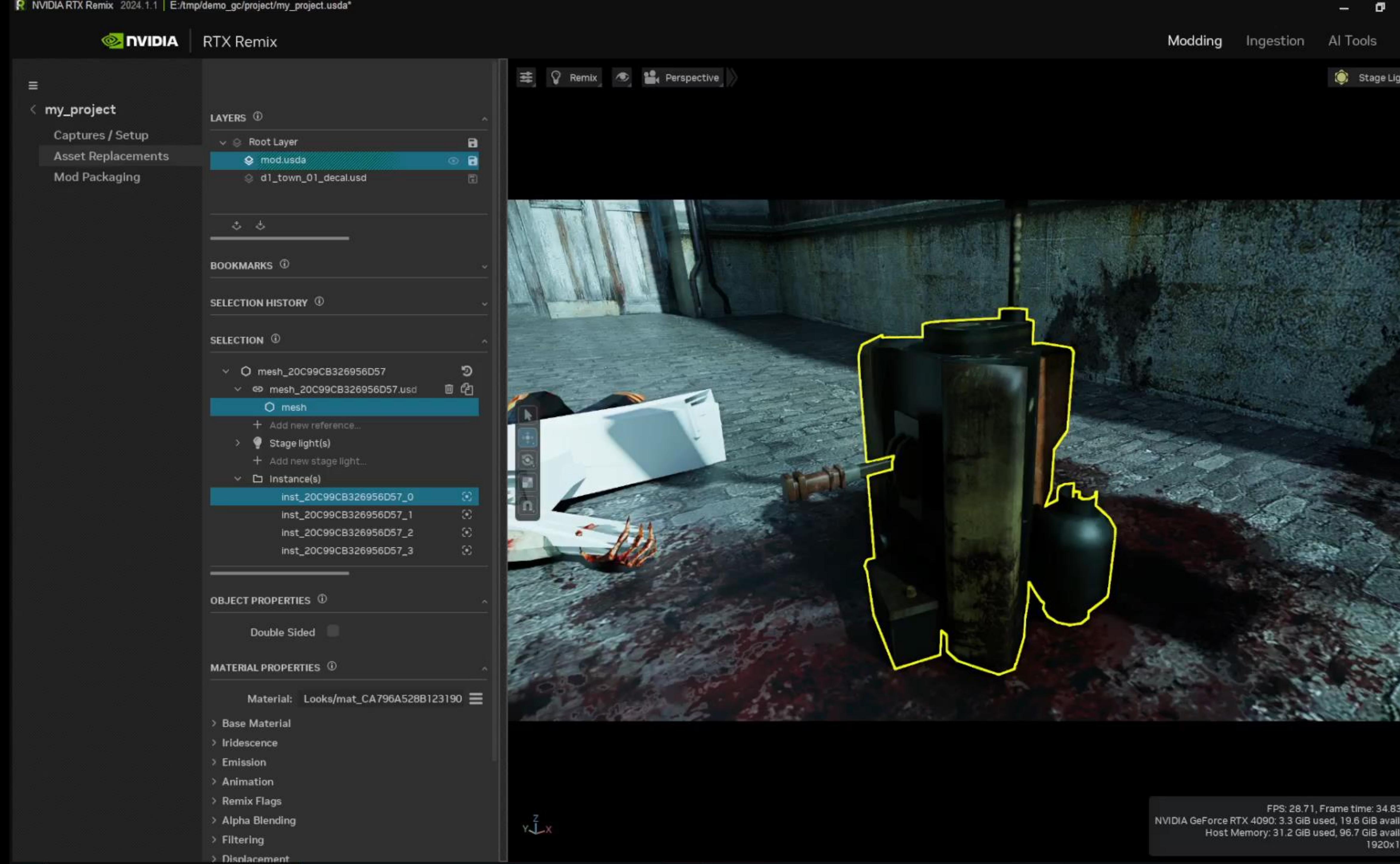
Omniverse Kit application to remaster games.



## Instances

# RTX Remix Toolkit

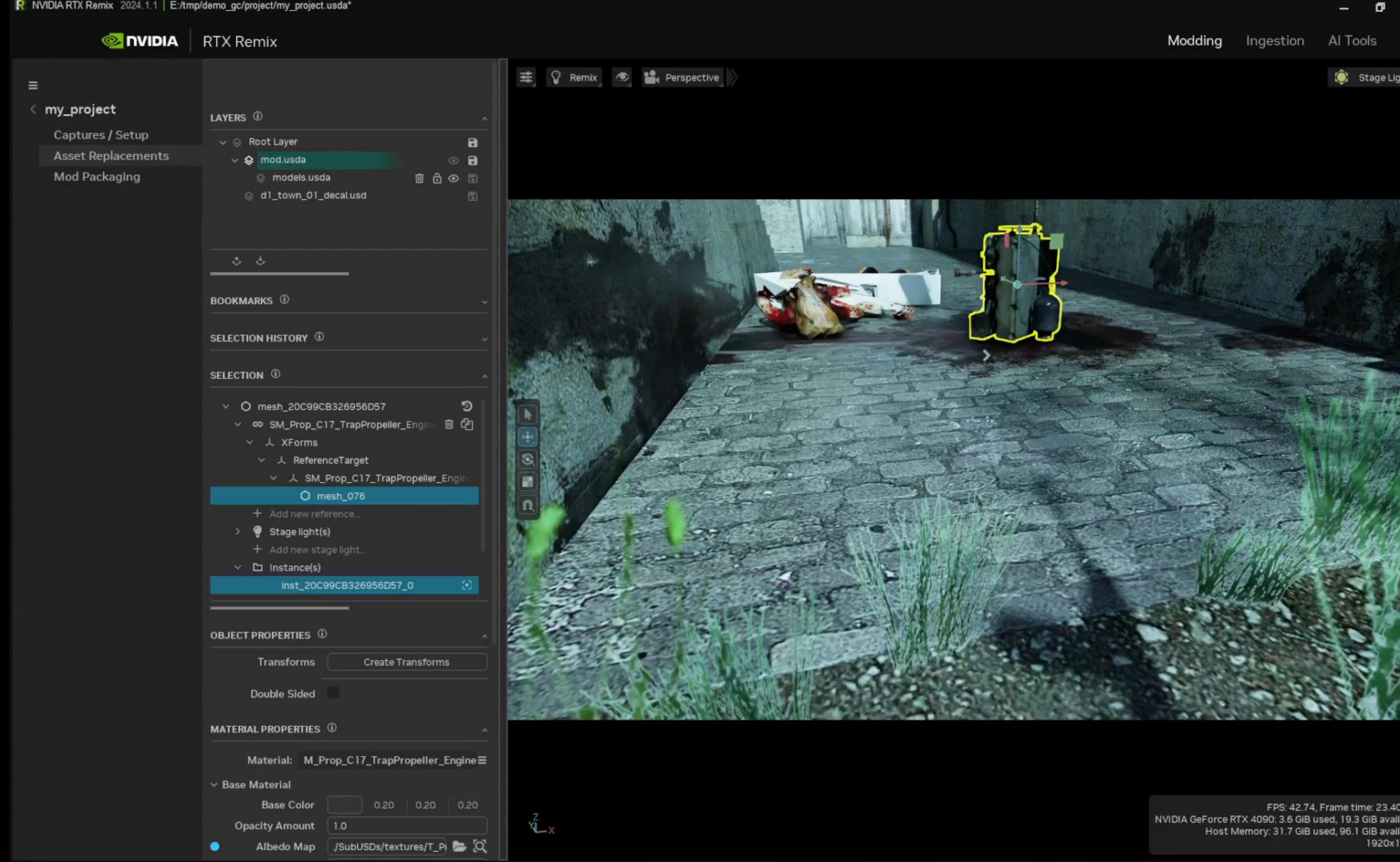
Omniverse Kit application to remaster games.



## Model Replacement

# RTX Remix Toolkit

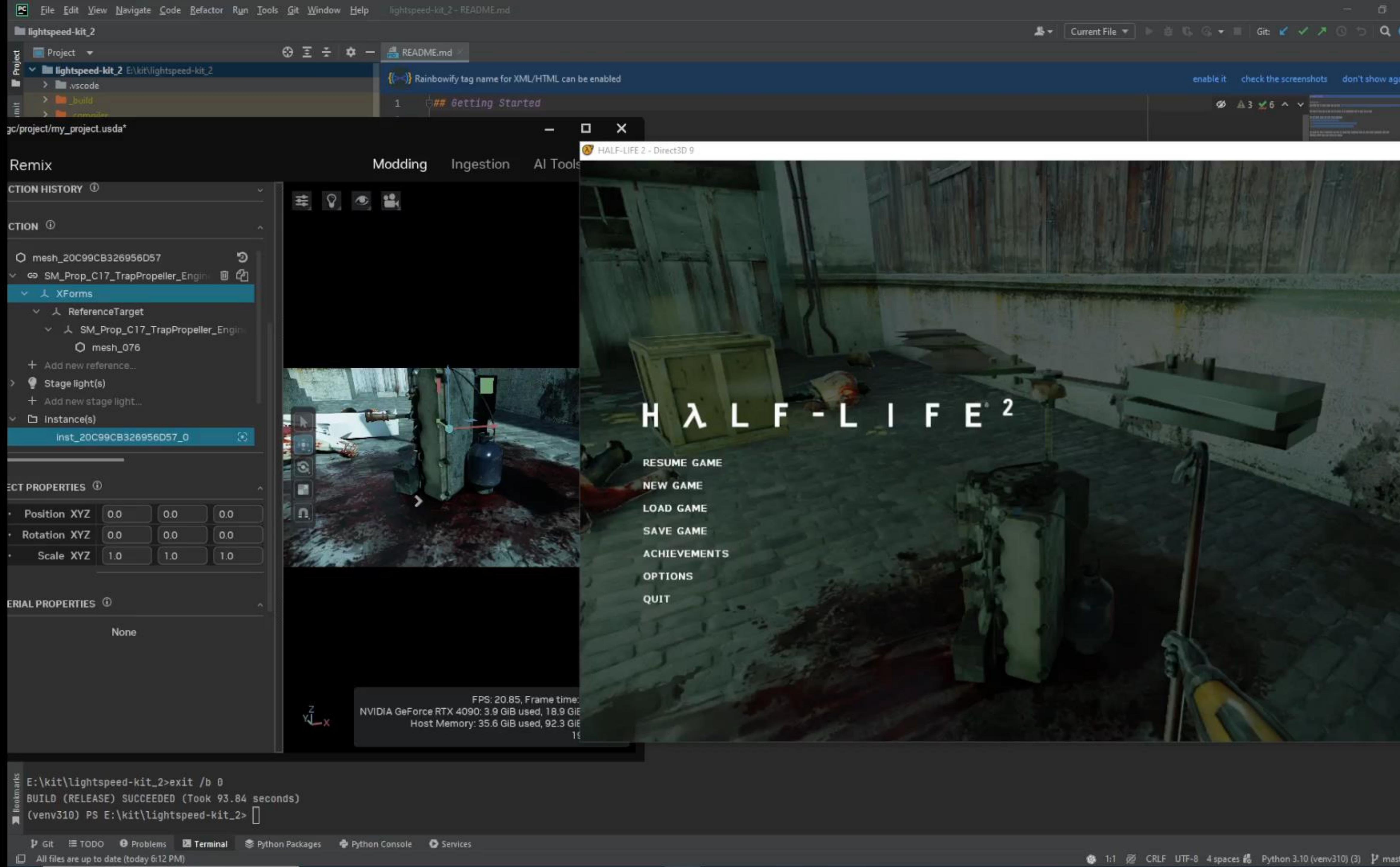
Omniverse Kit application to remaster games.



## Texture Replacement

# RTX Remix Toolkit

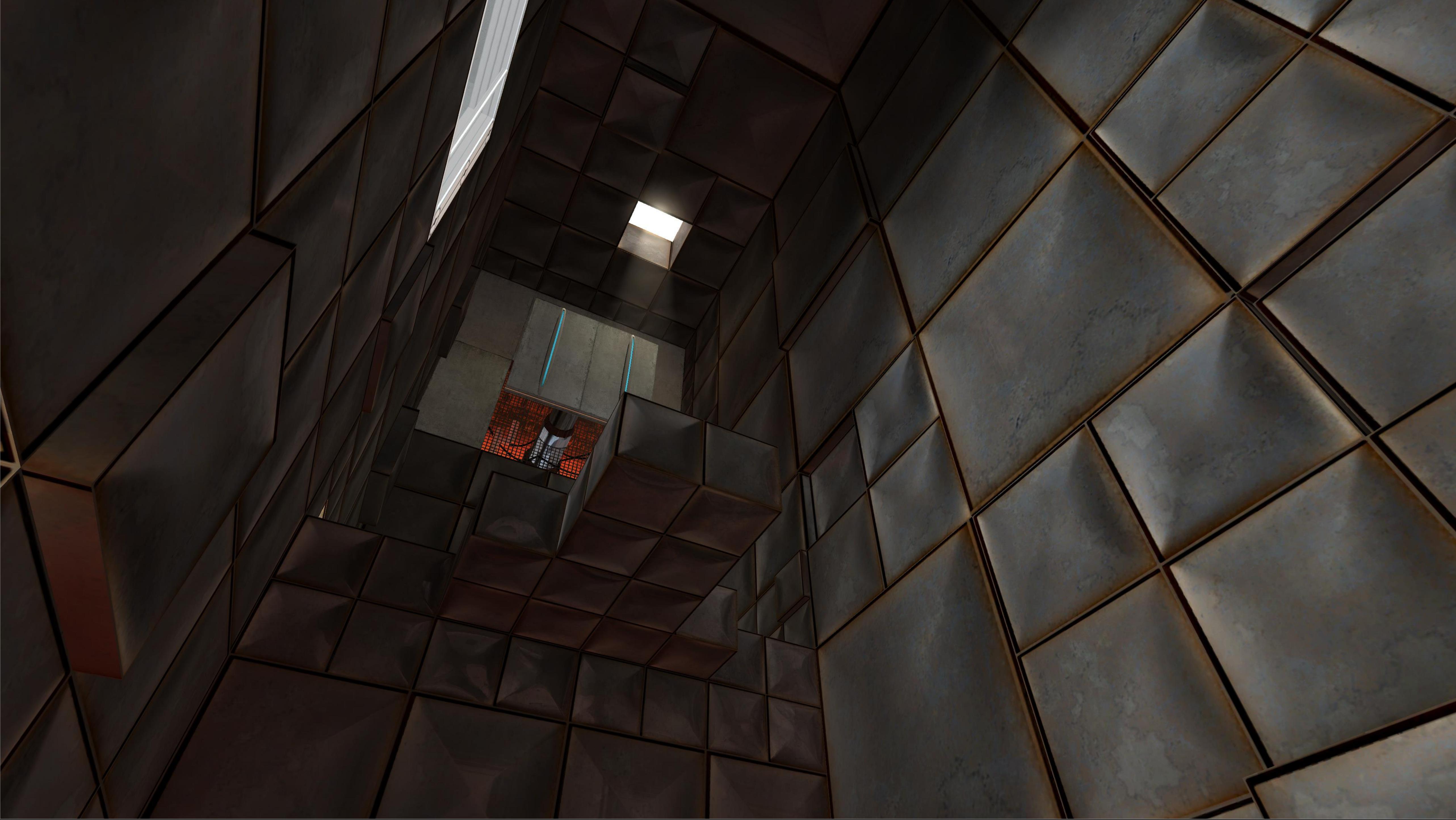
Omniverse Kit application to remaster games.



## In-Game Real-Time Updates

# RTX Remix Toolkit

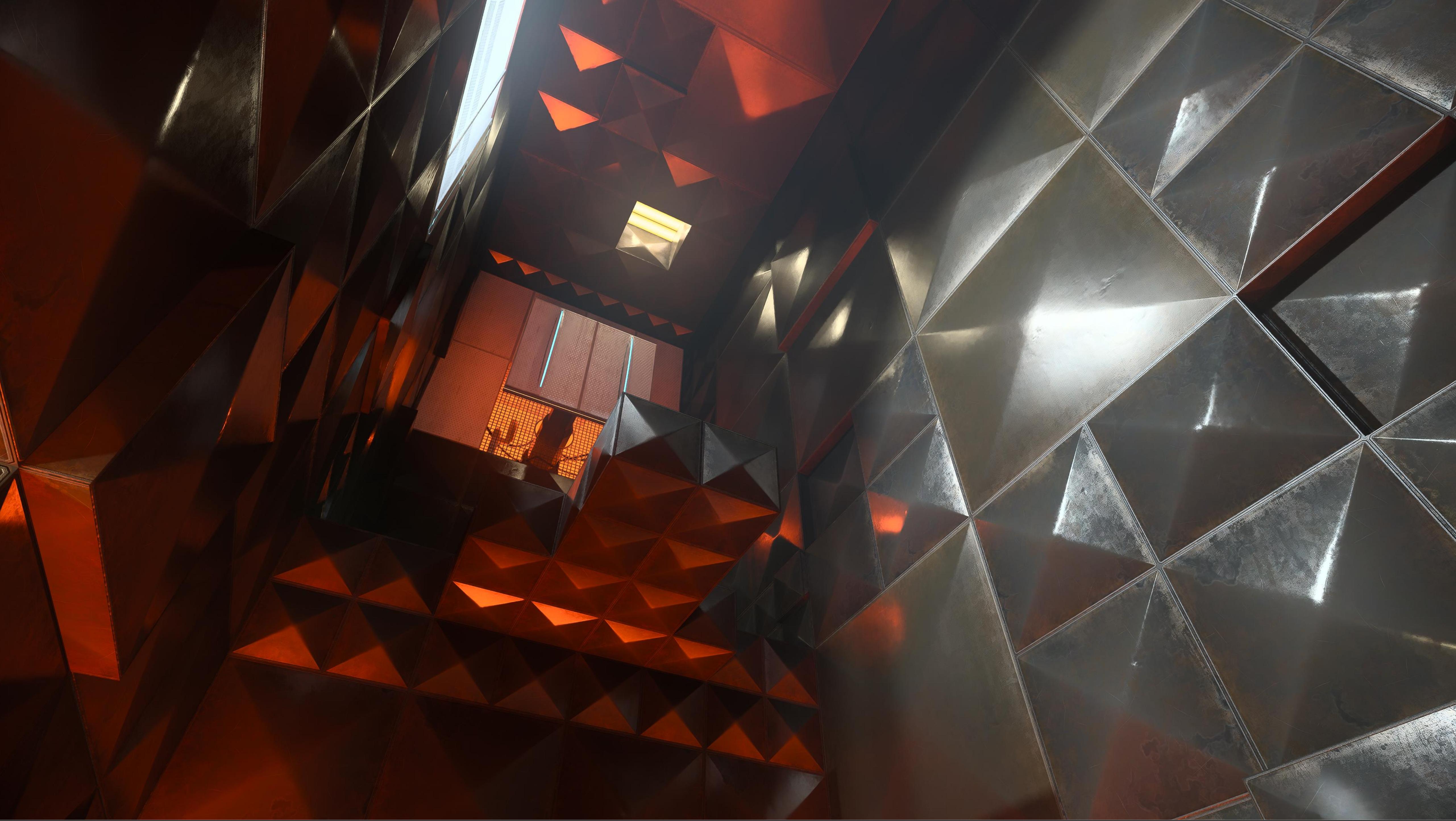
Omniverse Kit application to remaster games.



---

## In Game Results

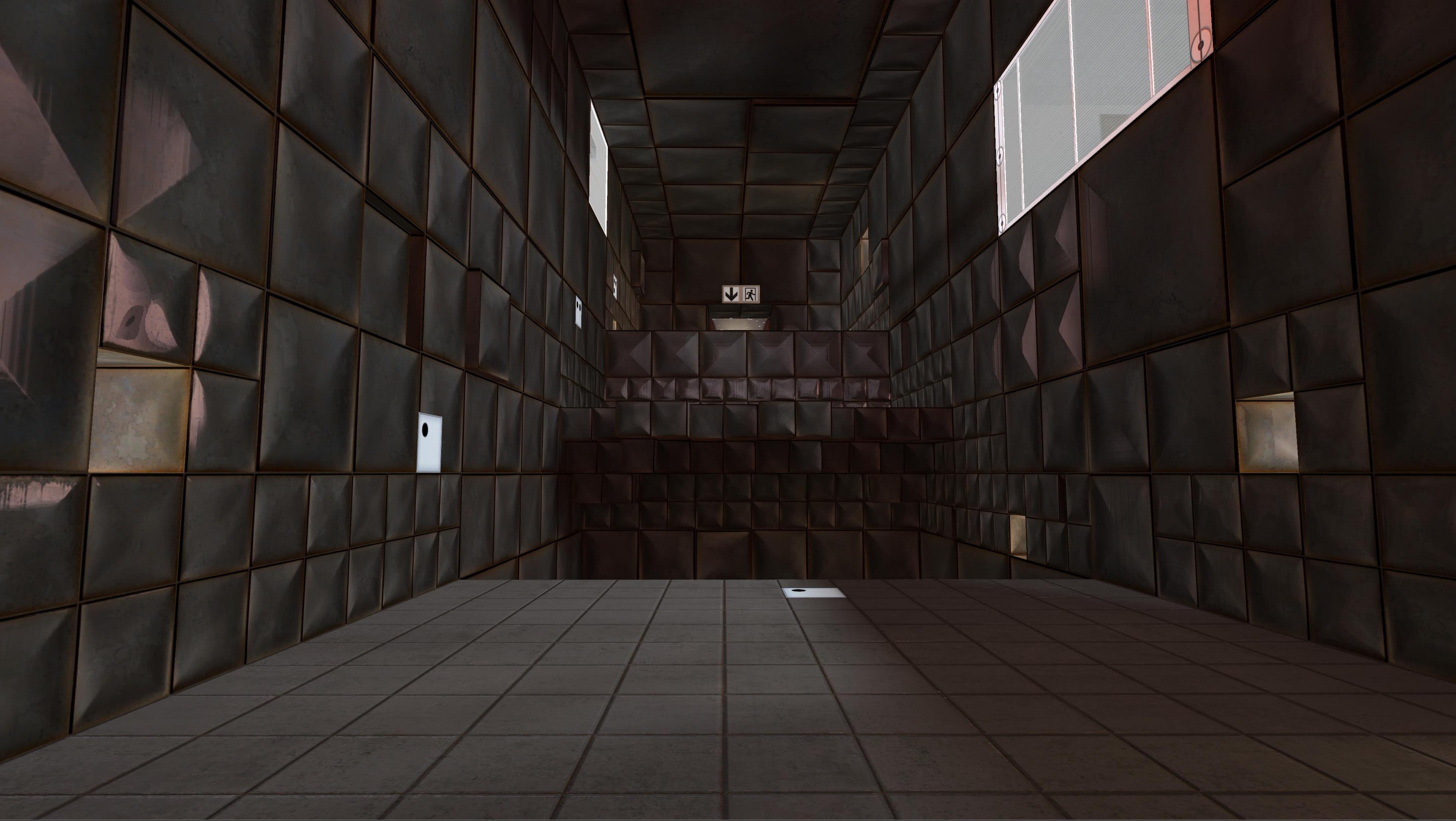
Without enhanced assets



---

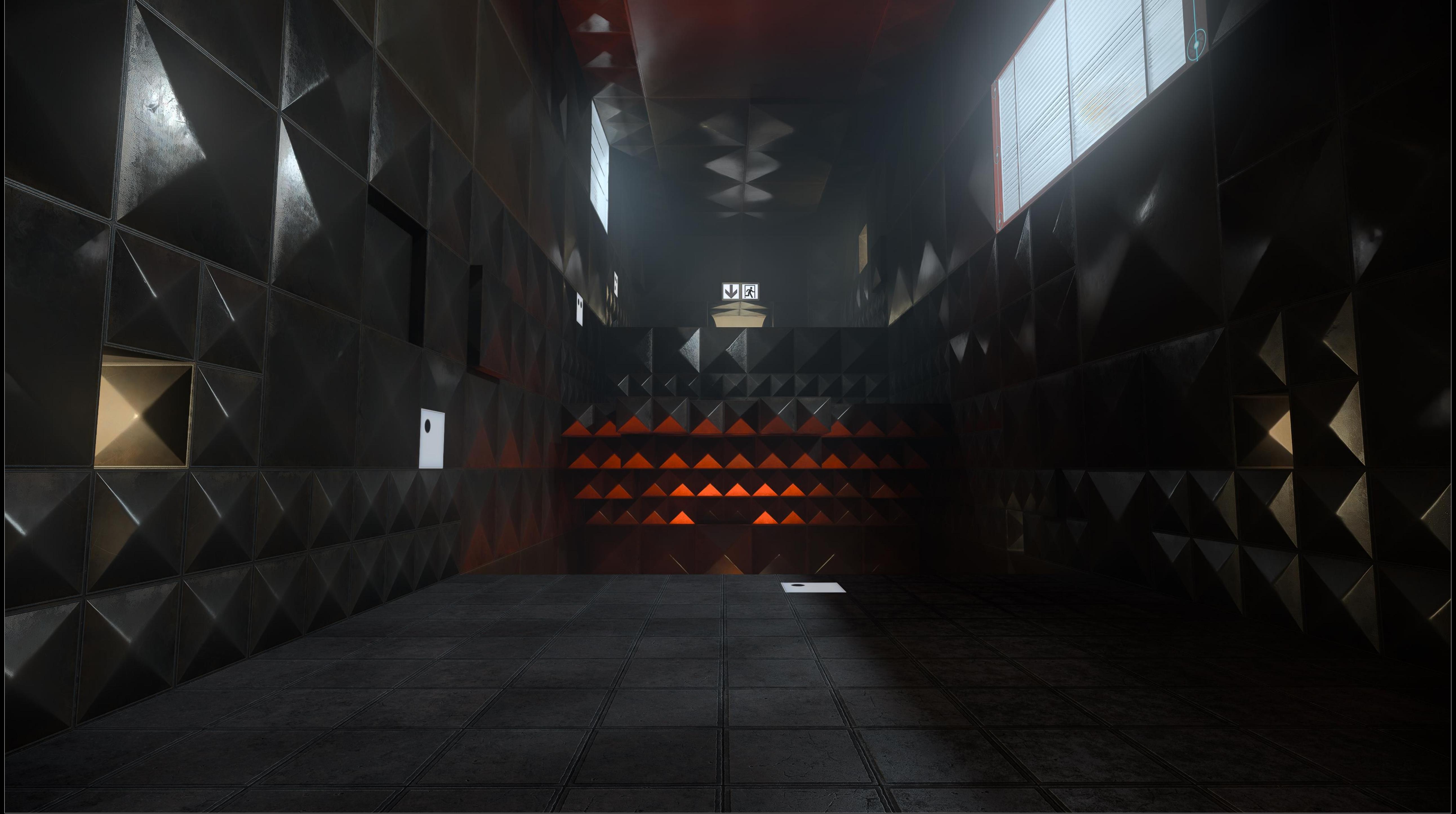
# In Game Results

With enhanced assets



# In Game Results

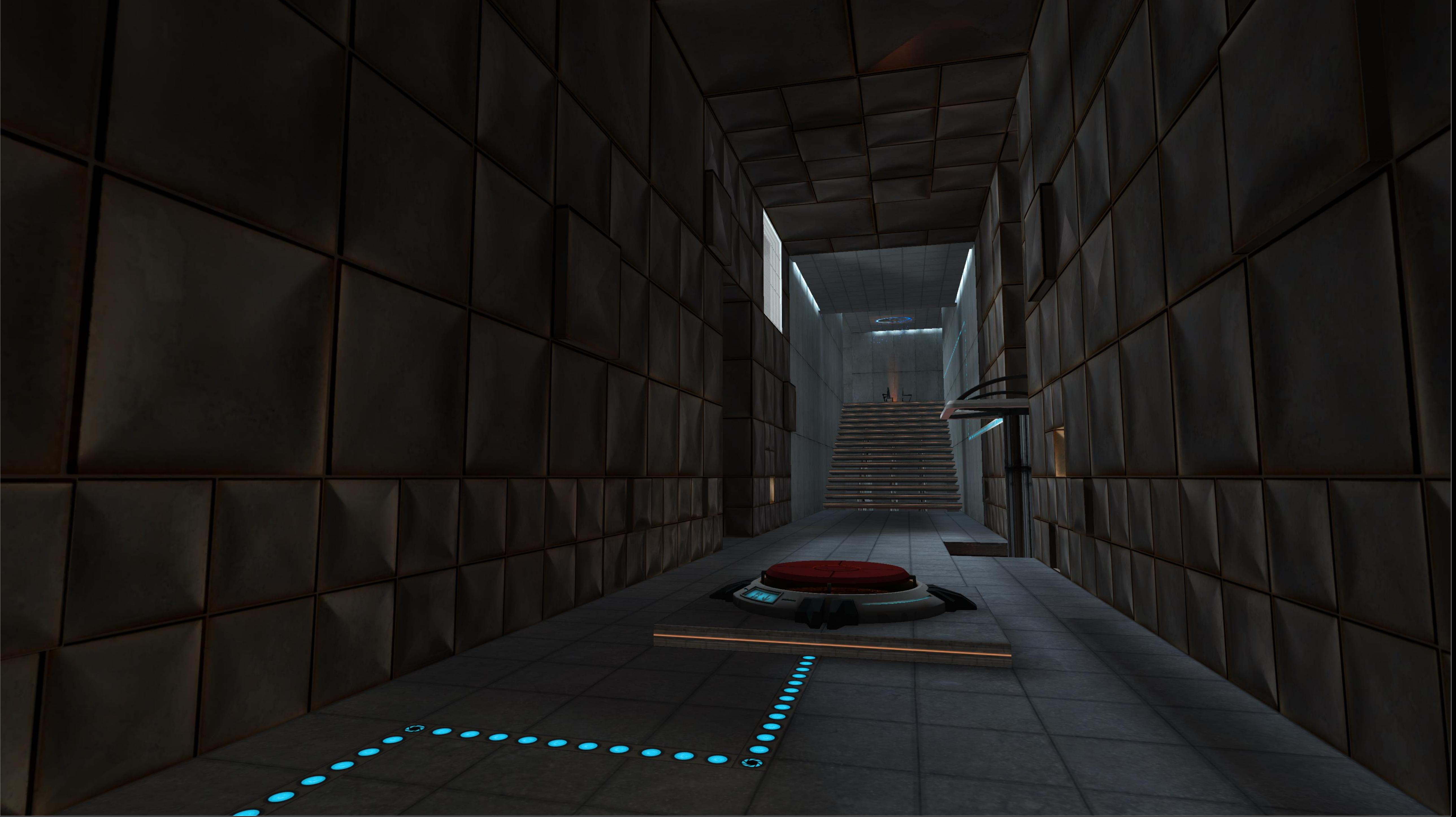
Without enhanced assets



---

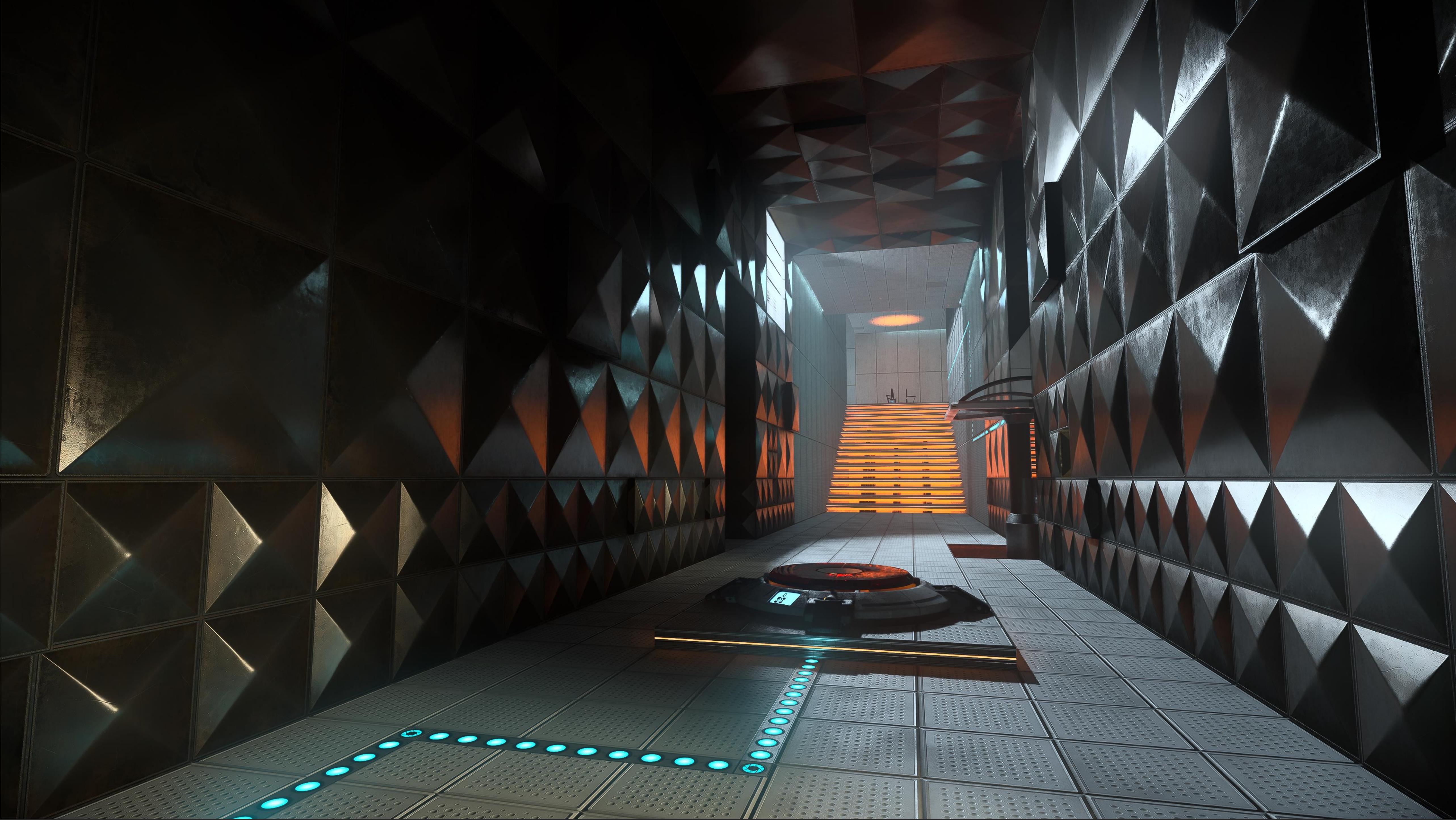
# In Game Results

With enhanced assets



# In Game Results

Without enhanced assets



# In Game Results

With enhanced assets

# Generative AI Tools

# Why use Generative AI in RTX Remix?

Batch processing of low-resolution color-only textures



**Low Resolution Textures**

Need to upscale low-resolution textures.



**Non-PBR materials**

Need to generate PBR components textures from baked color textures

# Upscaling Textures

While preserving important texture properties



Original Game Texture  
512x512



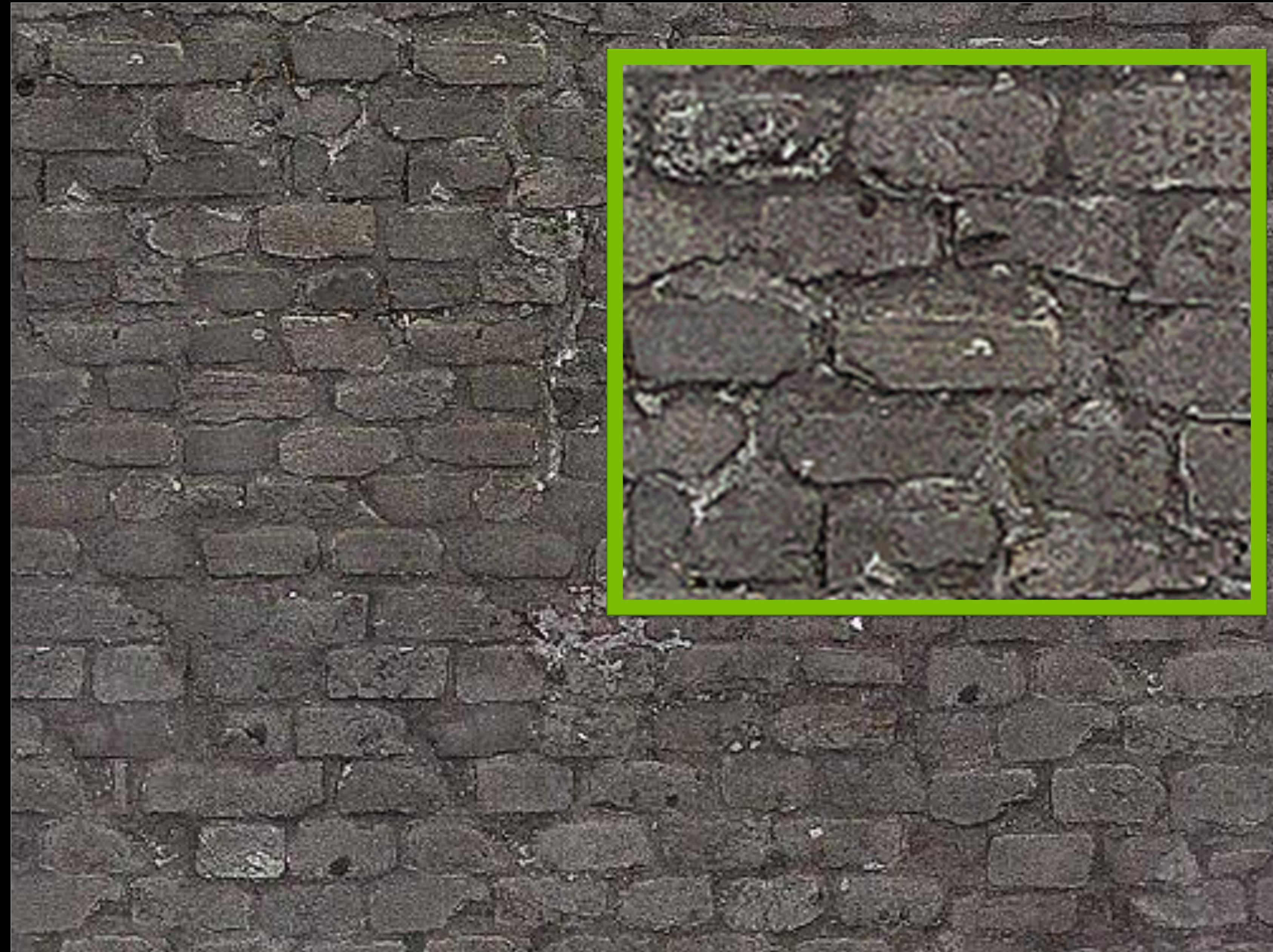
Upscaled Texture  
2048x2048

- DX9-era games tend to have low resolution textures
- Low resolution textures are upscaled by 4x (e.g. 512x512 → 2048x2048)
- Tileable properties should be conserved for tiling textures
- A variety of techniques are used to improve the visual quality of the generated textures while reducing the resources required for inference
  - Patch-Wise Inference with Circular Padding
  - Overlapping patches & Wrapped patches
  - Etc.

\* AI Tools are a work in progress and subject to change

# Upscaling Textures

Original Texture / AI Upscaled Texture



Low resolution  
Captured



AI Upscaled  
Generated

# Generating Physically Based Material Textures

From baked color textures



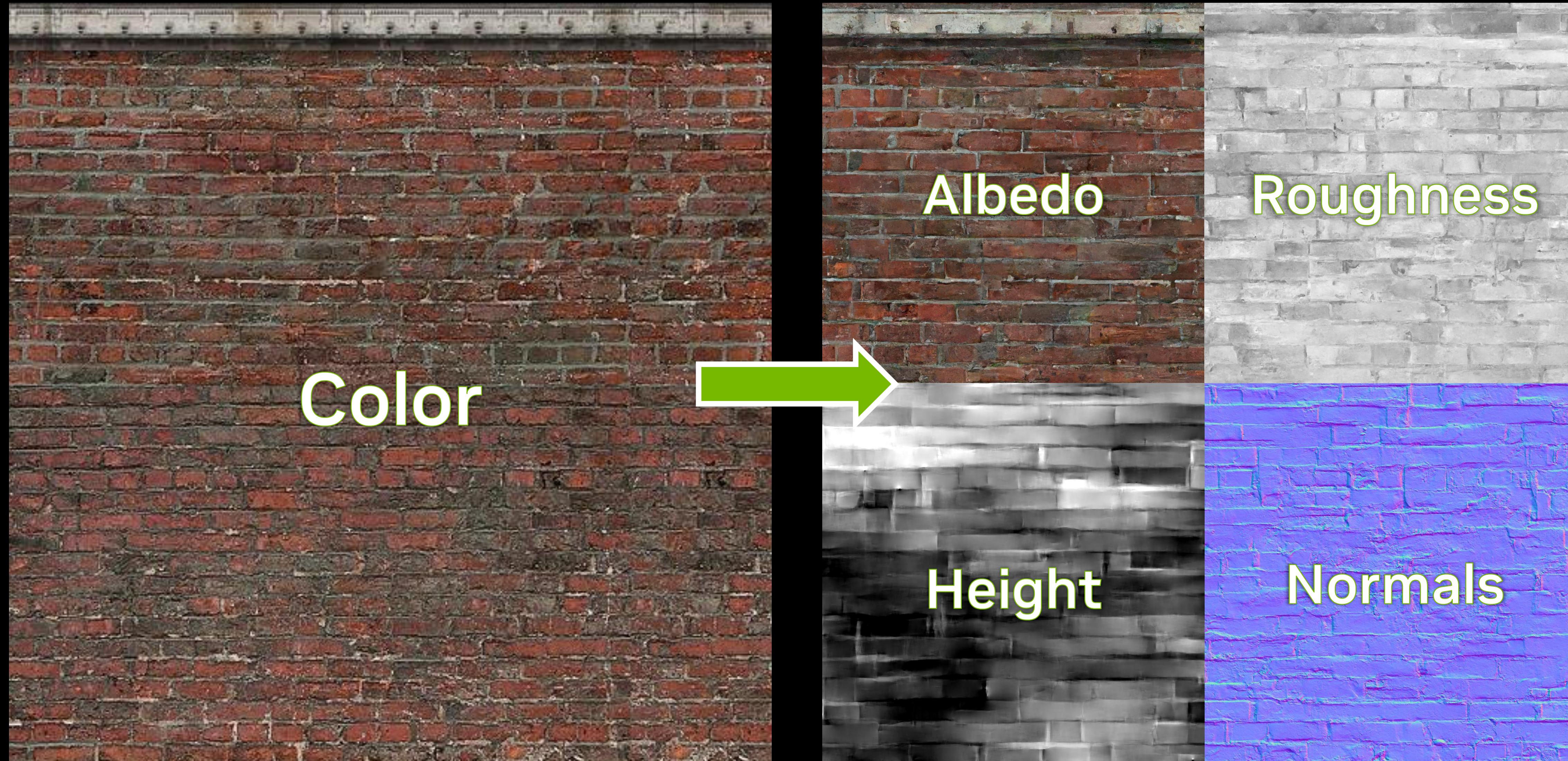
Color Texture with Baked Lighting  
AI Tools Input



Remix Generated PBR Material  
AI Tools Output

# Generating PBR Textures

From baked color textures



- DX9-era games tend to use color textures with baked lighting, these are not desirable in Path Traced renderers as the lighting is dynamic
- The renderer needs more information to know how the material should react to various lighting conditions
- Generating PBR components allows the renderer to produce more photorealistic results when compared to baked color textures
- The model currently generates 3 PBR Maps\*\*
  - Albedo
  - Roughness
  - Normals
- It is possible to derive Height maps from the generated maps as well

\* AI Tools are a work in progress and subject to change

\*\* More PBR Components to come

# Real-World Example

## Comparing Color Texture / AI Generated Material



Color Texture with Baked Lighting



AI Generated Material w/ Displacement Map

\* The displacements maps we generated using deconvolution of the AI-generated normal maps

# Real-World Example

## Comparing Color Texture / AI Generated Material



Color Texture with Baked Lighting



AI Generated Material w/ Displacement Map

\* The displacement maps we generated using deconvolution of the AI-generated normal maps

# Real-World Example

## Comparing Color Texture / AI Generated Material



Color Texture with Baked Lighting



AI Generated Material w/ Displacement Map

\* The displacements maps we generated using deconvolution of the AI-generated normal maps

# Ethically Trained Model

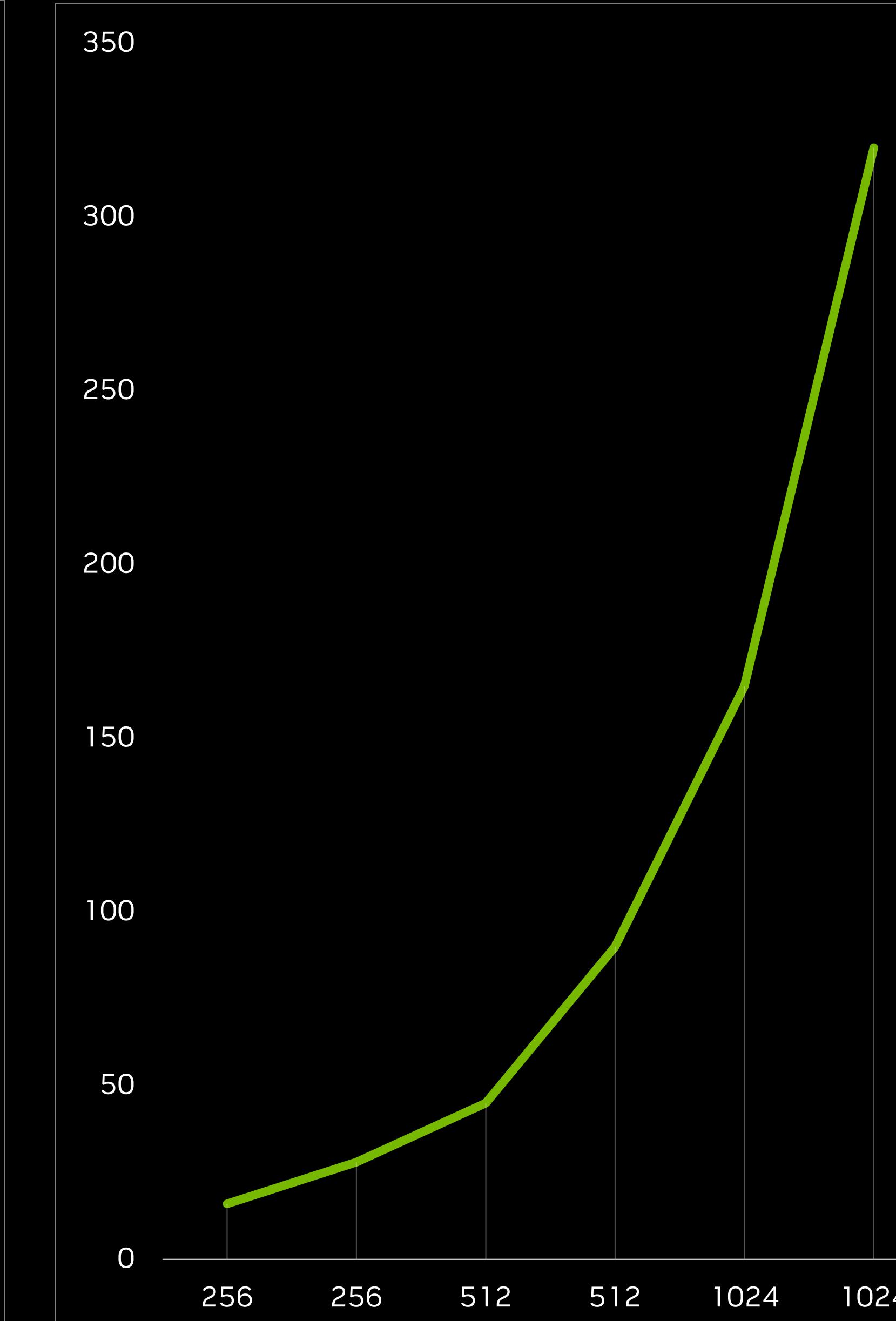
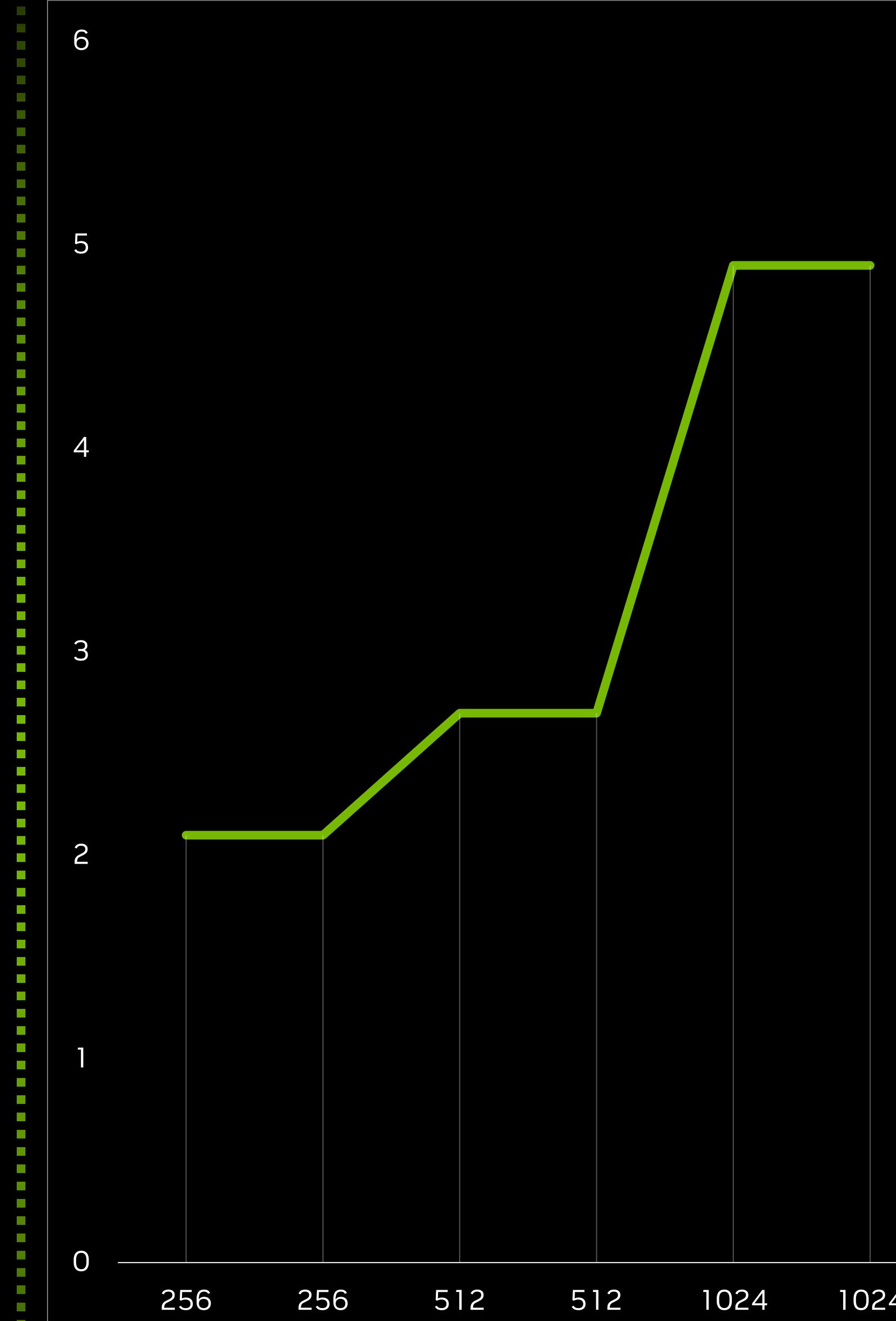
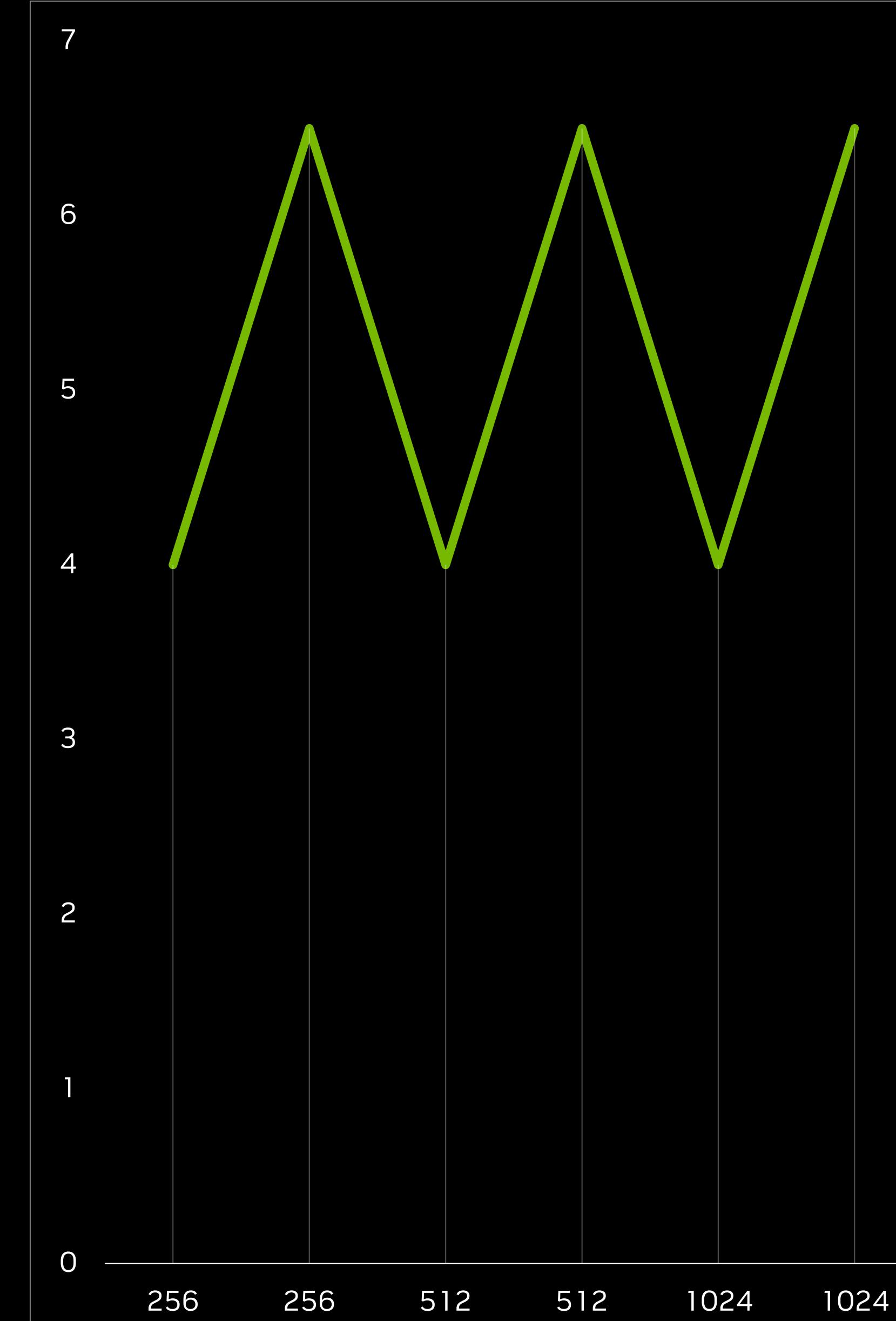
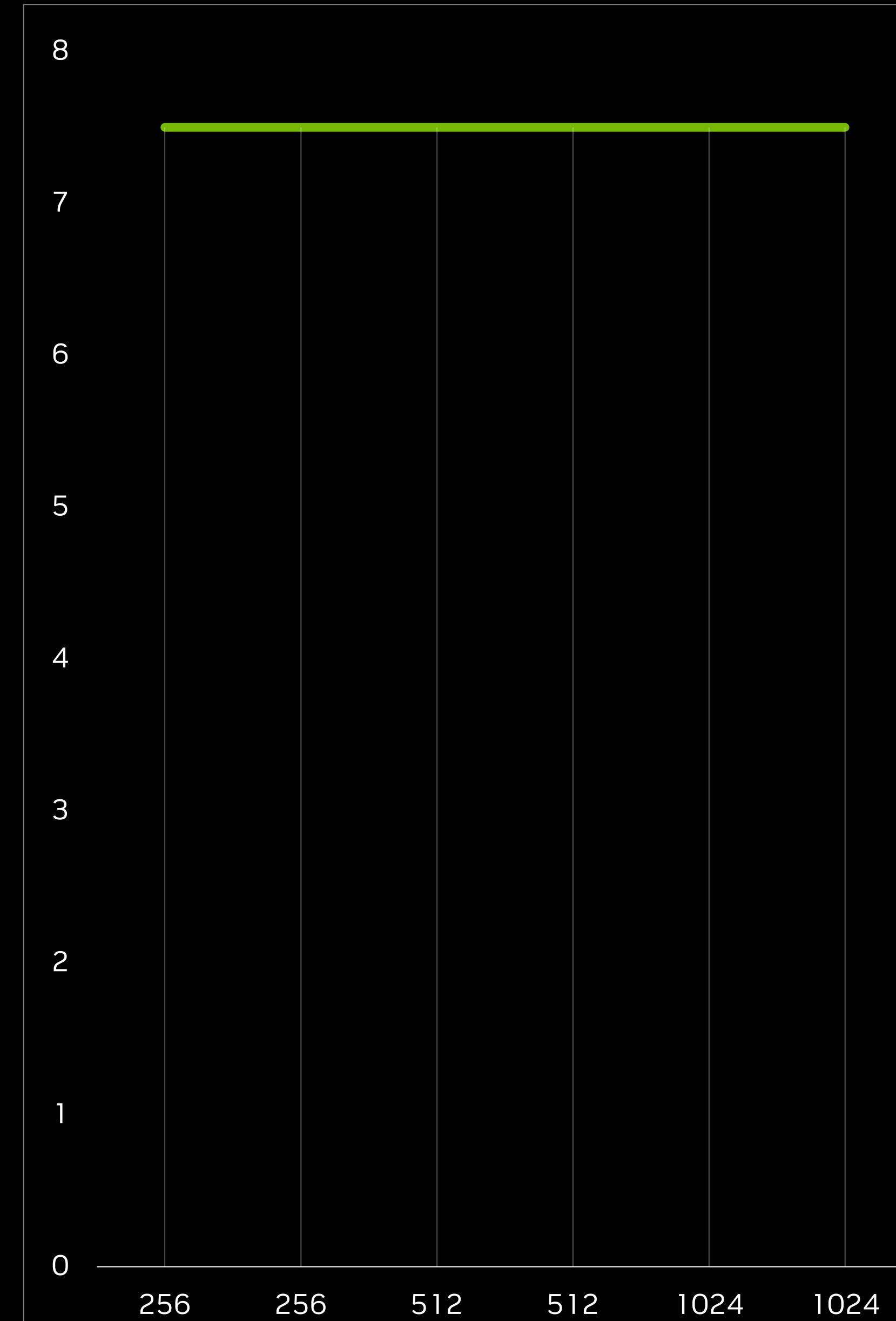
Ethically Sourced Datasets for an In-House Model

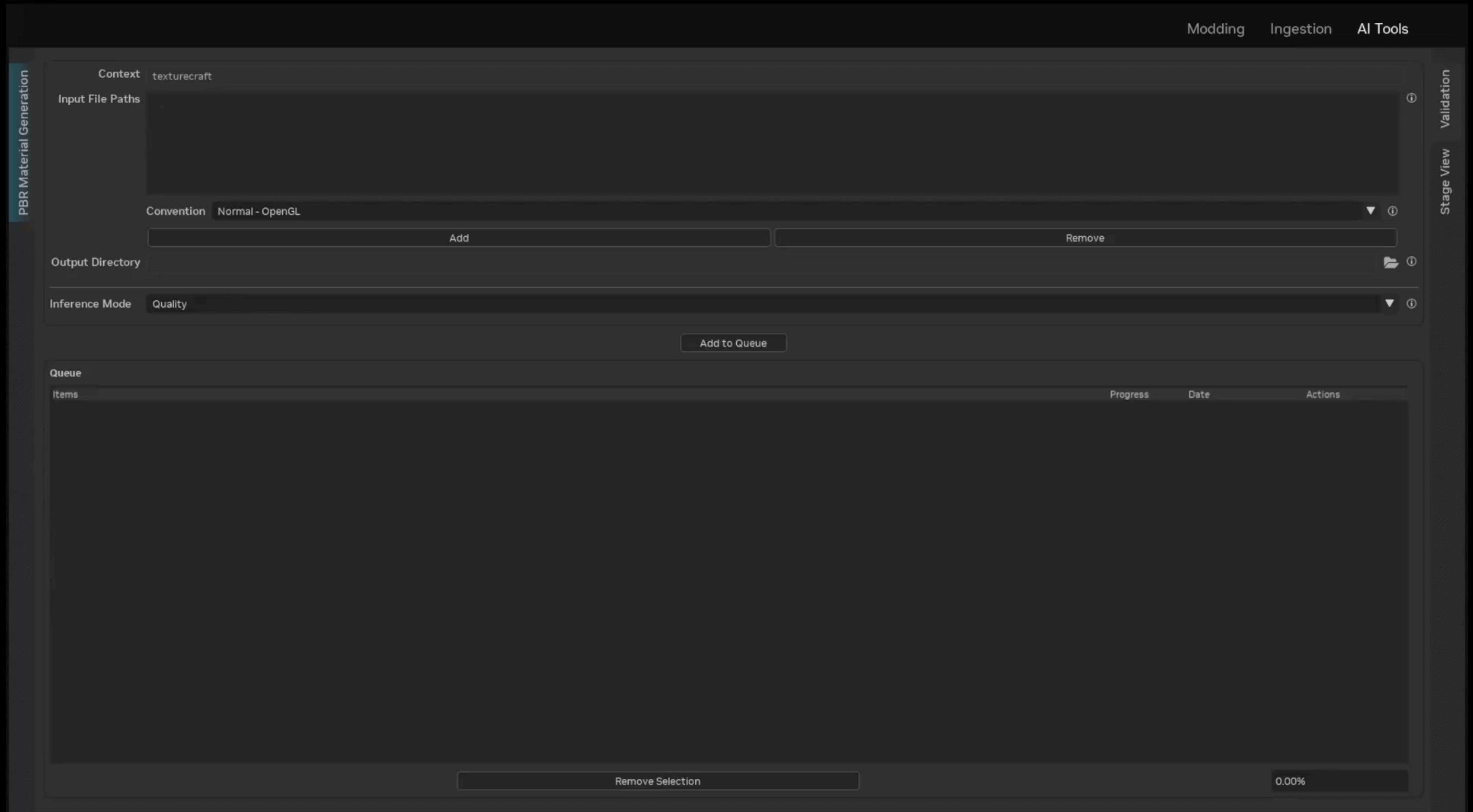


- Model is trained in-house, from scratch
- Trained on almost **7000 PBR materials** (Over **32 000 textures**)
- Dataset was sourced ethically:
  - Public domain datasets
  - In-house datasets
  - Licensed paid datasets
- We paid great attention to make sure all the data used was licensed properly.

# RTX Remix AI Model Performance

## Resource Usage and Inference Time





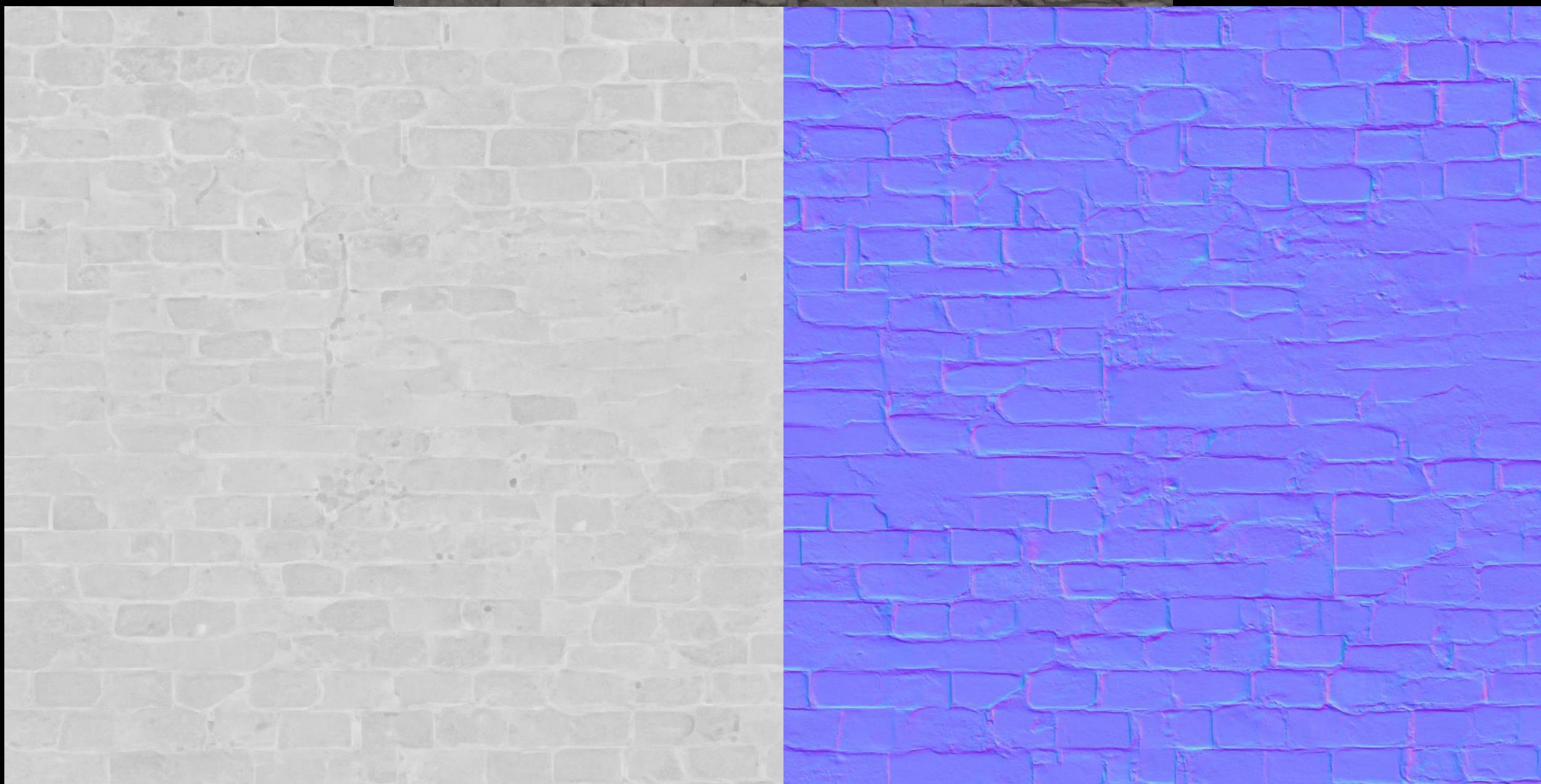
**Integrated directly in the NVIDIA RTX Remix Toolkit**  
Multiple Inference Modes & More User-Friendly Parameters to come

Free to use in: **RTX Remix Toolkit**  
<https://www.nvidia.com/en-us/geforce/rtx-remix/>

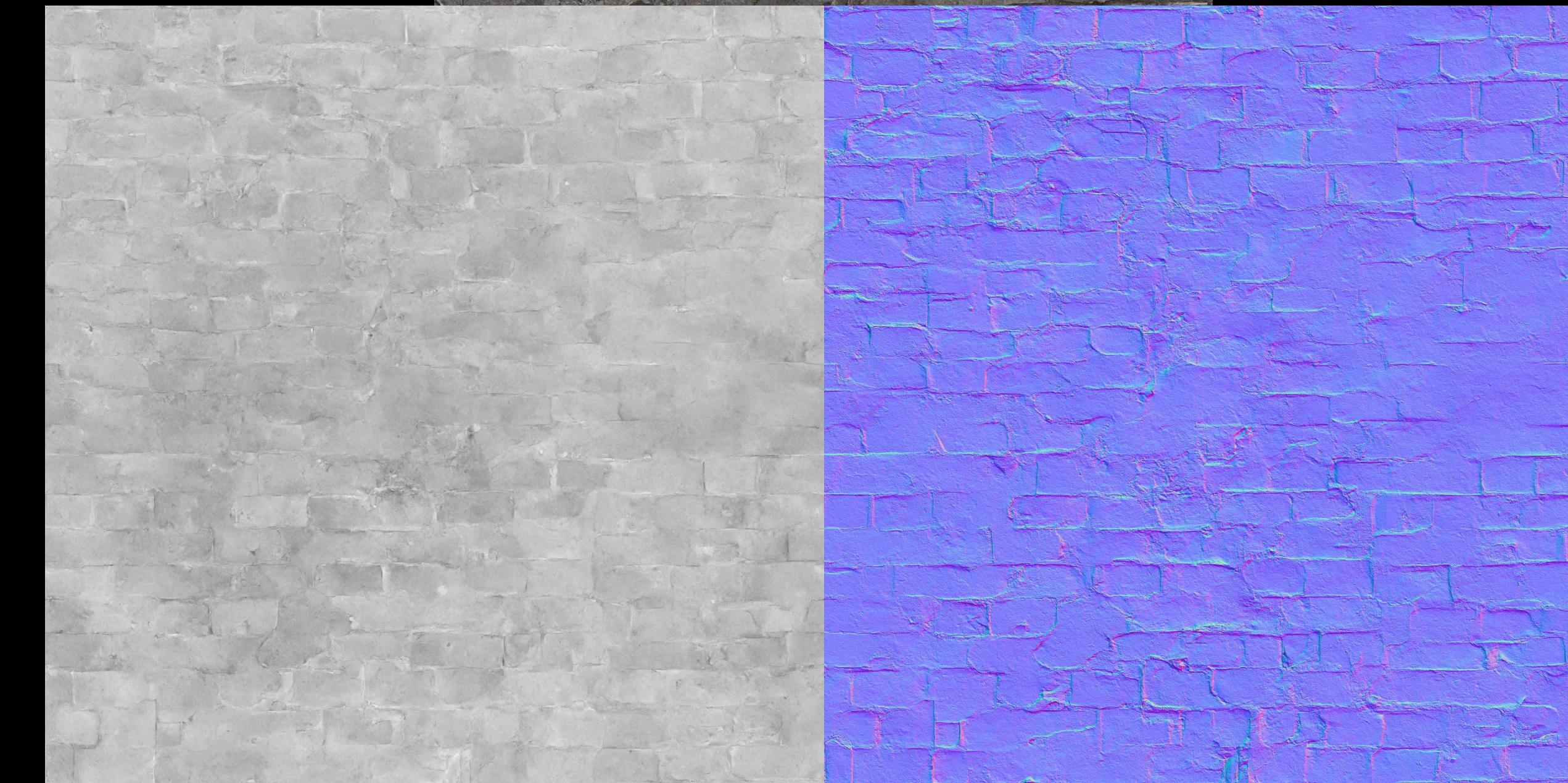


# AI Tools Inference Mode Comparison

## Speed VS Quality



**Speed Inference**  
512x512 input → 1024x1024 output



**Quality Inference**  
512x512 input → 2048 x 2048 output

# AI Tools Inference Mode Comparison

## Speed VS Quality



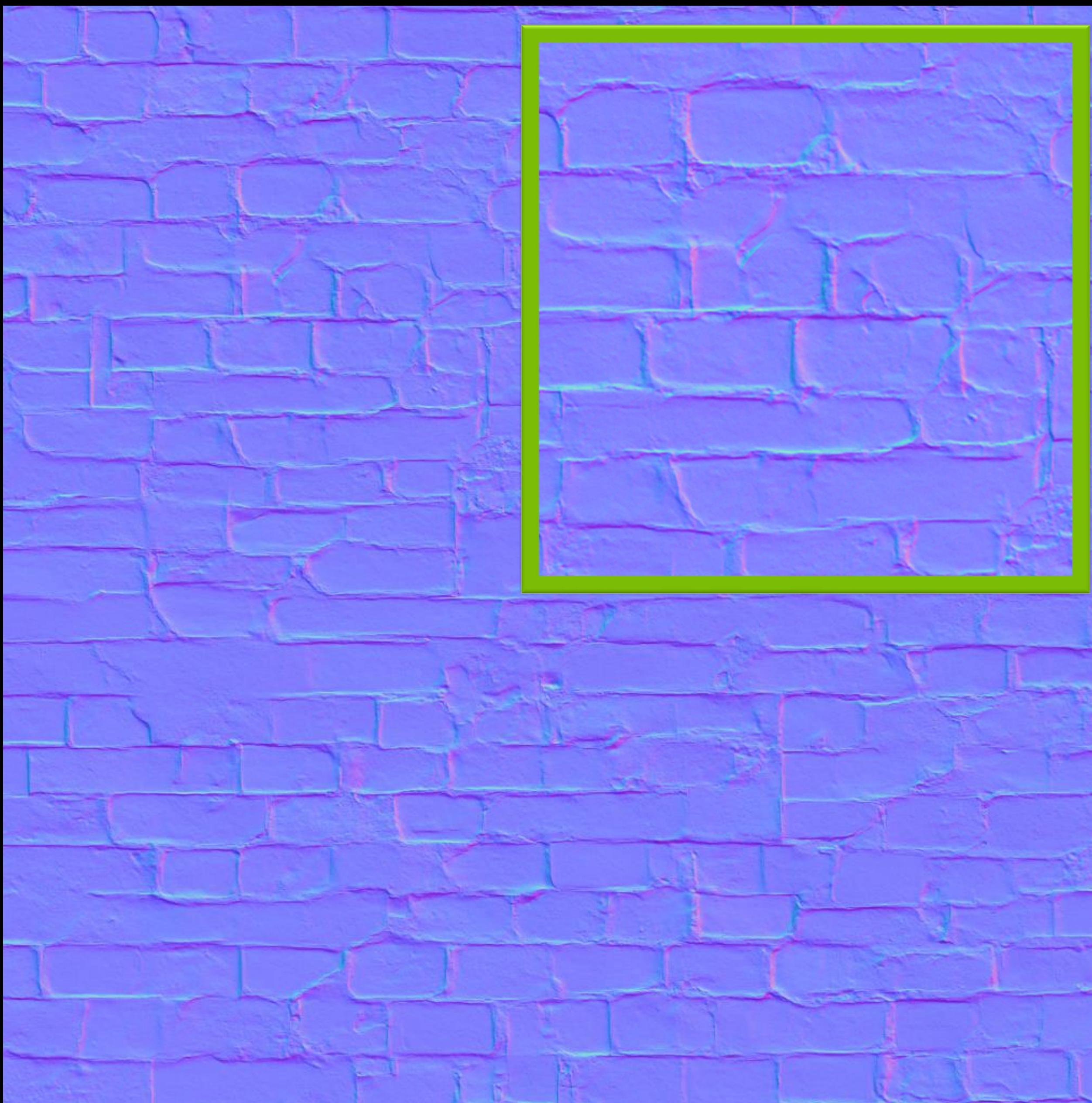
**Speed Inference**  
512x512 input → 1024x1024 output



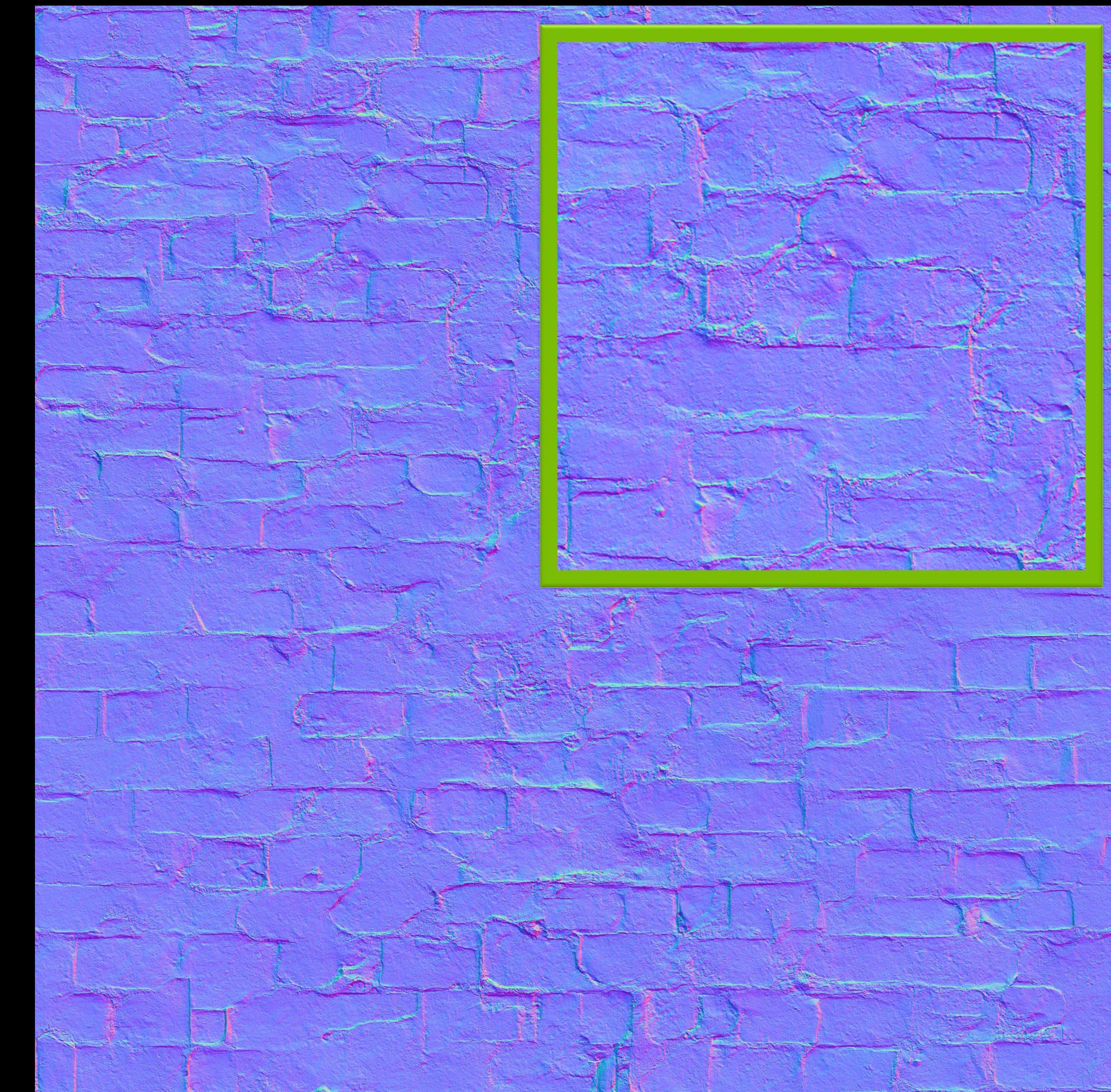
**Quality Inference**  
512x512 input → 2048 x 2048 output

# AI Tools Inference Mode Comparison

## Speed VS Quality



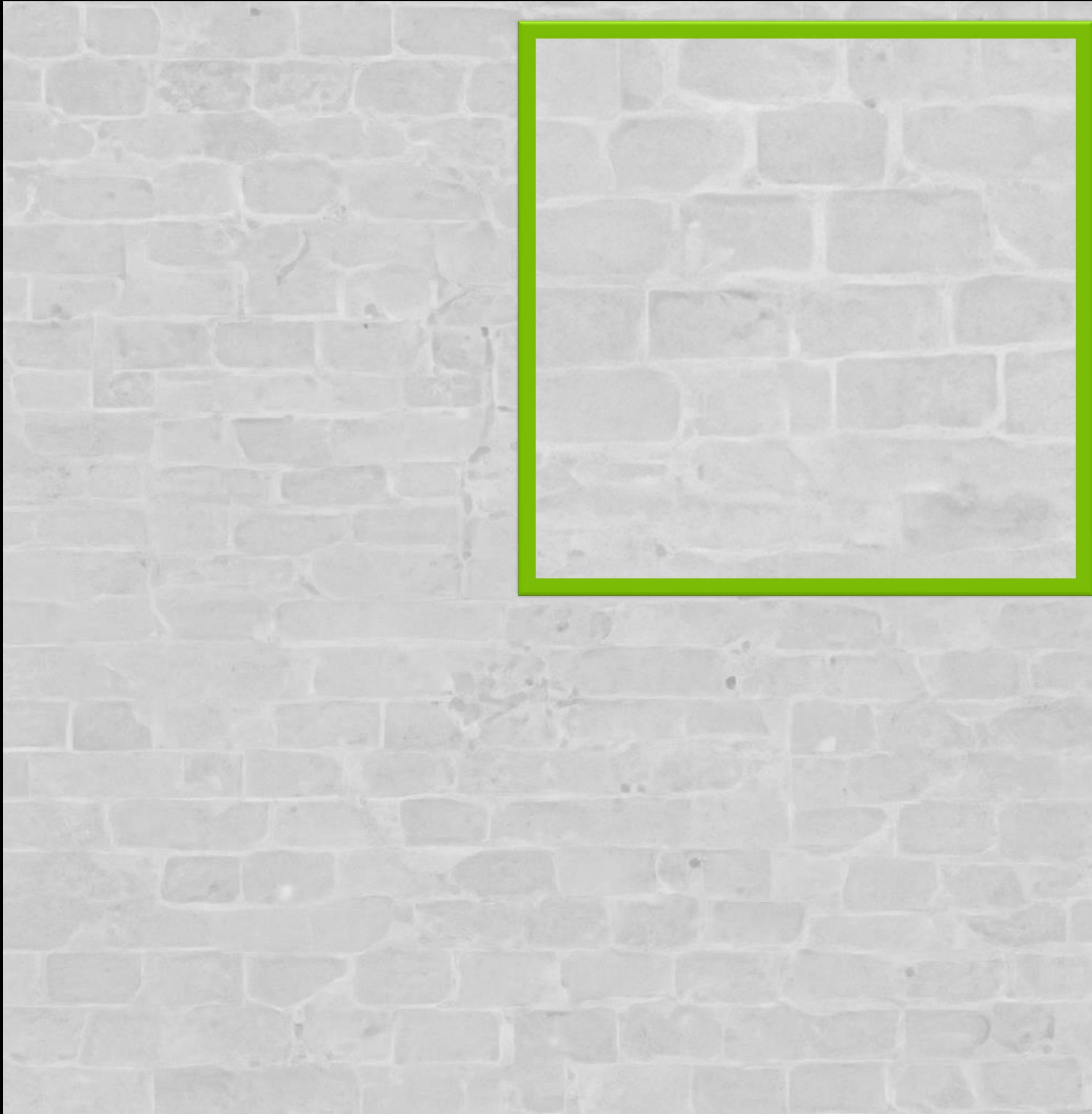
**Speed Inference**  
512x512 input → 1024x1024 output



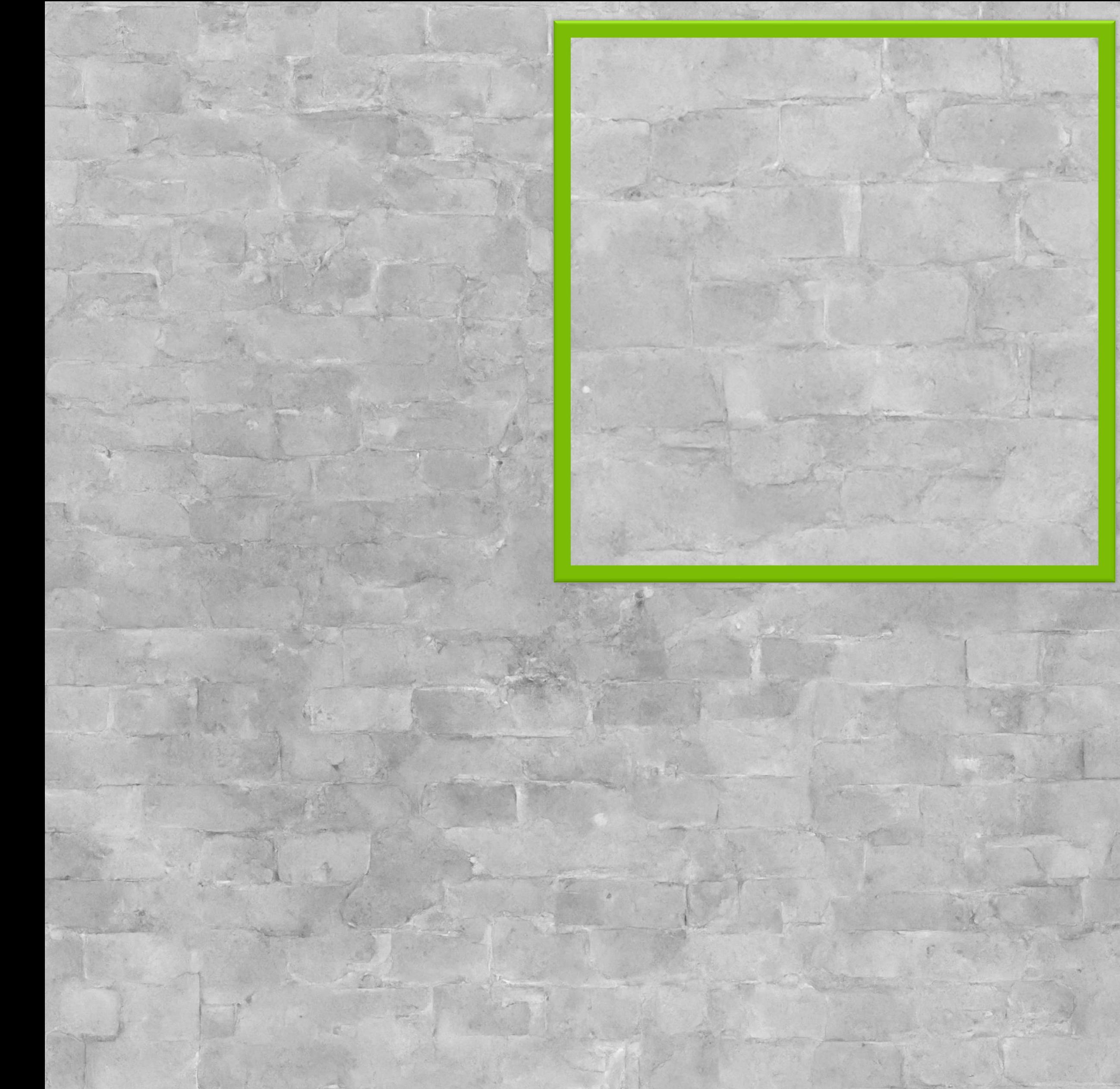
**Quality Inference**  
512x512 input → 2048 x 2048 output

# AI Tools Inference Mode Comparison

## Speed VS Quality



**Speed Inference**  
512x512 input → 1024x1024 output



**Quality Inference**  
512x512 input → 2048 x 2048 output

The screenshot shows two windows side-by-side. The left window is a Swagger UI interface for the Kit services core REST API, specifically the /stagecraft endpoint. It lists several endpoints:

- GET /stagecraft/prims/references/{prim\_path} Get References
- PUT /stagecraft/prims/references Replace Reference
- POST /stagecraft/prims/references Append Reference
- GET /stagecraft/textures/ Get Textures
- PUT /stagecraft/textures/override Override Texture

The 'Override Texture' endpoint is expanded, showing its description: "Override the given texture on the current edit target in the current stage." Below this is a 'Parameters' section with a 'Cancel' and 'Reset' button, and a 'Request body required' section with a dropdown set to 'application/json'. The request body example is:

```
{  
  "force": false,  
  "textures": [  
    {  
      "path": "/RootNode/Looks/mat_1C2933544D625F50/Shader_inputs/diffuse_texture",  
      "uri": "s:\Remix\hi2rtx\test\output\ingested\1C2933544D625F50 diffuse_a.rtex.dds"  
    }  
  ]  
}
```

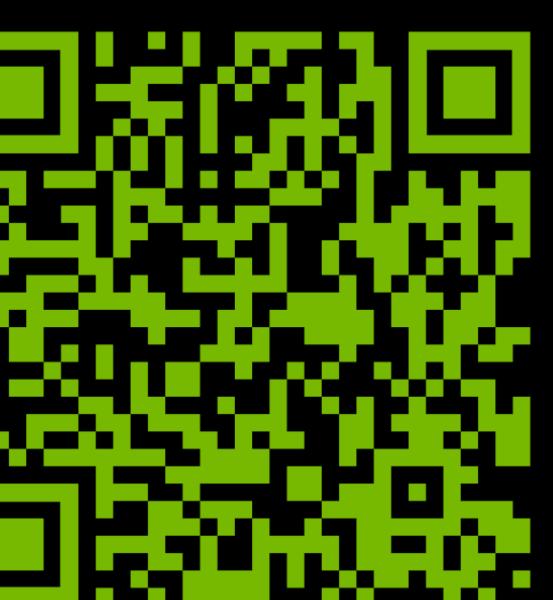
Below the request body is an 'Execute' button and a 'Responses' section for a 200 status code. The 'Media type' dropdown is set to 'application/json'. The 'Example Value' field contains the string "string".

The right window is the NVIDIA RTX Remix application. It shows a 3D scene of a wooden building with a door. A yellow bounding box highlights a specific area on the door. The interface includes a sidebar with sections like 'Captures / Setup', 'Asset Replacements', and 'Mod Packaging'. On the right, there's a 'BOOKMARKS' section, 'SELECTION HISTORY', and a detailed 'SELECTION' panel showing a tree node under 'mesh\_B13E5424FF329AA7.usd'. The 'OBJECT PROPERTIES' and 'MATERIAL PROPERTIES' panels are open, showing material settings for 'Looks/mat\_1C2933544D625F50'. The 'Base Material' section includes fields for 'Base Color', 'Opacity Amount', 'Albedo Map', 'Roughness Amount', 'Metallic Amount', 'Normal Map', and 'Iridescence'. The 'Material Properties' section includes 'filter\_mode', 'wrap\_mode\_u', and 'wrap\_mode\_v'. A status bar at the bottom right shows 'FPS: 34.71, Frame time: 28.81 ms' and system information: 'NVIDIA GeForce RTX 4090, 10.1 GB used, 12.7 GB available, Host Memory: 31.5 GB used, 32.3 GB available, 1920x1080'.

## Integrated within the RTX Remix REST API

Example using the REST API to update a Texture

In development: **RTX Remix Toolkit**  
<https://www.nvidia.com/en-us/geforce/rtx-remix/>



# RTX Remix Toolkit REST API

## Functionality Exposed by the API

- **Create**
  - Create/Insert a layer
  - Add a new prim reference
- **Read**
  - Get the currently opened project
  - Get the stage's layer stack
  - Get prims in the stage
- **Update**
  - Set the edit target layer
  - Set the viewport selection
- **Delete**
  - Delete a layer
  - Delete an asset or light
- **Etc.**

POST

GET

PUT

DELETE

- Start AI Inference
  - Ingest an asset
  - Open a new project
  - Save a layer
  - Lock/Unlock a layer
  - Mute/Unmute a layer
  - **Etc.**
- REST API is a work in progress and subject to change.

**CRUD Operations**  
Create/Read/Update/Delete

**Action Triggers**  
RPC-like triggers

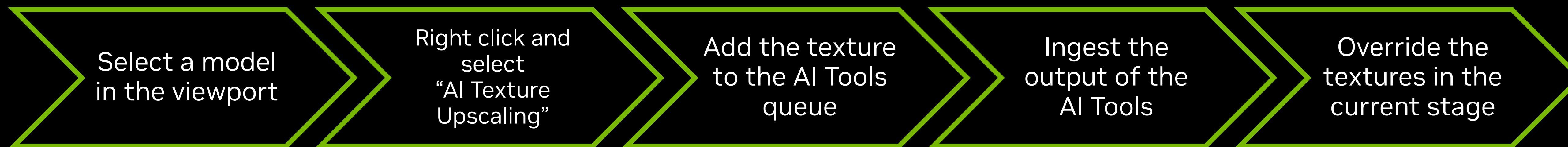
# RTX Remix Toolkit REST API

Examples of workflows made possible by the REST API

## Automatically Process All Textures in a Stage



## Process a Texture from the Modding Tab viewport



# Future plans for RTX Remix AI Tools

## Planned Features & Bug Fixes

- **REST API**

- Headless REST API process (Server Mode)
- Broader integration of the REST API
- Better integration of the various Remix functionalities
- Pipeline RTX Remix functionalities using REST API

- **AI Tools**

- Better quality inference output for AI Tools
- Faster inference for AI Tools
- More user-controls over the inference
- Ability to create texture variants using text prompts
- Ability to create textures from scratch using text prompts

- **UX**

- Project setup improvements
- Mod setup improvements
- Mod packaging improvements
- Property Trees improvements
- Better search capabilities within the stage

- **Stability**

- More input validation throughout the app
- General bug fixes
- General stability improvements

