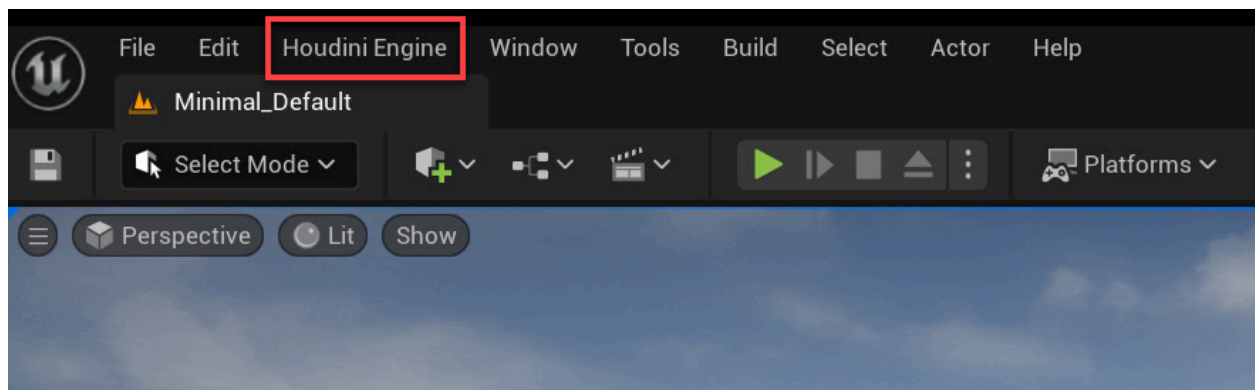


Installing world building example

1. Follow the instructions in [Install Houdini Engine for Unreal](#) to get the latest plugin.
2. Unzip the **Gamedev_Project.zip**.
3. Locate the Houdini Engine directory. The default location is `\Side Effects Software\Houdini Engine\X.Y.Z.\U.E..`
 - X.Y.Z. means your Houdini version.
 - U.E. means your Unreal Engine version.
4. Copy the Houdini Engine folder to the *Plugins* folder of the *Gamedev_Project* folder.
5. In the **SideFX_MOBA** folder, copy *SideFX_MOBA/Content/hda* to your local *Documents/houdini21.5/otl* folder. You can create the *otl* folder if it doesn't exist.
6. Launch the **SideFX_MOBA** Unreal project.
7. You can verify if Houdini Engine has installed properly if you see the Houdini Engine menu.



8. In the **Houdini Engine** menu, click **Start Session**.
9. A **Houdini Engine session connected** confirmation appears on the bottom right.

Getting Started

1. In the content browser, go to **HoudiniGames_Moba**, then **Blueprints**.
2. Right-click **HDA_World_Manager** and select Run Editor Utility Widget. This will pop up a new window for the HDA World Manager

SideFX
Software

World Builder

▼ Setup

Initialize HDAs

Step 1 Create Template HDA MOBA ▼

Step 2 Create Map Editor HDA

Step 3 Create World Manager HDA Preset 1 ▼

▼ Template

MOBA Template HDA Parameters

Lane Count 3

Map Size 1.0

Layout 4

Treeline Ratio 20.0

Treeline Seed 1

Rebuild Curves

▼ Map Editor

Level Curves Management

Update Curves From Level Delete All Curves

Create Custom Curve Lanes ▼

Set Curve Tags

Lanes Paths

Borders Bases

Treelines Rivers

World Manager HDA Parameters

Path Width 4.0

River Width 4.0

Lane Width 4.0

Treeline Width 1.0

Tree Scatter Density 1.0

Enclosed Treeline Distance 1.0

▼ World Manager

PDG - Procedural Dependency Graph

PDG Output HE_OUT_Edit ▼

HDA World Manager Parameters

Terrain Tiling 4

PDG Settings

Cook Node Network

Dirty Node Network Dirty All

Bake Final Output

UI Description

Setup

These three options initialize three different HDAs, executed one after the other.

Each stage has a corresponding tab below for editing. For example, creating the Template HDA allows you to modify it through the **Template** tab in the Detail panel.

- **Step 1** - Creates a mock design HDA that controls a series of curves used downstream in other areas of the world-building pipeline.
- **Step 2** - Cooks the previous template and uses the resulting curves as inputs for a **Preview** HDA. This allows you to edit curves directly in their Unreal level and quickly generate a proxy asset for the main cook. You can also begin adjusting parameters in the **World Manager** → **Map Editor** tab.
- **Step 3** - Removes the preview HDA and its proxy, then generates the final HDA. It applies any edits made in the **Map Editor** parameters to the final cook. This stage is the final result—offering less direct control—but any changes to the input curves are automatically reflected in the World Manager. Finally, the **World Manager** tab becomes available, providing PDG control options and allowing you to switch between **Edit**, **Preview**, and **Final** modes.

Template

These parameters modify the Template HDA. While Houdini Engine may expose many more parameters, this widget simplifies the interface—reducing hundreds of options to just a handful of the most relevant ones.

The current template creates a **MOBA-style** layout consisting of roads, paths, trees, rivers, and map borders. However, the playable area doesn't need to remain small, the constrained size is to help you visualize changes more easily. In practice, these tools can scale across areas spanning hundreds of square kilometers.

Map Editor

The Blueprint Utility Widget includes a **Map Editor** tab containing a set of tools built directly in Blueprint, others linked to HDA parameters. These tools streamline the editing process and automate common HDA management tasks that occur during in-level editing.

This example setup automatically creates and tags curves for the HDA, and exposes **World Manager** parameters that control how the curves are processed and interact within the HDA.

PDG Manager

If PDG is the preferred output method, you can select **TOP network nodes** from the widget's drop-down menu. This triggers the cooking of node dependencies. The three available nodes are designed to accelerate the level editing workflow.

PDG enables more granular control over the HDA's cooking process which allows you to cook only what's needed (e.g., terrain and rivers, but not trees or grass) and limit cooking to specific parts of the level. For example, editing a small wall curve in one corner of the map shouldn't require recooking all walls across the world.

Available Output Modes:

- **Edit Node:** Cooks only the terrain, allowing you to move curves and see which items are marked dirty and need future recooking.
- **Preview:** Cooks most asset placements and terrain while skipping heavy processing to save time.
- **Final:** Performs a full cook of all outputs used when baking the final results.

WORKFLOW

(Assuming Houdini Engine is installed)

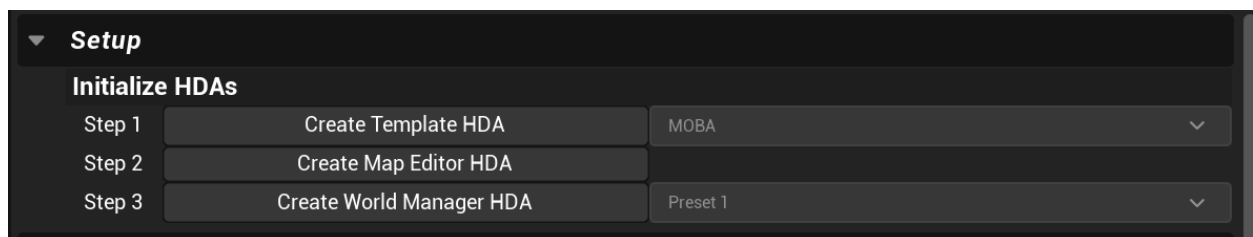
Create a new level

1. Open the HDA World Manager

To open the Editor Utility Blueprint, navigate to

[HoudiniGames_MOBA/Blueprints](#), right-click **HDA_World_Manager**, and select **Run Utility Widget**.

This will open the World Manager panel in your Unreal editor tabs.

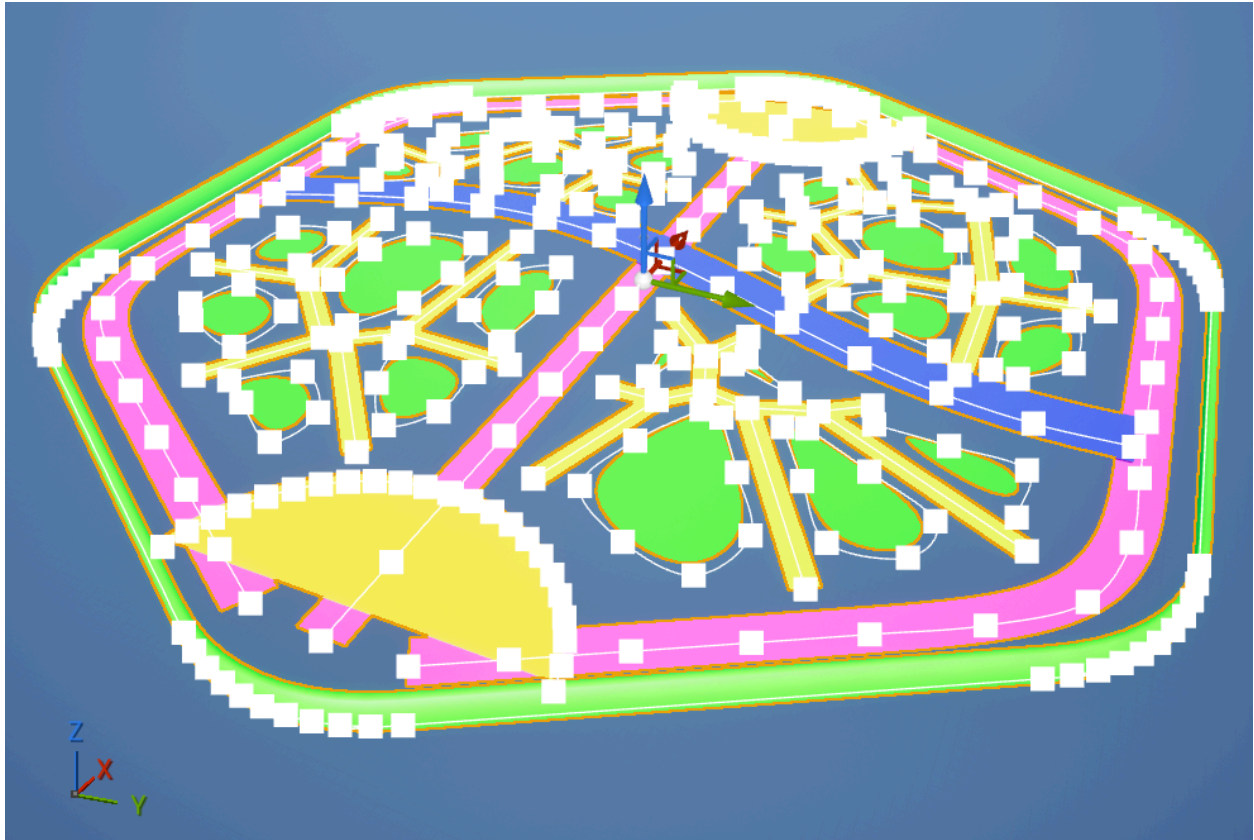


2. Create the Template HDA

With a new world opened, navigate to the World Manager widgets **Setup** tab, click **Create Template HDA**.

This may take a moment the first time, as it initializes a Houdini session and sets up the HDA.

Once complete, your map should resemble the image below.



Normally, you would adjust HDA parameters in the **Details** panel under **Houdini Parameters**, and each change would trigger a recook. However, the World Manager widget allows you to modify HDA parameters directly within the **Template** tab.

This provides a streamlined, user-friendly interface for editing the HDA, while still allowing designers to make pre- and post-cook changes within a familiar Blueprint environment.

▼

Template

MOBA Template HDA Parameters

Lane Count	3
Map Size	1.0
Layout	4
Treeline Ratio	20.0
Treeline Seed	1

Rebuild Curves

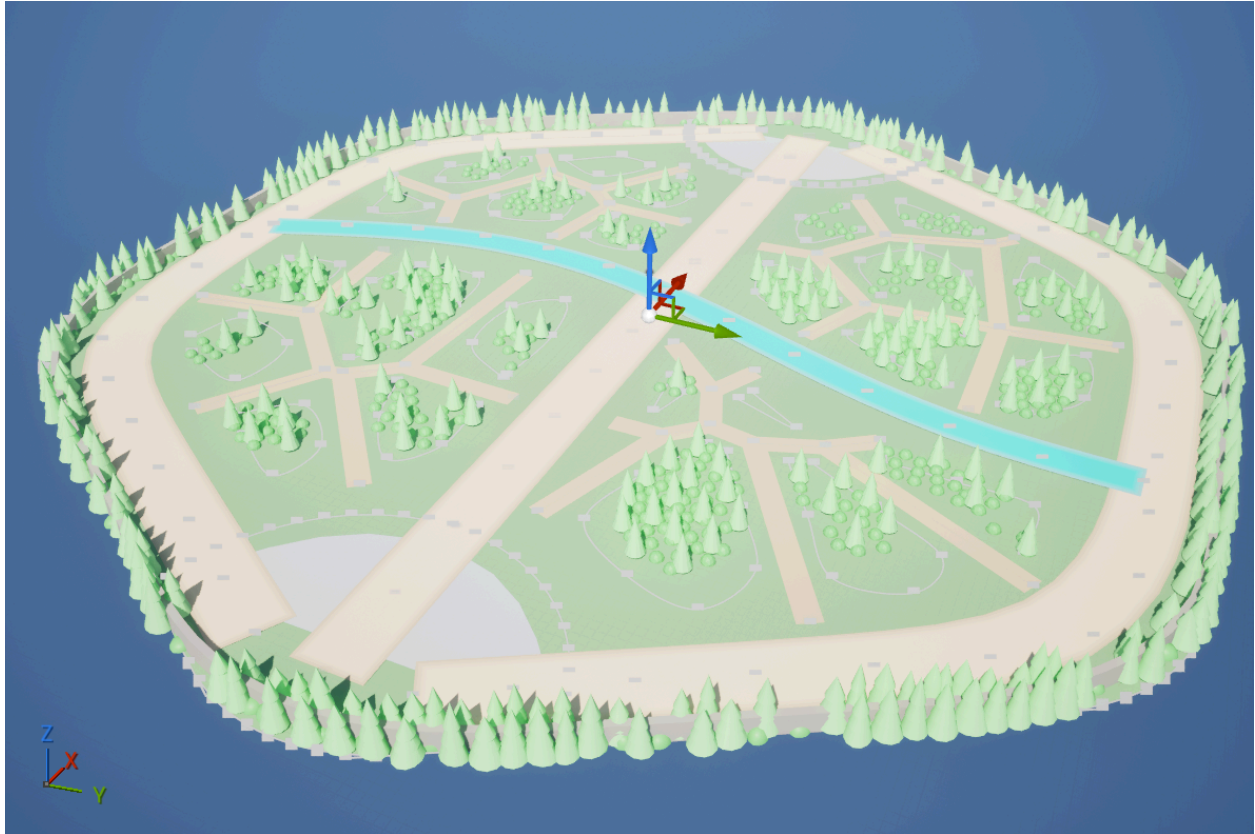
3. Create the Map Editor HDA

After finalizing your Template settings, click **Create Map Editor HDA** in the **Setup** tab.

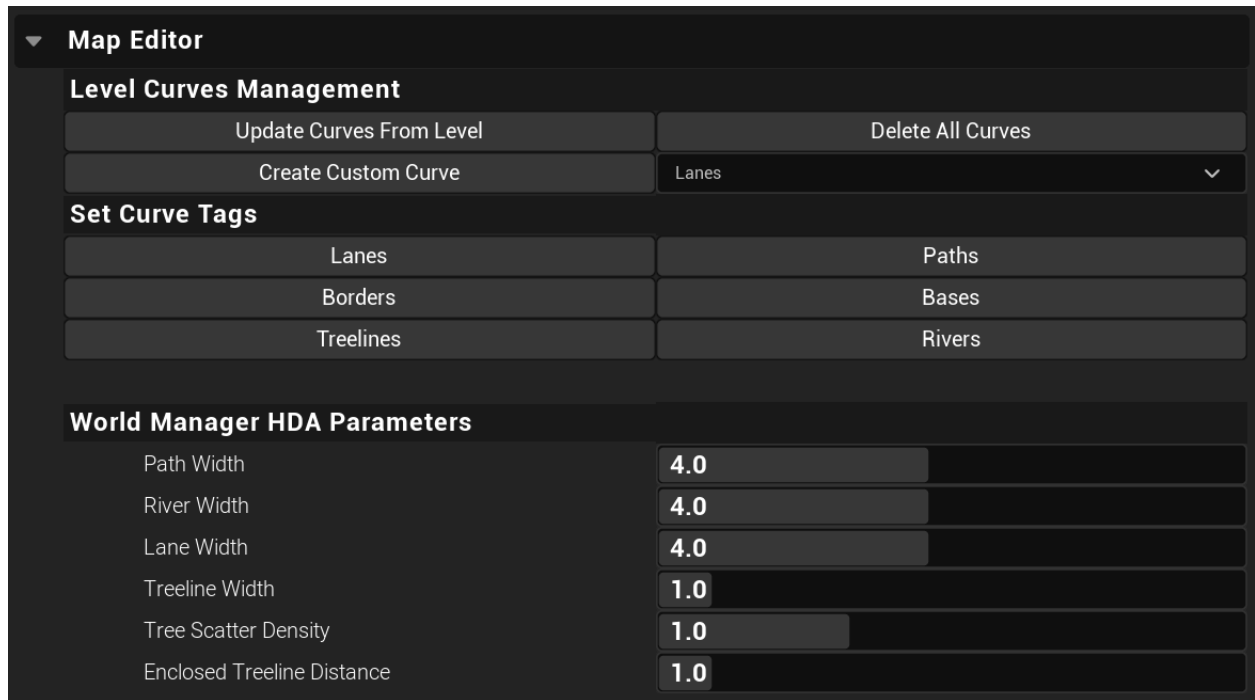
This action bakes the Template HDA results into your level—generating curves and assigning the correct tags.

These tags are automatically referenced in the Blueprint to link curves with the corresponding inputs on the **Preview HDA**.

Your level should now appear as shown below.

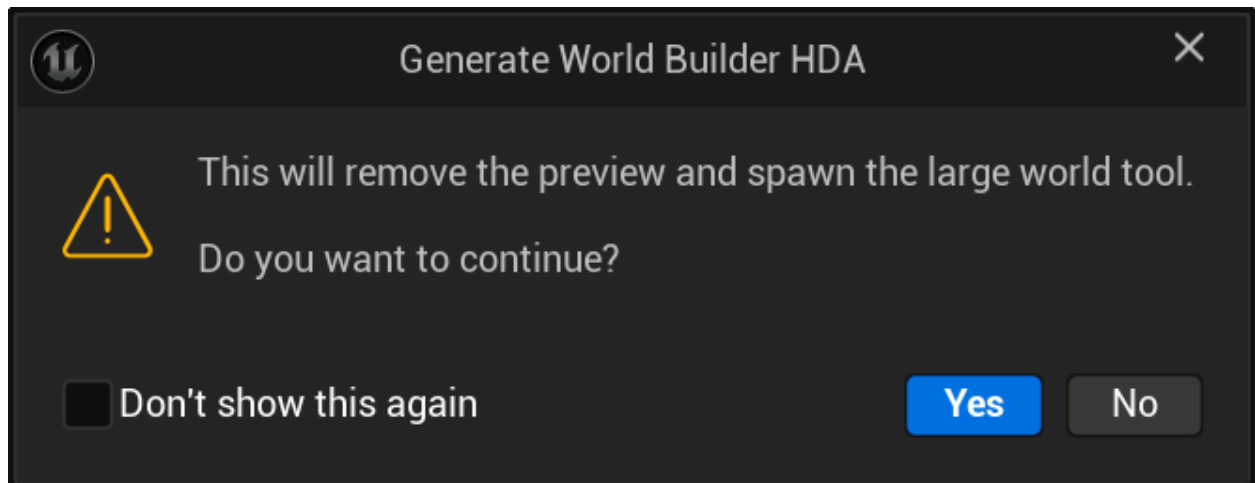


At this stage, you can freely edit level curves to quickly visualize how those adjustments will affect the final cook. The Preview HDA serves as a lightweight version of the full HDA, enabling faster iteration during editing.



4. Create the Final World Manager HDA

Once the preview edits are satisfactory, press **Create World Manager HDA** in the **Setup** tab. A confirmation window will appear—click **Continue** to proceed.



This process may take some time (hence the use of lightweight previews during editing). The tool will remove the Preview HDA and instantiate the final version. Like before, the Blueprint uses curve tags in the level to automatically connect HDA inputs.

When complete, your level should look like this.

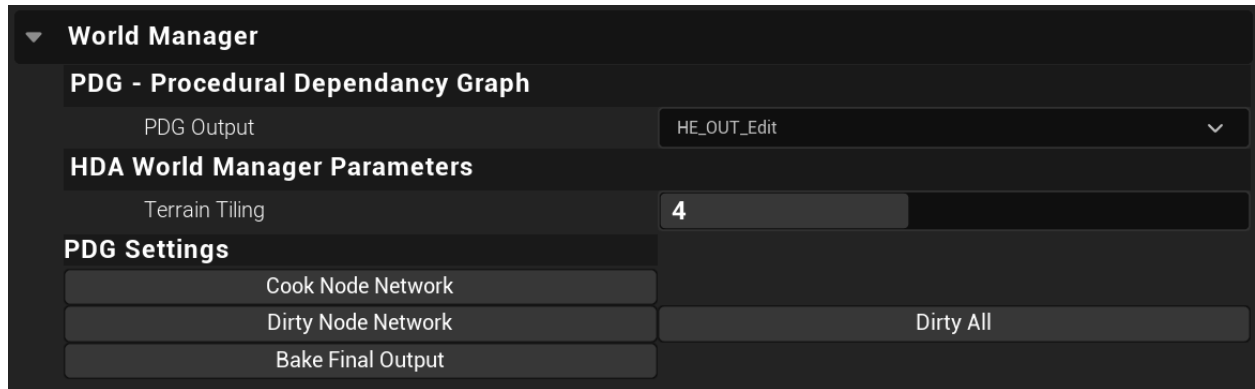


Note: You can continue editing the world's curves even after the final HDA is created. The generated meshes and assets remain responsive to curve changes. you can still add, copy, or delete curves manually or through the widget.

5. Using the World Manager Tab

The final tab of the Utility Widget—**World Manager**—provides additional control over how HDAs are processed.

you can choose between **SOP (Surface Operator)**-based cooking or **PDG (Procedural Dependency Graph)** workflows. This widget directly edits the PDG outputs.



6. Bake the Final Output

When ready, click **Bake Final Output**.

This will create baked actors inside your **Content Browser**, typically under:

[/Game/HoudiniEngine/Bake/](#)

These baked actors are **no longer linked** to the HDA, making them ideal for collaborative workflows where not all team members need to recook assets locally.

However, the HDA remains in the level—ready to be cooked and rebaked as needed.

Each rebake will automatically replace the previous actors, ensuring a **non-destructive workflow**.

Closing Thoughts

This project serves as an **inspirational pipeline** for world-building teams, demonstrating how Houdini Engine and Unreal Blueprints can complement each other.

While Houdini Engine excels at data exchange between Houdini and Unreal, many input and output processes typically require manual work. Blueprints help automate and streamline these steps, creating a smoother experience for artists.

Unreal offers excellent tools such as **Editor Blueprints**, **Editor Utility Widgets**, **Scriptable Tools**, and now **PCG**—all of which can enhance Houdini Engine workflows or even handle parts of the process directly.

The MOBA tool pipeline presented here is intended as inspiration for technical artists rather than a strict template. It demonstrates useful practices and potential workflows, but some implementations were designed specifically to meet this project's needs and may not apply universally.

