

Shops Generator (HDA + Unreal Interior Mapping Shader)

Asset Overview

Asset Name	Shops Generator
Category	Combined Procedural System
Unreal Version Tested	5.6
Version	Houdini Commercial 20.5.684
Houdini Version Tested	Houdini Commercial 20.5.684
Author	Amro Ahmed

Requirements

This asset is a Houdini Digital Asset, not a standalone Unreal asset. Running it requires all of the following:

- A licensed installation of Houdini (Commercial), build 20.5.684.
- The Houdini Engine for Unreal plugin, built against the same Houdini version. Houdini Engine is strict about the plugin build matching the Houdini build, so other versions may fail to load the asset.
- Unreal Engine 5.6 or above.
- Interior cubemaps of your own. Only a single example cubemap is included, provided so the material setup can be inspected and reproduced.

What This Asset Is For

This asset was made to make the process of making shop-heavy environments in game engines easier and less time-consuming for 3D artists by making it possible to place any number of shops along a curve input with endless variations of the distribution of the shops' materials.

What This Asset Is NOT For

This asset is not made to be used outside the Unreal Engine ecosystem. The shopfronts rely on an interior mapping shader to read as real shops; used anywhere else, the output is a flat facade with no depth behind the glass, which is not the intention of the asset.

It has two limitations when used along with interior mapping shaders:

- If used with the regular interior mapping material function in Unreal, the shader will break unless the camera vector transformation in the material function is set to Local instead of Tangent.
- When used with the interior mapping shader, the curve must be kept straight. The interiors break after the first corner, and the amount of distortion increases with the angle of the corner.

Inputs & Outputs

Index	Type	Description
0	Default Geometry Input	Pillars/Walls don't need a special input to be spawned, except for placing the HDA in the scene
1	Curve/Spline Input	A curve to define a path for the shops to be distributed on

Parameters

Parameter	Type	Description	Notes
Type	Ordered Menu	Choosing one of the two types that the asset provides (Pillar or Modular Shop)	
Height	Float	Controlling shops and pillars height	Should match the room height expected by the interior mapping material, otherwise the faked interior will not line up with the geometry
Shops Type	Ordered Menu	Controlling whether the shops width along the curve is set per segment (Free Size) or subdivided into a fixed count between points (Fixed Size)	Active only if “Type” is set to “Modular Shop” “Free Size” makes one shop between every 2 points, while “Fixed Size” makes a specific number of shops between every two points
Add Pillars	Boolean (Toggle)	Controlling whether shops have in-between pillars or not	Active only if “Type” is set to “Modular Shop”
Shops Number	Integer	Controlling number of shops between every two points in the spline	Active only if “Type” is set to “Modular Shop” and “Shops Type” is set to “Fixed Size”
Pillars Spacing	Float	Controlling the pattern of placing the pillars every 1, 2, or 3 shops etc.	Active only if “Type” is “Modular Shop”, “Shops Type” is “Fixed Size” and “Add Pillars” is ticked
Random Materials Seed	Float	Controlling the randomization seed of the materials throughout the shops. Changing the seed reshuffles which material lands on which shop without changing the set of materials used	Active only if “Type” is set to “Modular Shop”
Random Shops Materials	Multiparm (Integer)	Controlling the number of materials to be randomly distributed along the shops. Add as many entries as needed; each entry adds one material to the random pool	Active only if “Type” is set to “Modular Shop”
Value #	String	The material assigned to this entry in the random pool	Sub-parameter of Random Shops Materials
Weight #	Float	How often this material is picked relative to the others in the pool. A higher weight means the material appears on more shops	Sub-parameter of Random Shops Materials
Pillar/Wall Size	Vector 2 (Float)	Controlling the depth & width of the generated wall or pillar	
Pillar/Wall Material	String	The material input for the generated wall or pillar	

Interior Mapping Material Parameters

These parameters live on the Unreal material instance, not on the HDA. They control how the faked interior reads once the generated geometry is in the engine.

Parameter	Type	Description
Cubemap	Texture	The interior cubemap sampled to fake the room behind the glass. Only one example cubemap is supplied with this asset
Tiling X	Float	Horizontal tiling of the interior projection
Tiling Y	Float	Vertical tiling of the interior projection
Randomize Rotation (SB)	Static Bool	Randomizes the rotation of the interior so repeated cubemaps read as different rooms
Glass Roughness	Float	Roughness of the glass surface in front of the interior
Use Frame Texture	Bool	Enables the shopfront frame texture set
Frame Base Color Map	Texture	Base colour of the shopfront frame and signage
Frame ORM Map	Texture	Packed Occlusion / Roughness / Metallic map for the frame
Frame Emission Map	Texture	Emissive mask for lit signage on the frame
Emission Intensity	Float	Brightness multiplier for the frame emission
Use Layered material	Bool	Enables a second cubemap layered in front of the main interior
Layered Cubemap	Texture	The layered cubemap, used for foreground elements such as a counter or window display (this cubemap should have a mask in alpha channel)
Depth	Float	Depth offset of the layered element relative to the main interior
Tiling/Offset	Vector 4	Tiling and offset of the layered texture

Required Setup in Unreal

The generated geometry is only half of the asset. Without the interior mapping material applied in Unreal, the shopfronts render as plain walls.

- Make a material instance from the provided master material (it sets the camera vector transformation in the interior mapping material function to Local. With the default Tangent setting the interior projection will not resolve correctly on the Houdini geometry.)
- Keep the input curve straight. A curve that is straight but rotated as a whole works correctly; it is only a change of direction partway along the curve that breaks the interiors.
- Supply your own interior cubemaps. Only one example cubemap is included in the material instance shipped with this asset, provided so the setup can be inspected and reproduced.

Why the curve must stay straight

With the camera vector set to Local, the interior projection is evaluated in the object's local space and shares a single orientation across the whole asset. Rotating the entire curve rotates the geometry and the projection together, so the interiors stay correct. Introducing a corner leaves one section of geometry angled relative to the object's local axes while the projection orientation stays fixed, so the interior of that section is sheared by the angle of the corner. This is a consequence of the local-space projection rather than a fault in the generated geometry.

To build a shopfront run that turns a corner, place a separate instance of the asset for each straight section rather than bending a single curve.

Performance Notes

Because the shop interiors are faked entirely in the shader, the generated geometry itself stays light. The rendering cost sits in the interior mapping material and scales with the number of unique cubemaps in use rather than with the number of shops, so reusing a small set of cubemaps with Randomize Rotation enabled is cheaper than authoring a unique cubemap per shop.