SIMPLE
PAC

Main features

Simple Procedural Asset Creator allows you to create simple asset generators procedurally, using a universal node-based system, keeping all the workflow in-editor.
Main features:

- Possibility to create various types of simple asset generators
- 20 example asset generators included (+ Example how-to maps)
- Intuitive node-based system
- Easy spline shape based mesh generation nodes
- Large set of exposed parameters for mesh generation rules
- Custom parameter system with formulas, bool conditions and more
- Procedural mesh decals
- Angle-based mesh smoothing (+ All Smooth & All Flat options)
- HDRI lighting setup with many customization options
- Vertex Color control for easy material colorization
- Node instancing & ‘variations’ system
- Assets can be saved as a regular Static Meshes & exported as .FBX or .OBJ
- Full Blueprint project, clean & easy to extend
- … More, all described in this documentation.

Quick start guide

📗 **Simple P.A.C. comes as a standalone project and has most of important things already prepared for you (viewport layout, colors and other useful project settings).**

How to quickly check the included content and see it working:

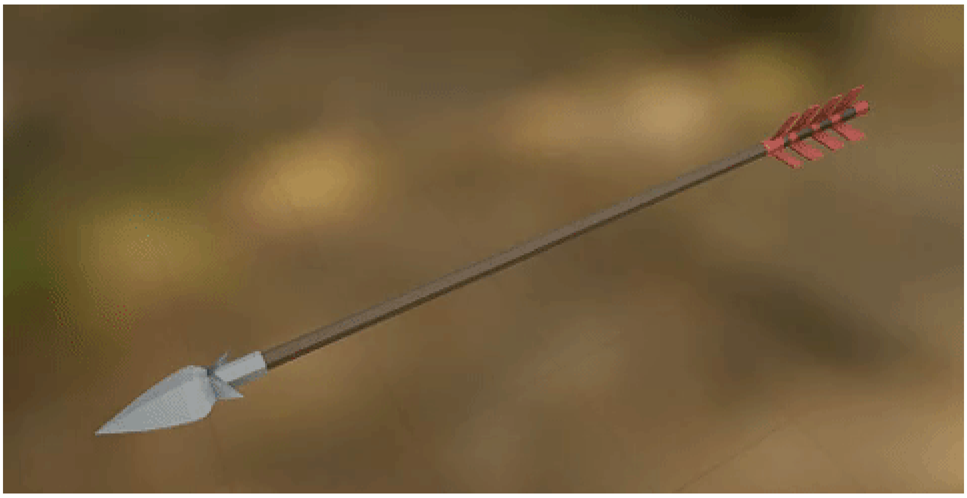
1. Open any generator map, e.g. the /Maps/BookGenerator/BookGenerator1_FlatShaded map
2. Click Perspective → ASSET in the upper left pane
3. Select the cogwheel icon and click Randomize to see it in action.

📗 **It might be possible to avoid the ASSET view setting step, if this bug is fixed. It's already reported, but no issue tracker is available yet.**

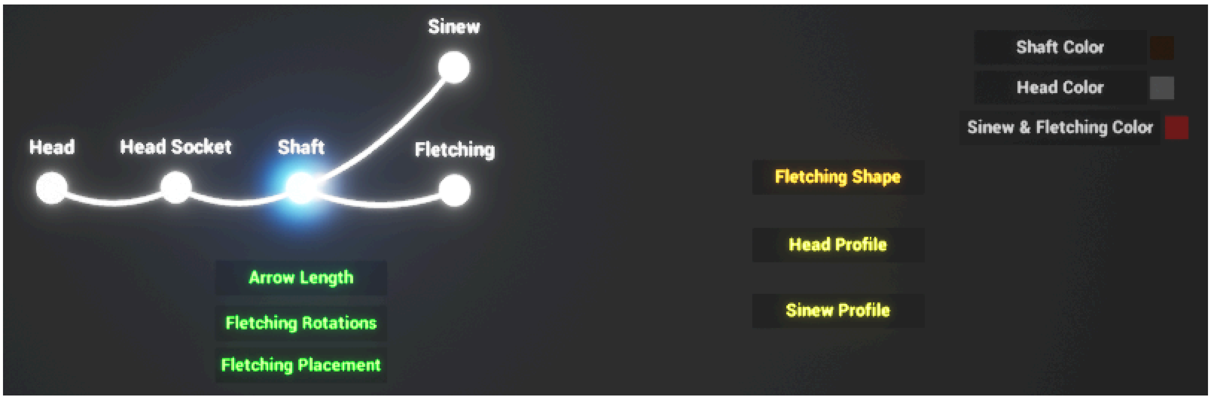
… You can check any included generator or example map this way.
A next good step would be to look at this documentation further or play with the examples.

Included example generators

Arrow Generator



How it's built:

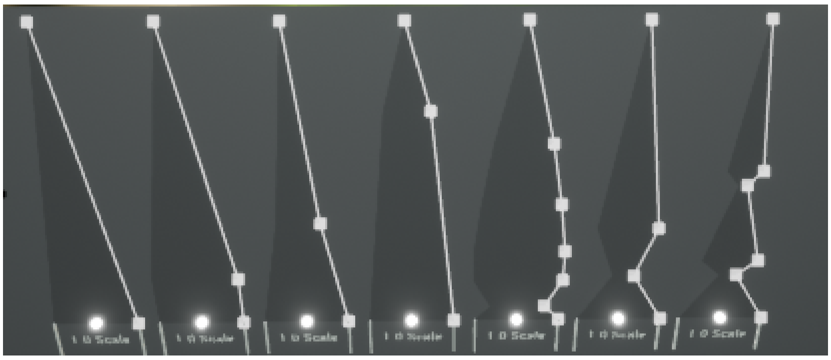


(Click on the image for full sized version)

Shaft: Spline Shaped Mesh (SSM) with simple straight Path and circle Shape.
Head Socket: Same as Shaft, but with Path Depth Start at 0.95 and smaller Shape Scale Offset.
Head: SSM with non-planar Shape to make the head arrow-like. It shows the possibility of having non-planar Shapes. It has simple straight Path and it's placed on Head Socket's Placement Spline, at 1.0 time. It also uses a Profile to define its shape and make it sharp.
Sinew: SSM with Path Depth End at 0.22 and a Profile to make it look like it's segmented.
Fletching: Contour Shaped Mesh, placed on Shaft's Placement Spline at 0.02 time. It uses one of the Fletching Shapes and instantiates itself 3 times, using Instance Rotation X with values -40, 90, 220 to make it properly angled.

Arrow Length: Controls X location of second point of main Path.
Fletching Rotations: Uses value pool with values: -40, 90, 220. It's assigned to Fletching's Instance Rotation X value, so the Fletching can instantiate itself on these rotation angles.
Fletching Placement: 0.07 value, assigned to Placement Spline Time of Fletching.

Fletching Shape: Random pool of 5 different shapes for Fletching.
Head Profile: Random pool of 7 different Profiles for Head.
Sinew Profile: Random pool of 3 different Profiles for Sinew.
Shaft/Head/Sinew Color: Random pool of different Colors for Shaft, Head & Sinew.

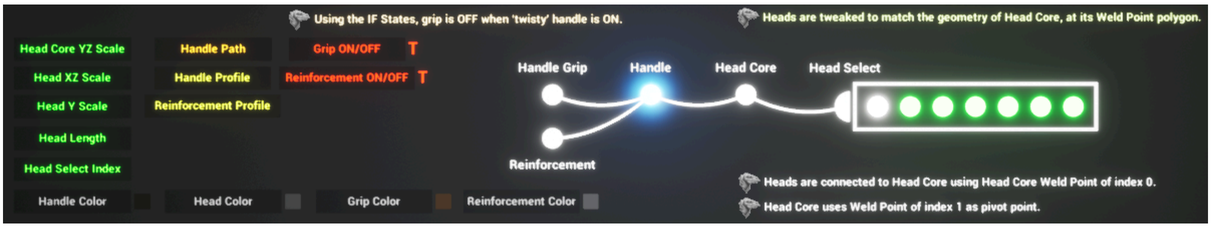


Example: Arrow’s Head Profiles

Map: /Maps/ArrowGenerator/ArrowGenerator1_FlatShaded

Axe Generator

How it's built:



Using the IF States, grip is OFF when 'twisty' handle is ON.

Heads are tweaked to match the geometry of Head Core, at its Weld Point polygon.

Heads are connected to Head Core using Head Core Weld Point of index 0.

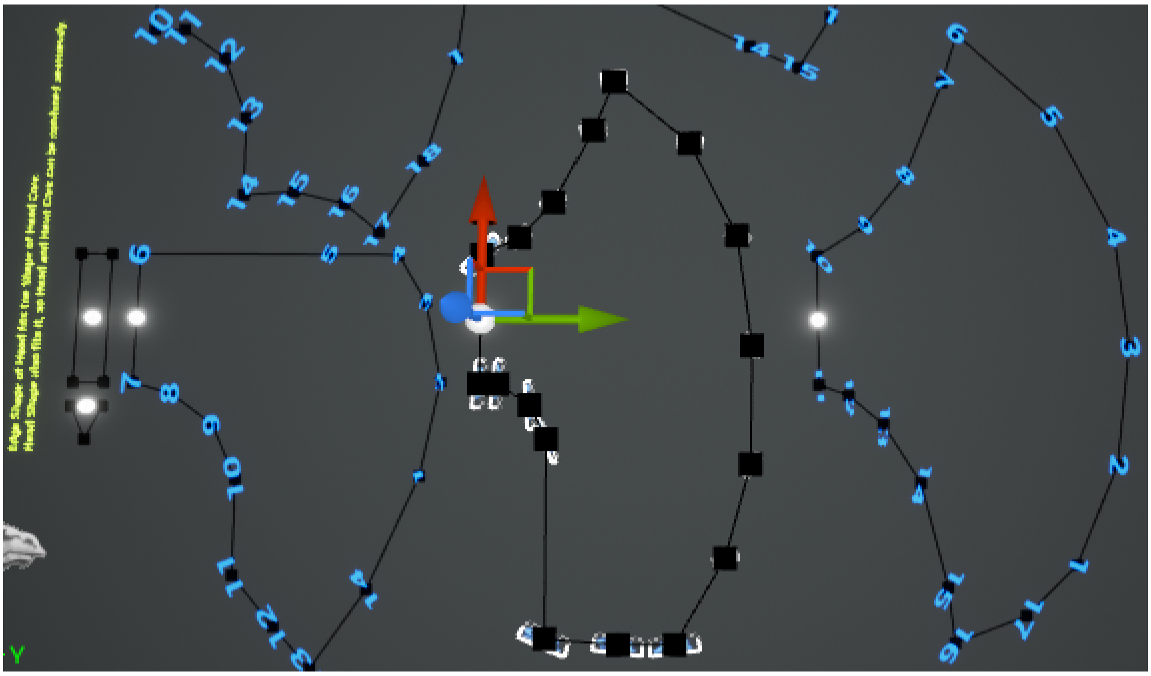
Head Core uses Weld Point of index 1 as pivot point.

(Click on the image for full sized version)

Handle: Spline Shaped Mesh (SSM) with handle-shaped **Path** and **Profile**.
Handle Grip: Same as Handle, but with less **Shape Scale Offset** value and **Path Depth Start** at **0.2**, **Path Depth End** at **0.5**.
Reinforcement: Same as **Handle**, but with custom **Profile** to make it appear like it's segmented, with **Path Depth Start** at **0.8** and **Path Depth End** at **0.9**.
Head Core: SSM placed on **Handle's Placement Spline** at **0.98** time. It's profiled to make it fit the handle and it has **Shape** that can seamlessly integrate with Head's (**Head Select**) Shape - more info in the comments.
Head Select: Select node with different axe heads to pick. Heads are Contour Shaped Meshes, where **Edge Shape** fits the **Shape** of **Head Core**. More info in the comments.

Head Core YZ Scale: Controls Y and Z scale of Head Core.
Head XZ Scale: Controls X and Z scale of axe heads. Uses a Formula with "**(C * 0.2)**", where C is **Head Core YZ Scale**. This way it always fits current **Head Core**, no matter what scale it has.
Head Y Scale: Controls Y scale of axe heads. Uses a Formula with "**(X * Length)**", where X is **Head XZ Scale** and Length is **Head Length**. This makes it always proportional and allows to control head length with separate **Head Length** parameter.
Head Length: Set to random value between **0.8** and **1.4**, used in the **Head Y Scale** parameter.
Head Select Index: Random value between **0** and **6**, used in the **Head Select** node to pick random Heads from it.
Grip ON/OFF: It uses IF Conditions to turn the grip on/off when needed. More info in comment.

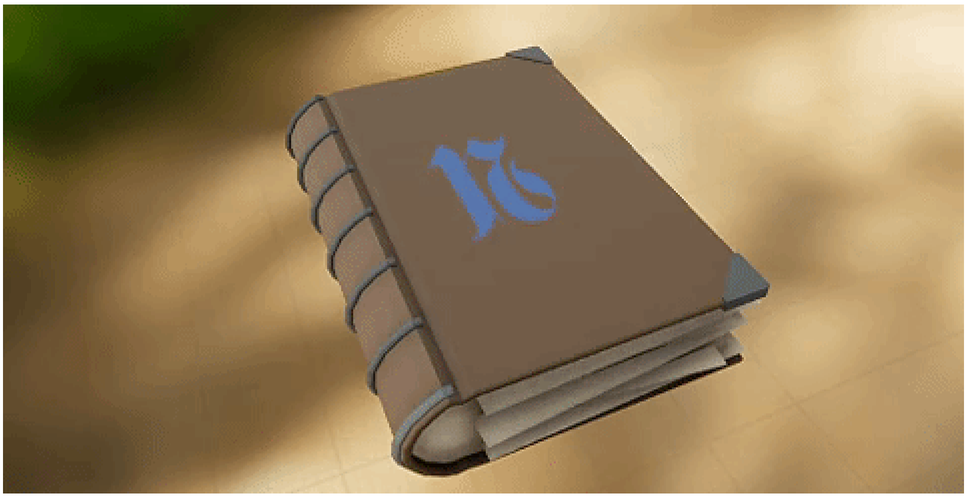
Reinforcement ON/OFF: Random bool value, to make the **Reinforcement** appear randomly.
… There are also the **Color**, **Material** & **Shape** parameters with random value pools.


Edge Shape of Head like the Shape of Head Core.
Head Shape when fits it, so Head and Head Core can be combined seamlessly
Y

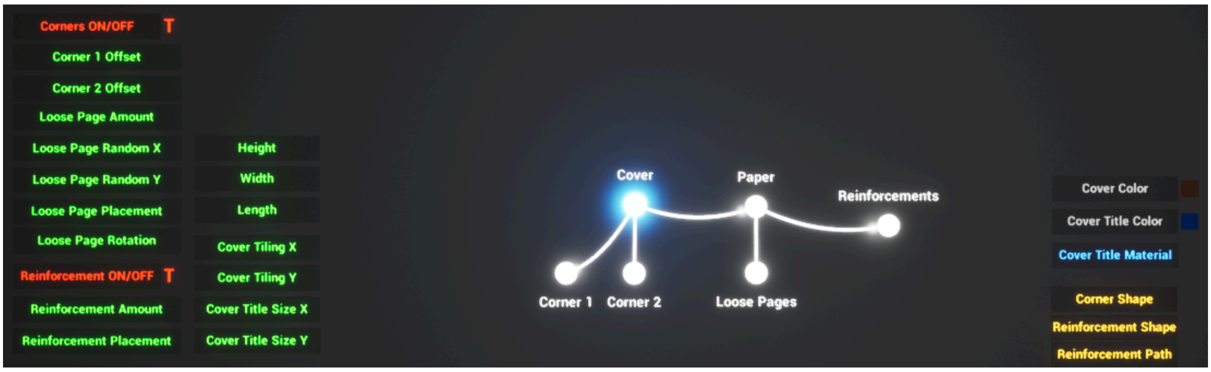
Example: Axe Head Shapes

Map: **/Maps/AxeGenerator/AxeGenerator1_FlatShaded**

Book Generator



How it's built:



(Click on the image for full sized version)

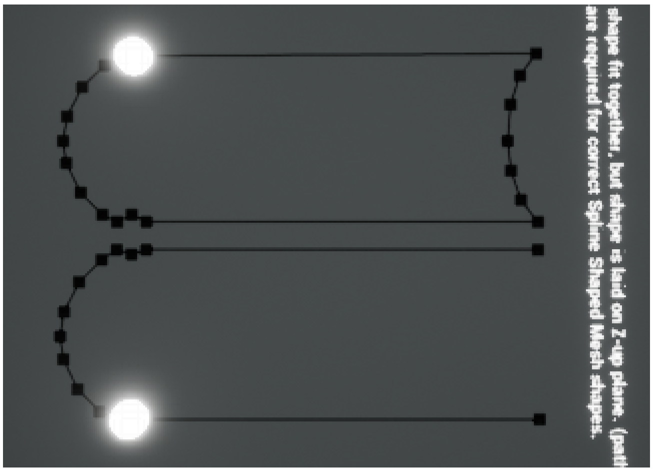
Cover: Spline Shaped Mesh (SSM) with simple rectangle Shape and cover-shaped Path.
Paper: SSM with Shape that fits Cover's Path and simple straight Path.
Reinforcements: Instanced on Paper's Placement Spline, SSM with custom Shape & Path.
Loose Pages: Adds loose pages to the book, it's an SSM with custom Shape and the same Path as Paper's Path. Instanced on Paper's Placement Spline.
Corner 1, Corner 2: Contour Shaped Meshes with custom contour Shapes, placed on Cover's Placement Spline at 1.0 time. Y location offset based on current book Length.

Height: Controls book height, by default set to random range 0.3 - 0.6.
Width: Controls book width, by default set to random range 0.47 - 0.53.
Length: Controls book length, by default set to random range 0.45 - 0.55.
Cover Tiling X/Y: UV tiling for Cover, set to random range 1.0 - 1.5.
Cover Title Size X/Y: Size of title symbol on Cover, set to random range 8 - 15.
Corner Offset X/Y: Y location offset for Corner 1 & 2, to make them fit current book length. Uses a Formula with "(L * 48)", where L is the Length parameter.
Loose Page Amount: Amount of Loose Pages node instances, set to random range 0 - 10.
Loose Page Random X/Y/Placement/Rotation: Random location & rotation offsets for Loose Pages node instances, to make the loose pages appear more natural.

Reinforcement Amount: Amount of Reinforcements node instances, set to random range 2 - 15.
Reinforcement Placement: Controls Reinforcements node instance placement along spline.

… There are also the ON/OFF parameters for Corners and Reinforcements (so they appear randomly) and random Color, Shape & Material pools for different asset parts.

Book generator also uses Global Mesh Decal for the title symbol.

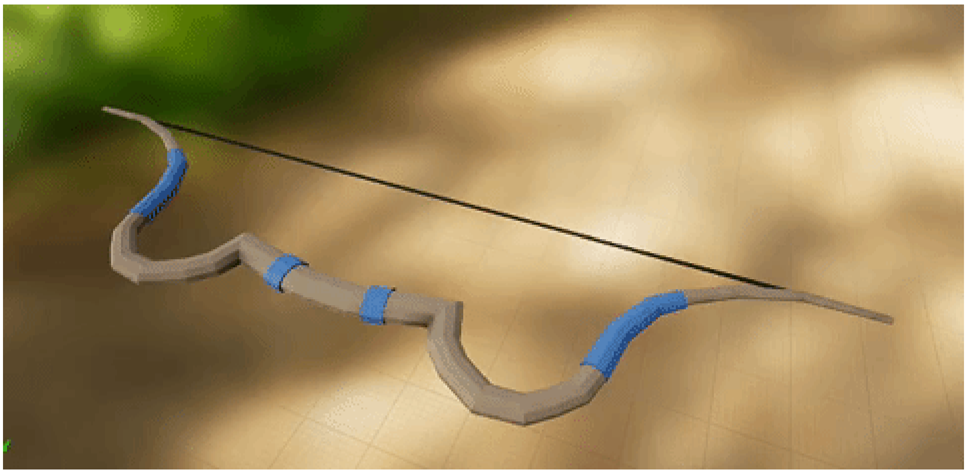

shape fit together, but shape is laid on Z-up plane. (path
are required for correct Spline Shaped Mesh shapes.

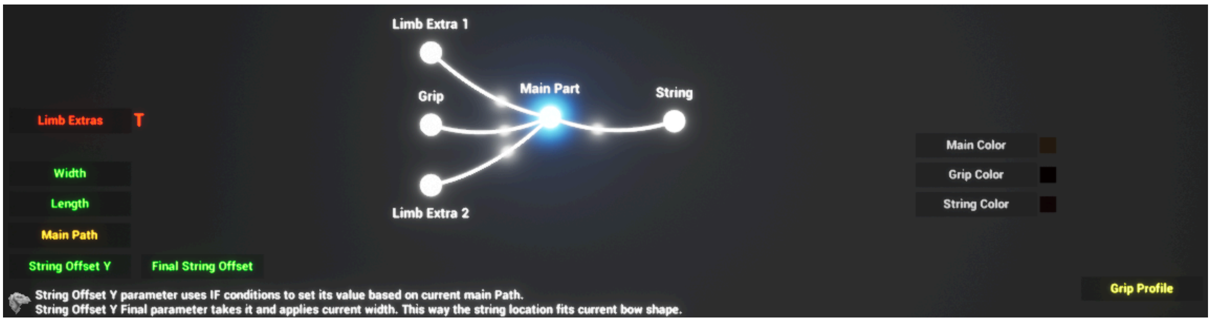
Example: Book Paper Shape & Cover Path

Map: /Maps/BookGenerator/BookGenerator1_FlatShaded

📗 … Rest of the generators work similarly, all based on the parent-placement methods or shape-fitting. More information available in the Main Concepts.
See the included generator maps to have access to all details, to see how it's made.
In case of any questions, feel free to ask on the support email.

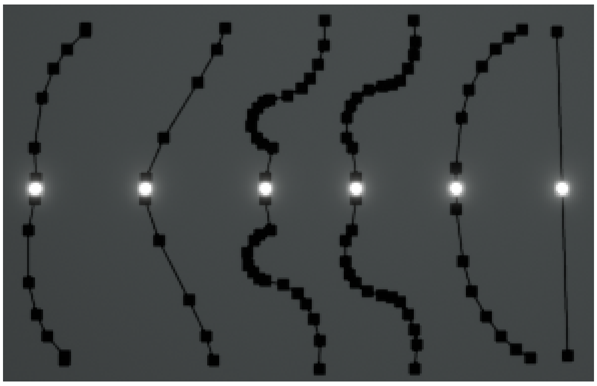
Bow Generator





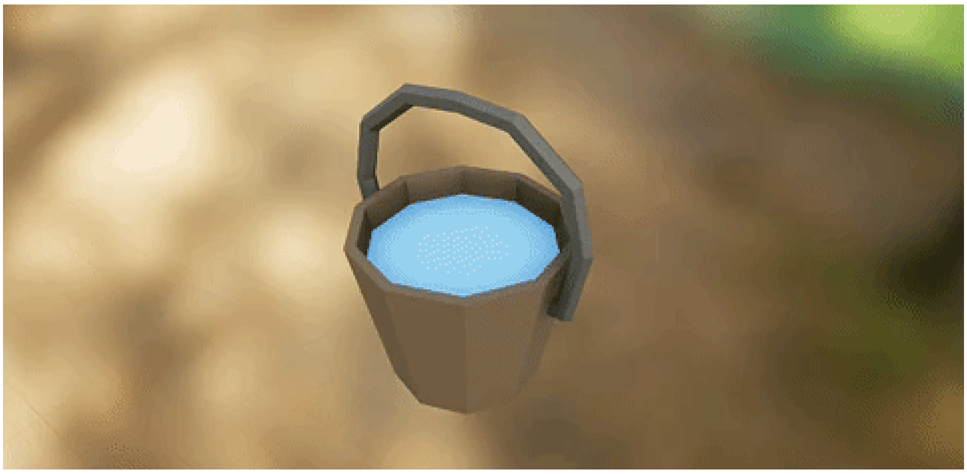
String Offset Y parameter uses IF conditions to set its value based on current main Path.
String Offset Y Final parameter takes it and applies current width. This way the string location fits current bow shape.

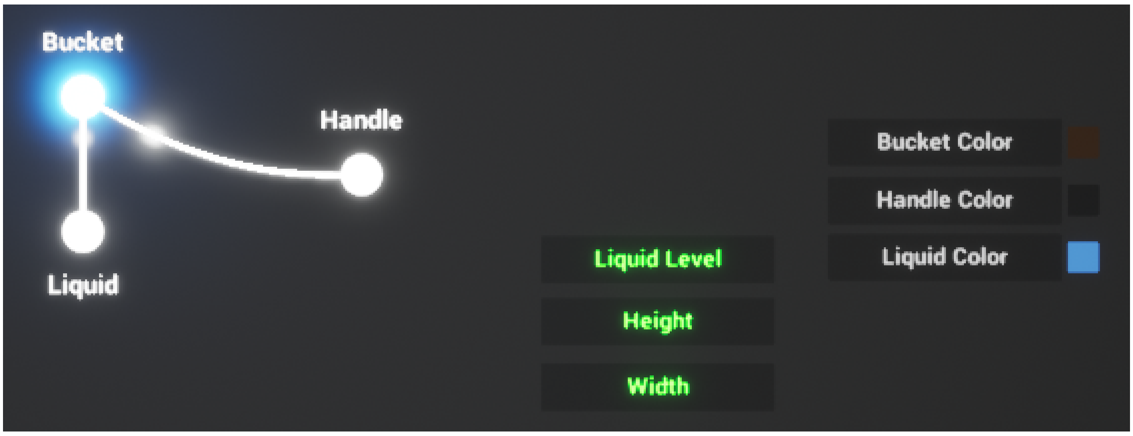
(Click on the image for full sized version)

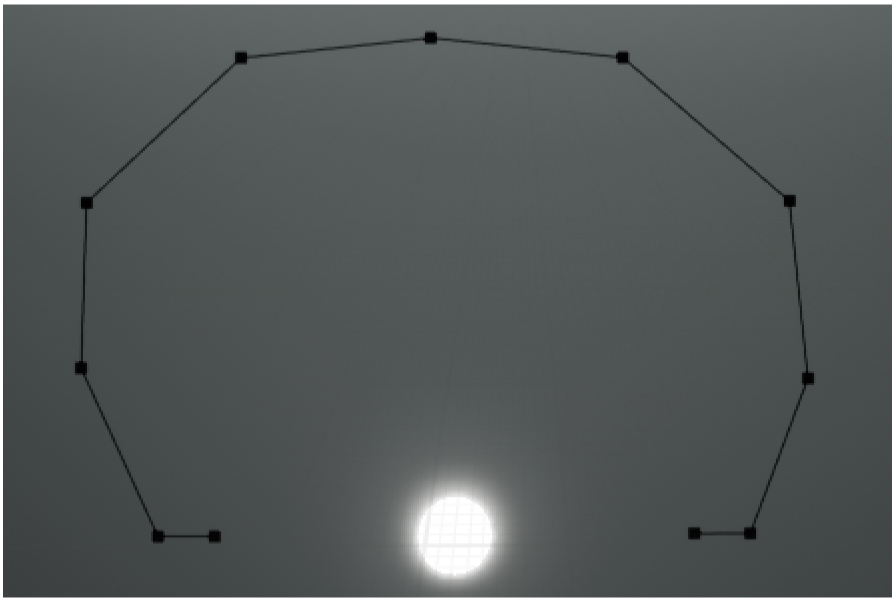


Map: /Maps/BowGenerator/BowGenerator1_FlatShaded

Bucket Generator



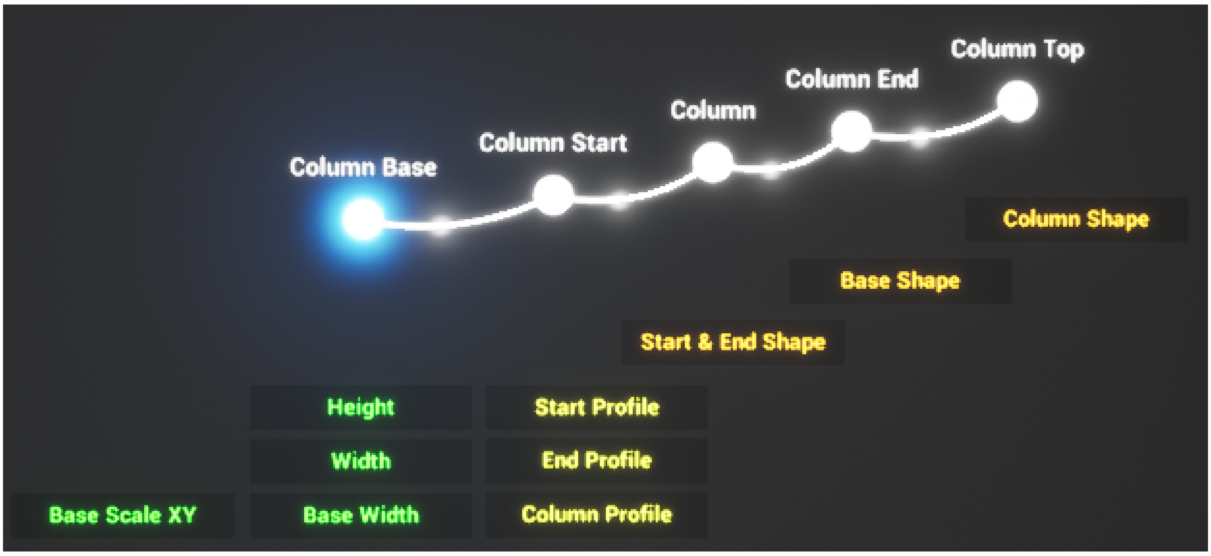


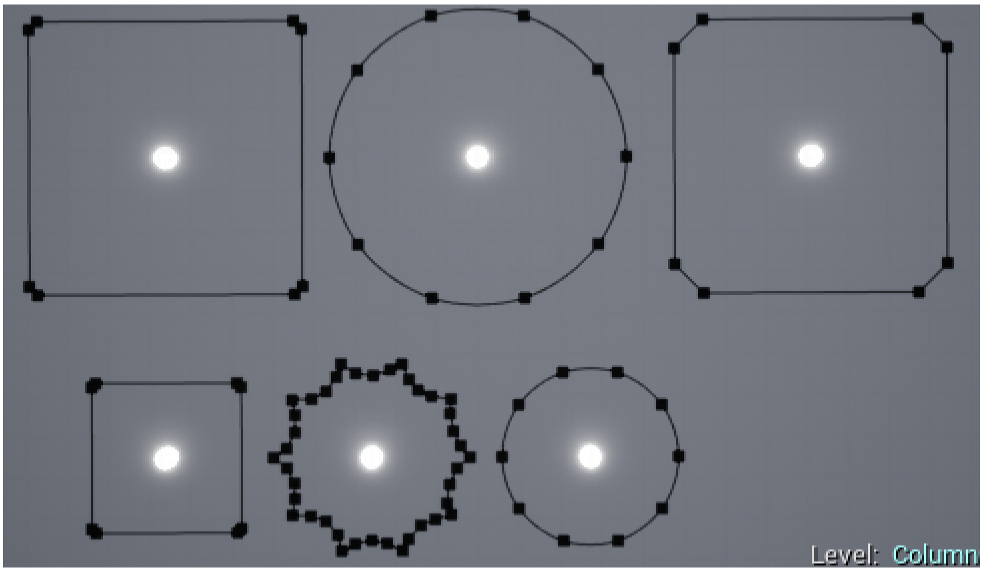


Example: Handle Path

Map: /Maps/BucketGenerator/LowPolyBucketGenerator

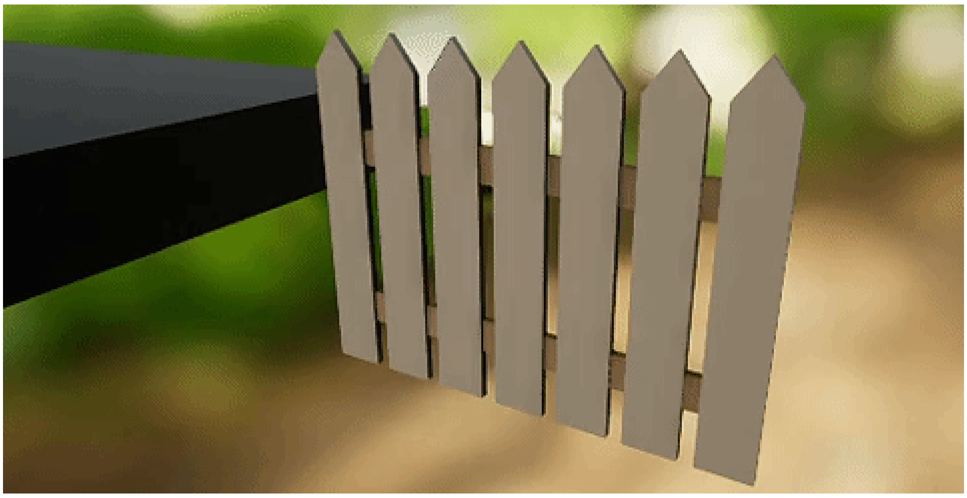
Column Generator

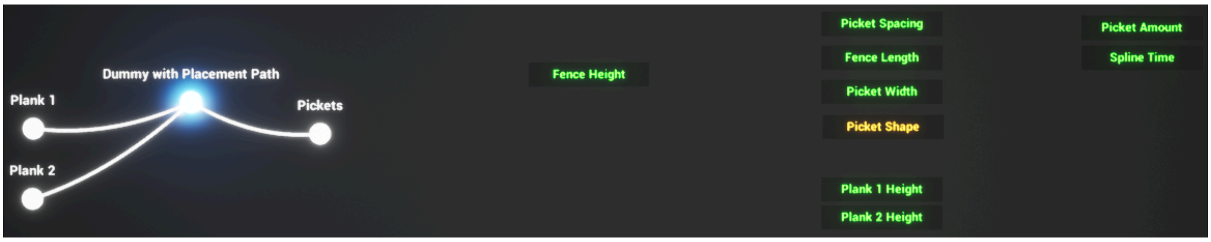




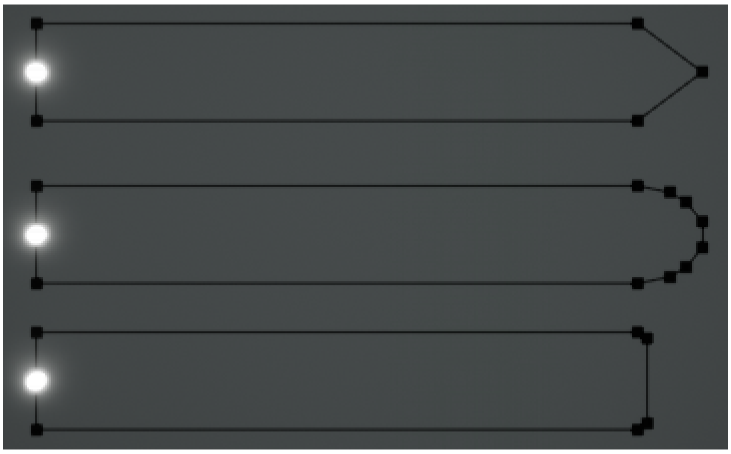
Example: Column Shapes for main Spline Shaped Mesh.
Map: /Maps/ColumnGenerator/ColumnGenerator1_FlatShaded

Fence Generator





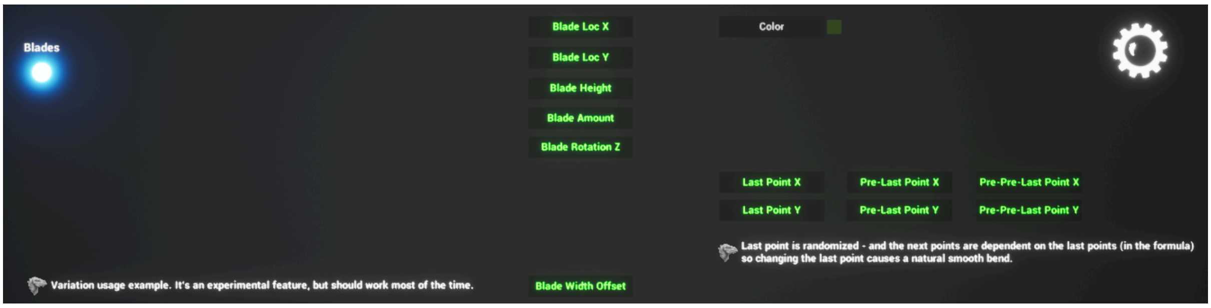
(Click on the image for full sized version)



Example: Fence Picket Shapes.

Map: /Maps/FenceGenerator/FenceGenerator1_FlatShaded

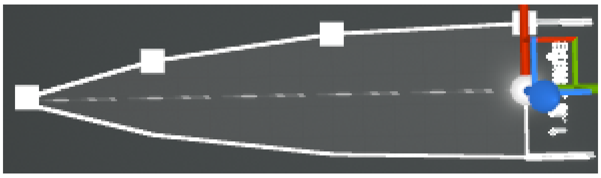
Grass Generator



Last point is randomized - and the next points are dependent on the last points (in the formula) so changing the last point causes a natural smooth bend.

Variation usage example. It's an experimental feature, but should work most of the time.

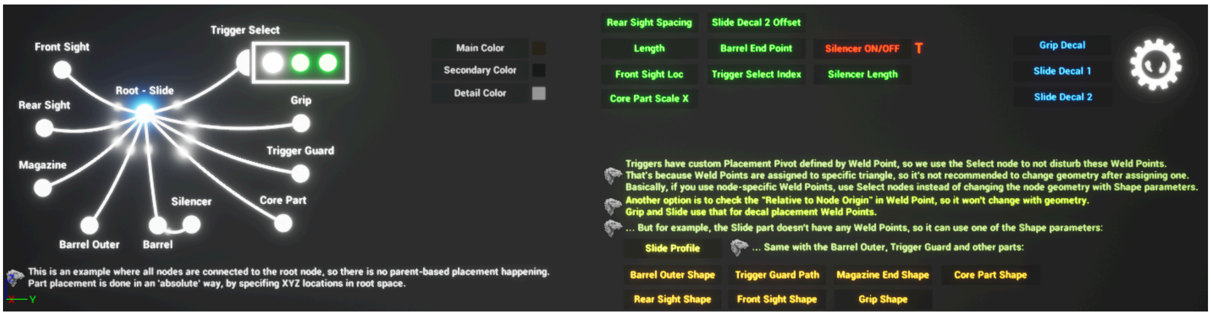
(Click on the image for full sized version)



Example: Profile for grass blades.
Map:
/Maps/GrassGenerator/GrassGenerator_FlatShaded

This generator serves as an example of the Variations feature, where each blade is different.

Gun Generator



This is an example where all nodes are connected to the root node, so there is no parent-based placement happening.
Part placement is done in an 'absolute' way, by specifing XYZ locations in root space.

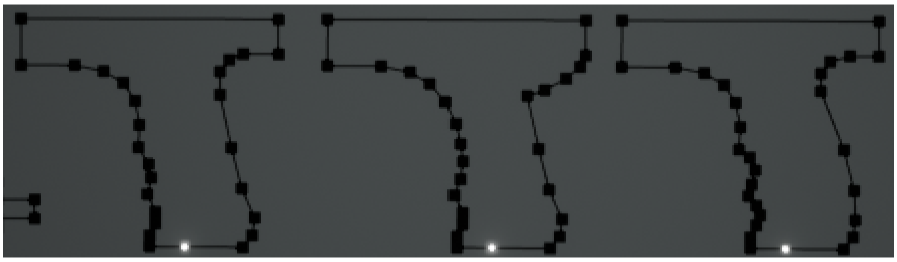
Triggers have custom Placement Pivot defined by Weld Point, so we use the Select node to not disturb these Weld Points.
That's because Weld Points are assigned to specific triangle, so it's not recommended to change geometry after assigning one.
Basically, if you use node-specific Weld Points, use Select nodes instead of changing the node geometry with Shape parameters.

Another option is to check the "Relative to Node Origin" in Weld Point, so it won't change with geometry.
Grip and Slide use that for decal placement Weld Points.

... But for example, the Slide part doesn't have any Weld Points, so it can use one of the Shape parameters:

... Same with the Barrel Outer, Trigger Guard and other parts:

(Click on the image for full sized version)



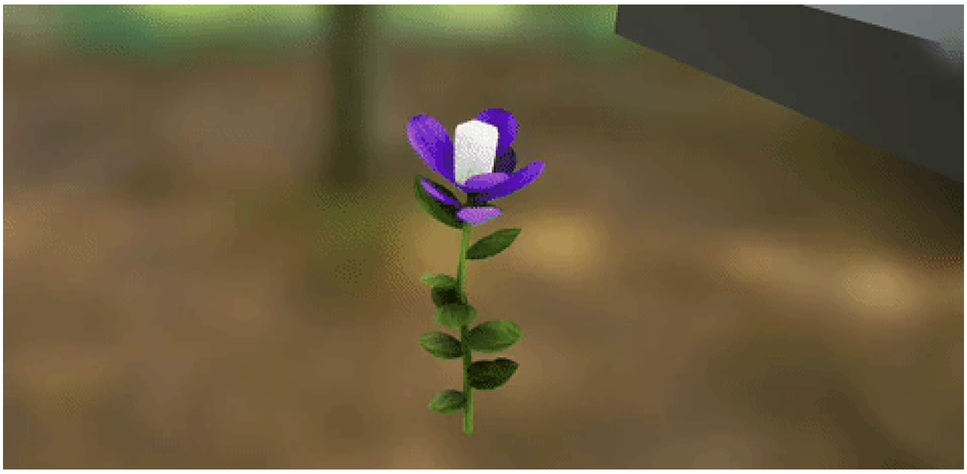
Example: Gun grip Shapes.

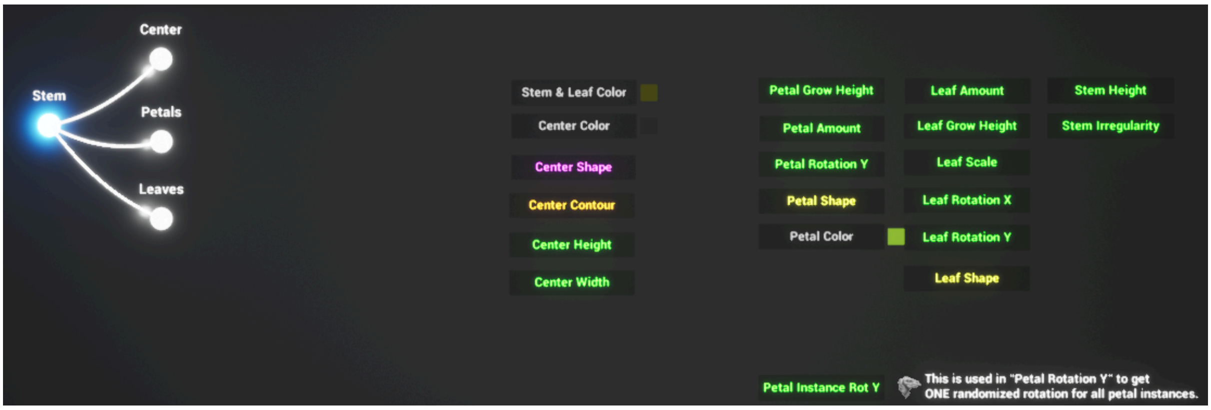
Map: /Maps/GunGenerator/GunGenerator1_FlatShaded

Gun generator is an example where all nodes are connected to the root node and there is no parent-based placement happening. Part placement is done in an 'absolute' way, by specifying XYZ locations in root space for each asset part.

Extras: Adding silencer to the Gun Generator.

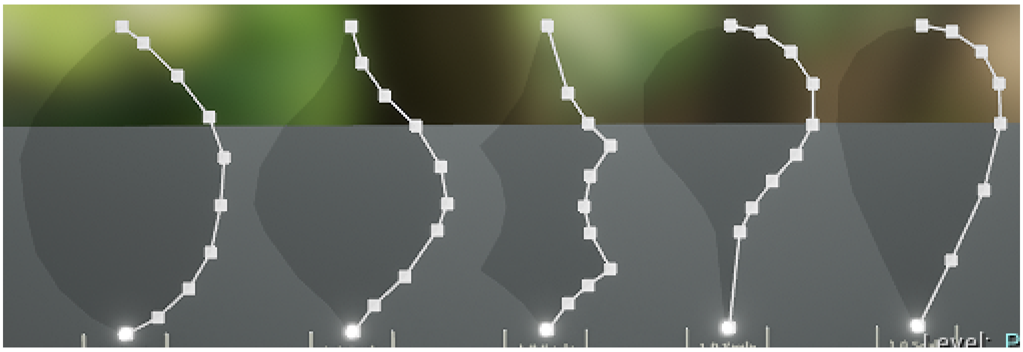
Plant Generator





This is used in "Petal Rotation Y" to get ONE randomized rotation for all petal instances.

(Click on the image for full sized version)

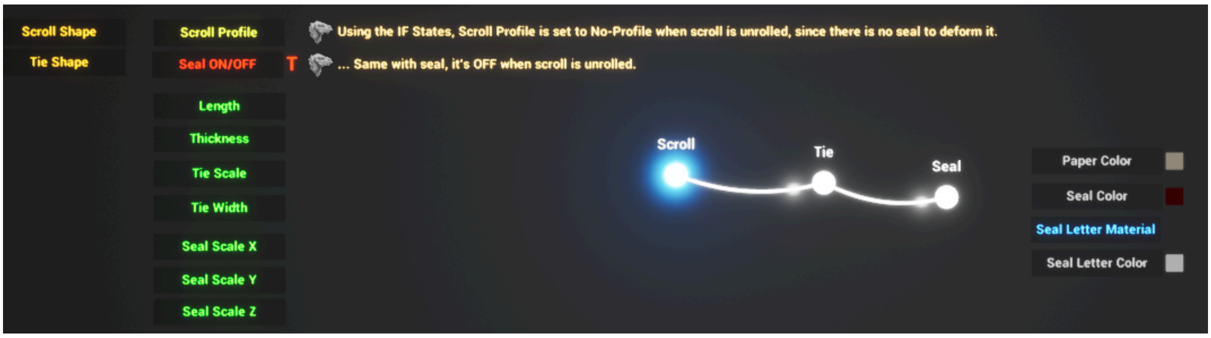


Example: Plant leaf Profiles.

Map: /Maps/PlantGenerator/PlantGenerator1_FlatShaded

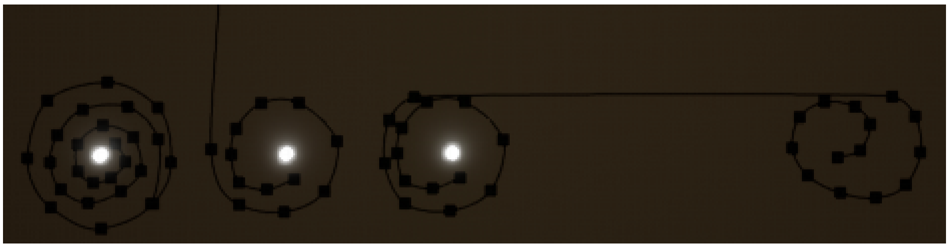
Leaves and petals on the plant are a good instancing example.

Scroll Generator


Using the IF States, Scroll Profile is set to No-Profile when scroll is unrolled, since there is no seal to deform it.

... Same with seal, it's OFF when scroll is unrolled.

(Click on the image for full sized version)

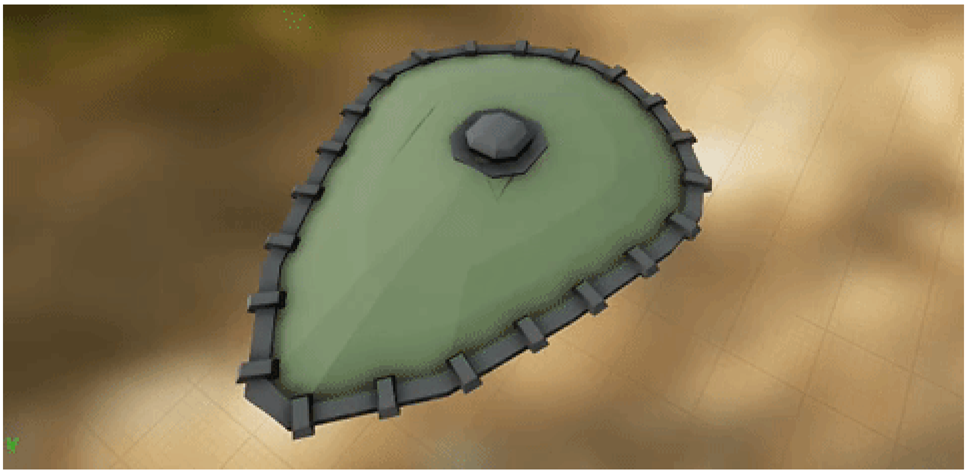


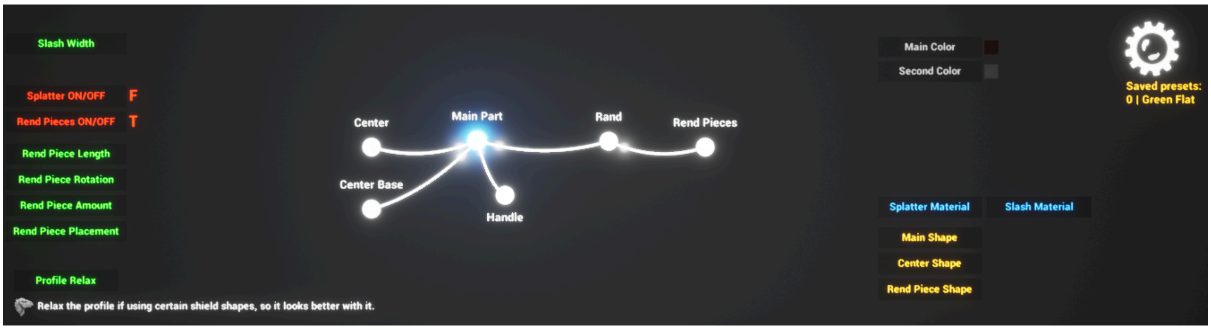
Example: Scroll Path shapes.

Map: /Maps/ScrollGenerator/ScrollGenerator1_FlatShaded

Scroll generator uses Mesh Decal for the wax seal imprint.

Shield Generator

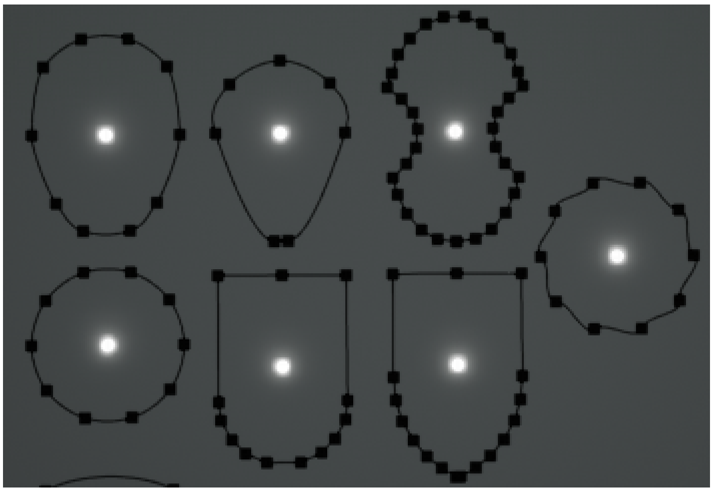




Relax the profile if using certain shield shapes, so it looks better with it.

Saved presets:
0 | Green Flat

(Click on the image for full sized version)

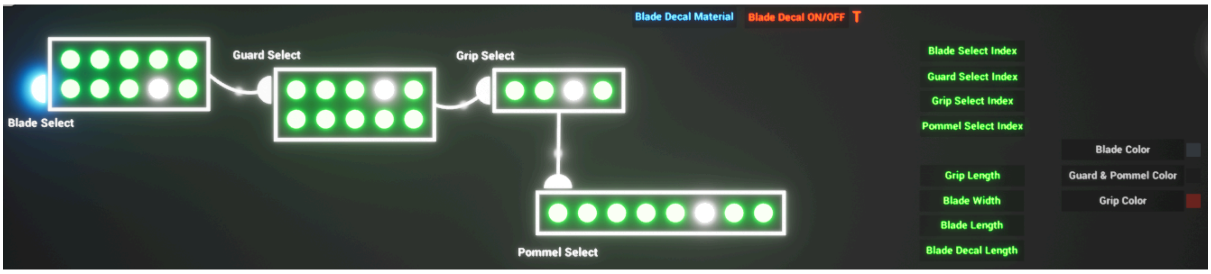


Example: Default shield shapes.

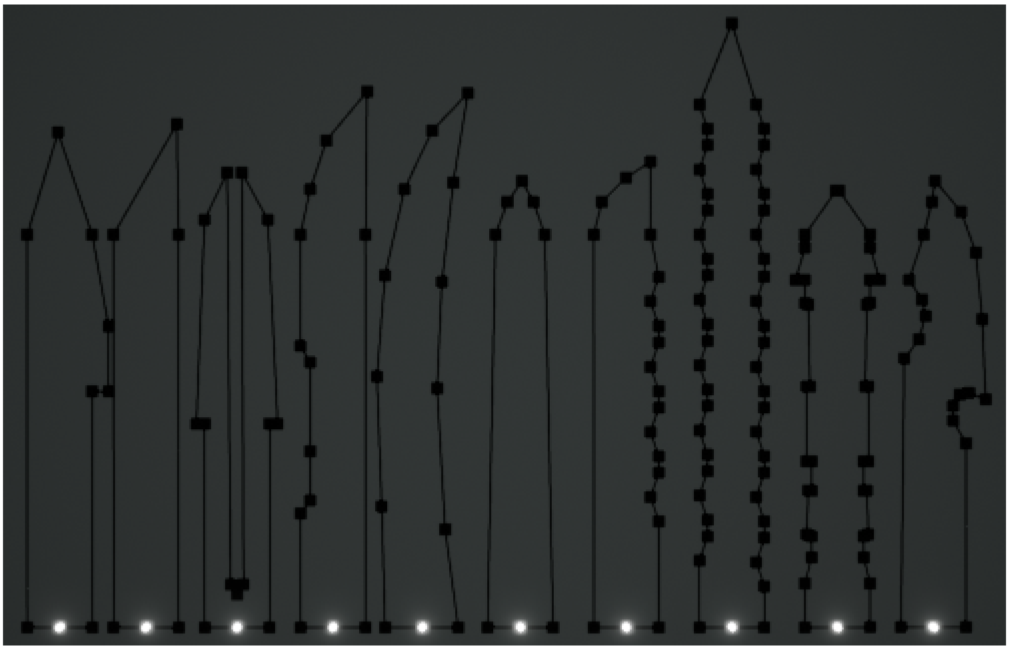
Map: /Maps/ShieldGenerator/ShieldGenerator1_FlatShaded

Optional damage & blood decals are generated using the global Mesh Decals.

Sword Generator



(Click on the image for full sized version)



Example: Sword blade shapes.

Map: /Maps/SwordGenerator/SwordGenerator1_FlatShaded

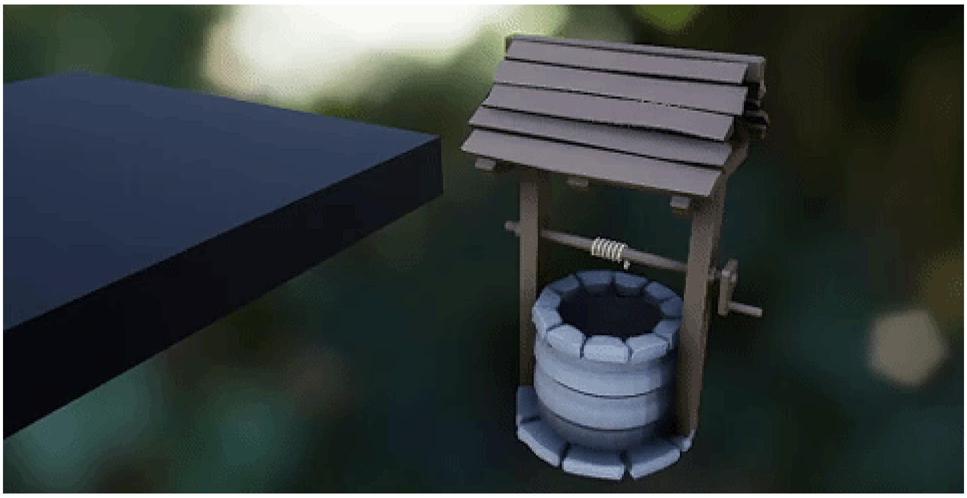
Sword generator uses Select nodes for all the asset parts.

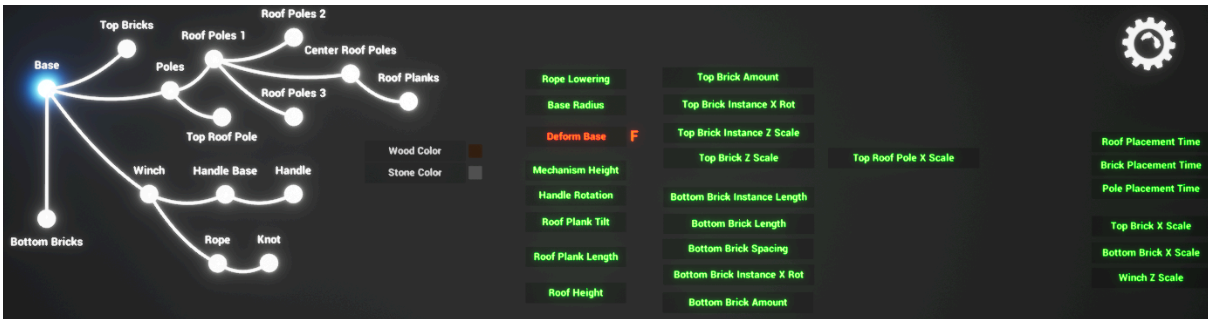
This is preferred when node placement rely on geometry-assigned placement points, e.g. Weld Points or Placement Splines. In this case all the node Shape parameters don’t change per node. It generates once, saves all its placement points and stays this way. Then, Select node just picks specified assigned node...

… In other cases it’s more convenient to parameterize these Shape values e.g. with Shape Parameters. But then the node geometry will drastically change due to its Shape value change, so all saved placement points can be misplaced, since they are assigned to closest geometry triangles, via triangle indexes.
It should work fine anyway (in most cases), because when geometry changes, node tries to re-assign relative triangle indexes to retain their true placement - but using Select nodes is just a 100% safe option, it’s also a bit more performant.

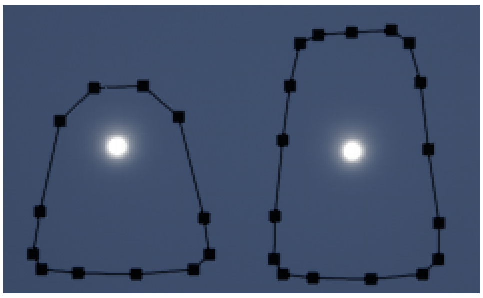
Optional damage & engraving decals are generated using the global Mesh Decals.

Well Generator





(Click on the image for full sized version)



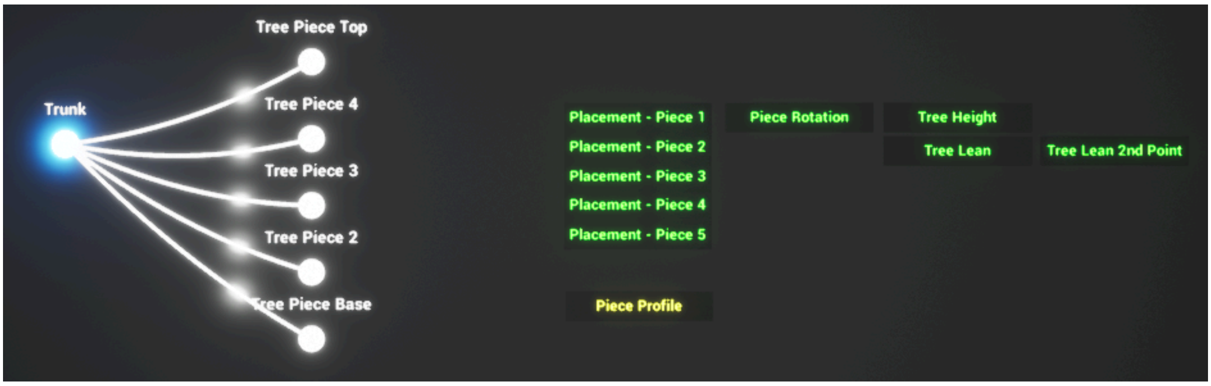
Example: Stone shapes for well top & bottom part.

Map: /Maps/WellGenerator/WellGenerator_FlatShaded

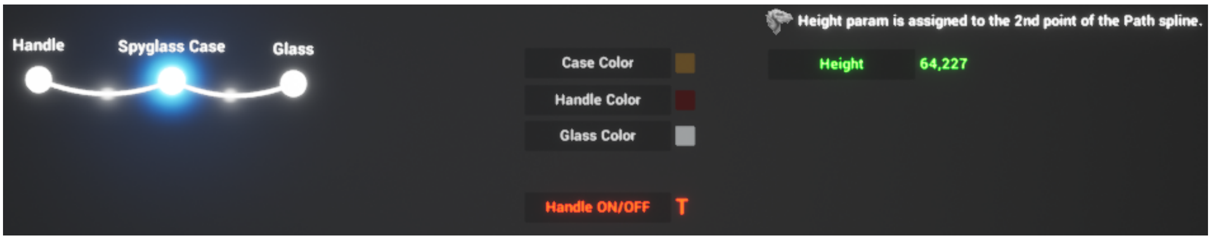
Well generator uses Instancing for stones (placed on Placement Spline) and the Deform modifier for stones & wood planks (+ optionally for the well base).

Formulas are utilized to e.g. fit the bottom & top stones nicely by altering their width based on current stone amount.

Simple Tree Generator



Simple Spyglass Generator

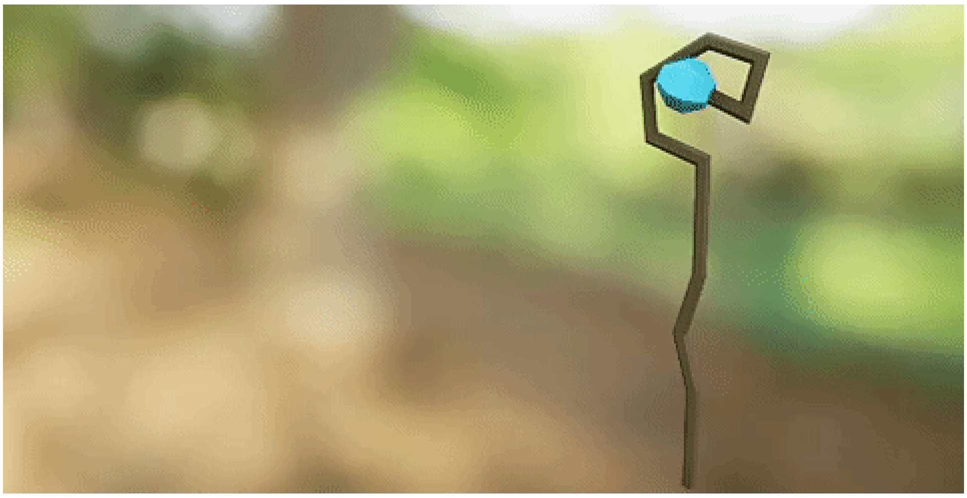


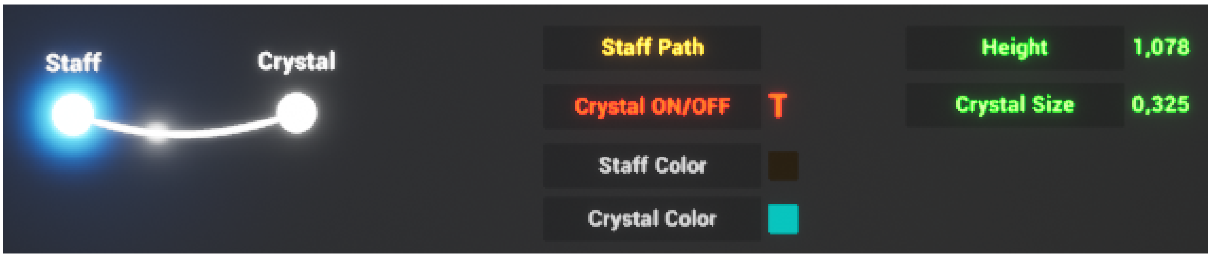
Height param is assigned to the 2nd point of the Path spline.

Height 64,227

Handle ON/OFF T

Simple Staff Generator





Staff Path

Crystal ON/OFF T

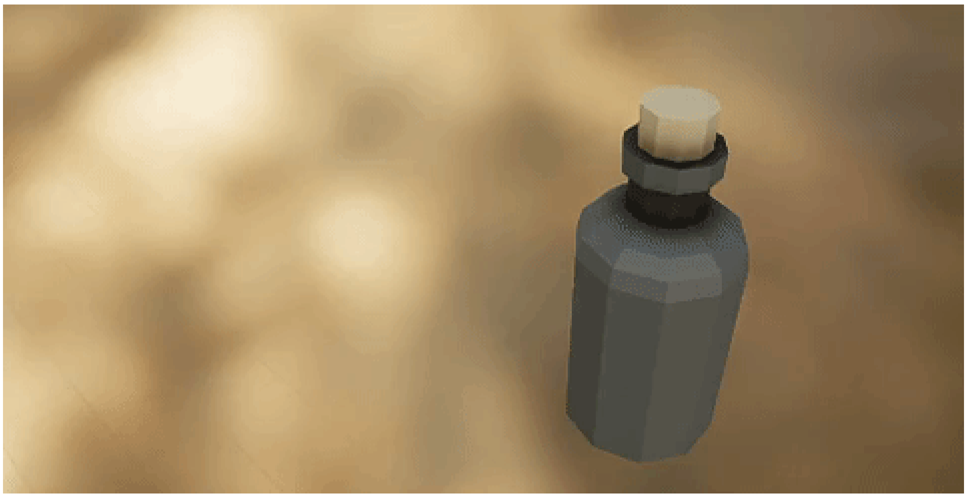
Staff Color 🟫

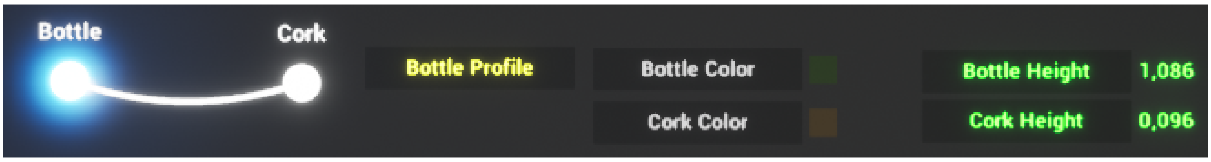
Crystal Color 🟦

Height 1,078

Crystal Size 0,325

Simple Bottle Generator





Bottle Profile

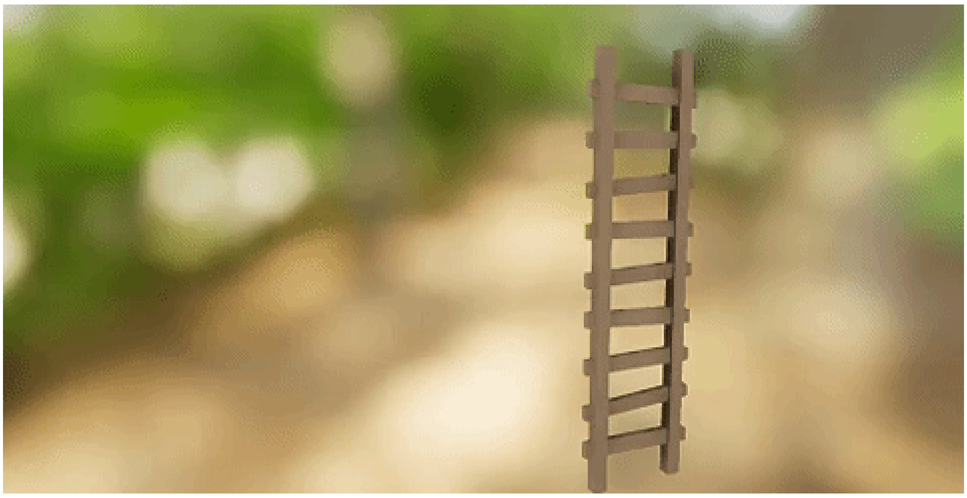
Bottle Color 🟩

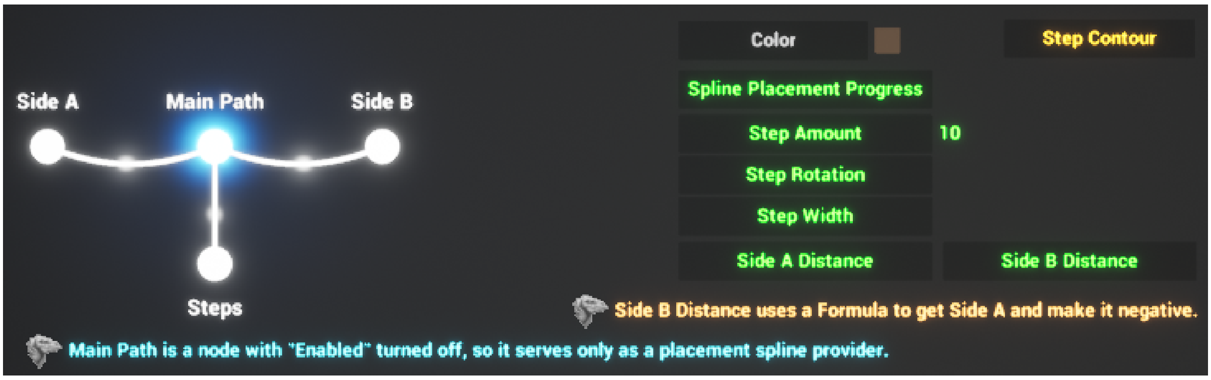
Cork Color 🟫

Bottle Height 1,086

Cork Height 0,096

Simple Ladder Generator





Side B Distance uses a Formula to get Side A and make it negative.

Main Path is a node with "Enabled" turned off, so it serves only as a placement spline provider.

… The simple generators are available in the: /Maps/SimpleGenerators/ folder.

Bonus: WIP House, included in the project.

Main concepts

Geometry placement

Asset generation in P.A.C. is based on a parent-placement concept. Basically, each node represents one part of an asset and it's placed on its parent node, using specified parameters & placement settings.
Both location and rotation can be set this way, via the **Placement on Parent** section in the **Details** panel of every node.
These are the default placement methods:

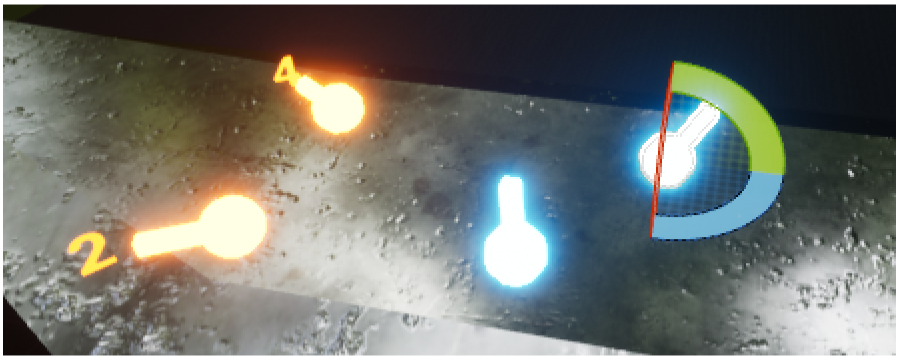
On Weld Point

Weld Points can be used to specify locations & rotations relative to current node geometry.

Every node can have multiple Weld Points.

> 📗 **When node geometry changes, it tries to update its Weld Points, so they always stay in correct location. This is possible because Weld Points detect closest triangle and calculate their location based on that, always relatively to the triangle.**

Weld Points can be added by drag&dropping a Weld Point from **/Blueprints/Misc/** folder into Part Inspector geometry and clicking **Save Weld Points**.



Saved points are locked and marked with different color.
To edit saved points, use the **Edit Weld Point** button after specifying **Weld Point Index**.
To remove saved point, click **Remove Weld Point** after specifying **Weld Point Index**.

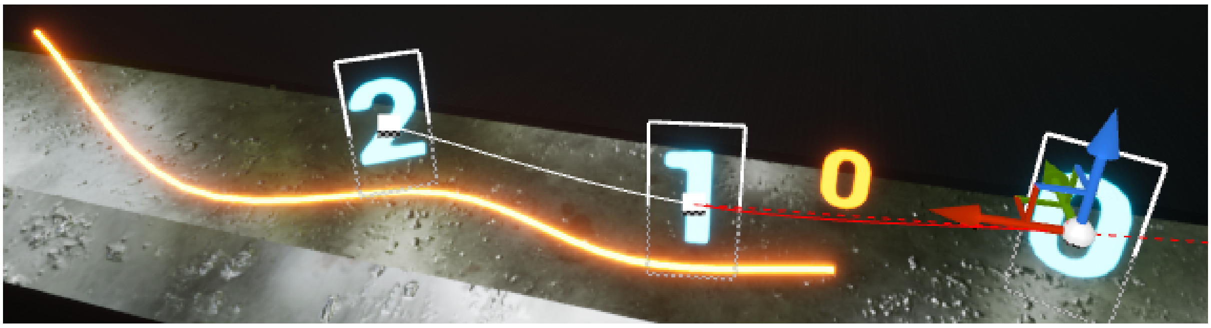
Use the node's **Placement Point Index** to specify which Weld Point on parent node you want to use. Weld Points are indexed from 0.

On Placement Spline

Placement Splines, similarly to Weld Points, can be assigned to node geometry, but instead of single location & rotation, they allow to specify a custom spline. This spline can be used for geometry placement, using spline-specific parameters. Every node can have multiple Placement Splines.

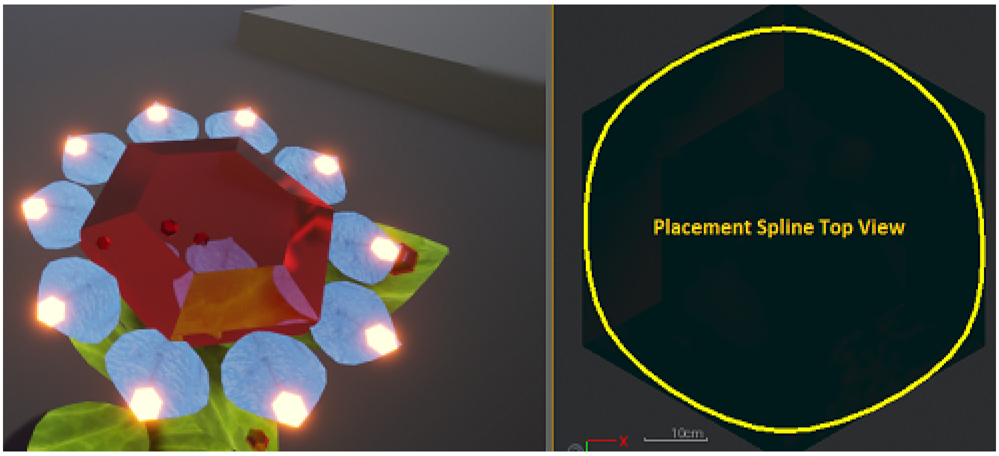
📗 **Similarly to Weld Points, when node geometry changes, it tries to update its Placement Spline points, so they always stay in correct location.**

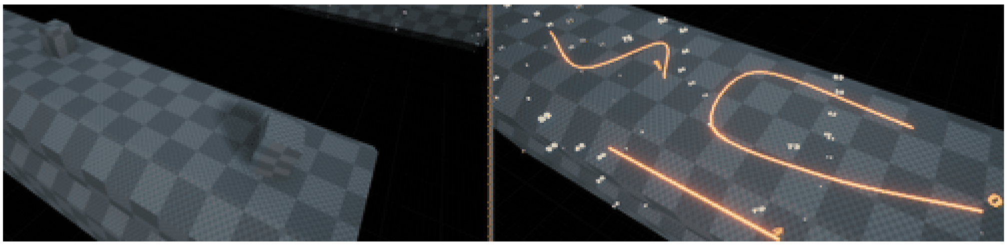
Placement Splines can be added by drag&dropping a Placement Spline from **/Blueprints/Misc/** folder into Part Inspector geometry and clicking **Save Placement Splines**.



Saved splines are locked and marked with different color.
To edit saved splines, use the **Edit Placement Spline** button after specifying **Spline Index**.
To remove saved spline, click **Remove Placement Spline** after specifying **Spline Index**.

Use the node's **Placement Point Index** (it's under Placement on Parent) to specify which Placement Spline on parent node you want to use. Placement Splines are indexed from 0. You can specify spline time (0-1 progress along spline) at which it will sample location & rotation from it, using the **Placement Spline Time** parameter.





On Vertex

Placement on vertex causes child node to be placed on specified parent vertex.
Use the **Placement Point Index** to specify vertex index.
To see vertex indexes, use the Part Inspector.

On Triangle

Placement on triangle causes child node to be placed on specified parent triangle.
Use the **Placement Point Index** to specify triangle index.
To see triangle indexes, use the Part Inspector.

Between Verts

This placement method places child node between specified parent vertices.
Use the **Between Verts** array to specify vertex indexes.
To see vertex indexes, use the Part Inspector.

Vert Path

Vert Path can be defined in the **Vert Path → Path Verts** array and the **Time Along Path** parameters.
In general, it works like a placement path created by specified vertices, placing node contents at specified path time (0.0 - 1.0 progress).

Pivot Point

Each node can "sit" on a parent placement point using its local (0,0,0) location or a custom pivot point.
Pivot Point can be specified using node's Weld Point, vertex, triangle or other methods, under the **Placement Pivot** section.

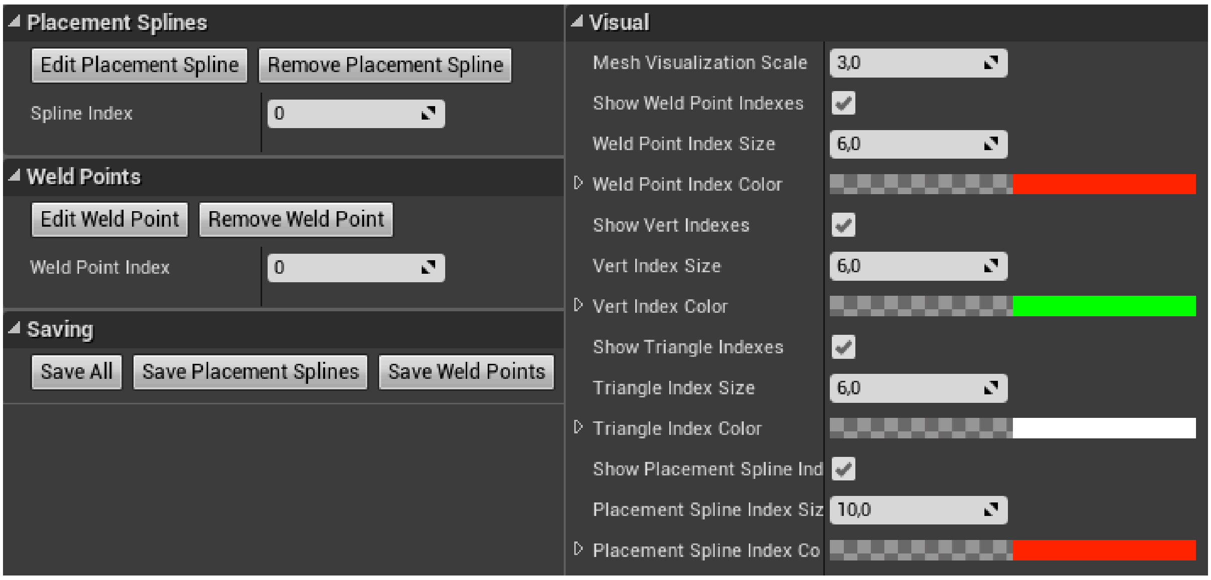
Part Inspector

Part Inspector allows to inspect current node geometry and add Weld Points & Placement Splines to it.
In the node **Details** panel → **General** section, click **Send to Inspector**. Node contents will be displayed in the designated Part Inspector area, you can quickly access it by choosing the **Part Inspector** camera from the **Perspective** viewport menu.

Weld Points & Placement Splines can be placed on node geometry by using the Part Inspector:

1. In the node **Details** panel → **General** section, click **Send to Inspector**
2. Node geometry will show in the Inspector view: To see it, select **Perspective** → **Part Inspector** e.g. in the upper right viewport pane.
3. Drag & drop Weld Point or Placement Spline from **/Blueprints/Misc/** into desired location.
4. Select the Part Inspector geometry and click e.g. **Save Weld Points**.

Part Inspector can also be used to see all the current geometry info like vert & triangle indexes and more. You can toggle it by selecting the Part Inspector geometry, in the **Details** panel:



Spline Index 0

Weld Point Index 0

Mesh Visualization Scale 3,0

Show Weld Point Indexes ✓

Weld Point Index Size 6,0

Weld Point Index Color 🟥

Show Vert Indexes ✓

Vert Index Size 6,0

Vert Index Color 🟩

Show Triangle Indexes ✓

Triangle Index Size 6,0

Triangle Index Color ⬜

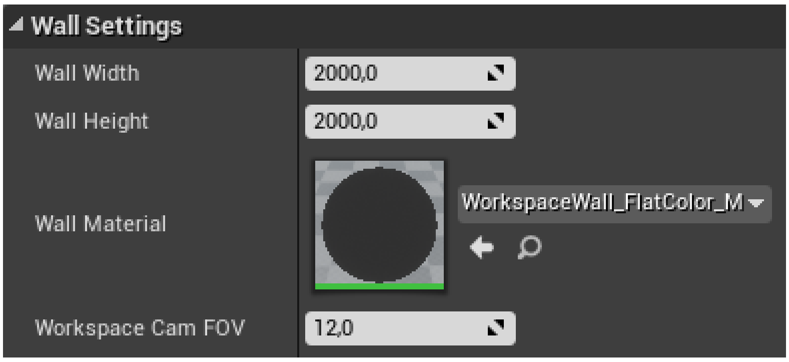
Show Placement Spline Ind ✓

Placement Spline Index Siz 10,0

Placement Spline Index Co 🟥

Workspace & node hierarchy

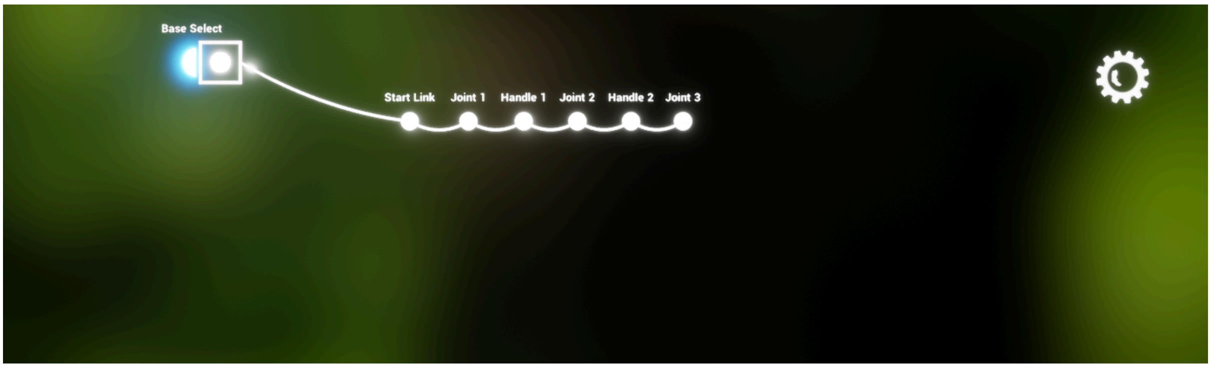
Workspace is a wall where you can drop nodes, parameters & comments to create a generator.
Bottom viewport pane is recommended for the Workspace view, but you can customize it to your liking, e.g. by putting a whole separate viewport to your second screen (if you have one) for even more space.
You can also click anywhere on the Workspace to display its Details panel - and change the wall size (if you need more space), camera FOV (practically zoom) or wall material:



Wall Settings

Wall Width	2000,0
Wall Height	2000,0
Wall Material	WorkspaceWall_FlatColor_M
Workspace Cam FOV	12,0

By changing the Wall Material you can customize the Workspace look.
For example, using the Translucent1_M material will result in this, where background represents HDRI map (blurred by DoF) currently used in your scene:



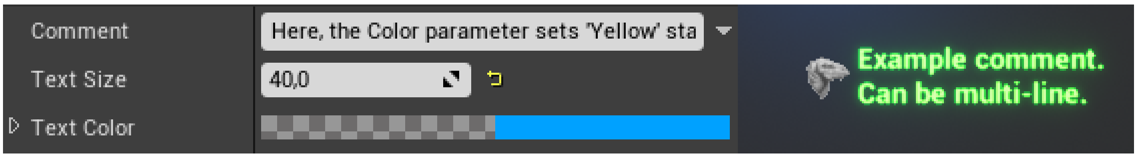
📗 **Hold middle mouse button to pan around the Workspace view**

There is also a rotating cogwheel icon - it's the Global Settings, where you can randomize your asset, add global decals, set presets and more:

More info here.

When node is dropped into Workspace, it will get closest node and set it as its parent. If there are no nodes on the Workspace yet, the first drag&dropped node will become a root node. Root nodes have different color.
There's always one root node (top node in hierarchy) and child nodes, where every node can have multiple child nodes.
You can reparent nodes using the **Parent** field under the General section.

📗 **Workspace can contain multiple Comments, using /Blueprints/Misc/Comment_BP:**



Comment: Here, the Color parameter sets 'Yellow' sta
Text Size: 40,0
Text Color

Example comment.
Can be multi-line.

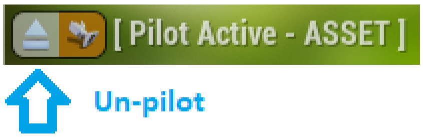
Workflow & asset saving

The recommended workflow for creating custom asset generators:

- In 3-pane viewport layout have following views: Workspace in bottom one, Asset preview in upper left one and Shape Room or Part Inspector in upper right one
- Drag & drop nodes/parameters into the Workspace, tweak them and see the realtime asset preview in another pane
- In the Shape Room, create or modify shapes if needed, seeing realtime asset preview in another pane
- Test your generator by clicking Randomize in Global Options.

📗 Always change parameter values by click → enter new value → Enter. Using sliders can cause lag with more complex generators.

❗ Sometimes UE resets piloted cameras in other viewport panes (UE bug). To avoid that, un-pilot the camera after setting the ASSET, Shape Room or Part Inspector views:



Saving assets

Generated assets can be saved as regular Static Meshes:

1. Select generated asset
2. In the Components panel, select Procedural Mesh and click Convert to Static Mesh
3. [UE bug workaround] After saving the mesh into a specified folder, right click it and click Save.

… Saved meshes can be exported or migrated to other projects.

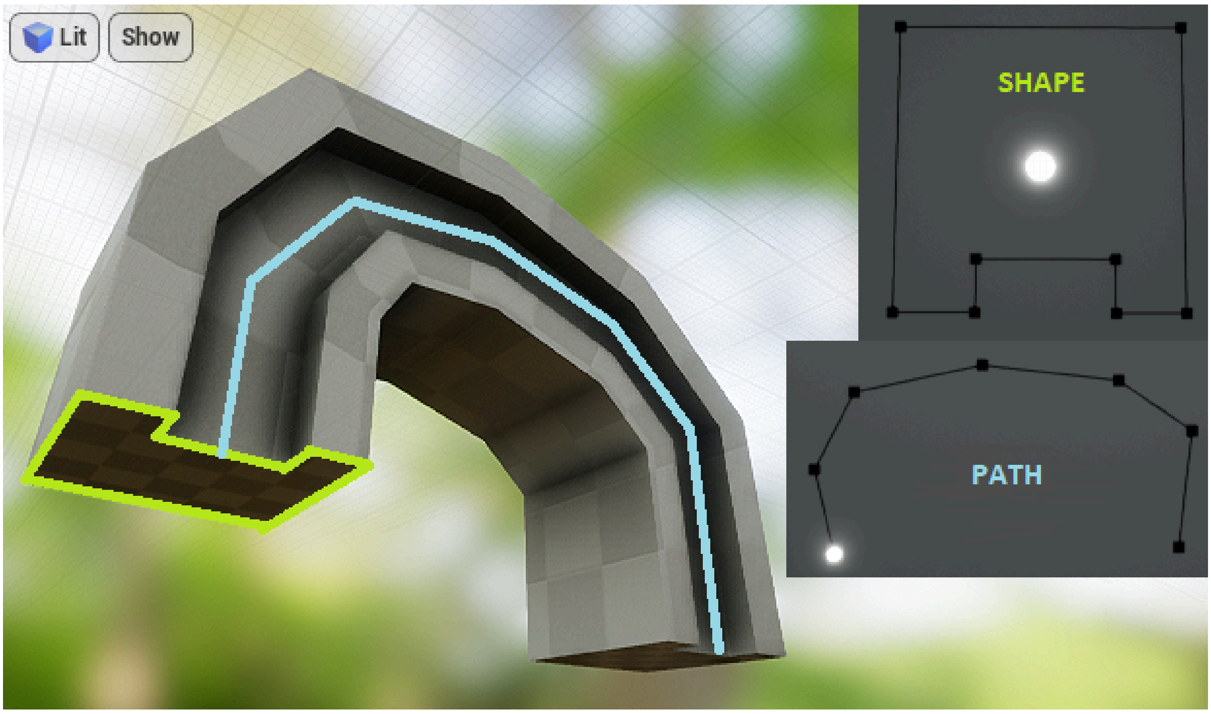
Creating a new generator

1. Create a new folder for your generator.
2. Open the Empty Starter Map. It can be found in the main /Maps/ folder.
3. Click File → Save Current As… and save it with desired name.
4. Now you can start creating by dragging nodes & parameters from the /Blueprints/Nodes/ and /Parameters/ folders into the workspace. For more info, check the tutorials and the included example generators in the /Maps/ folder.

Nodes

Spline Shaped Mesh Node

Spline Shaped Mesh is based on 2 shapes: Shape & Path. It works by 'traversing' the Shape along the Path. Both Path & Shape shapes can be closed or not.



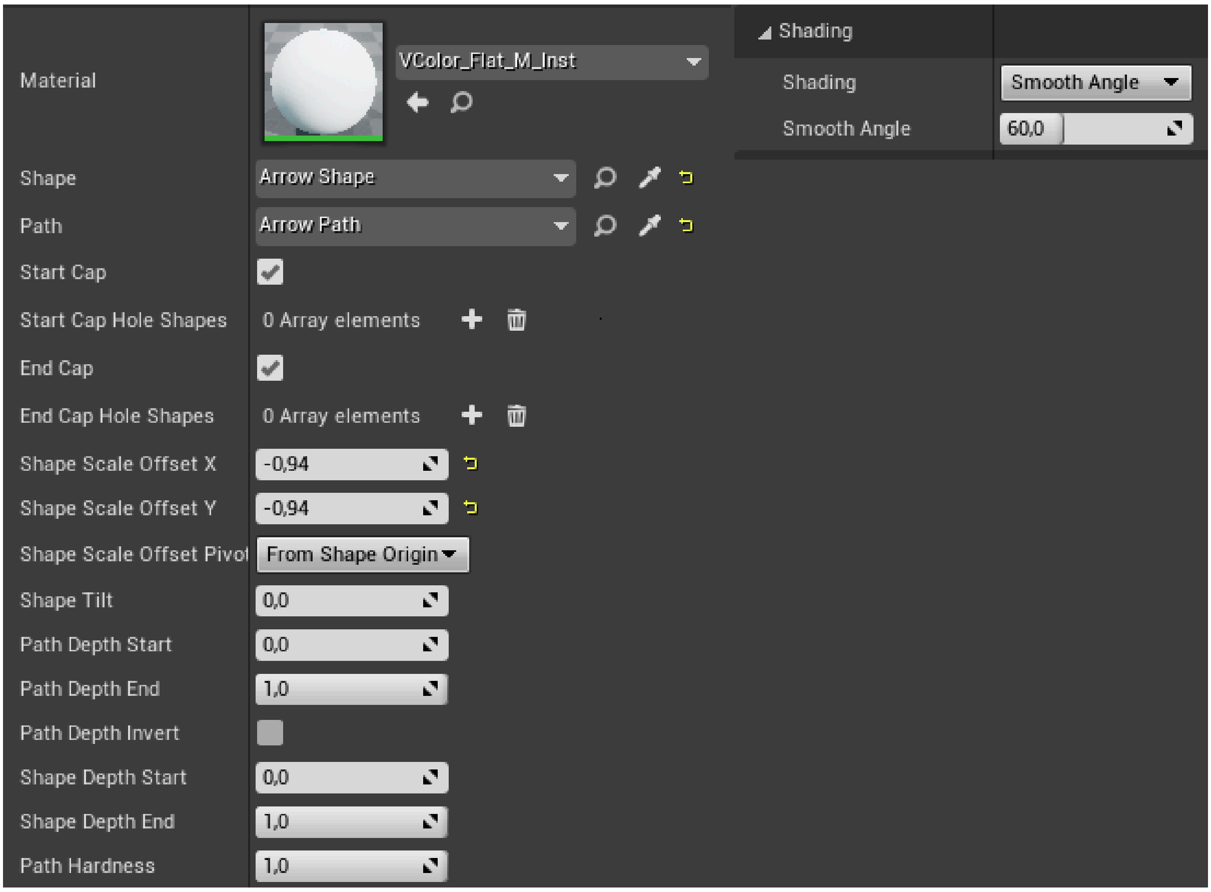
Example available in the /Maps/SimpleExamples/SplineShapedMeshExample

📗 **Generating a shape along path spline uses a custom along-path plane orientation system. It fixes the infamous gimbal lock problem that is present in e.g. the default UE Spline Mesh Component or other rotator-based solutions, when not using quaternions. This solution also improves the final look, allowing for more control & things like precisely oriented corners, without 'skewing' the geometry, maintaining true shape width all the way along the path spline.**

Shape must have Z-up orientation to work properly along Path!

SSM provides automatic vert shading methods: Flat, Smooth and Angle Smoothing with controllable value to set edge smoothing based on angle.

Main parameters:



Material	VColor_Flat_M_Inst
Shape	Arrow Shape
Path	Arrow Path
Start Cap	☑
Start Cap Hole Shapes	0 Array elements
End Cap	☑
End Cap Hole Shapes	0 Array elements
Shape Scale Offset X	-0,94
Shape Scale Offset Y	-0,94
Shape Scale Offset Pivot	From Shape Origin
Shape Tilt	0,0
Path Depth Start	0,0
Path Depth End	1,0
Path Depth Invert	□
Shape Depth Start	0,0
Shape Depth End	1,0
Path Hardness	1,0

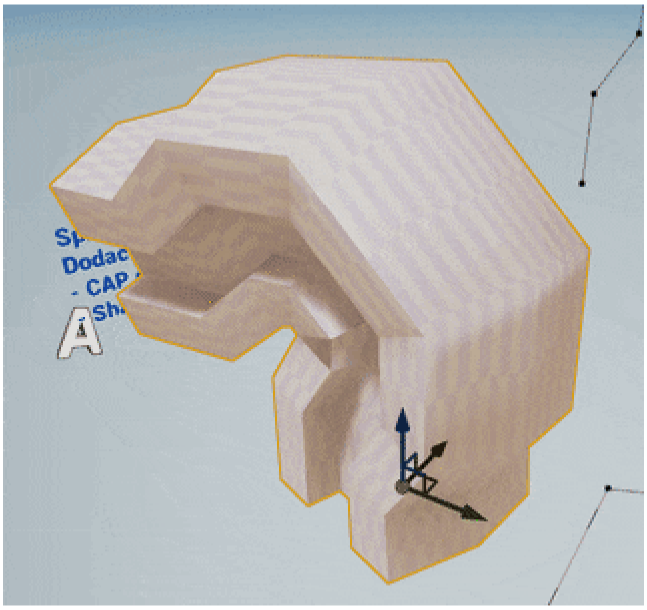
Shading

Shading	Smooth Angle
Smooth Angle	60,0

💡 **All the parameters have tooltips with explanation, just hover over any parameter:**

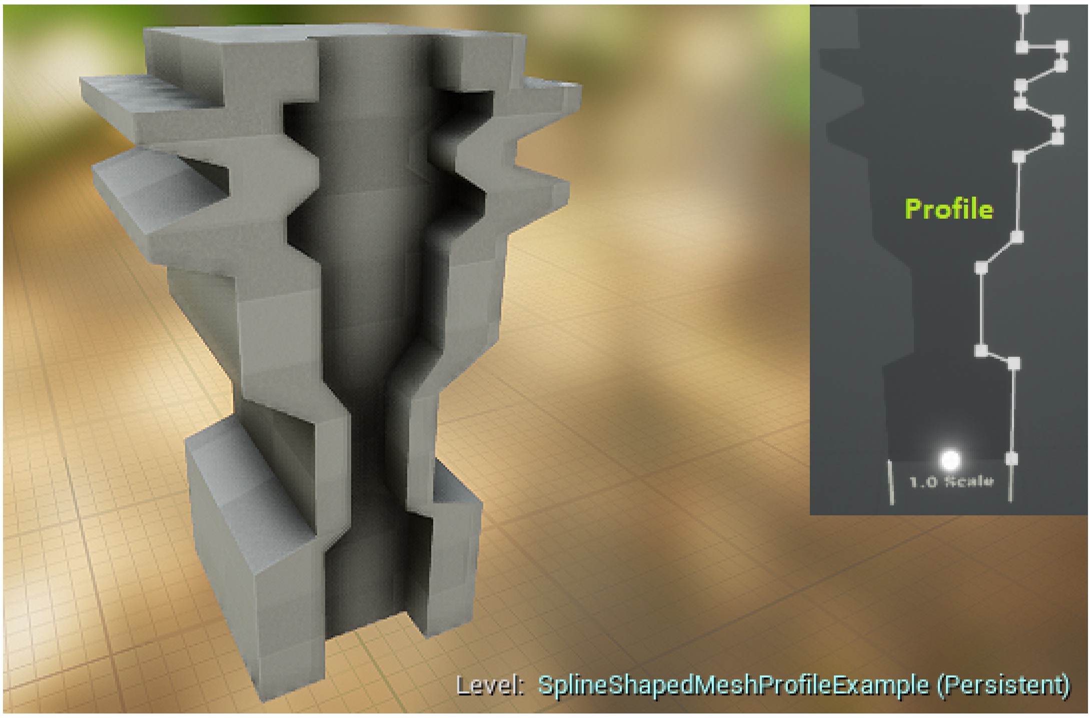
Path Depth Start	0,0
Path Depth End	1,0

Shape Tilt and Angle Smoothing in action:



SSM Profiles

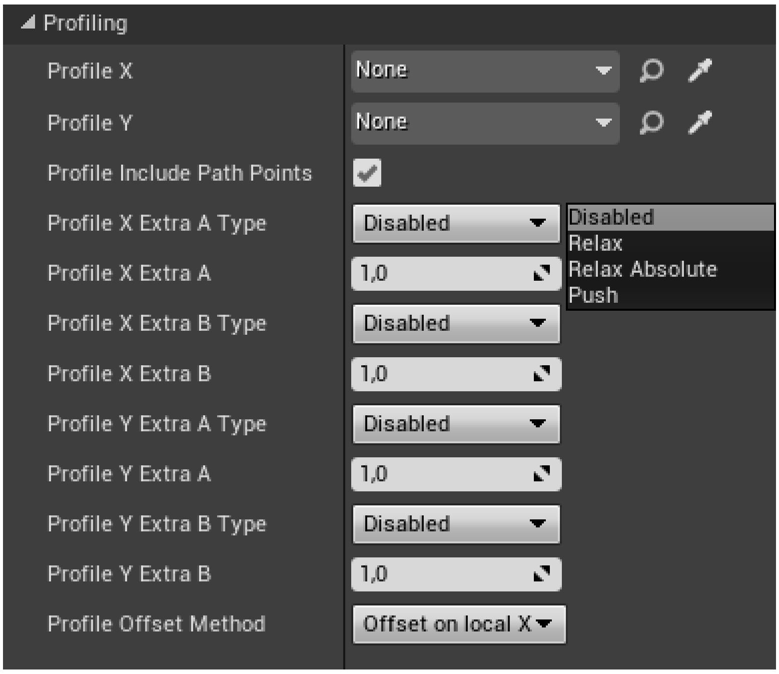
Profiles allow to define additional shape on X and Y axis, working with any Shape and Path.



Example available in the /Maps/SimpleExamples/SplineShapedMeshProfileExample

Profile shapes can be symmetrical (default) or not, it's defined by the Symmetry parameter. You can assign different profiles for both X & Y under the Profiling section. It's also possible to add extra profiling modifiers there.

SSM Profiling Parameters:

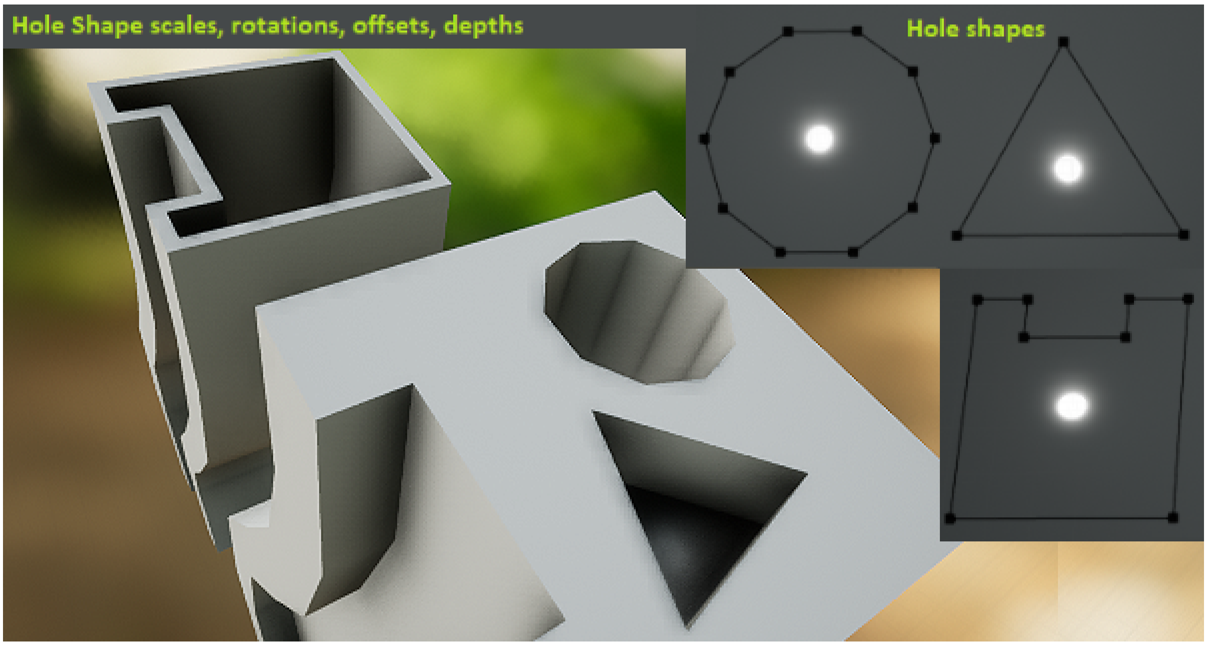


Profiling

Profile X	None
Profile Y	None
Profile Include Path Points	☑
Profile X Extra A Type	Disabled
Profile X Extra A	1,0
Profile X Extra B Type	Disabled
Profile X Extra B	1,0
Profile Y Extra A Type	Disabled
Profile Y Extra A	1,0
Profile Y Extra B Type	Disabled
Profile Y Extra B	1,0
Profile Offset Method	Offset on local X

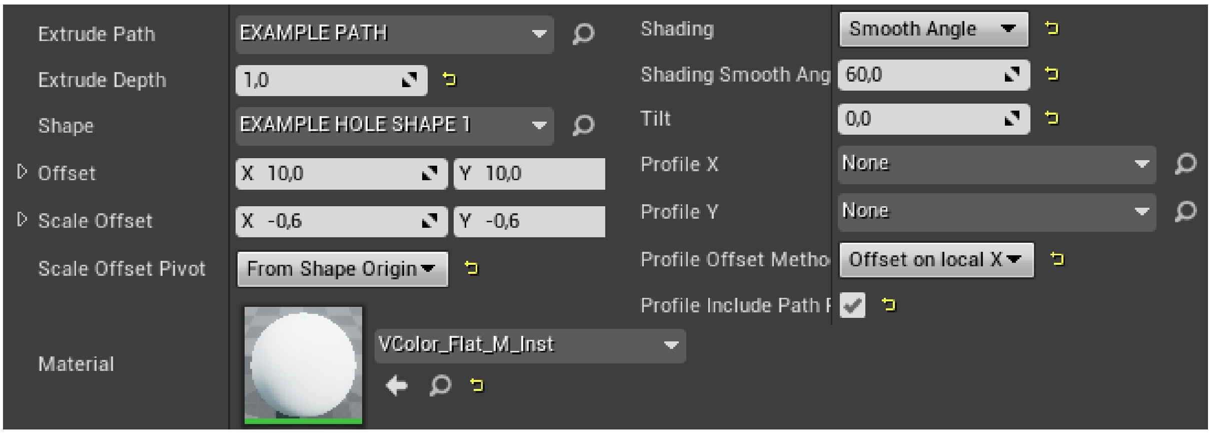
Hole Shapes

You can define Hole Shapes that will go from specified mesh end (start or end cap) all the way through the path or for specified depth along path.
They can have different shapes, scales, rotations, offsets, depths and more. Holes can start from both ends ('caps') of the Spline Shaped Mesh.



Example available in the /Maps/SimpleExamples/SplineShapedMeshHolesExample

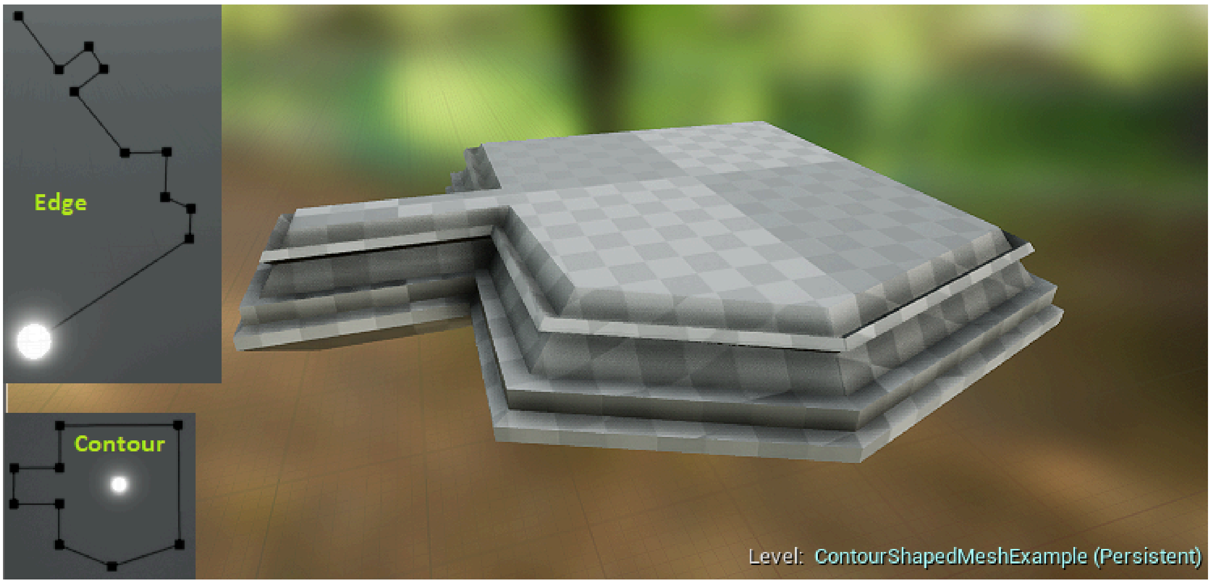
Hole Shape parameters:



Extrude Path	EXAMPLE PATH	Shading	Smooth Angle
Extrude Depth	1,0	Shading Smooth Ang	60,0
Shape	EXAMPLE HOLE SHAPE 1	Tilt	0,0
Offset	X 10,0 Y 10,0	Profile X	None
Scale Offset	X -0,6 Y -0,6	Profile Y	None
Scale Offset Pivot	From Shape Origin	Profile Offset Metho	Offset on local X
		Profile Include Path F	☑
Material	VColor_Flat_M_Inst		

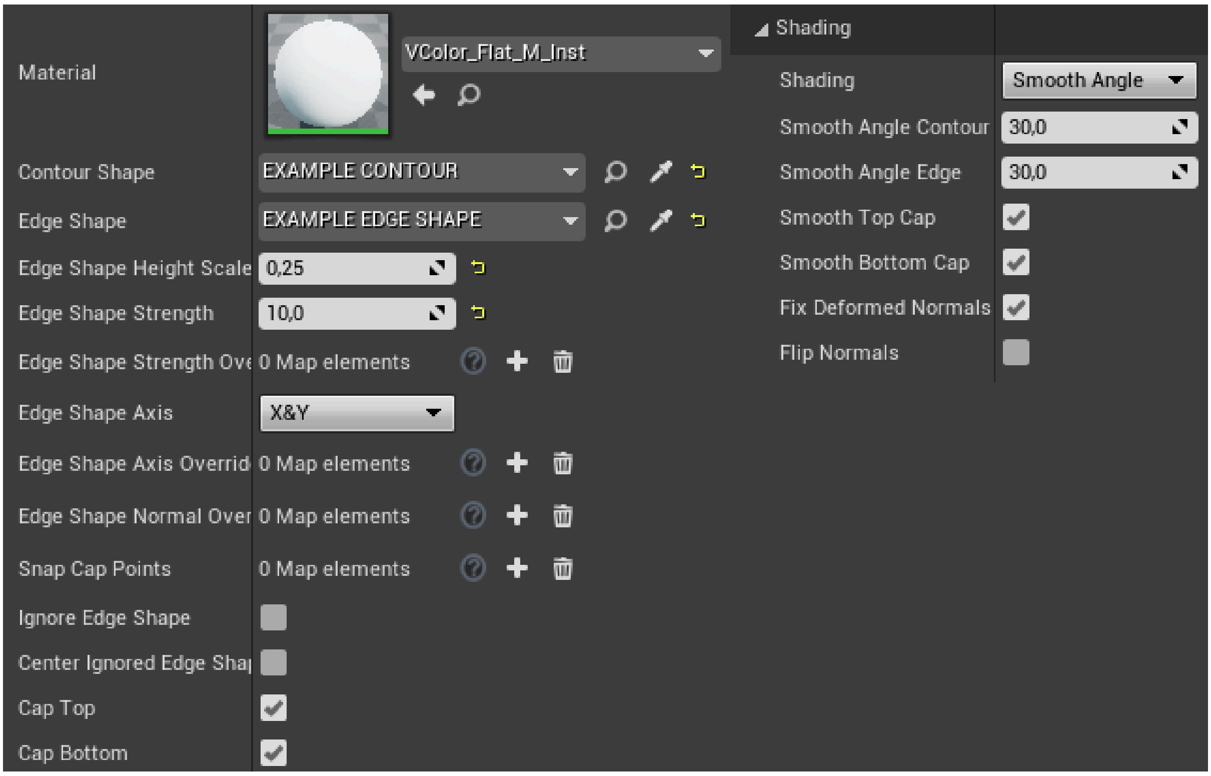
Contour Shaped Mesh Node

Contour Shaped Mesh is based on two shapes: Contour and Edge shape.
In general, it acts like a contour-defined extruded polygon with custom shaped edge.
Example: Contour for sword blade shape, Edge to make it sharp on sides.



Example available in the /Maps/SimpleExamples/ContourShapedMeshExample

Main parameters:



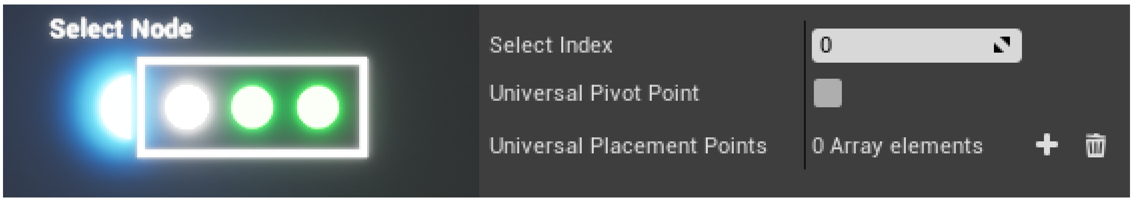
Material	VColor_Flat_M_Inst
Contour Shape	EXAMPLE CONTOUR
Edge Shape	EXAMPLE EDGE SHAPE
Edge Shape Height Scale	0,25
Edge Shape Strength	10,0
Edge Shape Strength Ove	0 Map elements
Edge Shape Axis	X&Y
Edge Shape Axis Overrid	0 Map elements
Edge Shape Normal Over	0 Map elements
Snap Cap Points	0 Map elements
Ignore Edge Shape	□
Center Ignored Edge Sha	□
Cap Top	☑
Cap Bottom	☑

Shading

Shading	Smooth Angle
Smooth Angle Contour	30,0
Smooth Angle Edge	30,0
Smooth Top Cap	☑
Smooth Bottom Cap	☑
Fix Deformed Normals	☑
Flip Normals	□

Select Node

Select Node can be used to assign many nodes and select which one should be active, using the Select Index parameter.
It's used to e.g. pick random asset parts during procedural generation.
It's also a useful optimization tool, because all the assigned nodes are already calculated, the Select Node only picks mesh data from them and passes it further.



Select Index 0
Universal Pivot Point □
Universal Placement Points 0 Array elements

Example available in the /Maps/SimpleExamples/SelectNodeExample

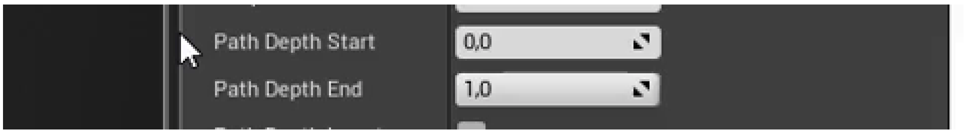
Currently selected node is always visually marked as a white pulsating node, you can change it under the Editor Visuals to customize it to your liking.

To assign a node to Select node, just drag it inside the rectangle - it works both from content browser and from the Workspace view, when node is already placed there.

📗 You can easily duplicate assigned nodes via Ctrl+C → Ctrl+V when one of them is selected. New copies will be automatically assigned and the Select node expanded.

Universal Node Parameters

💡 **All the parameters have tooltips with explanation, just hover over any parameter:**

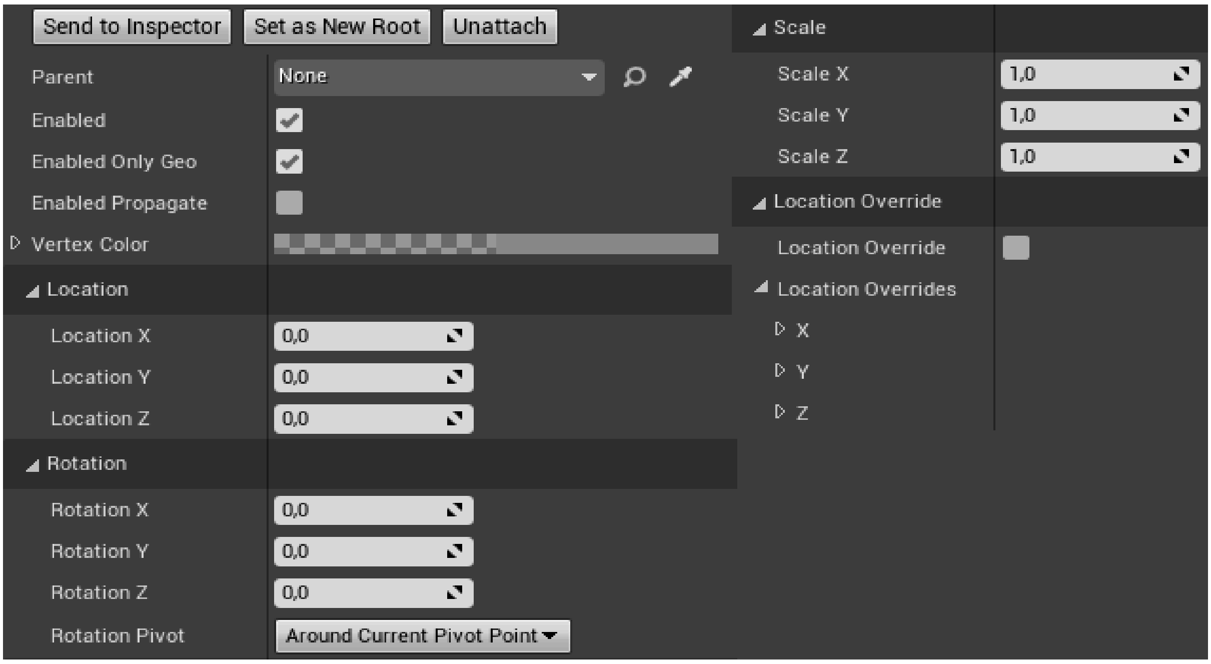


Path Depth Start 0,0

Path Depth End 1,0

All nodes have a common set of universal parameters. More info available in the tooltips.

General



Parent None

Enabled ☑

Enabled Only Geo ☑

Enabled Propagate ☐

Vertex Color

Location

Location X 0,0

Location Y 0,0

Location Z 0,0

Rotation

Rotation X 0,0

Rotation Y 0,0

Rotation Z 0,0

Rotation Pivot Around Current Pivot Point

Scale

Scale X 1,0

Scale Y 1,0

Scale Z 1,0

Location Override

Location Override ☐

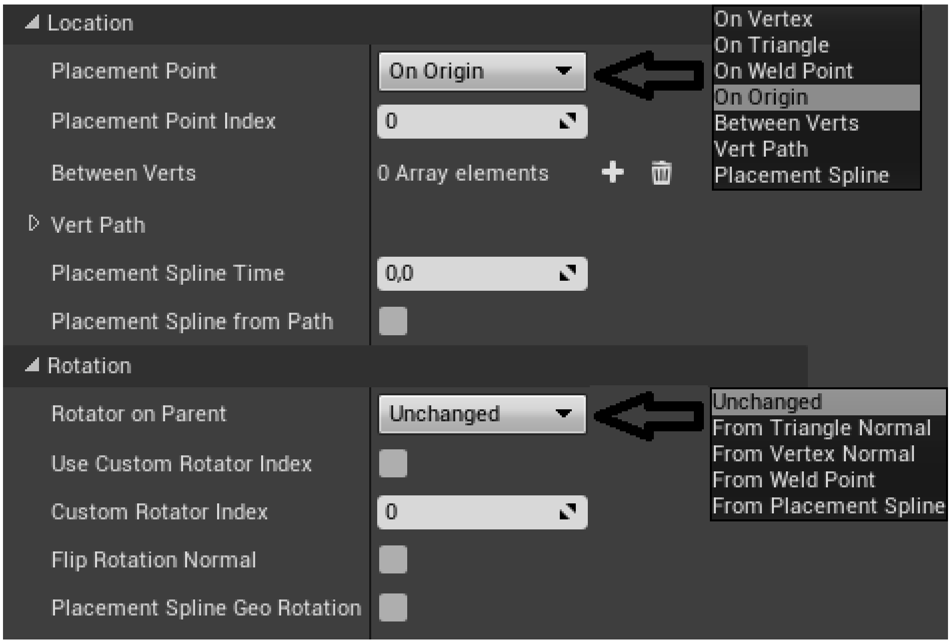
Location Overrides

X

Y

Z

Placement on Parent



Location

Placement Point On Origin

Placement Point Index 0

Between Verts 0 Array elements

Vert Path

Placement Spline Time 0,0

Placement Spline from Path ☐

Rotation

Rotator on Parent Unchanged

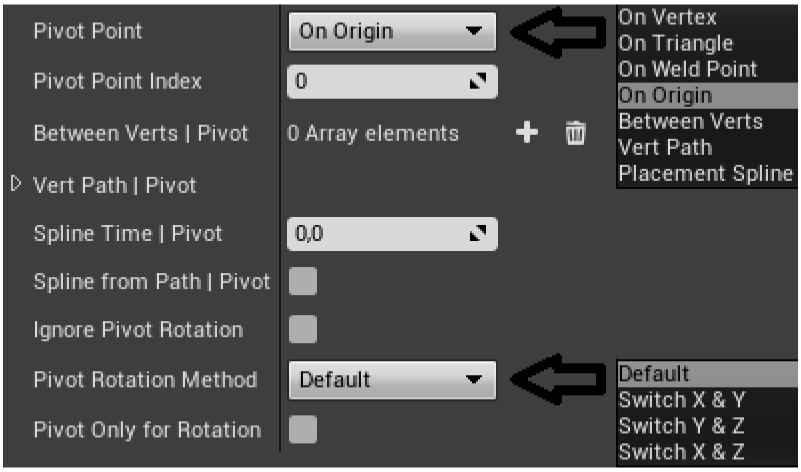
Use Custom Rotator Index ☐

Custom Rotator Index 0

Flip Rotation Normal ☐

Placement Spline Geo Rotation ☐

Placement Pivot

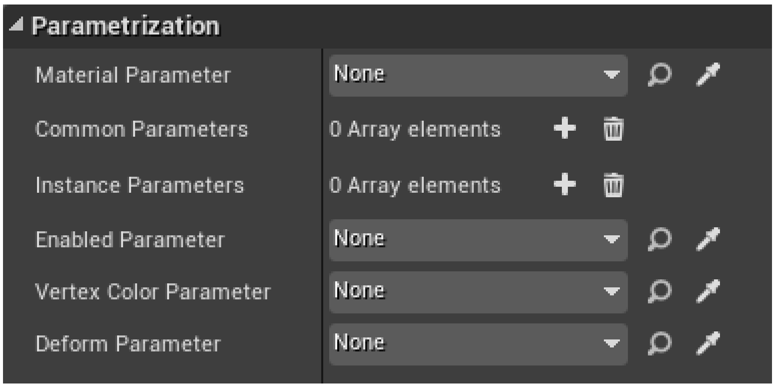


Pivot Point: On Origin
Pivot Point Index: 0
Between Verts | Pivot: 0 Array elements
Vert Path | Pivot
Spline Time | Pivot: 0,0
Spline from Path | Pivot: □
Ignore Pivot Rotation: □
Pivot Rotation Method: Default
Pivot Only for Rotation: □

Parameterization

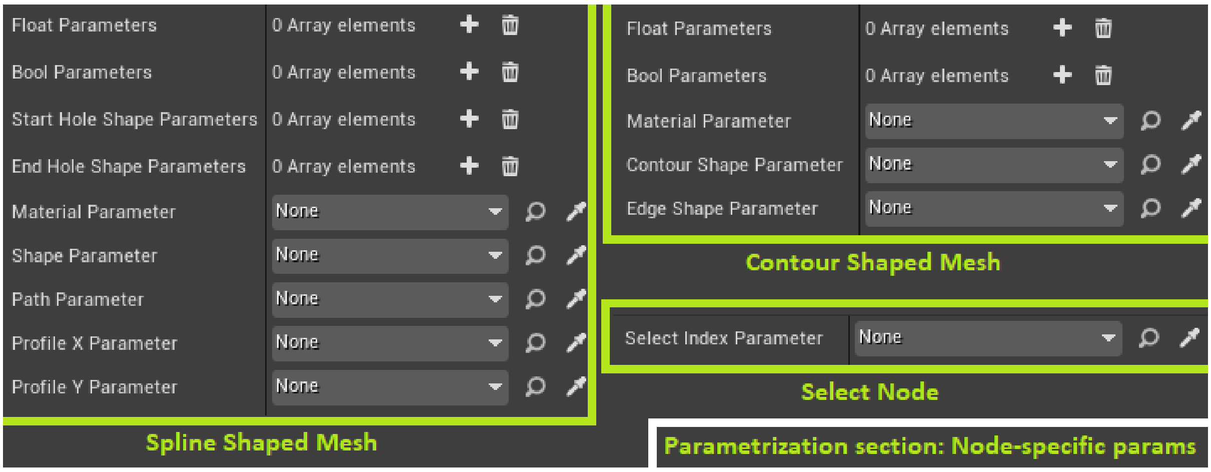
This section allows to assign Parameter Actors.
It has both universal (common to all nodes) and unique (node-specific) parameters.

Universal:



Parametrization
Material Parameter: None
Common Parameters: 0 Array elements
Instance Parameters: 0 Array elements
Enabled Parameter: None
Vertex Color Parameter: None
Deform Parameter: None

Node-specific:



Spline Shaped Mesh
Float Parameters: 0 Array elements
Bool Parameters: 0 Array elements
Start Hole Shape Parameters: 0 Array elements
End Hole Shape Parameters: 0 Array elements
Material Parameter: None
Shape Parameter: None
Path Parameter: None
Profile X Parameter: None
Profile Y Parameter: None

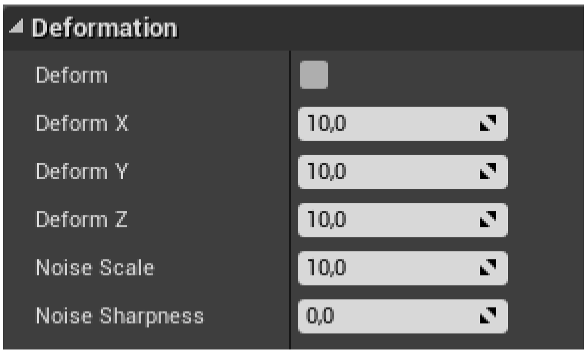
Contour Shaped Mesh
Float Parameters: 0 Array elements
Bool Parameters: 0 Array elements
Material Parameter: None
Contour Shape Parameter: None
Edge Shape Parameter: None

Select Node
Select Index Parameter: None

Parametrization section: Node-specific params

… The **Float Parameters** & **Bool Parameters** can contain almost all node-specific values that can be parameterized with external Parameter actor.

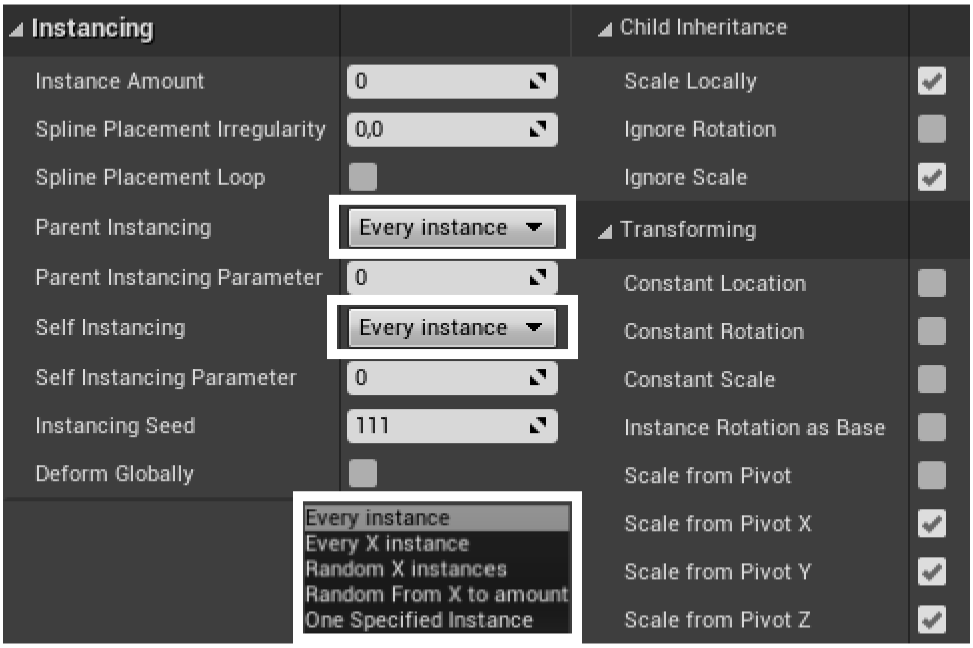
Deformation



Deformation	
Deform	□
Deform X	10,0
Deform Y	10,0
Deform Z	10,0
Noise Scale	10,0
Noise Sharpness	0,0

This modifier allows to deform node's mesh with specified values, using noise.

Instancing



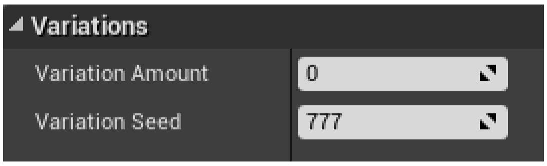
Instancing	
Instance Amount	0
Spline Placement Irregularity	0,0
Spline Placement Loop	□
Parent Instancing	Every instance
Parent Instancing Parameter	0
Self Instancing	Every instance
Self Instancing Parameter	0
Instancing Seed	111
Deform Globally	□

Child Inheritance	
Scale Locally	☑
Ignore Rotation	□
Ignore Scale	☑

Transforming	
Constant Location	□
Constant Rotation	□
Constant Scale	□
Instance Rotation as Base	□
Scale from Pivot	□
Scale from Pivot X	☑
Scale from Pivot Y	☑
Scale from Pivot Z	☑

Instancing section. More info on instancing available here.

Variations

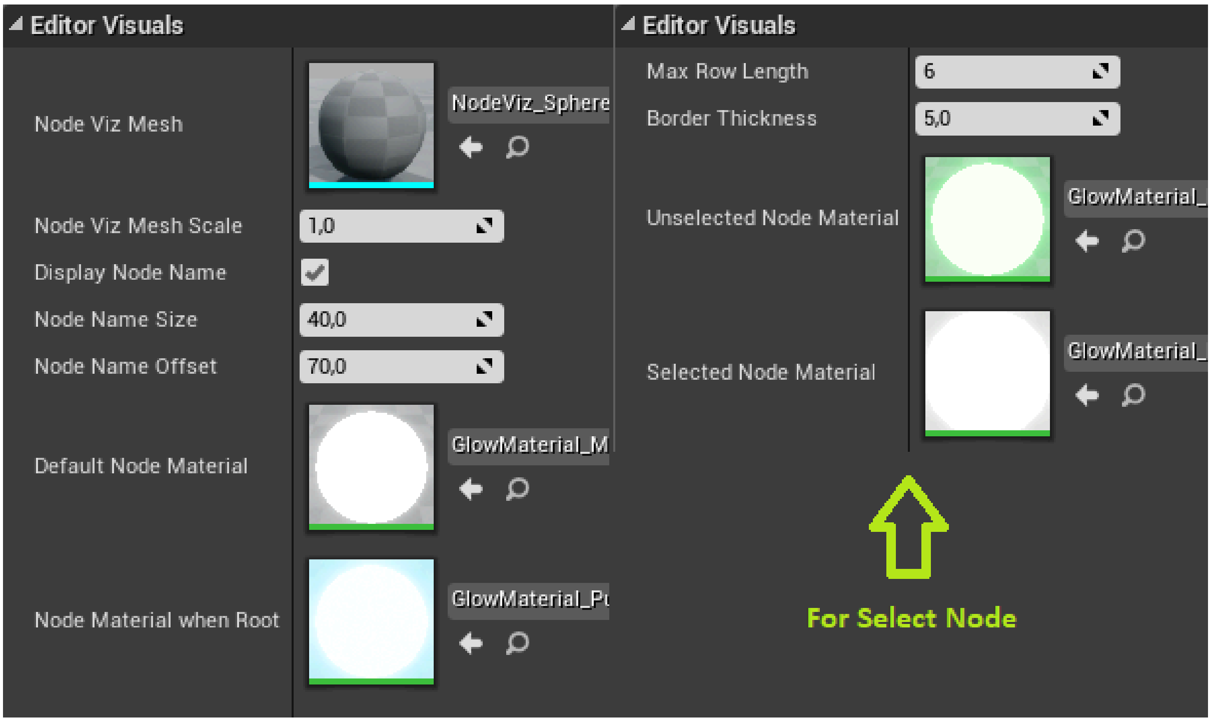


Variations	
Variation Amount	0
Variation Seed	777

Variations section. More info on variations available here.

Editor Visuals

Here you can change all the visual aspects of selected node:



Editor Visuals

Node Viz Mesh: NodeViz_Sphere

Node Viz Mesh Scale: 1,0

Display Node Name: ☑

Node Name Size: 40,0

Node Name Offset: 70,0

Default Node Material: GlowMaterial_M

Node Material when Root: GlowMaterial_Pu

Editor Visuals

Max Row Length: 6

Border Thickness: 5,0

Unselected Node Material: GlowMaterial_

Selected Node Material: GlowMaterial_

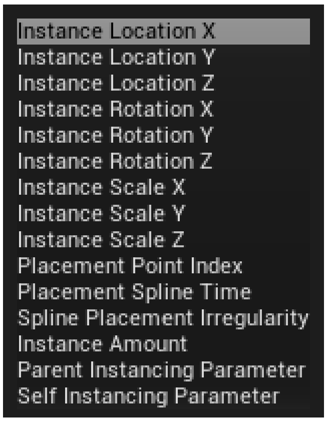
For Select Node

Instances & Variations

Every node can be instanced with specified instance amount & placement settings.
It allows to easily place node contents in larger amounts, e.g. place many leaves along stem or many pickets for a fence.

📗 Node geometry is calculated only once, then it is copied for each instance transform, saving CPU time during asset generation.

There are many instancing examples included in the project.



To add instances, set Instance Amount and select instance placement method.

Then parameterize one of the Instance parameters.
They can be set in the **Parametrization → Instance Parameters**.

For example, the included Plant Generator adds leaf instances along stem, using stem spline as placement method, with parameterized **Placement Spline Time**.

Example map: **/Maps/SimpleExamples/InstancingExample**

Variations feature works similarly, but it recalculates node mesh for every variation.
This can be used to generate X amount of different node variations, by parameterizing their generation rules.
Example usage: Grass Generator, where every grass blade is a Spline Shaped Mesh node variation with slightly altered spline.

📗 When using SSM (Spline Shaped Mesh) Variations on parent SSM, don’t use same Path shape for both variations and parent node, to avoid bad placement.

Parameters

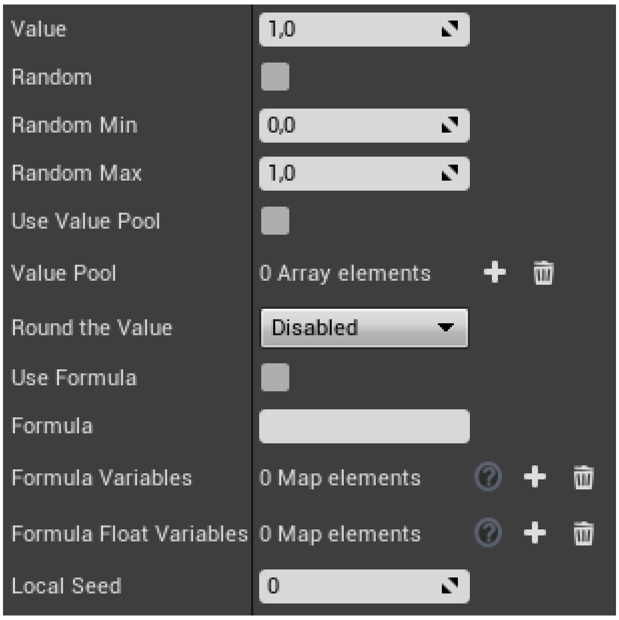
You can parameterize node variables in the node Details panel → Parametrization section, by assigning a custom Parameter actor that can be dropped into Workspace.
This allows to easily control desired node variables, straight from the Workspace view.

Parameters can also be used to specify random value ranges (or pools), set values based on current asset state (using the IF Conditions) or use the Formulas for Float parameters.

Example map: /Maps/SimpleExamples/ParameterizationExample

Float Parameter

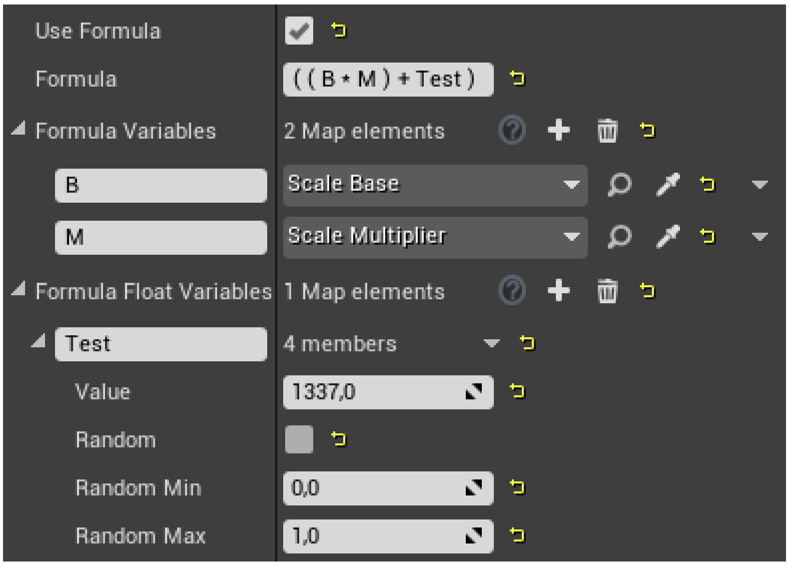
Float Parameters can set value of any node Float variable, e.g. location/rotation/scale XYZ values and all the other exposed variables.



Value	1,0
Random	□
Random Min	0,0
Random Max	1,0
Use Value Pool	□
Value Pool	0 Array elements
Round the Value	Disabled
Use Formula	□
Formula	
Formula Variables	0 Map elements
Formula Float Variables	0 Map elements
Local Seed	0

Formulas

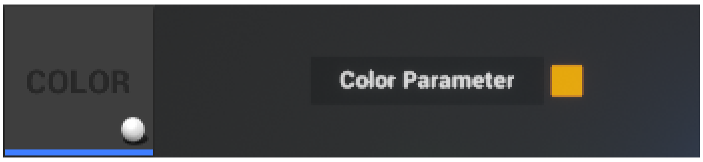
Formulas allow to set Float Parameter value from custom math expression, with possibility to create local variables (e.g. from other Float Parameter values, which allows to create parameter value dependencies).
Formulas can evaluate fully parenthesized arithmetic expressions, with spaces between every operand & operator.



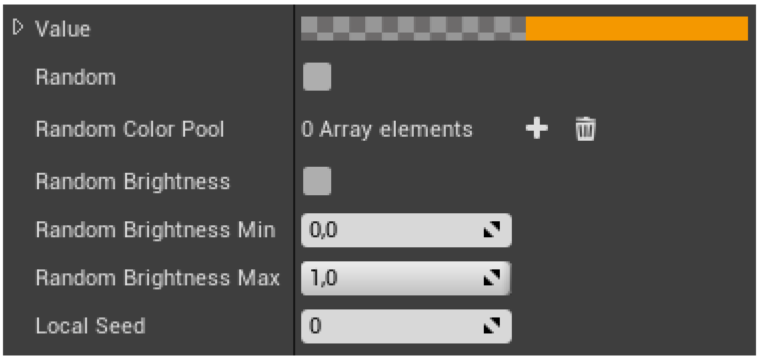
Use Formula	☑
Formula	((B * M) + Test)
Formula Variables	2 Map elements
B	Scale Base
M	Scale Multiplier
Formula Float Variables	1 Map elements
Test	4 members
Value	1337,0
Random	□
Random Min	0,0
Random Max	1,0

See the tooltips for more info.
Various example formula usages are available in the included generator maps.
Example map: /Maps/SimpleExamples/ParameterFormulaExample

Color Parameter

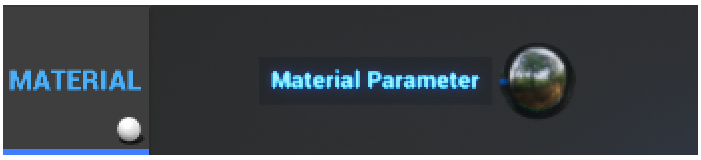


Color Parameters can set value of any node color variable, e.g. the Vertex Color.
It’s possible to specify random Color pool, with optional brightness values.

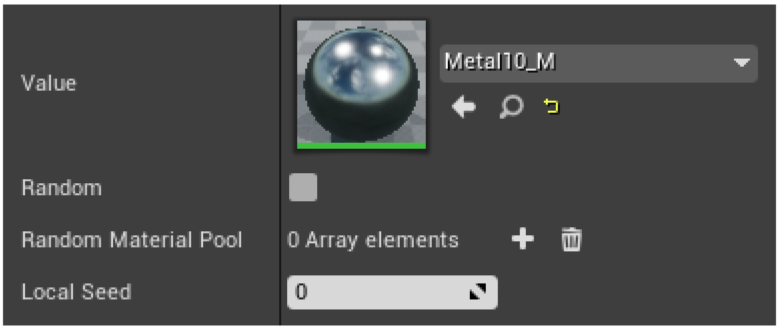


Value	
Random	□
Random Color Pool	0 Array elements
Random Brightness	□
Random Brightness Min	0,0
Random Brightness Max	1,0
Local Seed	0

Material Parameter

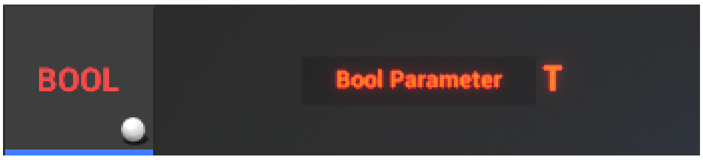


Material Parameters can set value of specified node & decal **Material** variables.
It’s possible to specify random Material pool.

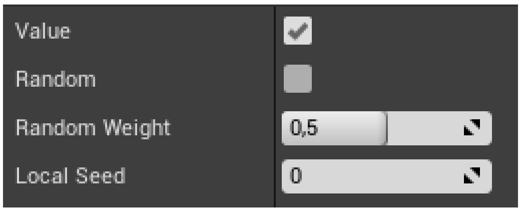


Value	Metal10_M
Random	□
Random Material Pool	0 Array elements
Local Seed	0

Bool Parameter

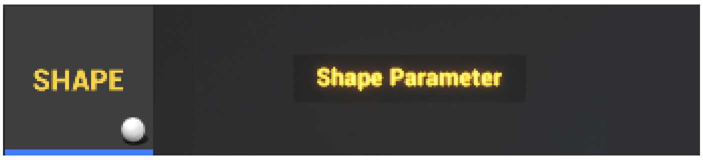


Bool Parameters can set value of any node Bool variable, e.g. the **Enabled** (ON/OFF) variable.
It’s possible to make the bool value random, with specified weight.

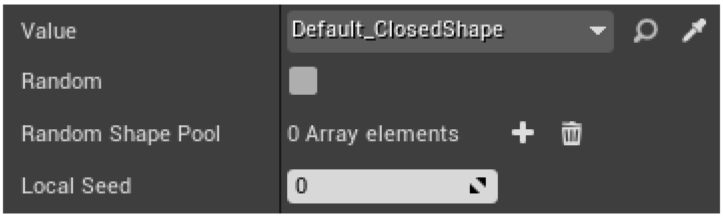


Value	☑
Random	□
Random Weight	0,5
Local Seed	0

Shape Parameter

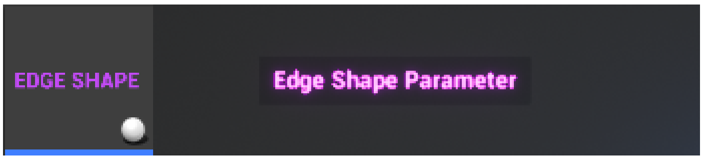


Shape Parameters can set value of any node shape variable, e.g. the **Contour Shape** in Contour Shaped Mesh node, **Shape** in Spline Shaped Mesh node, etc.
It’s possible to specify random Shape pool.

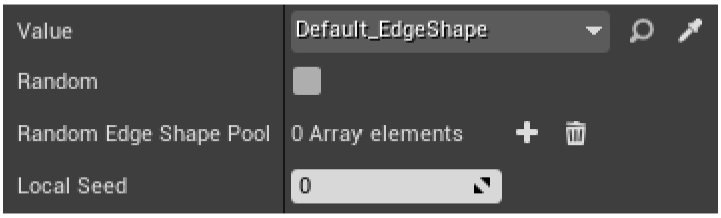


Value	Default_ClosedShape
Random	□
Random Shape Pool	0 Array elements
Local Seed	0

Edge Shape Parameter

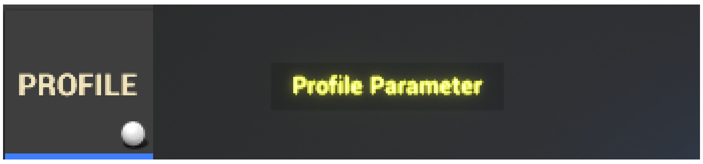


Edge Shape Parameters can set value of the **Contour Edge Shape** variable in any Contour Shaped Mesh node.
It’s possible to specify random Edge Shape pool.

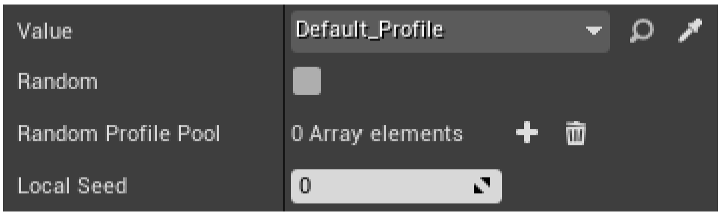


Value	Default_EdgeShape
Random	□
Random Edge Shape Pool	0 Array elements
Local Seed	0

Profile Parameter



Profile Parameters can set value of the **Profile X** & **Profile Y** variables in any Spline Shaped Mesh node.
It’s possible to specify random Profile pool.

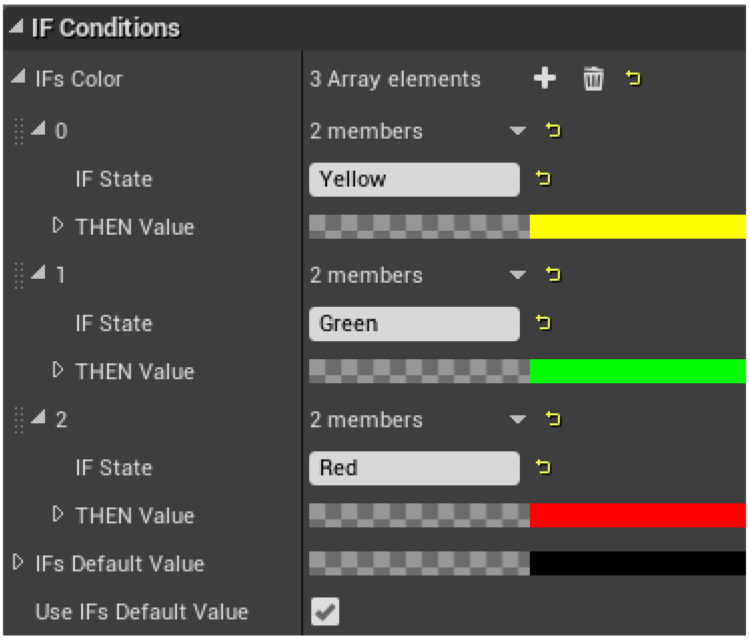


Value	Default_Profile
Random	□
Random Profile Pool	0 Array elements
Local Seed	0

IF Conditions

IF Conditions allow to specify custom-named asset states, based on current parameter values. These states can be checked to set parameter values, depending on current true/false state of specified state.
For example, you can turn plant leaves off if current stem height is lower than a specified value, set asset part scale based on currently selected Shape parameter, etc.

Example IF section:



IF Conditions

IFs Color: 3 Array elements

0: 2 members
- IF State: Yellow
- THEN Value: 🟨

1: 2 members
- IF State: Green
- THEN Value: 🟩

2: 2 members
- IF State: Red
- THEN Value: 🟥

IFs Default Value: ⬛

Use IFs Default Value: ☑

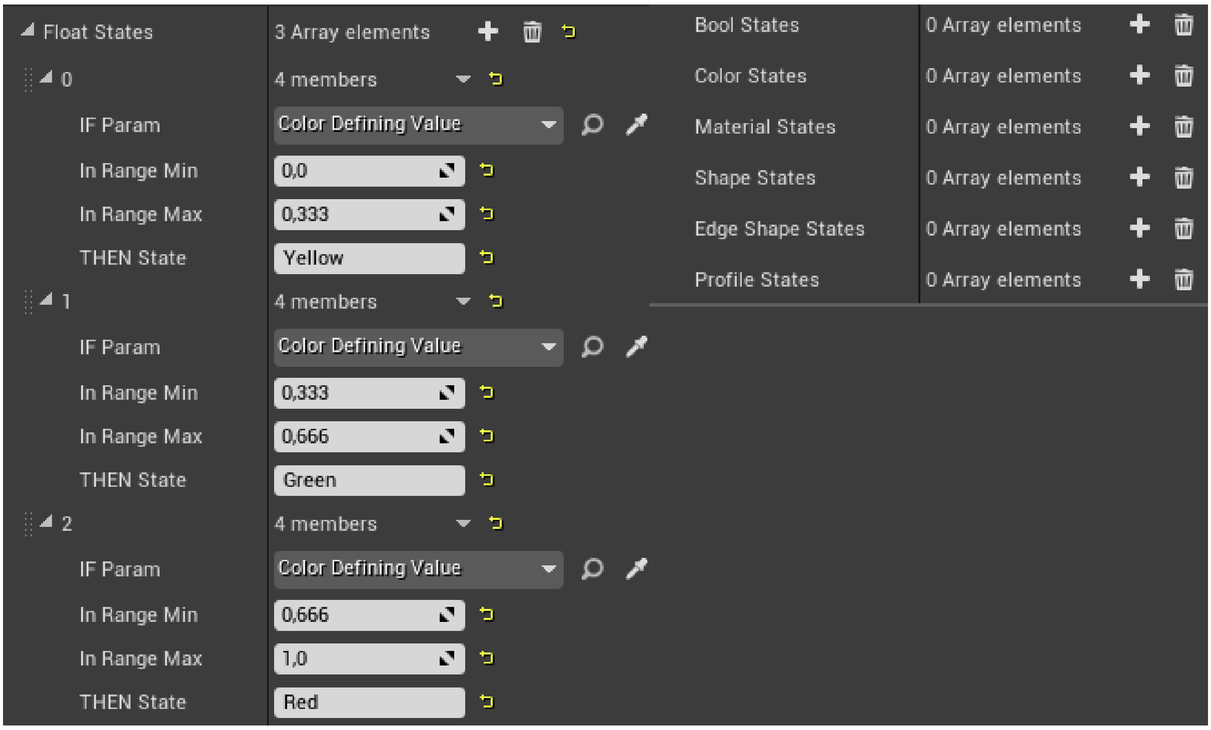
Basically, it works like an ‘if / else if’ stack, setting specified value based on current state.
More info available in the tooltips.

💡 **All the parameters have tooltips with explanation, just hover over any parameter:**

Path Depth Start: 0,0

Path Depth End: 1,0

Example State section:

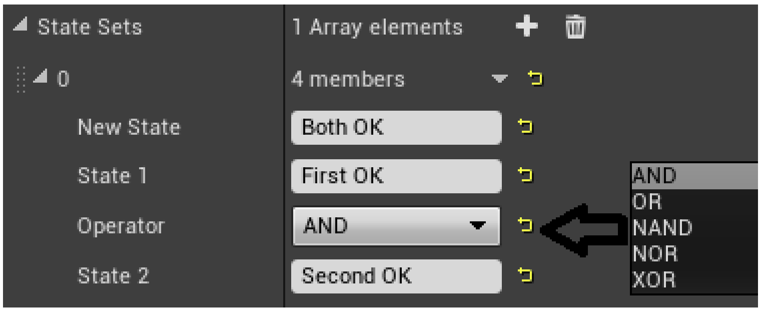


Float States	3 Array elements	Bool States	0 Array elements
0	4 members	Color States	0 Array elements
IF Param	Color Defining Value	Material States	0 Array elements
In Range Min	0,0	Shape States	0 Array elements
In Range Max	0,333	Edge Shape States	0 Array elements
THEN State	Yellow	Profile States	0 Array elements
1	4 members		
IF Param	Color Defining Value		
In Range Min	0,333		
In Range Max	0,666		
THEN State	Green		
2	4 members		
IF Param	Color Defining Value		
In Range Min	0,666		
In Range Max	1,0		
THEN State	Red		

In this example, it sets Yellow, Green or Red state depending on current float value.
States can be defined using all parameter types, not only Float Parameters.

📗 **You can use exclamation mark in front of any state name to negate it.**

State Sets allow to combine existing states using simple two-element boolean expression:



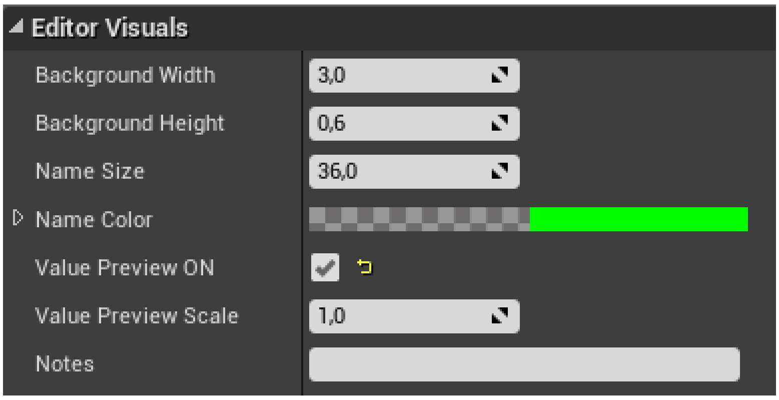
State Sets	1 Array elements
0	4 members
New State	Both OK
State 1	First OK
Operator	AND
State 2	Second OK

Example map: /Maps/SimpleExamples/IFConditionsExample

Another example is available in the included Bow Generator, where ‘String Offset Y’ parameter uses IF conditions to set its value based on current main Path.
‘String Offset Y Final’ parameter takes it and applies current width.
This way the string location fits current bow shape.

Editor Visuals

This section allows to set custom parameter color, width and other visual things:



Editor Visuals

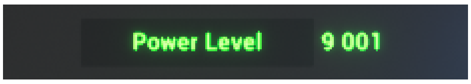
Background Width	3,0
Background Height	0,6
Name Size	36,0
Name Color	🟩
Value Preview ON	☑
Value Preview Scale	1,0
Notes	

To set parameter name, just rename the actor - the name display text will match it.

Value Preview allows to preview current parameter value straight from workspace view.

The **Notes** field can be used to store some info that is important for this parameter.
It's just an additional option, you can also use workspace Comments for that.

📗 **Parameters can be named by changing the actor name.**
The name display text will match it:

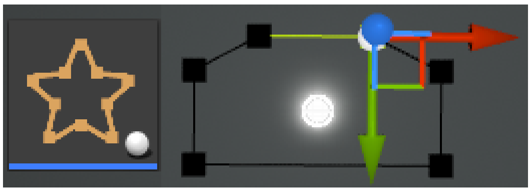


Power Level 9 001

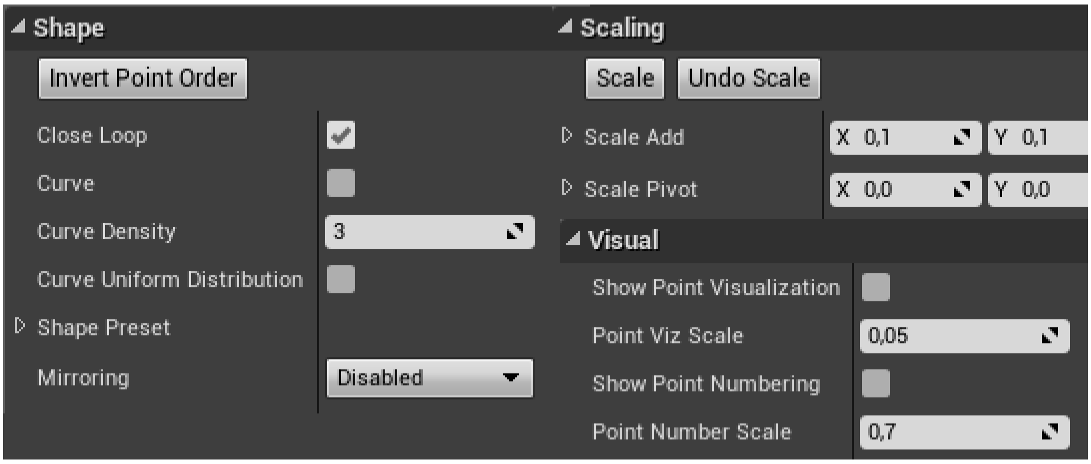
Shapes

Shapes are used to define mesh geometry when assigned to nodes like Spline Shaped Mesh or Contour Shaped Mesh.
There is a designated space for shapes in the default level, you can go there quickly by clicking Perspective → ShapeRoom in viewport.

Shape



Spline-based shape. You can add points by holding left Alt & dragging out a point or use one of the other methods - more info available here.
White point visualizes the (0,0,0) relative shape location, default pivot point.



Shape

Invert Point Order

Close Loop	☑
Curve	□
Curve Density	3
Curve Uniform Distribution	□
▷ Shape Preset	
Mirroring	Disabled

Scaling

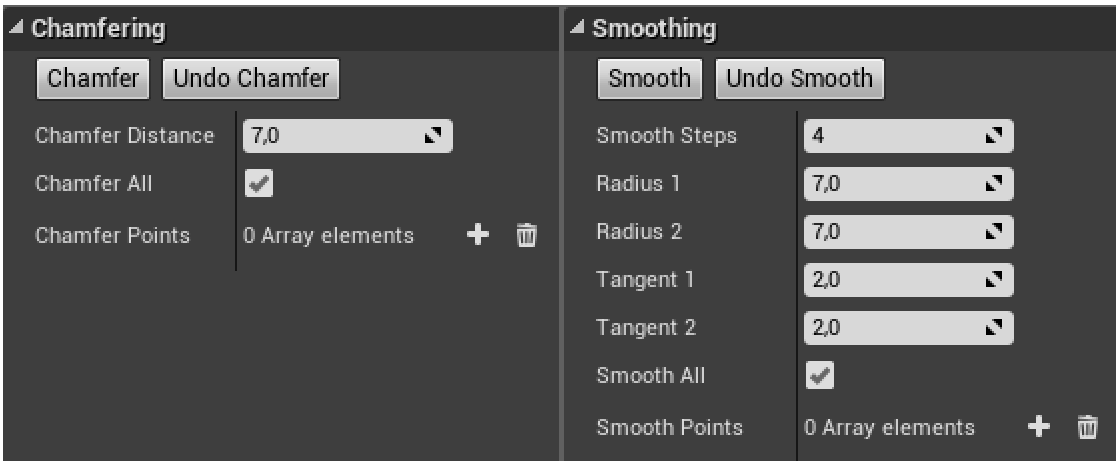
Scale | Undo Scale

▷ Scale Add	X 0,1	Y 0,1
▷ Scale Pivot	X 0,0	Y 0,0

Visual

Show Point Visualization	□
Point Viz Scale	0,05
Show Point Numbering	□
Point Number Scale	0,7

Besides scaling, shapes can also be chamfered or smoothed:



Chamfering

Chamfer | Undo Chamfer

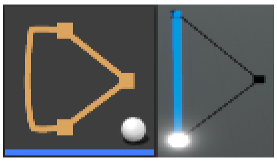
Chamfer Distance	7,0
Chamfer All	☑
Chamfer Points	0 Array elements

Smoothing

Smooth | Undo Smooth

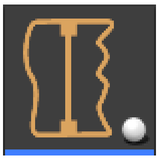
Smooth Steps	4
Radius 1	7,0
Radius 2	7,0
Tangent 1	2,0
Tangent 2	2,0
Smooth All	☑
Smooth Points	0 Array elements

Edge Shape



Child of the Shape class.
Used to define **Edge Shape** in Contour Shaped Mesh.

Profile



Child of the Shape class.
Used to define optional profile in Spline Shaped Mesh.

Shape generators

Shape generators are designed to generate different shapes based on specified parameters.
Included shape generators: Circle, Equilateral Triangle, Helix, Rectangle.

Included example generators can be found in **/Maps/ShapePresetMakingMap**
The generator blueprints are available in the **/Blueprints/Internal/ShapeGen/** folder.

Shape presets

Any Shape can be saved as a preset which then can be easily loaded into any other Shape.
To save it, under the **Generated Data to Copy** section right click the **Spline Data** and click **Copy**.
Then you can paste it into the **/Blueprints/Internal/ShapeGen/SplineShapePresets** data table by creating a new row, right clicking the **Spline Data** and clicking **Paste**.

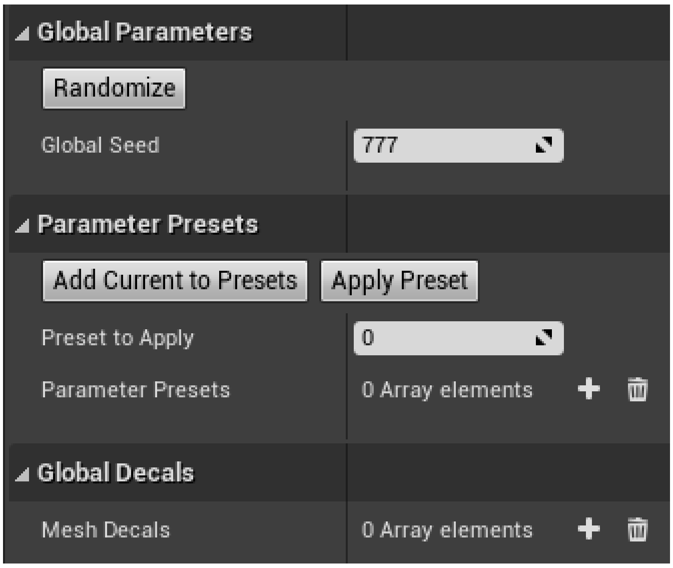
To load a preset, under the **Shape** → **Shape Preset** section just pick a Row Name with desired preset.

Shape parameterization

Shape points can be parameterized - basically, you can control any shape point using Float Parameters.
To parameterize a shape point, add an element to the **Point Parameters** array under the **Parametrization** section and assign a Float Parameter to X,Y,Z location values.

Global Options

The rotating cogwheel contains all the global options:



Global Parameters

Global Seed: 777

Parameter Presets

Preset to Apply: 0

Parameter Presets: 0 Array elements

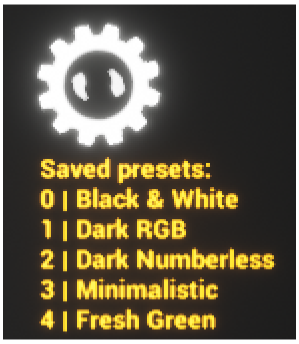
Global Decals

Mesh Decals: 0 Array elements

More info is available in the tooltips.

The **Randomize** button randomizes all parameters that are set to random value range. Basically, each click generates new version of your asset.

Parameter Presets allow to save current parameter values as a preset. Presets can be renamed, altered and loaded at any time, using the **Apply Preset** button.

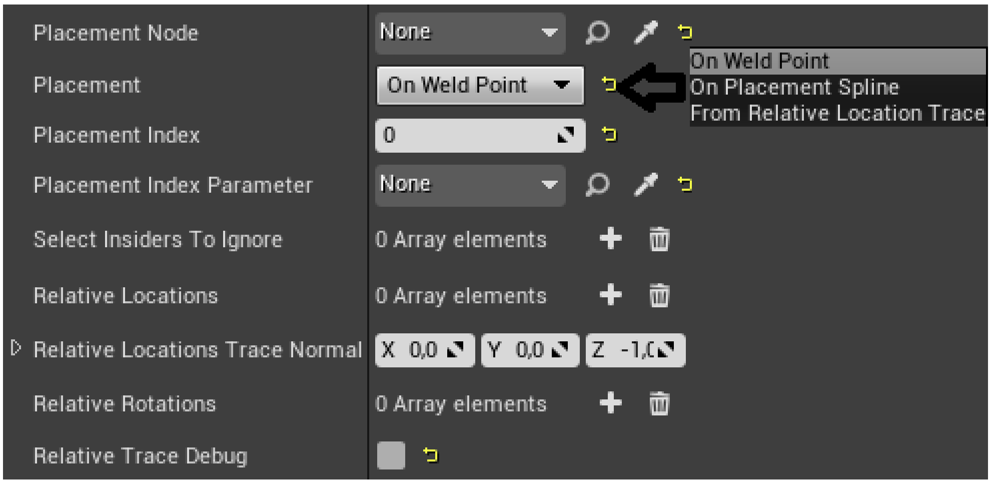


Saved presets:
0 | Black & White
1 | Dark RGB
2 | Dark Numberless
3 | Minimalistic
4 | Fresh Green

Global Mesh Decals

Global Decals can be used to add mesh decals to your asset.

Decal placement is calculated from location relative to specified node Weld Point, Placement Spline or from node-relative line traces:

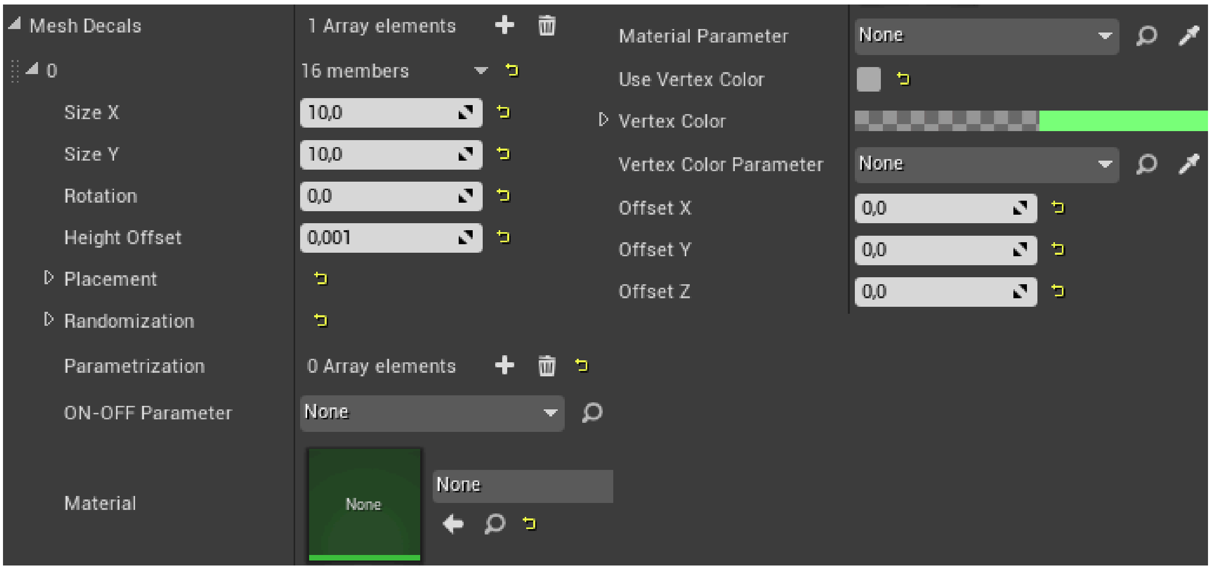


Placement Node	None
Placement	On Weld Point
Placement Index	0
Placement Index Parameter	None
Select Insiders To Ignore	0 Array elements
Relative Locations	0 Array elements
Relative Locations Trace Normal	X 0,0 Y 0,0 Z -1,(
Relative Rotations	0 Array elements
Relative Trace Debug	☐

Example usage: The included Gun Generator.

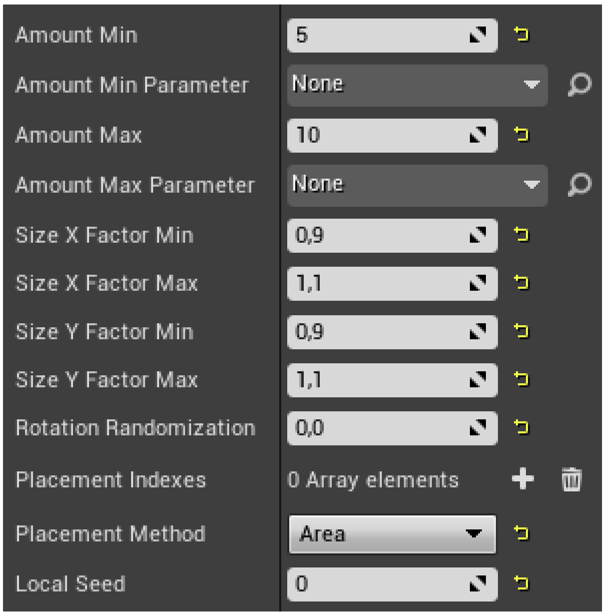
Mesh Decals

Single plane mesh decals placed at specified point.



Mesh Decals	1 Array elements
0	16 members
Size X	10,0
Size Y	10,0
Rotation	0,0
Height Offset	0,001
Placement	
Randomization	
Parametrization	0 Array elements
ON-OFF Parameter	None
Material	None
Material Parameter	None
Use Vertex Color	□
Vertex Color	🟩
Vertex Color Parameter	None
Offset X	0,0
Offset Y	0,0
Offset Z	0,0

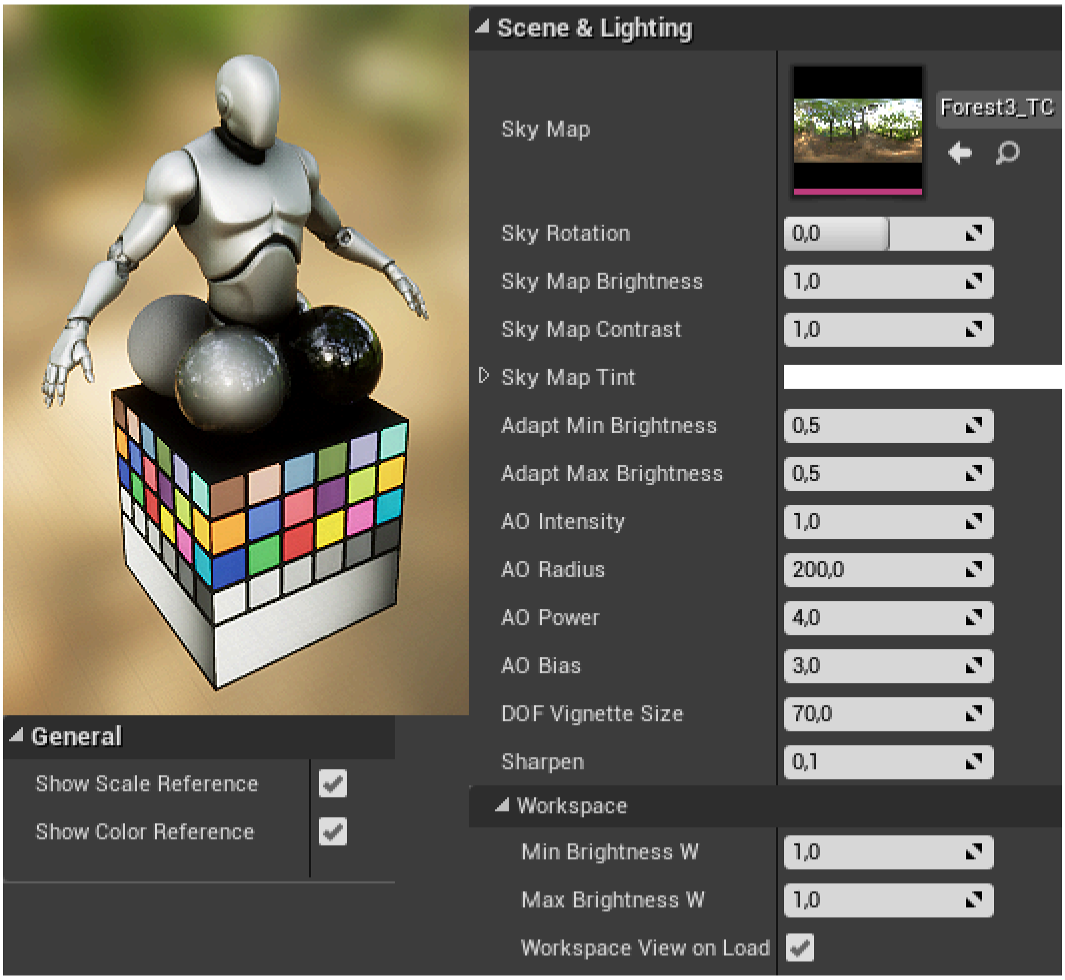
Randomization options:



Amount Min	5
Amount Min Parameter	None
Amount Max	10
Amount Max Parameter	None
Size X Factor Min	0,9
Size X Factor Max	1,1
Size Y Factor Min	0,9
Size Y Factor Max	1,1
Rotation Randomization	0,0
Placement Indexes	0 Array elements
Placement Method	Area
Local Seed	0

Scene Settings

Scene Settings actor serves as a human scale reference (using the standard UE Mannequin) and a color reference, both toggleable.
It also provides HDRI lighting for your scene and it's customizable:



General

Show Scale Reference ☑
Show Color Reference ☑

Scene & Lighting

Sky Map: Forest3_TC
Sky Rotation: 0,0
Sky Map Brightness: 1,0
Sky Map Contrast: 1,0
Sky Map Tint
Adapt Min Brightness: 0,5
Adapt Max Brightness: 0,5
AO Intensity: 1,0
AO Radius: 200,0
AO Power: 4,0
AO Bias: 3,0
DOF Vignette Size: 70,0
Sharpen: 0,1

Workspace

Min Brightness W: 1,0
Max Brightness W: 1,0
Workspace View on Load ☑

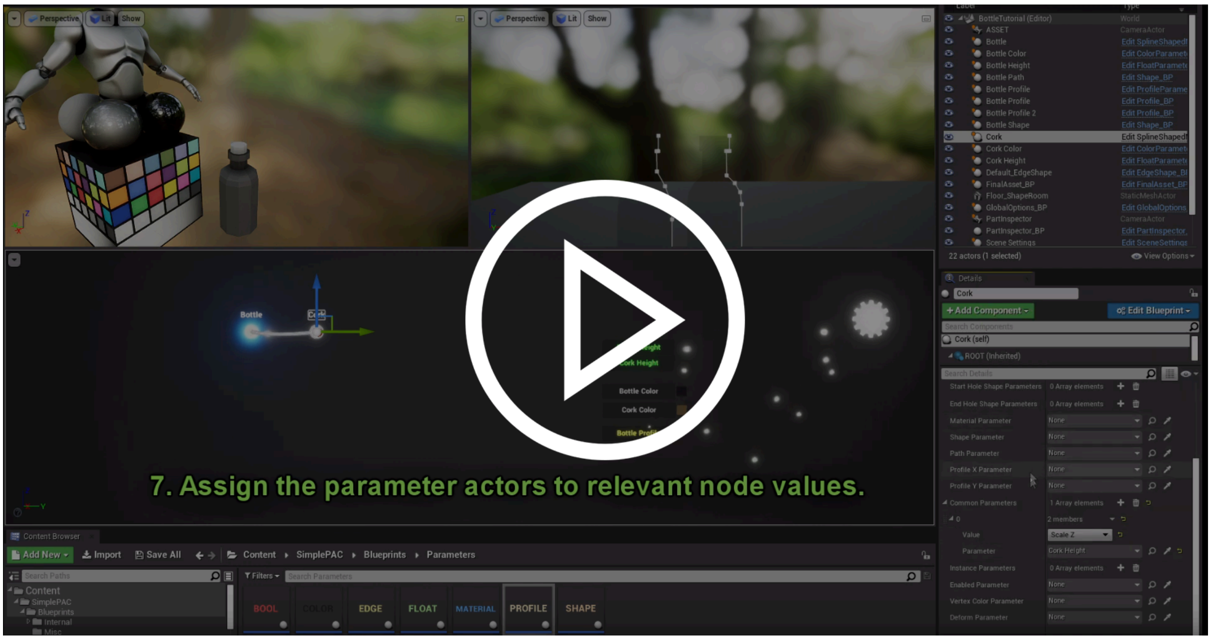
By default it's located near final asset mesh, but you can move it freely.

📗 Changing **Sky Rotation** will affect whole scene lighting, not only the skybox.

Tutorials

Creating the Simple Bottle Generator

Step 1: [Create new map for the generator](#) & [set the viewports.](#)
Step 2: Please follow the video:



7. Assign the parameter actors to relevant node values.

Label	Type
BottleTutorial (Editor)	World
ASSET	CameraActor
Bottle	Edit SplineShaped
Bottle Color	Edit ColorParamet
Bottle Height	Edit FloatParamete
Bottle Path	Edit Shape_BP
Bottle Profile	Edit ProfileParame
Bottle Profile	Edit Profile_BP
Bottle Profile 2	Edit Profile_BP
Bottle Shape	Edit Shape_BP
Cork	Edit SplineShaped
Cork Color	Edit ColorParamet
Cork Height	Edit FloatParamete
Default_EdgeShape	Edit EdgeShape_BI
FinalAsset_BP	Edit FinalAsset_BP
Floor_ShapeRoom	StaticMeshActor
GlobalOptions_BP	Edit GlobalOptions
PartInspector	CameraActor
PartInspector_BP	Edit PartInspector
Scene Settings	Edit SceneSettings

22 actors (1 selected)

(https://www.youtube.com/watch?v=ycu9CYdmDiM)

This video shows an example simple asset creation process, in this case creating the [Simple Bottle Generator](#) that is included in the project.

The result is available in the /Maps/SimpleGenerators/SimpleBottleGenerator1 map.

Adding height control to the Fence Generator

This example shows how you can use Float Parameters and Formulas to add more control to your generators.
Open the included /Maps/FenceGenerator/FenceGenerator1_FlatShaded_PreTutorial map and set the workspace view like it's shown in the Quick Start guide.

1. Drag one /Blueprints/Parameters/FloatParameter into the workspace and name it e.g. 'Fence Height', set its Value to 0.5.
2. In the Pickets node, add Scale X to the Parametrization → Common Parameters list and assign previously added Fence Height parameter there.

Now you can control fence height by changing the Fence Height parameter value...

… But when changing it to e.g. 0.3, you can see that the horizontal planks don't reflect this change. Here we can use the float parameter Formula:

1. Select the Plank 2 Height parameter and add one element to the Formula Variables list. Name it e.g. "H" and assign the Fence Height parameter to it. This way we can access values from other parameters.
2. In the Formula field type in "(140 * H)" and check Use Formula.

Now the horizontal plank should be correctly aligned and when you change the Fence Height, it will react accordingly.
Of course you can change the “140” value to your liking, or even make it a separate parameter if you wish.

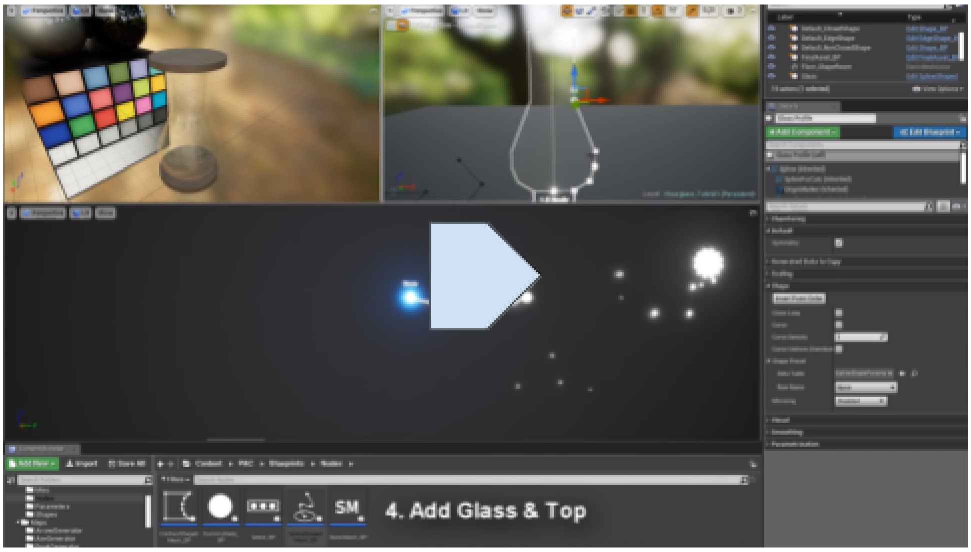
The final result of these steps is available in the /Maps/FenceGenerator/FenceGenerator1_FlatShaded map.

Creating Hourglass Generator

This tutorial was created in the extended PAC version, but it's still useful here, as you can replicate most of this in the SimplePAC - just ignore all the material/UV steps and use vertex colors instead.

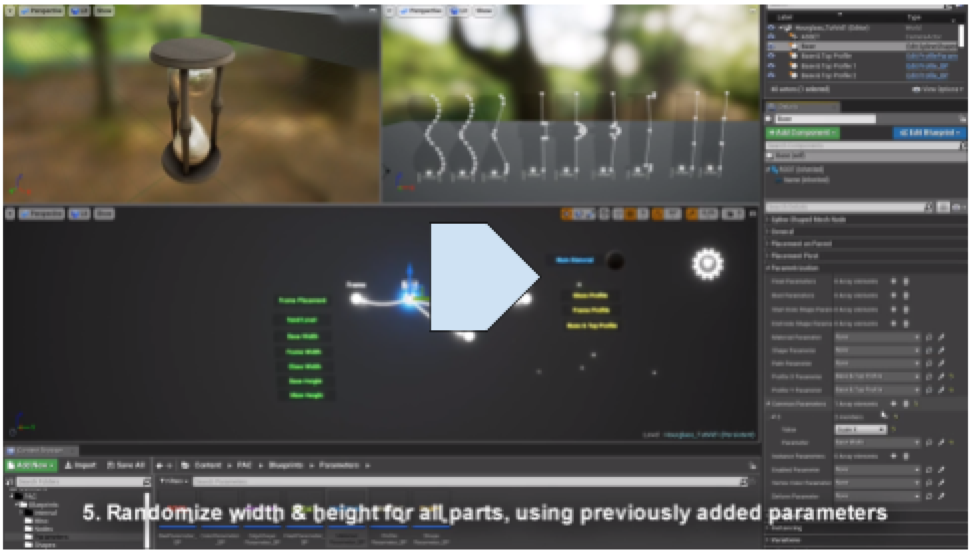
Step 1: Create new map for the generator & set the viewports.

Part 1, click the image or this link: https://youtu.be/alN-ZxcstRA



4. Add Glass & Top

Part 2, click the image or this link: https://youtu.be/k806gXruURM



5. Randomize width & height for all parts, using previously added parameters

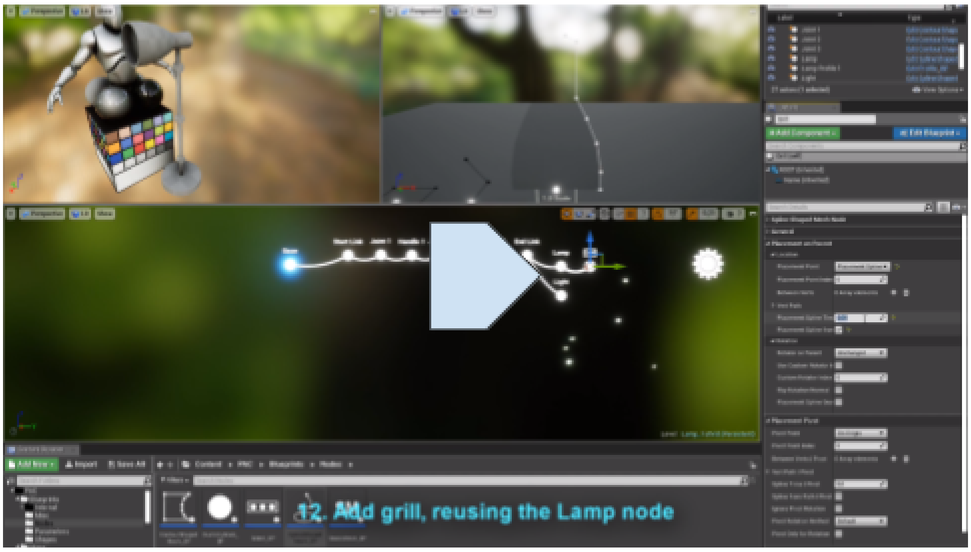
Demonstrated features: Spline Shaped Mesh, Part Inspector, Weld Point placement, Placement Spline placement, Instancing, Parameters and more.

Creating Lamp Generator

This tutorial was created in the extended PAC version, but it's still useful here, as you can replicate most of this in the SimplePAC - just ignore all the material/UV steps and use vertex colors instead.

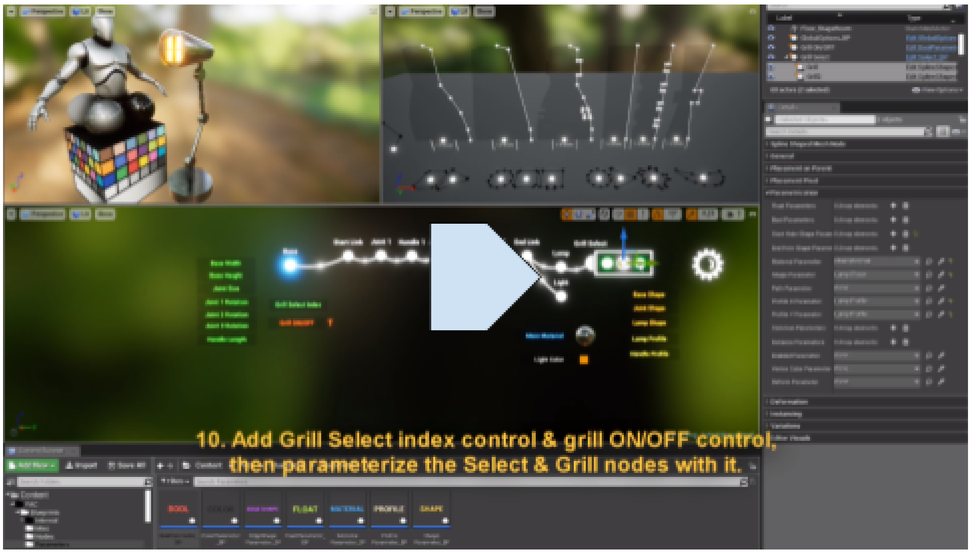
Step 1: Create new map for the generator & set the viewports.

Part 1, click the image or this link: https://youtu.be/i9MeI6b0h5g



12. Add grill, reusing the Lamp node

Part 2, click the image or this link: https://youtu.be/4aTr-kz1iaM



10. Add Grill Select index control & grill ON/OFF control, then parameterize the Select & Grill nodes with it.

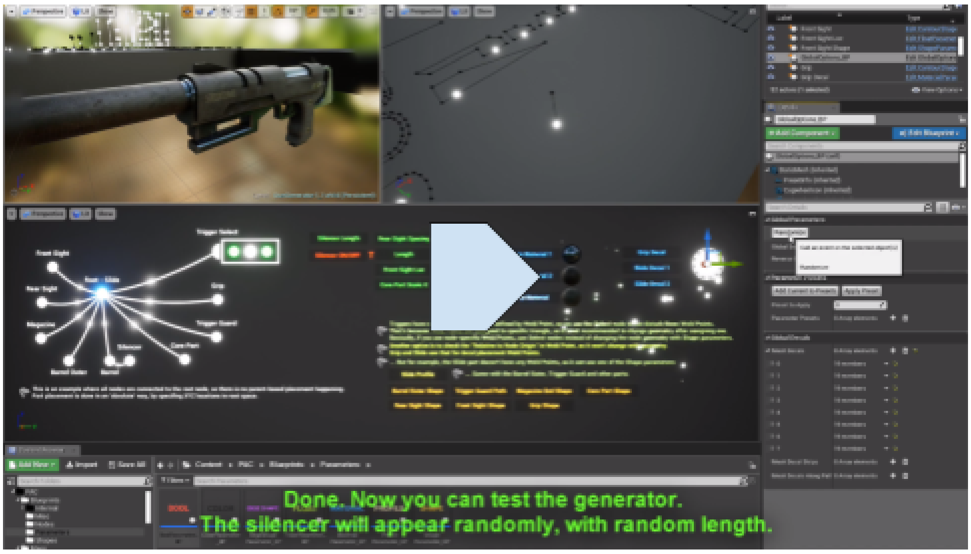
Demonstrated features: Select Node, Spline Shaped Mesh, Holes, Contour Shaped Mesh, Shape Scaling, Workspace appearance, Placement Spline placement, Parameters and more.

Adding silencer to the Gun Generator

This tutorial was created in the extended PAC version, but it’s still useful here, as you can replicate most of this in the SimplePAC - just ignore all the material/UV steps and use vertex colors instead.

Step 1: Save the Gun Generator as a new map & set the viewports.

Click the image or this link: https://youtu.be/SGyNeAAnvng



Done. Now you can test the generator.
The silencer will appear randomly, with random length.

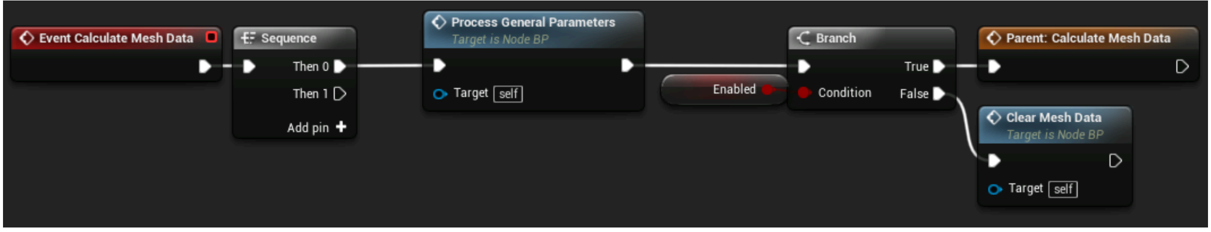
Demonstrated features: Modifying the existing example generators, Spline Shaped Mesh, Holes, Placement Spline placement, Parameters and more.

Advanced: Creating custom node

You can create custom nodes by deriving from the base Node_BP class.
This allows to create your own nodes, with custom geometry generators inside.

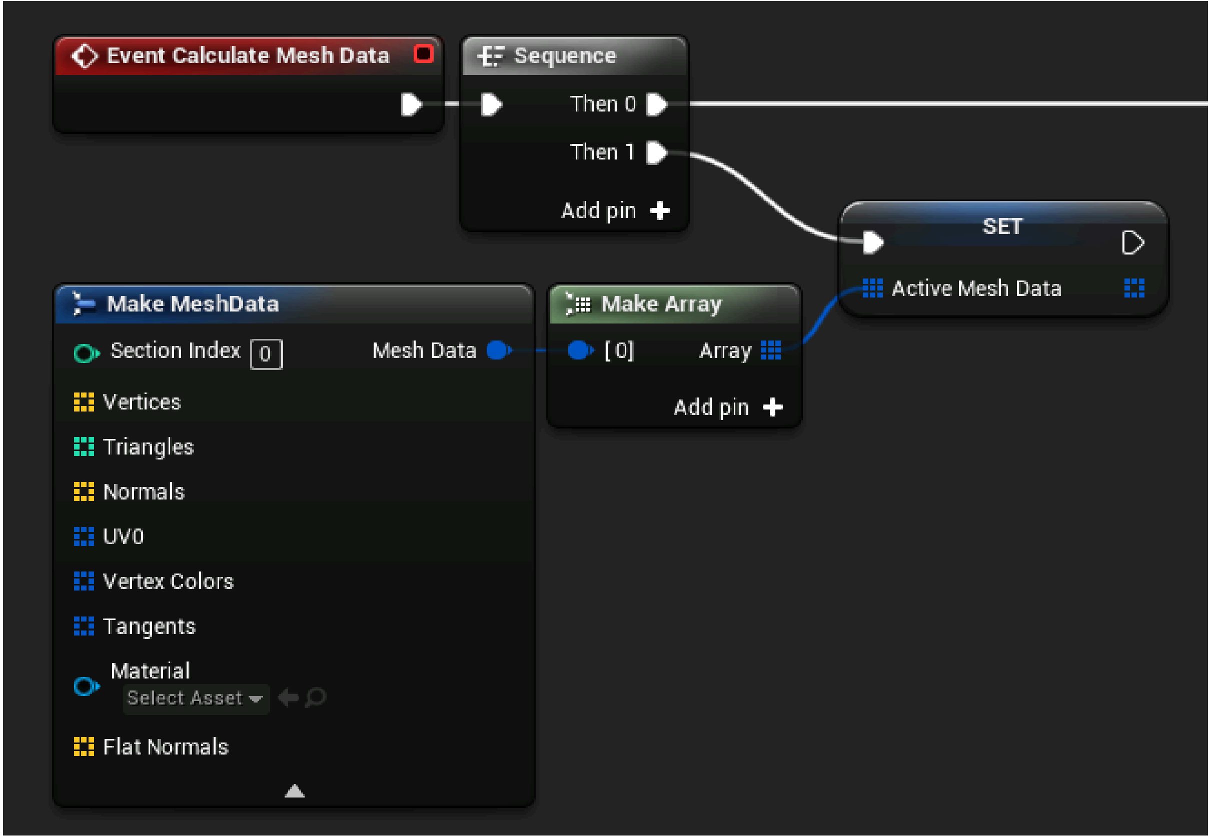
Here's a simple Box node example that shows where to return the geometry data:

1. Create new Blueprint Class, pick the Node_BP in the Pick Parent Class window (search for it in the All Classes tab)
2. In the blueprint, click Add New → Override Function → Calculate Mesh Data
3. Add a Sequence node and this logic:



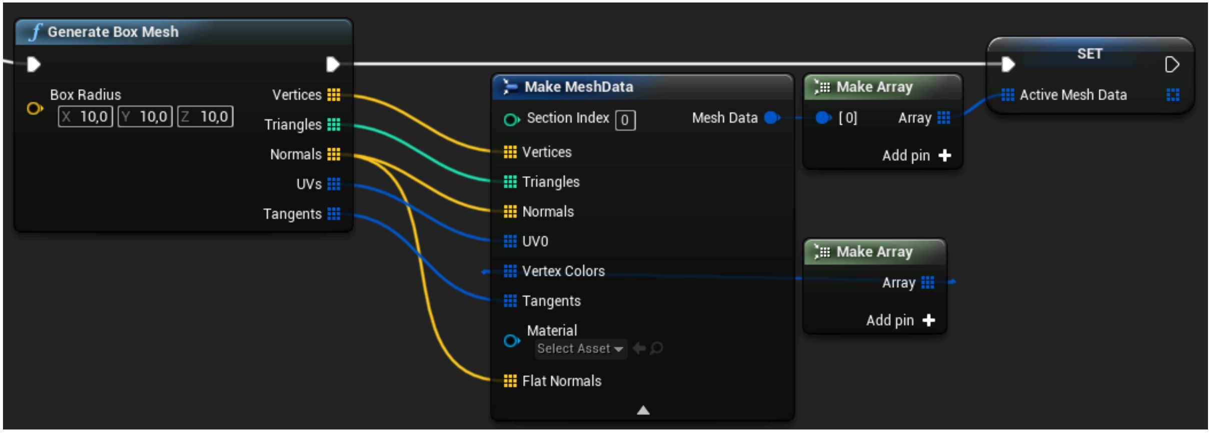
Available to copy from here.

4. In the next step, set Active Mesh Data variable like that, here you can pass any mesh geometry data that you've calculated:



Available to copy from here.

5. As an example, let’s add Box geometry, using the **Generate Box Mesh** node:



Available to copy from here.

… The node is ready to use.
After compiling, it can be dragged into the Workspace just like the included nodes.

Now you can e.g. expose the **Box Radius** and the **Material** values as public variables, so they can be accessed from Workspace view, when your node is selected.

To make these variables parameterizable, see how it’s implemented in the included nodes (**Blueprints/Nodes/** folder). The logic can be node specific, so it all depends on what kind of geometry your node generates and what other features it has.

Simple Examples

Most of the main features have simple usage examples (besides the included generators) available in the /Maps/SimpleExamples/ folder, all with explanations/comments.

Ending

In case of any questions or issues, please contact me on the support email that is listed on my Marketplace profile.

Marketplace profile: https://www.unrealengine.com/marketplace/profile/S.+Krezel

Forum thread:
https://forums.unrealengine.com/unreal-engine/marketplace/1584105-procedural-asset-creator

Thank you,
S. Krezel

Table of contents

📗 For better online navigation, please use the Document Outline.
If you don't see it on the left, enable it by clicking **View → Show document outline.**