String Light Plugin

Important Notice for Update 1.2:

You can ignore this message if this is your first time installing this plugin.

Plugin Folder is renamed from

[EngineDirectory]/Engine/Plugins/Marketplace/StringLightPack

To

[EngineDirectory]/Engine/Plugins/Marketplace/StringLightPlugin

Please follow these steps:

1. **If you have modified the original plugin Content, make a backup of StringLightPack Content (we don't recommend modifying the original Content as they can be overwritten with newer updates.)**
2. **Uninstall the plugin.**
3. **Install the new plugin version 1.2**
4. **If you have modified the original plugin Content, copy the StringLightPack Content backup to StringLightPlugin.**
5. Please don't copy the **Maps** and **DemoRoom** folder.

Release Notes

Version 1.2

Summary

New

- Group Instanced Meshes of different String Lights for higher performance
- Create Static Mesh(es) from String Lights with different merge options
- Specify Lights to add to strings by Index

- Distance Based Optimization
- Improved Directional Wind
- Segment Collision for String Lights (Capsule Sweeps)
- Support Per Particle Force.
- Better Debug View for String Lights.
- More Blueprint nodes.
- More Features, Improvements and Optimizations.

Fixed

- Transform of Attached Light Bulbs
- Socket Transforms of String Light Components.
- Shadow Casting of String Light Components.
- Calculated Bounds of String Light Components.
- Line Trace on Light Bulbs.
- Shadow casting option on sections of static mesh assets.
- Other minor issues.

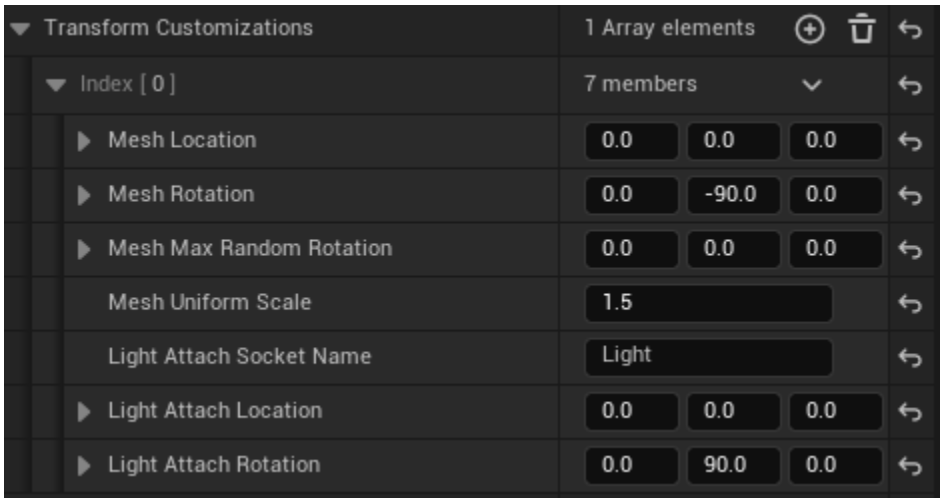
Important Changes

There are some changes that might affect your project but **not necessarily**.
You may have to re-adjust some settings but this should be easy to fix.

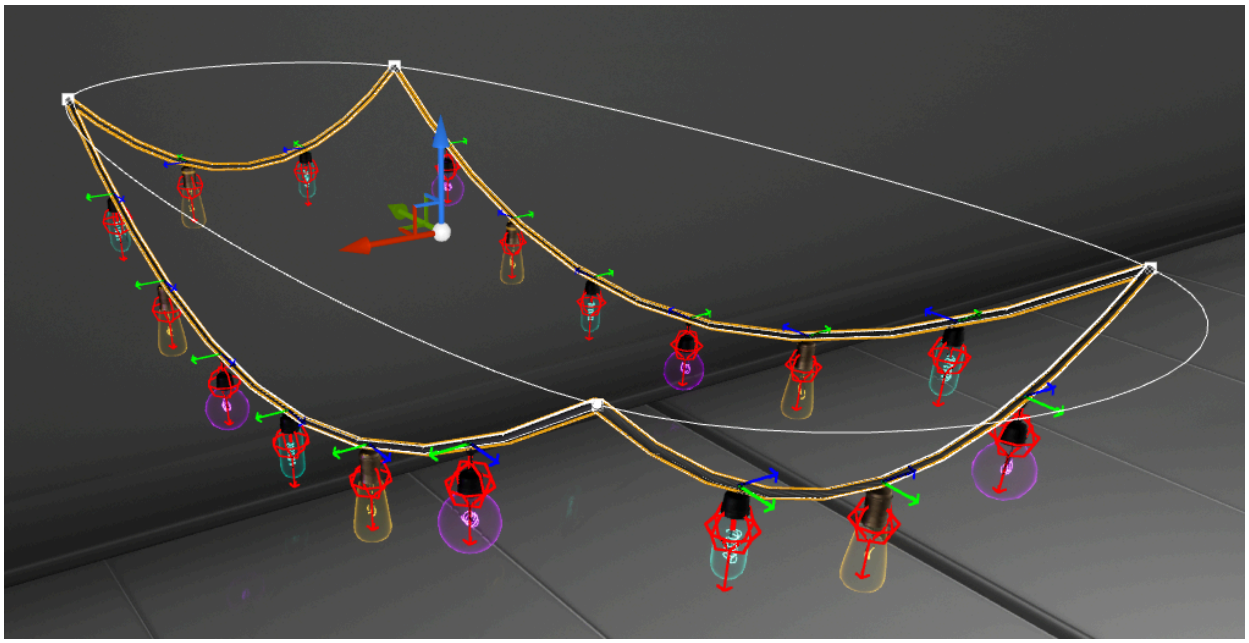
- Transforms
- Showcase Content
- Plugin Folder Name

Transforms

Transform calculation of attached light bulbs and other components to string lights has been fixed. As a result, you may have to re-adjust transform settings.

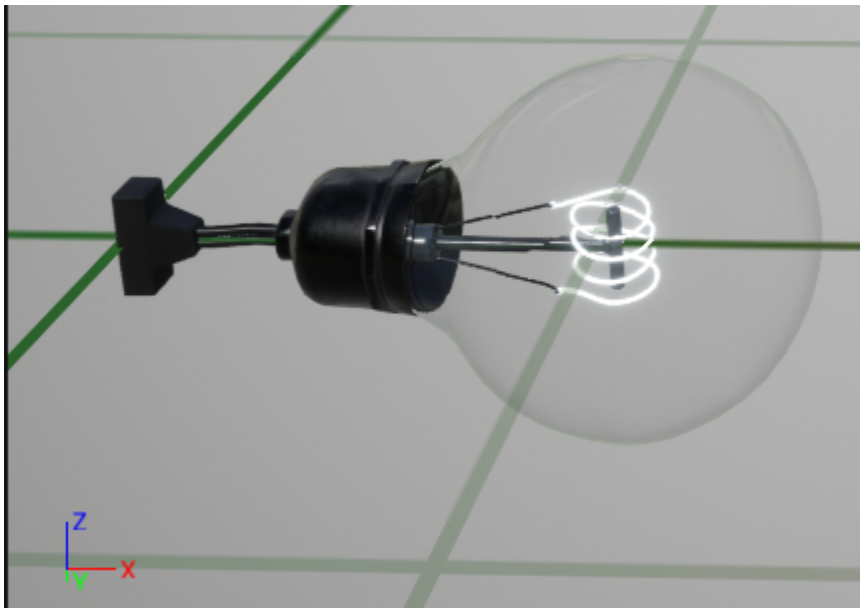


In order to easily find out what rotation or location offset to use, you can use the new draw debug feature.



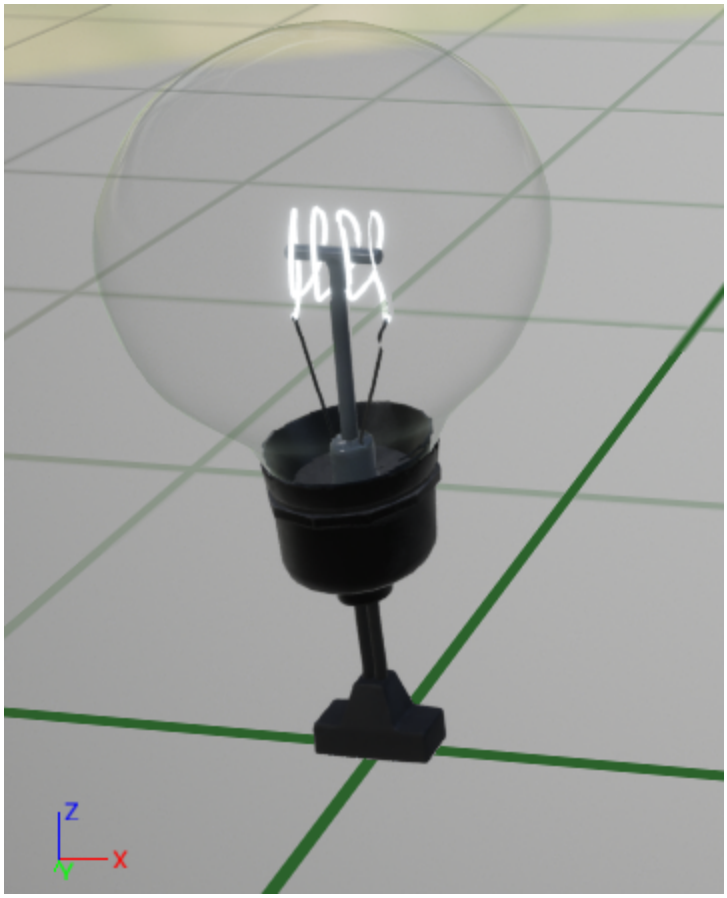
Here are some examples on how to configure rotation for your light bulls.

If your light bulb is pointing forward (X axis)

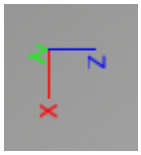


All String Light Sockets have their **X axis** pointing **downwards**.
So the Rotation Offset in Transform Customizations should remain **(0, 0, 0)**

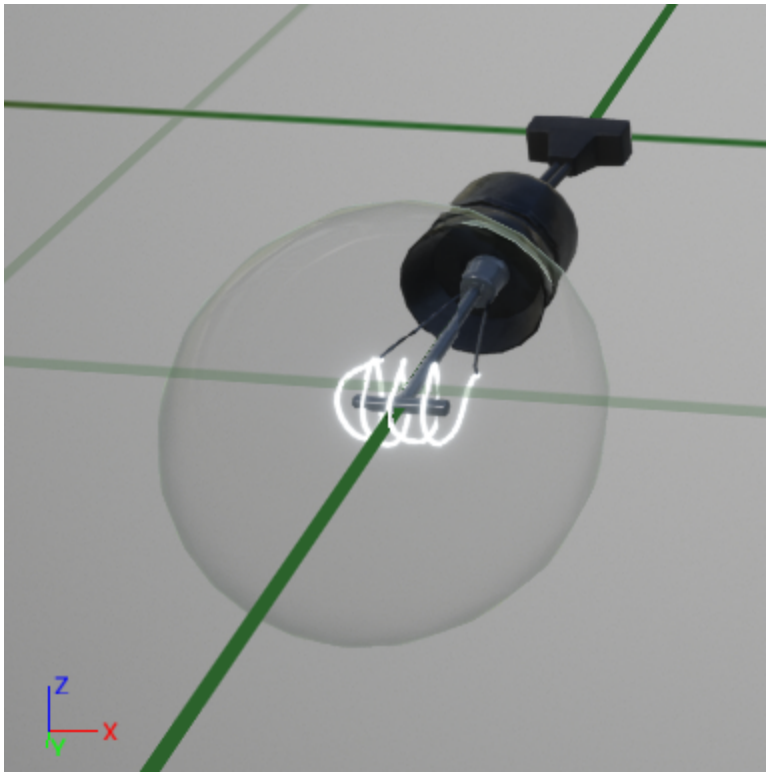
If your light bulb is pointing up (Z axis)



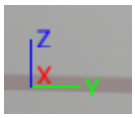
Find a rotation offset that leads the **Z axis** to **coincide** with the **non-rotated X axis**.
So the Rotation Offset in Transform Customizations should be **(0, -90, 0)**.



If your light bulb is pointing right (Y axis)



Find a rotation offset that leads the **Y axis** to **coincide** with the **non-rotated X axis**.
So the Rotation Offset in Transform Customizations should be **(0, 0, -90)**.



Showcase Content

Showcase Content (only **Maps** and **DemoRoom** folders) are removed from the Plugin Content directory. They can be accessed from the link provided on our product's main description called **Showcase Project**.

Plugin Folder Name

You can ignore this section if this is your first time installing this plugin.

Plugin Folder is renamed from

[EngineDirectory]/Engine/Plugins/Marketplace/StringLightPack

To

[EngineDirectory]/Engine/Plugins/Marketplace/StringLightPlugin

Please follow these steps first:

6. **If you have modified the original plugin Content, make a backup of StringLightPack Content (we don't recommend modifying the original Content as they can be overwritten with newer updates.)**
7. **Uninstall the plugin.**
8. **Install the new plugin version 1.2**
9. **If you have modified the original plugin Content, copy the StringLightPack Content backup to StringLightPlugin.**
10. Please don't copy the **Maps** and **DemoRoom** folder.

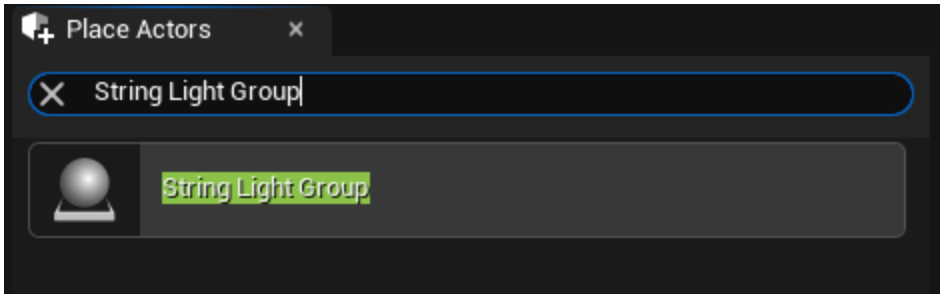
New Features

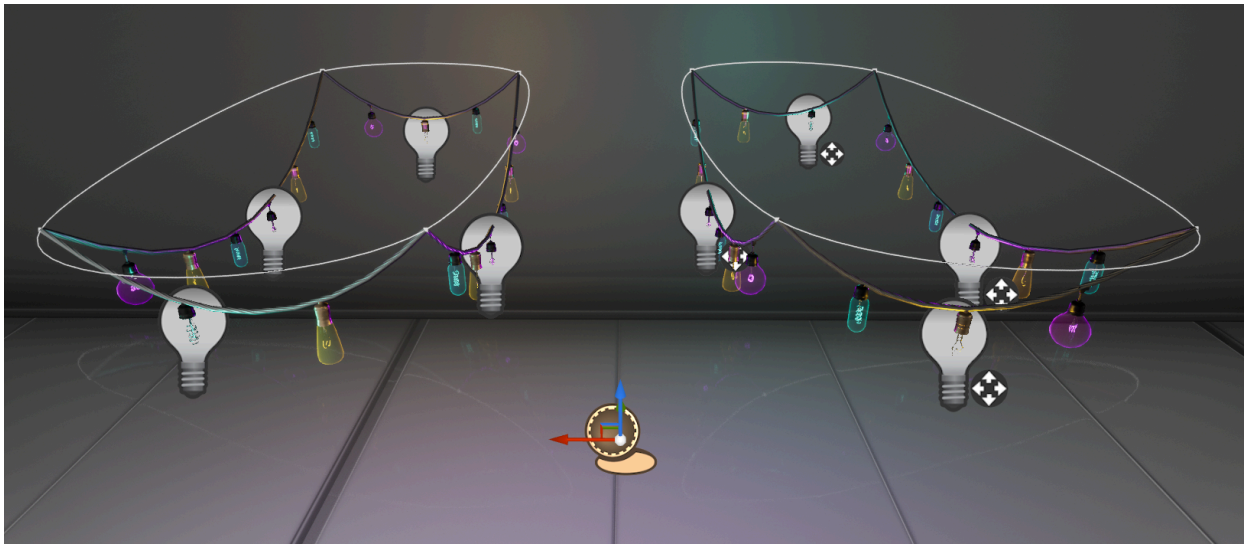
String Light Group

You can improve the performance of String Lights by sharing HISM components between multiple String Lights.

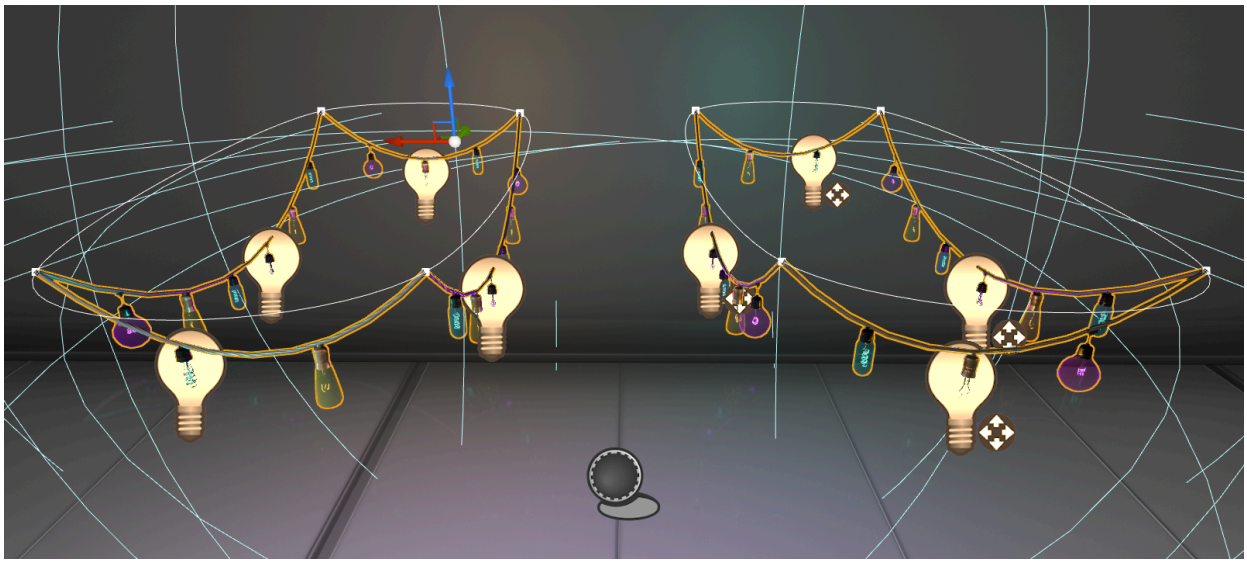
- String Light Groups can greatly **improve rendering performance**.
- String Light Groups will automatically group similar mesh instances together in one HISMC.
- String Lights and the String Light Group must be on the **same level**.
- There can be **more than one** group in the scene.

In order to do this, place an instance of the **String Light Group** actor in the scene.

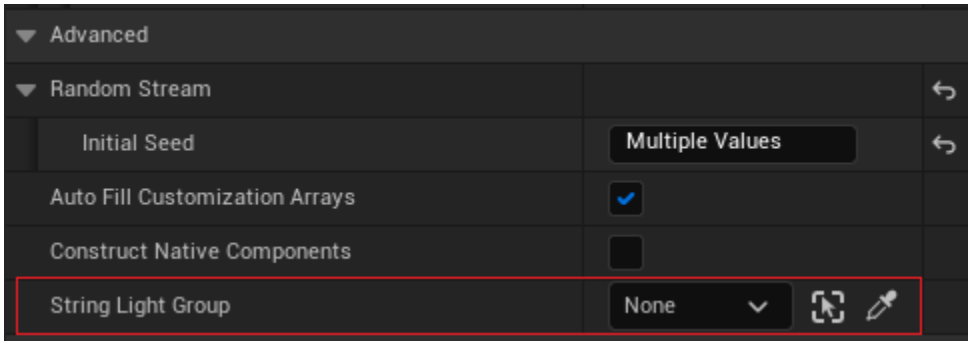




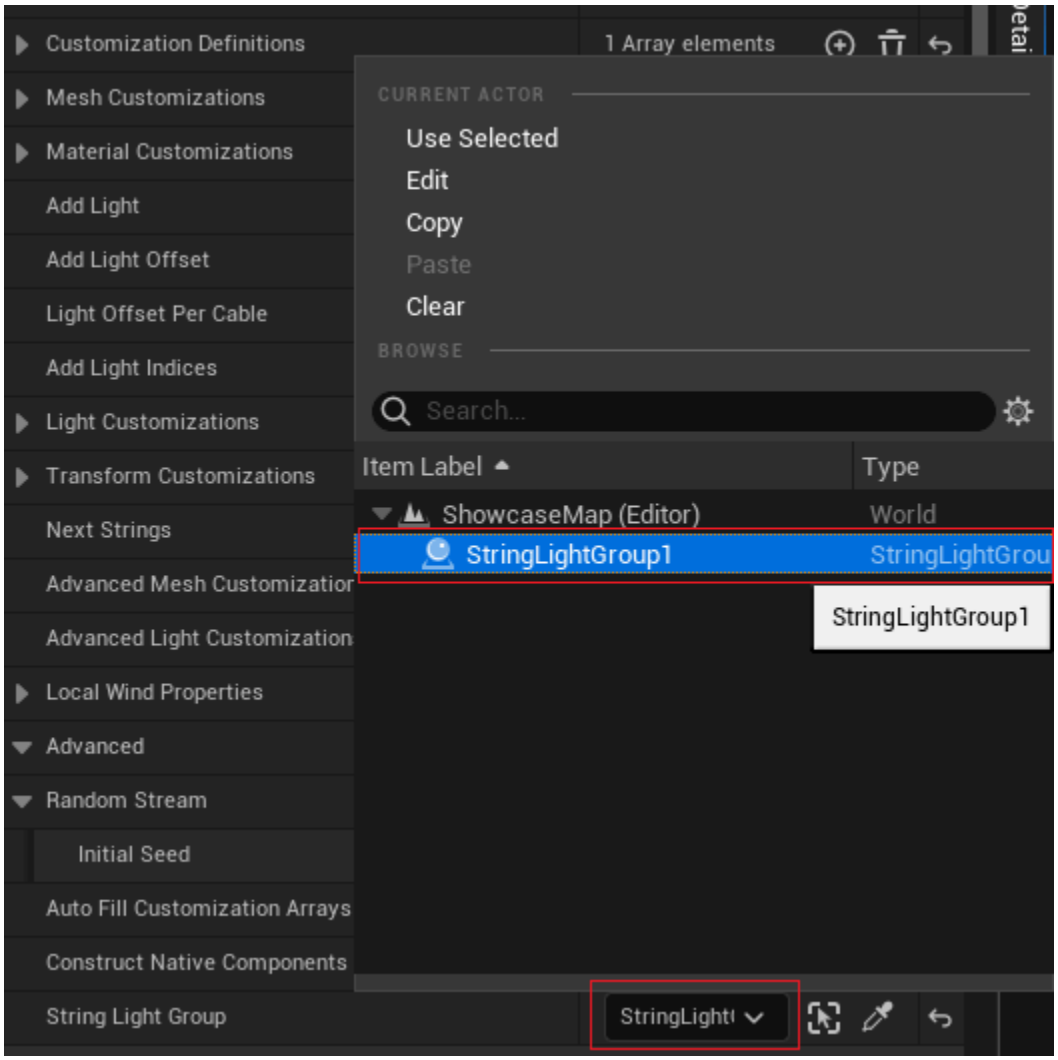
Next, select String Lights for which you want to group their Instanced Meshes together.



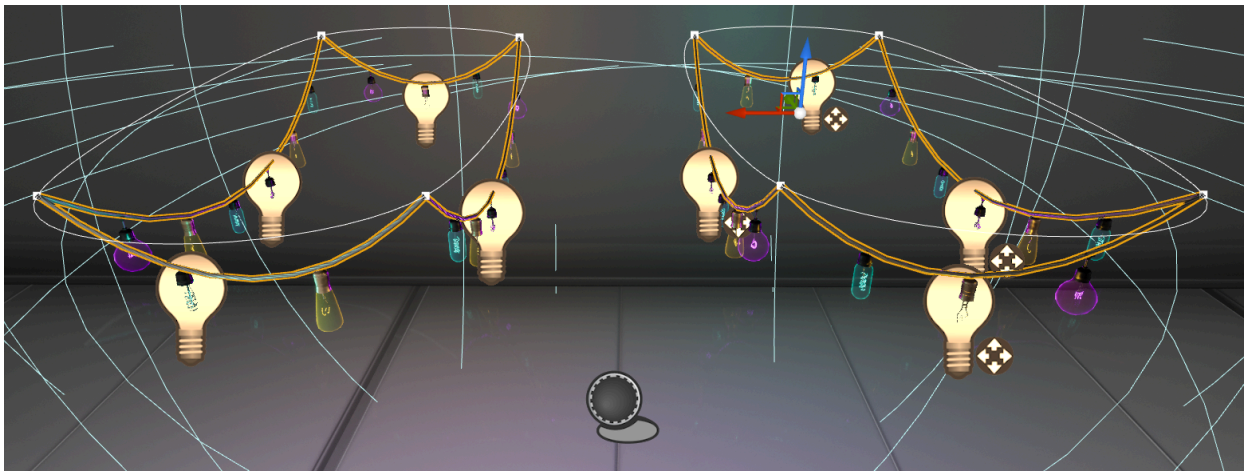
Locate the **String Light Group** setting in the **Advanced Section** at the bottom of the **Setup** category from the Details Panel.



Finally, select the String Light Group actor.

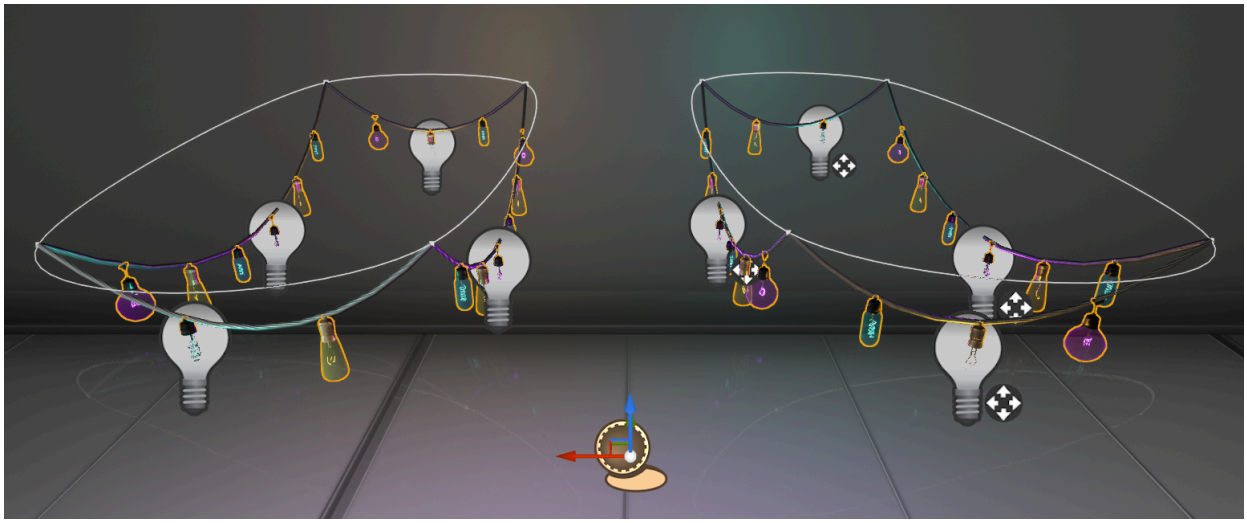


Mesh Instances will be automatically moved into the **String Light Group** actor. As you can see Instances are no longer highlighted with selected String Lights because they are part of the String Light Group.

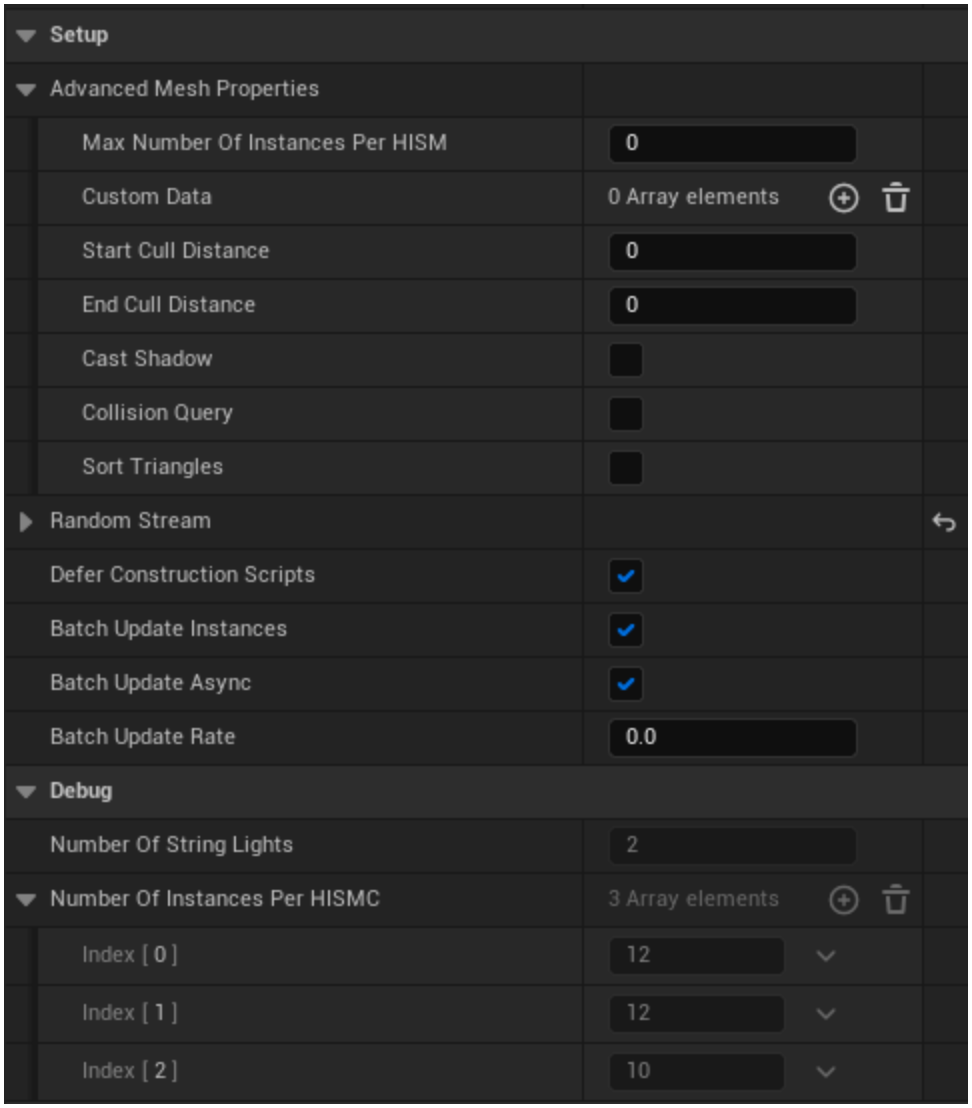


You can select the String Light Group by clicking on the billboard or any related instanced mesh. Notice that Mesh Instances live on the String Light Group and are attached to different String Lights.

You can move the String Light Group billboard anywhere that you like.



Here is the overview of String Light Group settings. All properties are provided with tooltips.



Advanced Mesh Properties: This works just like the Advanced Mesh Properties on the String Lights, except that this setting is shared for all grouped Instances.

Random Stream: This is used to feed mesh Instances with random values which can be accessed from the material.

Defer Construction Scripts: Improves Construction Script Performance.
Whenever a String Light actor changes, Construction Scripts for the String Light Group actor should rerun. This option will defer this action onto the next frame to avoid multiple rerun's when editing multiple String Lights at a time.

Batch Update Instances: by default, String Light Components handle updating the transform of the attached Mesh Instances. When this option is enabled, String Light Group will handle

updating of mesh instances instead. This will avoid rebuilding HISM trees multiple times per frame.

In practice, performance improvement from this option is negligible, but allows for following optimizations.

Batch Update Async: Should be enabled for games, but if there exist thousands of instances in this group, disabling this can fix some frame drop issues **in the editor**. (When moving the mouse in the editor over instances, frame rate can drop, disabling this option fixes that problem, but In game this problem does not exist and you can always use async update. Let us know if you experience otherwise.)

Batch Update Rate: Controls the update rate of instances. Note that instances can not be updated more than once per frame, and they are not updated when it's not needed.

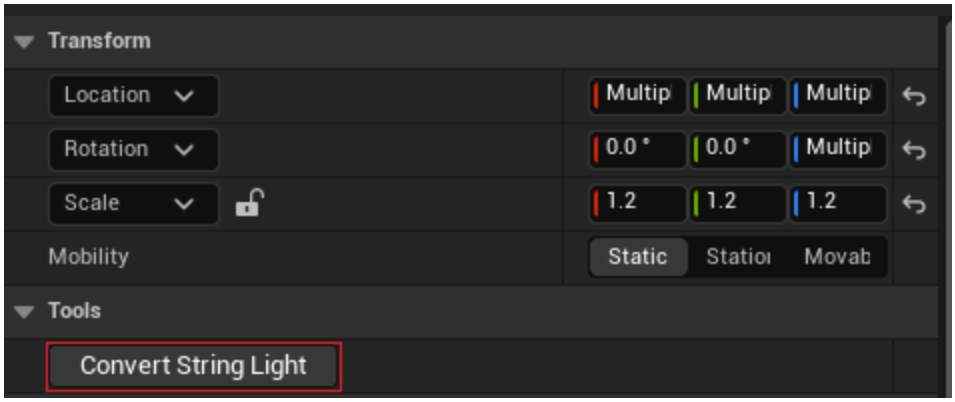
Number Of String Lights: This only exists in the editor and shows the number of String Lights that are using this group.

Number Of Instances Per HISMC: This only exists in the editor. Number of elements of the array shows how many groups (HISM Components) are created. Value of each element shows the number of Instances in that group. In this example, there are 3 groups created for String Lights. Each group contains 10-12 instanced meshes.

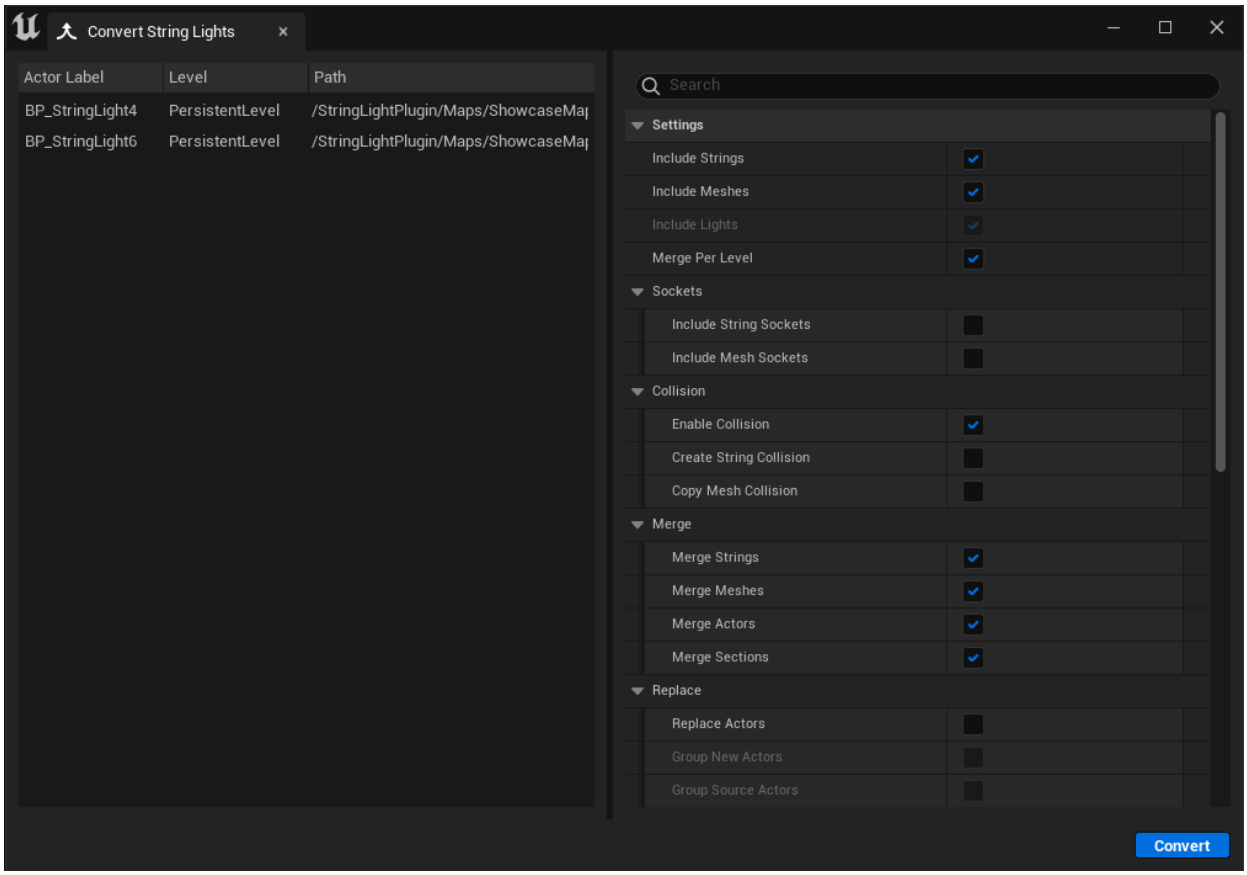
Merge String Lights

You can now convert String Lights into Static Meshes and replace them to completely remove the cost of String Lights. Optionally you can keep source String Light actors hidden and editor only which can be restored later.

You can bring up the convert window by clicking on the **Convert String Light** button



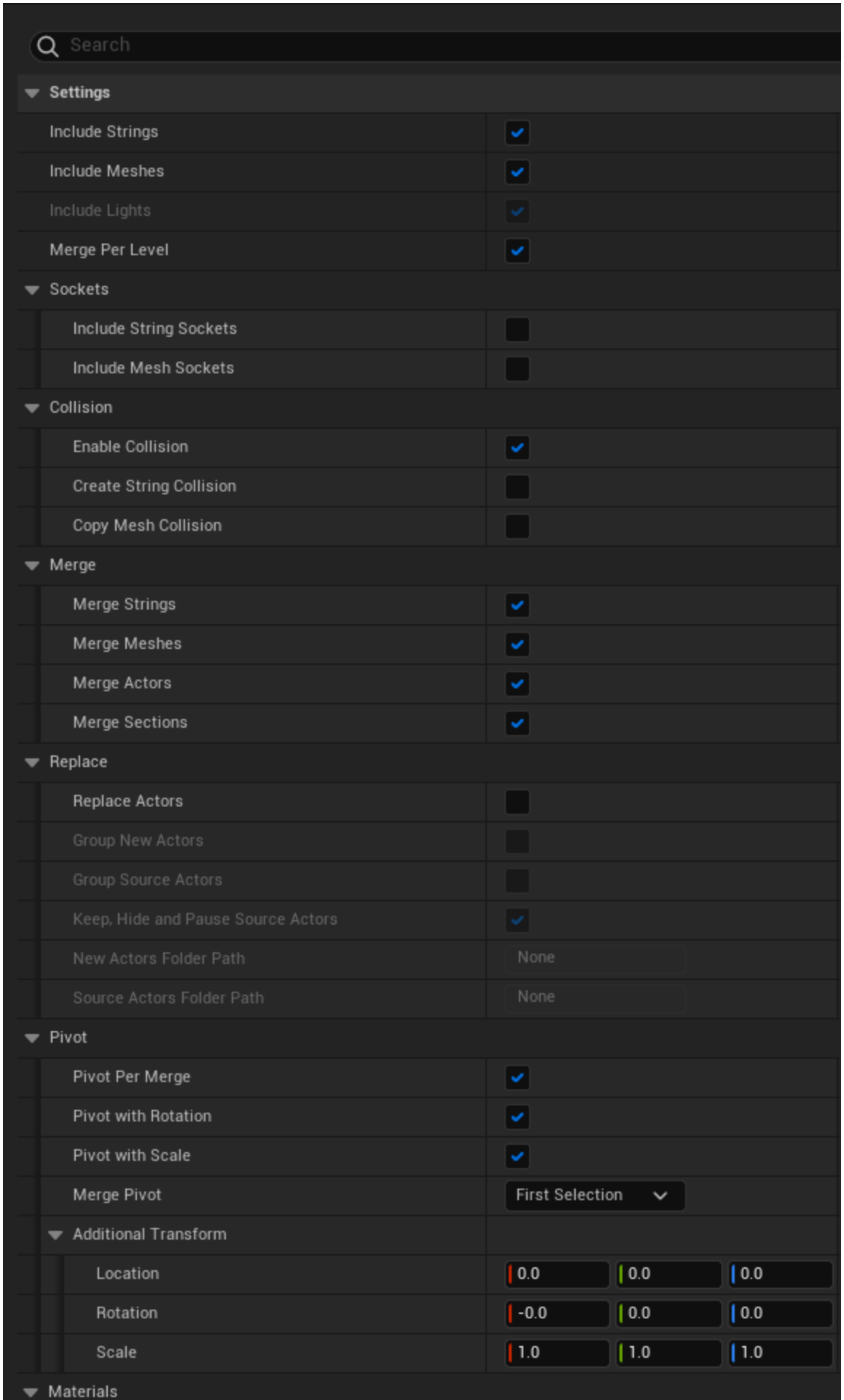
On the left you have the list of selected string lights in the **selected order**.
On the right you have the convert settings. All the settings are provided with tooltips.



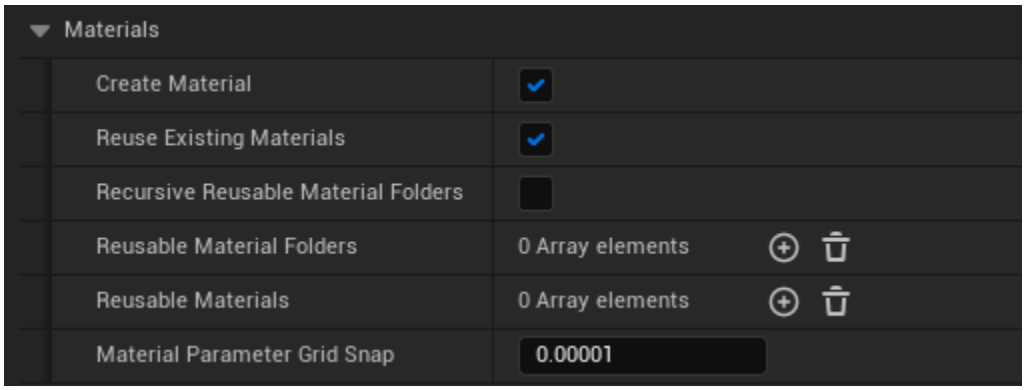
By clicking on the Convert button, you will get to choose a path and base name for assets that will be generated. Asset Names might change based on the kind of asset and number of assets being generated (names will still include the given base name). All generated assets will have a unique name and can not overwrite other assets.

*Converting String Lights while Play in Editor is active is not tested and might cause problems or crashes.

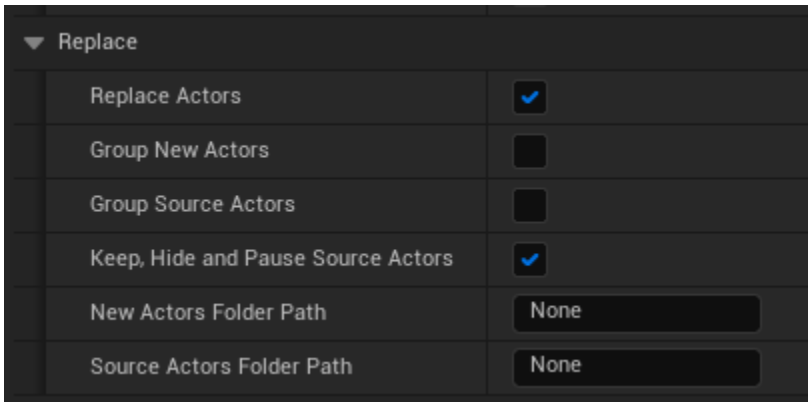
Here is the list of available settings.


Search
Settings
Include Strings
Include Meshes
Include Lights
Merge Per Level
Sockets
Include String Sockets
Include Mesh Sockets
Collision
Enable Collision
Create String Collision
Copy Mesh Collision
Merge
Merge Strings
Merge Meshes
Merge Actors
Merge Sections
Replace
Replace Actors
Group New Actors
Group Source Actors
Keep, Hide and Pause Source Actors
New Actors Folder Path
None
Source Actors Folder Path
None
Pivot
Pivot Per Merge
Pivot with Rotation
Pivot with Scale
Merge Pivot
First Selection
Additional Transform
Location
0.0
0.0
0.0
Rotation
-0.0
0.0
0.0
Scale
1.0
1.0
1.0
Materials

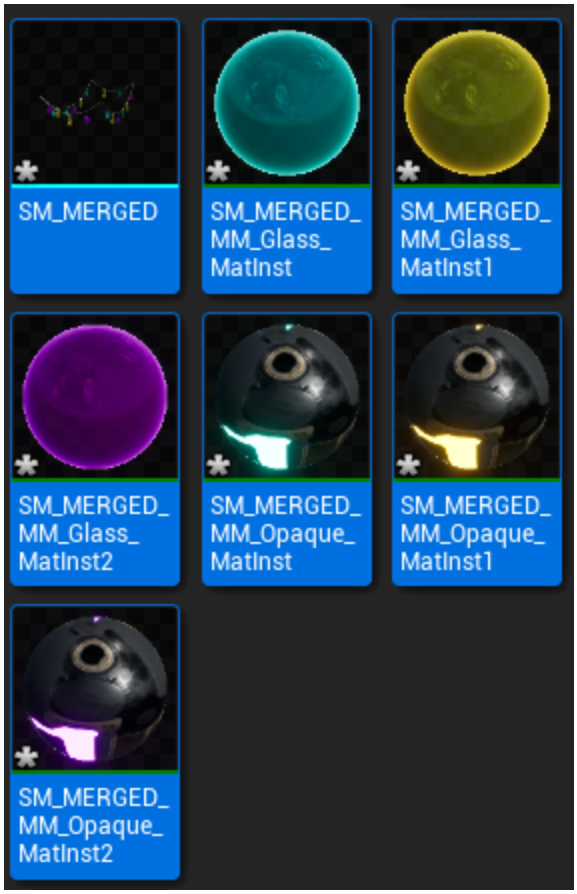
You can enable Create Material to generate non-dynamic material instance assets from dynamic material instances.



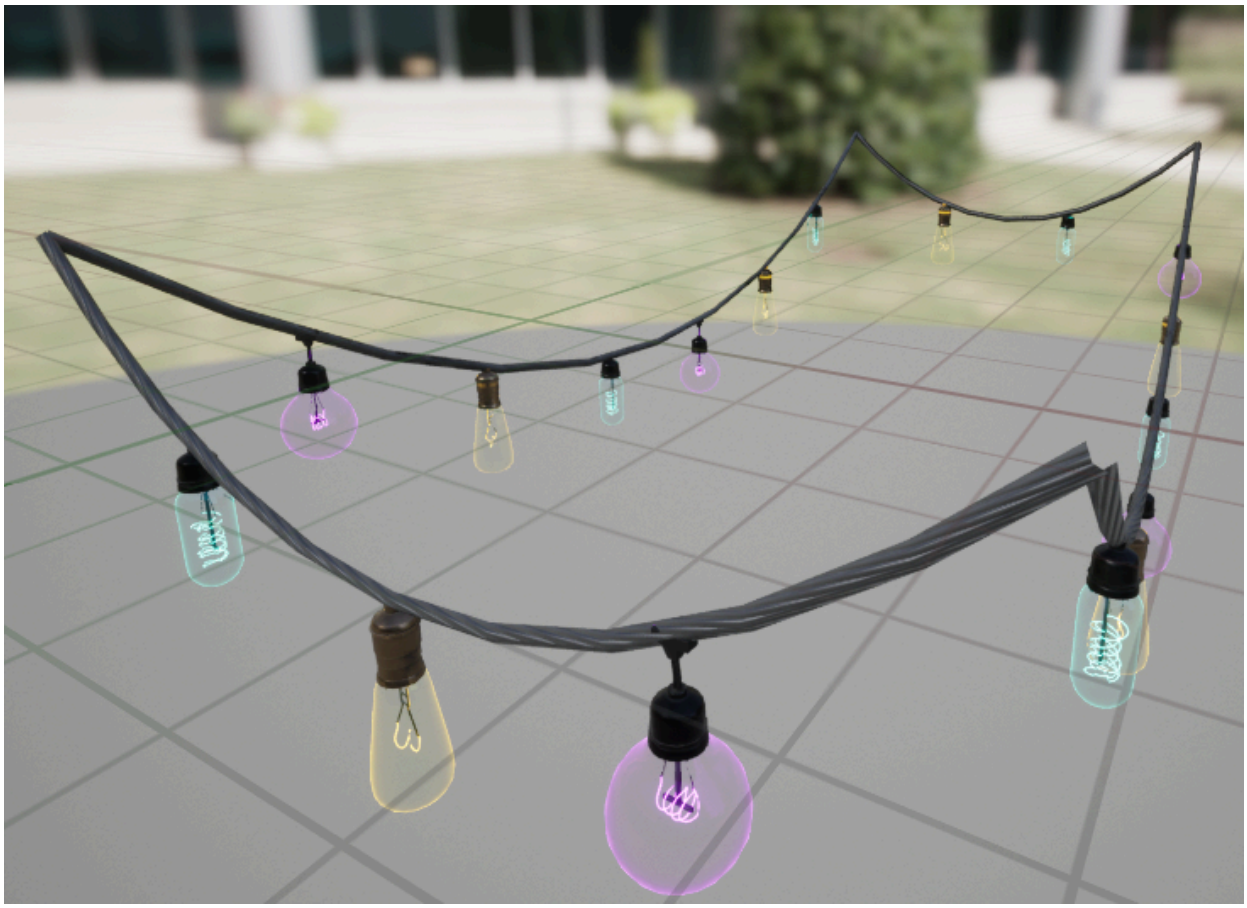
You can optionally Replace actors and Keep source String Lights for non destructive workflow.



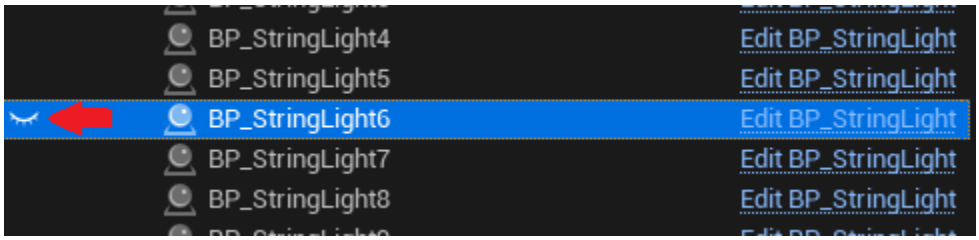
After selecting a path to generate assets, and replacing, we have.



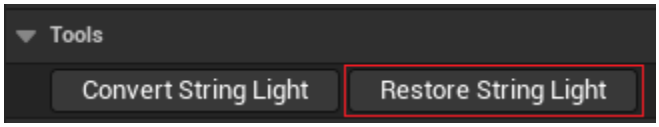
These assets work just like other Static Meshes and Material Instances.



If you choose to Keep, Hide and Pause Source Actors, after replacing you will find the actor is hidden. Note that this actor is Editor Only and will not be included in your game package. Also the simulation is paused which means this actor has no performance cost.

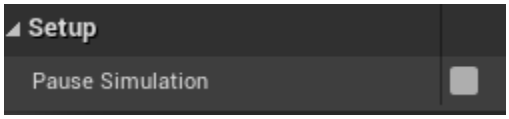


By clicking on the hidden actor, You can restore it back to the scene by clicking on the Restore button (this button is only visible for converted string lights)



After clicking on the button, String Light Actor will be restored back and work just like other String Lights. Restore button should disappear.

The simulation remains paused and you can resume simulation manually from the Setup category.

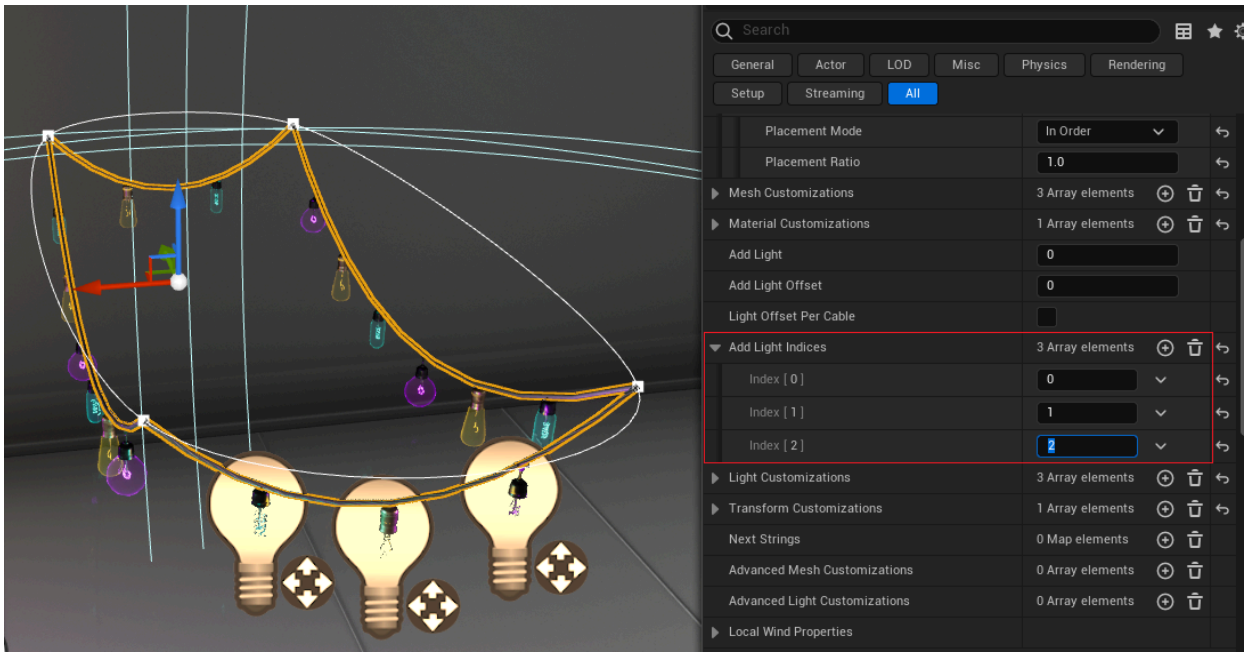


Add Light Indices

You can now specify exact indices for lights to attach to the string. This option overrides the Add Light option.

Add Light Offset can shift these indices. This can now be negative to shift lights by any amount.

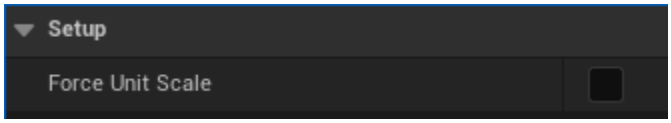
If Light Offset Per Cable is true, lights will be added for each cable.



Force Unit Scale

If you enable this option, all components except the spline component will have a unit scale of 1. This is useful for scaling the Spline component without affecting the scale of other components.

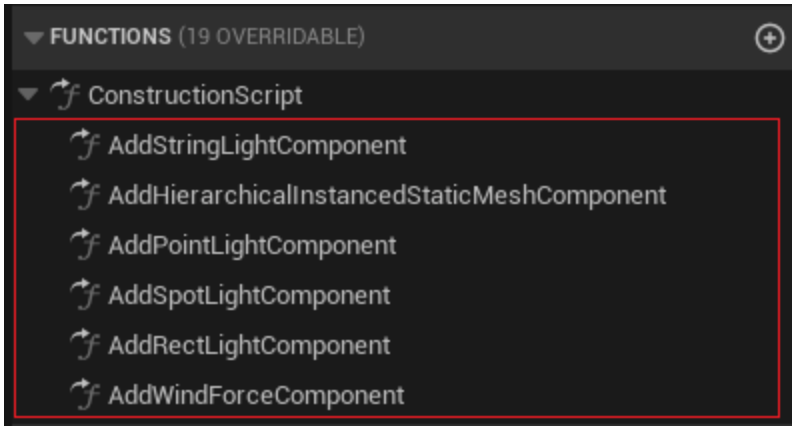
For backwards compatibility, this is disabled by default, but we recommend you to enable it.



Native Components

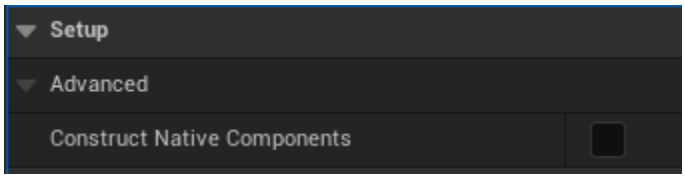
Previously because of some issues with Undo/Redo operations in the editor, we decided to add components via blueprint.

BP_StringLight has overridden these functions:



But the issue with Undo/Redo has been fixed. If you enable the new option Construct Native Components, These blueprint functions will no longer be called and all components will be constructed natively which is slightly faster to load.

Note that native components constructed by String Light will be tagged with "__native_" tag for some special handling.

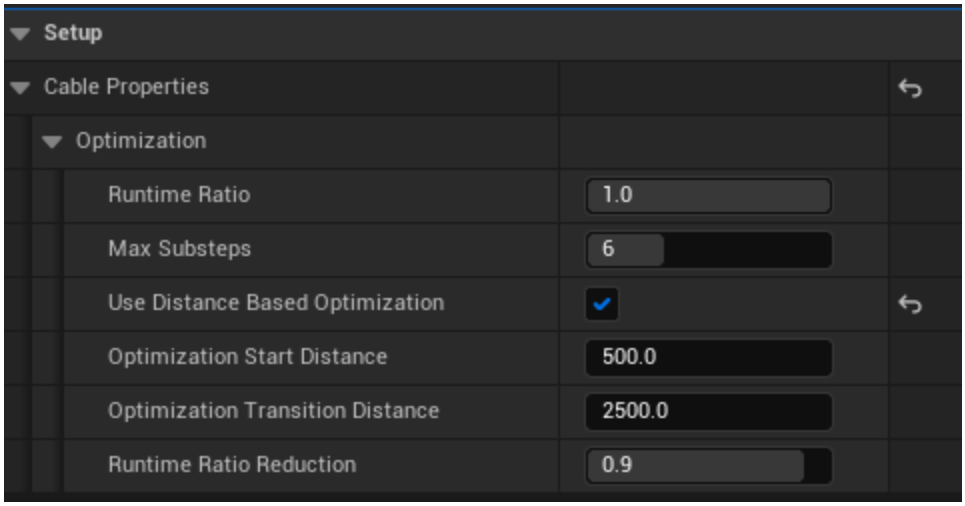


Don't use the "__native_" tag if you want to add custom components to the String Light when this option is enabled.

Runtime and Distance Based Optimization

You can now specify how much resource the Simulation can use by specifying the Runtime Ratio. Unlike substep time, this will not affect the stiffness of the String Lights.

You can also reduce runtime ratio based on the nearest player view distance to the String Light.



Runtime Ratio: Lower values can reduce the simulation cost. value of 0 will halt the simulation (does not disable the component tick). default is 1.

Simulation will wait for (SubstepTime*RuntimeRatio) before updating, but still uses the accumulated SubstepTime for updating the simulation. Unlike SubstepTime, this does not affect the string stiffness.

Max Substeps: The maximum number of substeps allowed per frame for the simulation. This can prevent expensive ticks when the simulation delta time is large (for example when the frame rate or RuntimeRatio is very low).

Lower limits can cause the simulation to run at a slower rate therefore reducing the simulation cost. This has no effect if the limit is not reached. The default value is 6.

Use Distance Based Optimization: reduce Runtime Ratio based on nearest camera distance to the String Light Component.

Optimization Start Distance: Specify at which distance, Runtime Ratio Reduction should occur. (in centimeters)

Optimization Transition Distance: Specify the length of transition between no reduction and maximum reduction in Runtime Ratio. (in centimeters)

If 0, the transition will be immediate. Otherwise linearly interpolates.

Runtime Ratio Reduction: Specify the amount of reduction in Runtime Ratio when the String Light is at farthest distance from the nearest camera.

Minimum ratio will be (RuntimeRatio - RuntimeRadioReduction) but it can not be less than 0. (clamped)

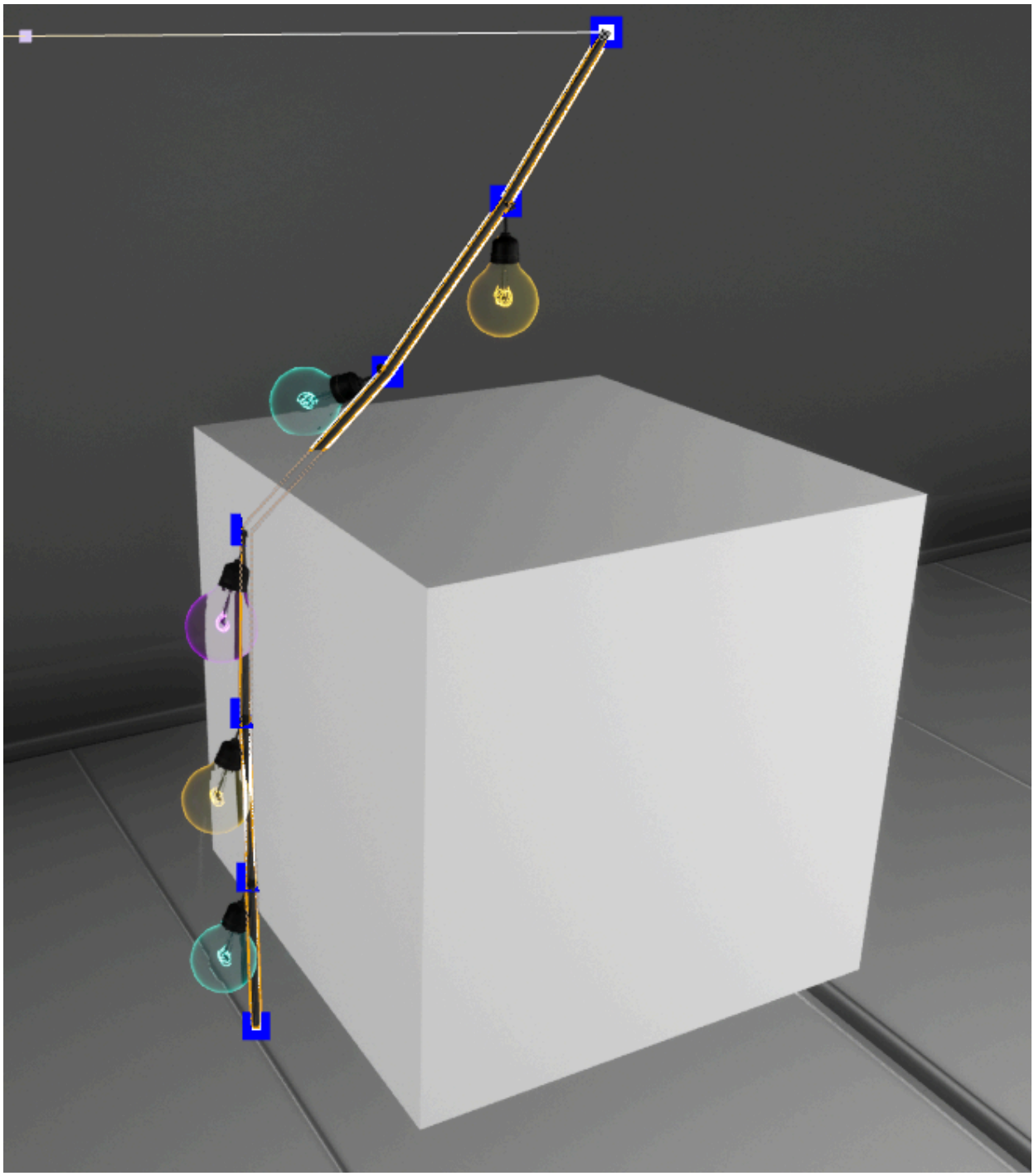
Collision

Segment Collision: Reduces gaps in the collision detection but is much more costly.

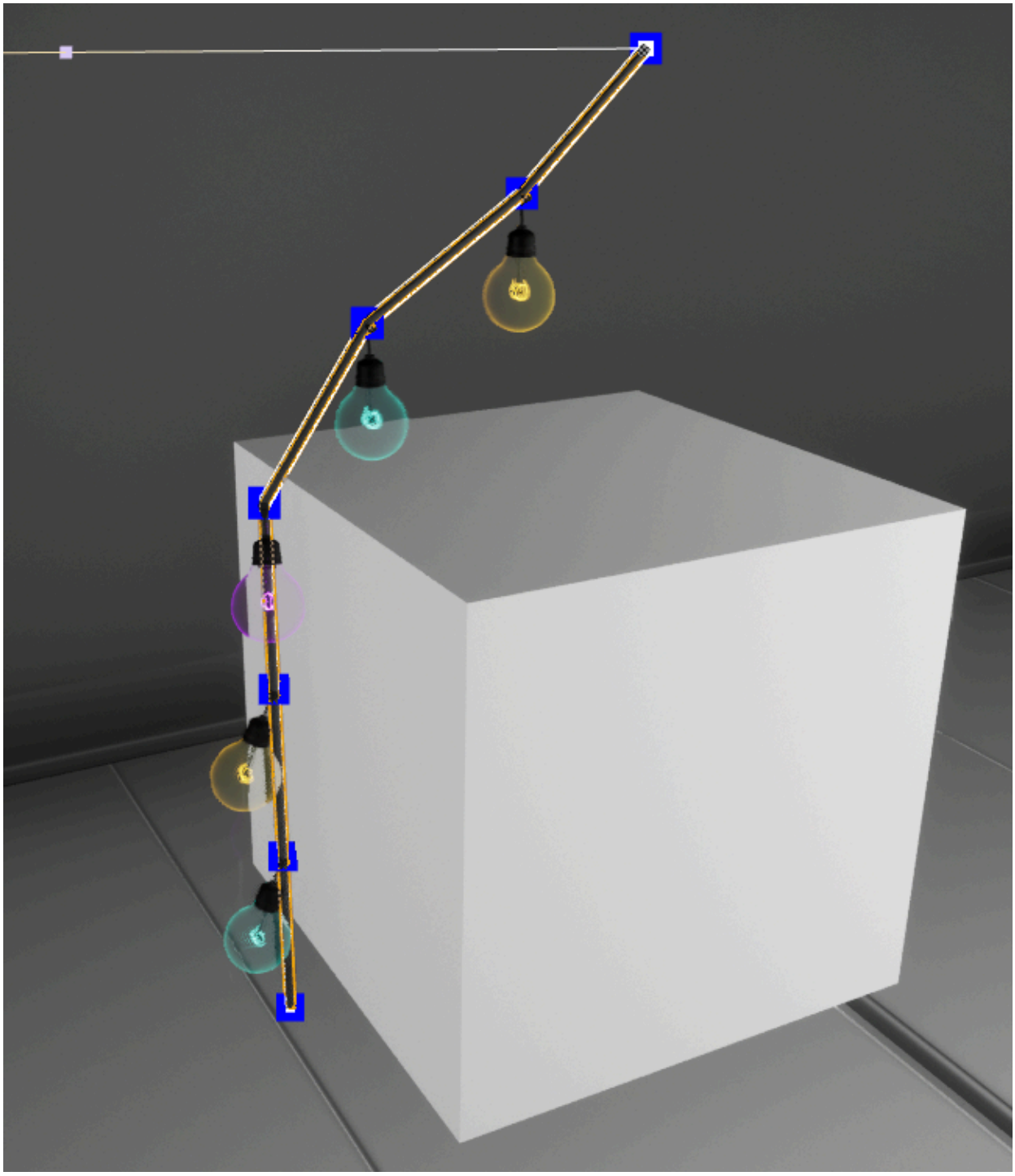
Performs additional capsule sweeps for string segments. This will be used along with sphere sweeps because capsule sweeps alone can not resolve particle locations.

The logic behind Segment Collision is not perfect and in some cases this might not give stable results.
If you encounter stability issues, try to use more segments or disable this option.

Not Enabled: Blue points are string particles. Only particles collide with the box.



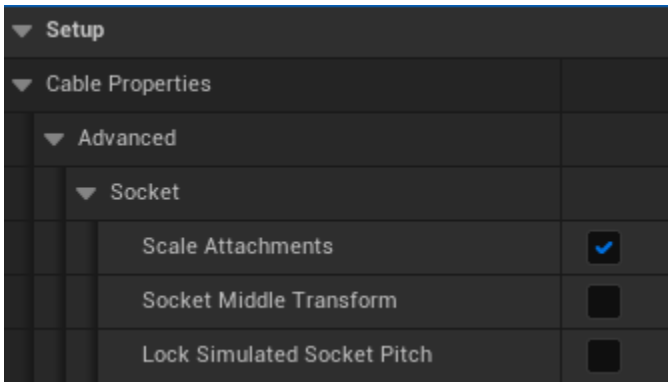
Enabled: Segments also collide with the box.



Hanging Light Collision: you now have the option to disable world collision for hanging lights to save some performance. Note that this does not disable Self Collision.

Socket Options

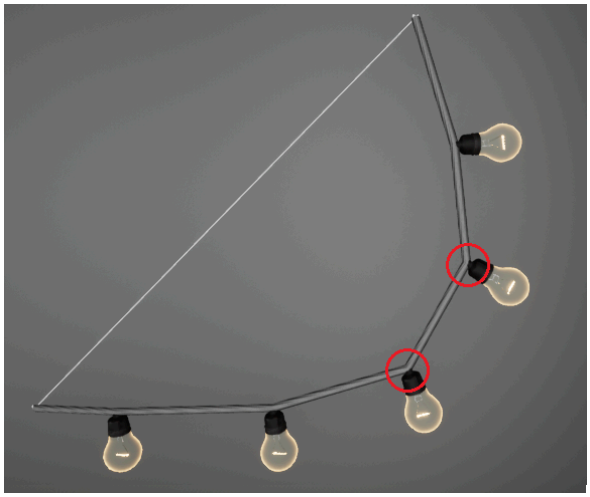
There are new options for sockets which affect all attached components and light bulbs to String Lights.



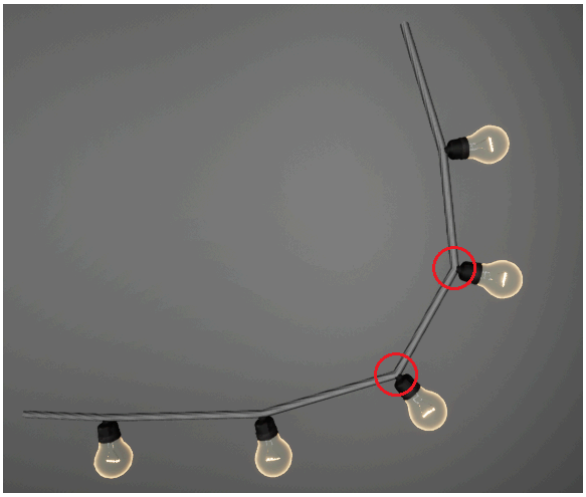
Scale Attachments: if true, sockets and attached mesh instances will be scaled by the string light component. call Mark Render State Dirty after changing this option from the blueprint.

Socket Middle Transform: if true, sockets on string particles get their transform from previous and next segments, otherwise only one segment. call Mark Render State Dirty after changing this option from the blueprint.

Not Enabled:

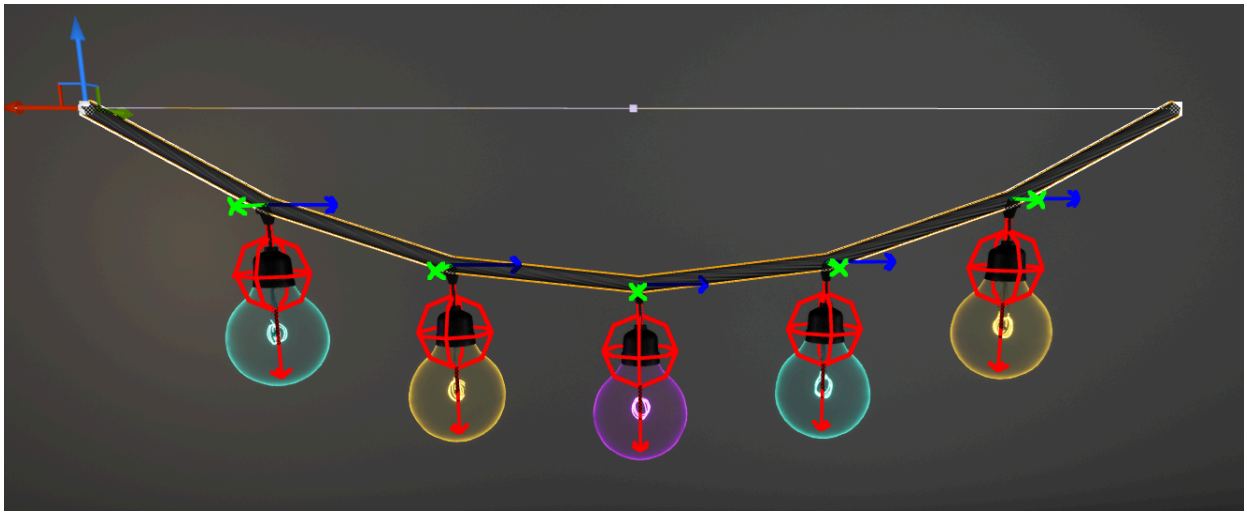


Enabled:

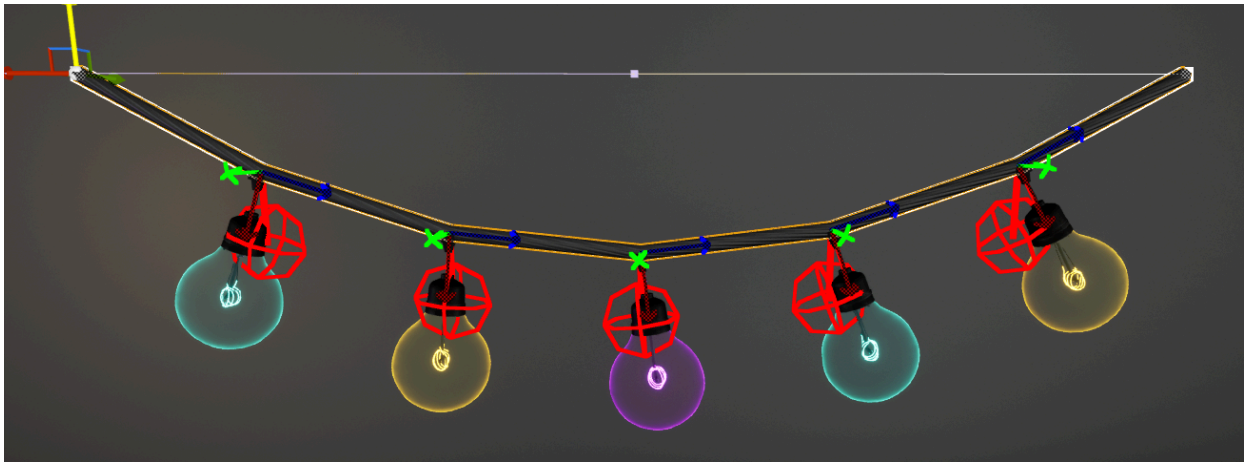


Lock Simulated Socket Pitch: whether to lock pitch of simulated hanging light sockets or not. This does not affect performance. call Mark Render State Dirty after changing this option.

Not Enabled:

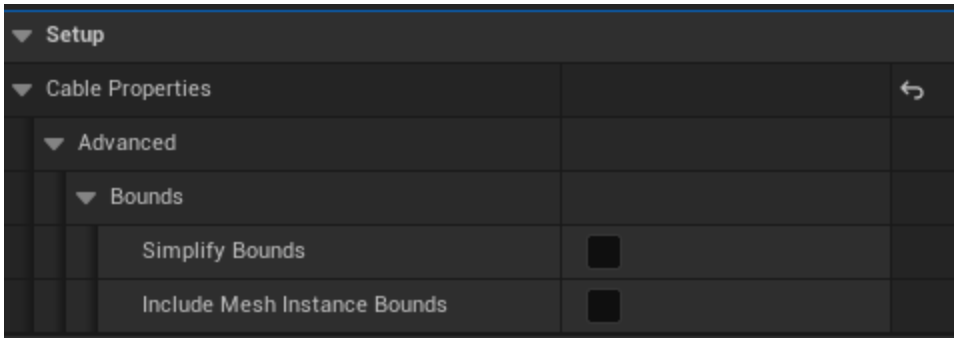


Enabled:
Hanging lights are still simulated but these light bulbs can only rotate around their Z axis.



Bounding Box Options

There are new options for the Bounding Box of String Lights.



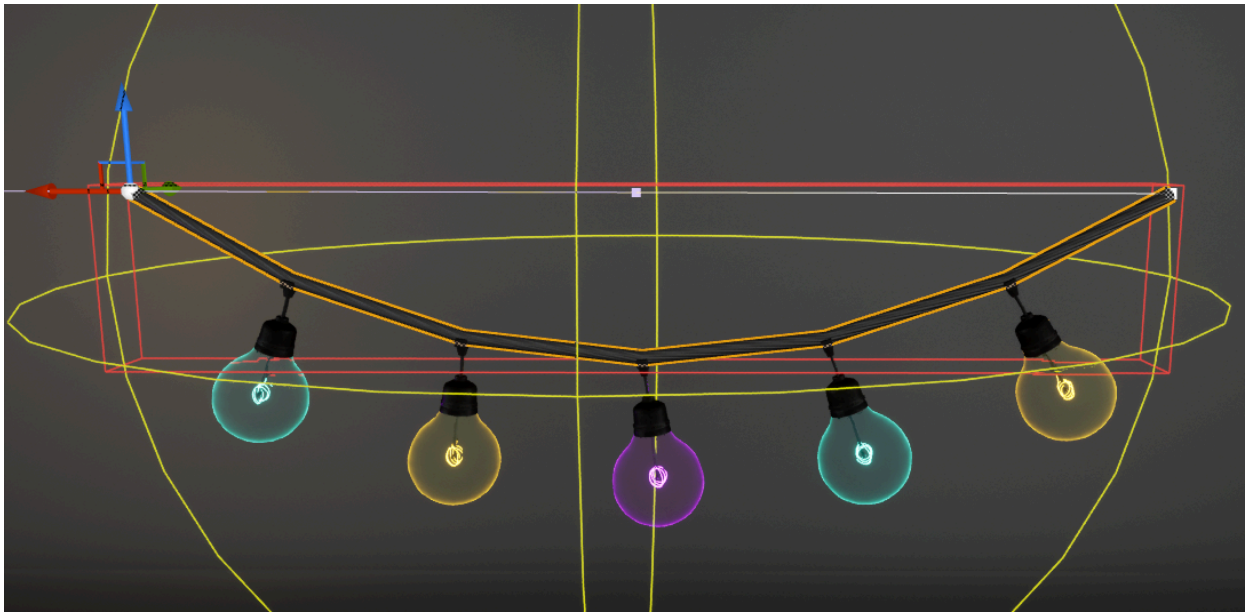
Simplify Bounds: Uses faster bounds calculation by only considering start, middle and end points of the cable therefore calculated bounds may not be accurate. This may affect shadow casting as well. only viable when the number of particles is high.

Include Mesh Instance Bounds: Extend the bounds of the string by the bounds of attached mesh instances.
This is useful in situations where the string is not visible but hanging lights are visible and bSkipUpdateWhenNotVisible is true.

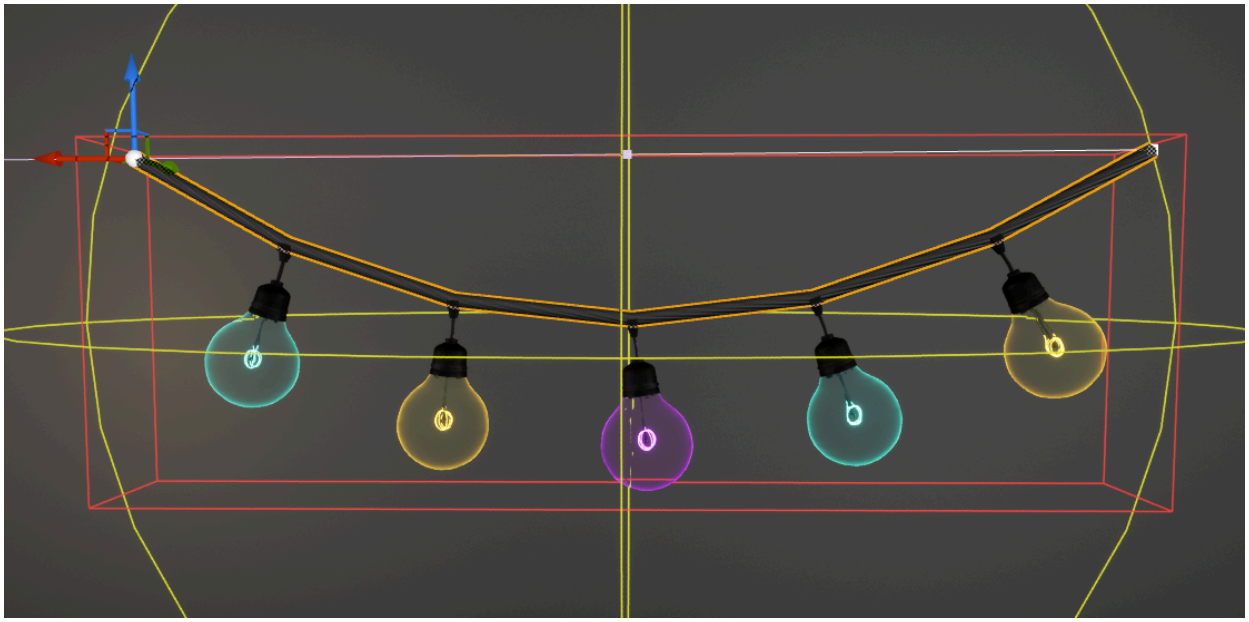
This can be left off if you don't run into this issue or other components are already attached to the string as the engine will already extend the bounds by the attached children.

if Simplify Bounds is true, only mesh instances attached to first, middle and last hanging particles are considered.

Not Enabled:

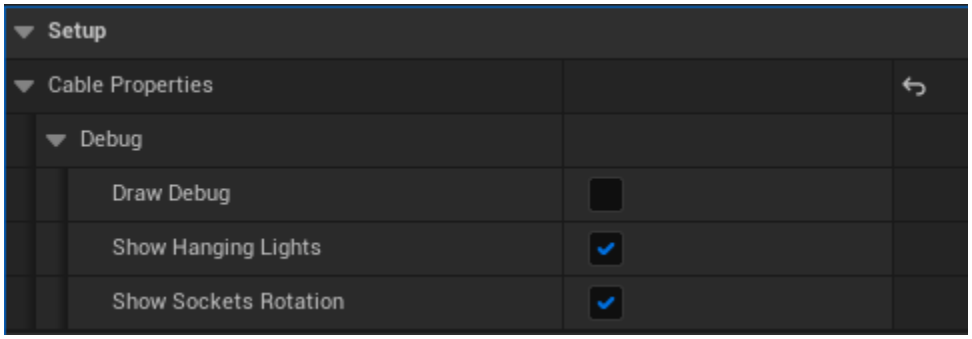


Enabled:



Draw Debug Options

There are new options for Debugging of String Lights.



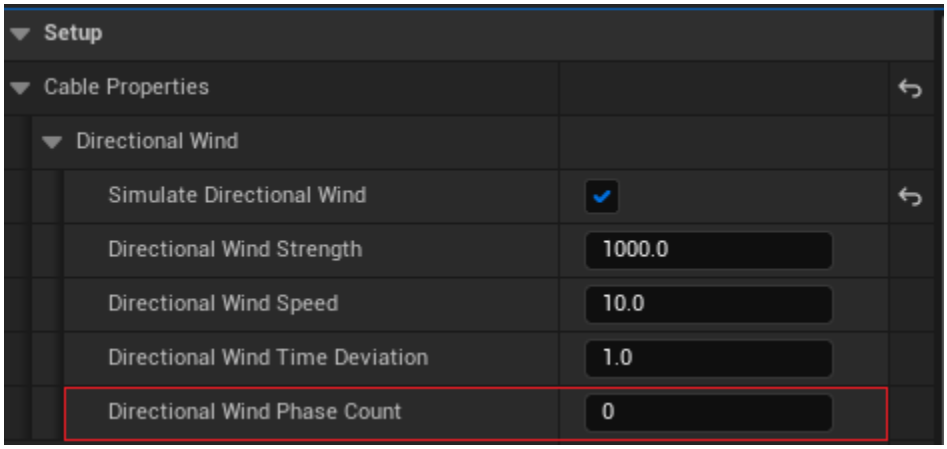
Show Hanging Lights: Whether to draw debug elements for Hanging Lights or String Particles.

Show Sockets Rotation: Whether to show rotation of sockets using arrows or not.

Directional Wind Phase Count

This new option specifies how many simulation updates it takes before recalculating directional wind forces. the value of 0 will update wind with every simulation update, value of 1 will update wind with every other simulation update and so on.

This can help reduce the cost if there are many strings and directional wind forces in the scene.



Per Particle Forces

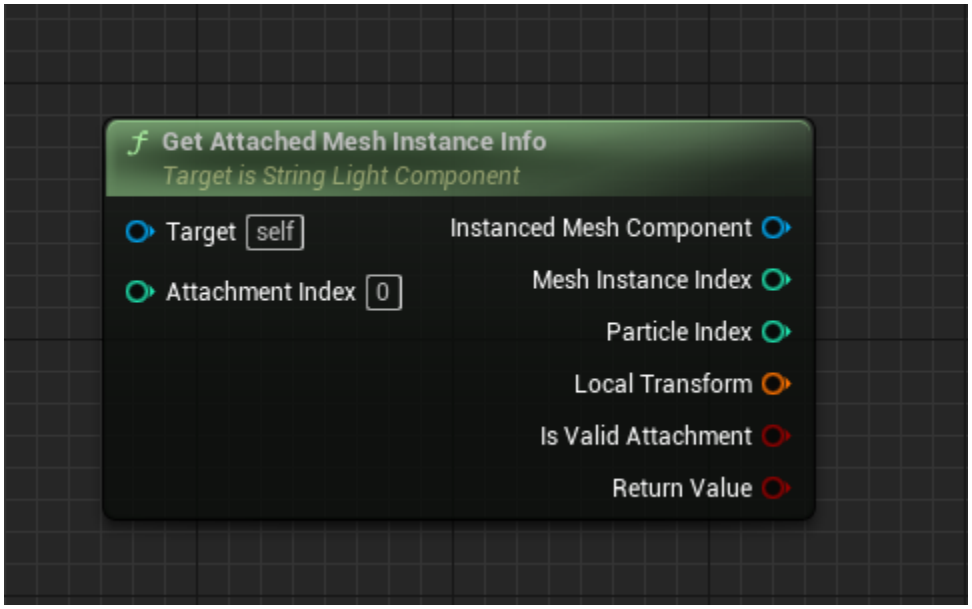
You can now add force to individual String Particles using this Add Force to Socket function. If Socket name is None, this will apply force to all particles.

Note that the force will be multiplied by substep delta time so you should not multiply the input force by delta time. All forces are zeroed every frame and need to be applied again.

Blueprint

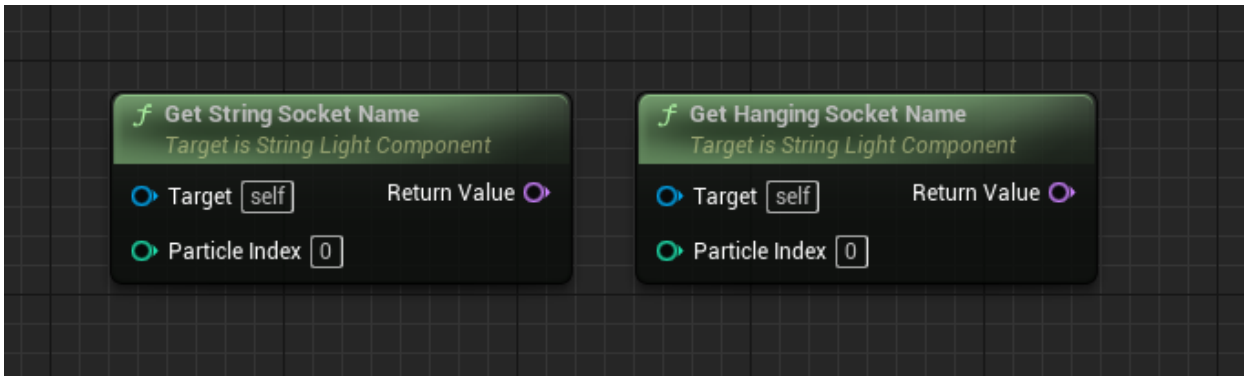
Some existing nodes are improved and there are new nodes for new functionalities.

You can now get all the information from attached mesh instances such as indices and transform.

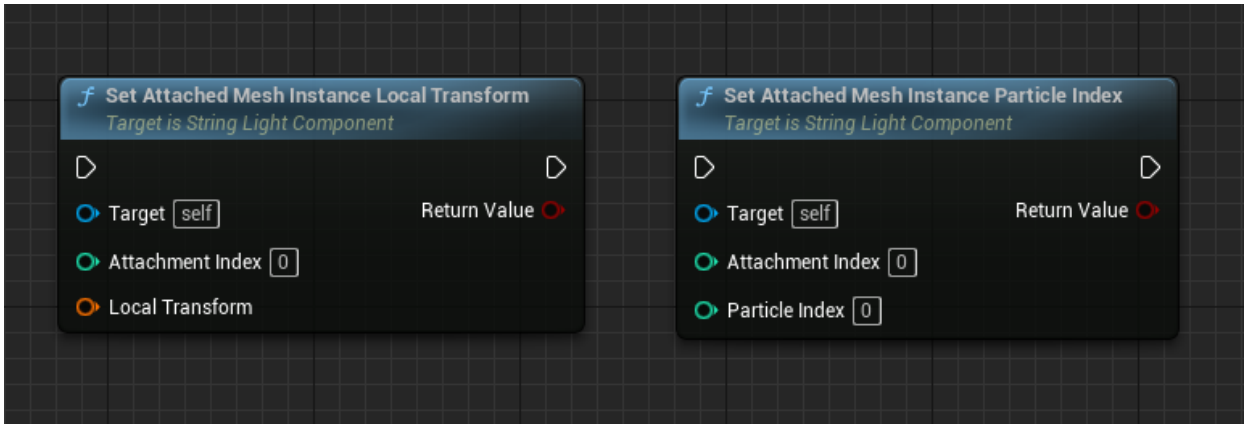


You can now get the name of string or hanging particle sockets using these functions Instead of manually appending “SP_” or “HP_” with the particle index.

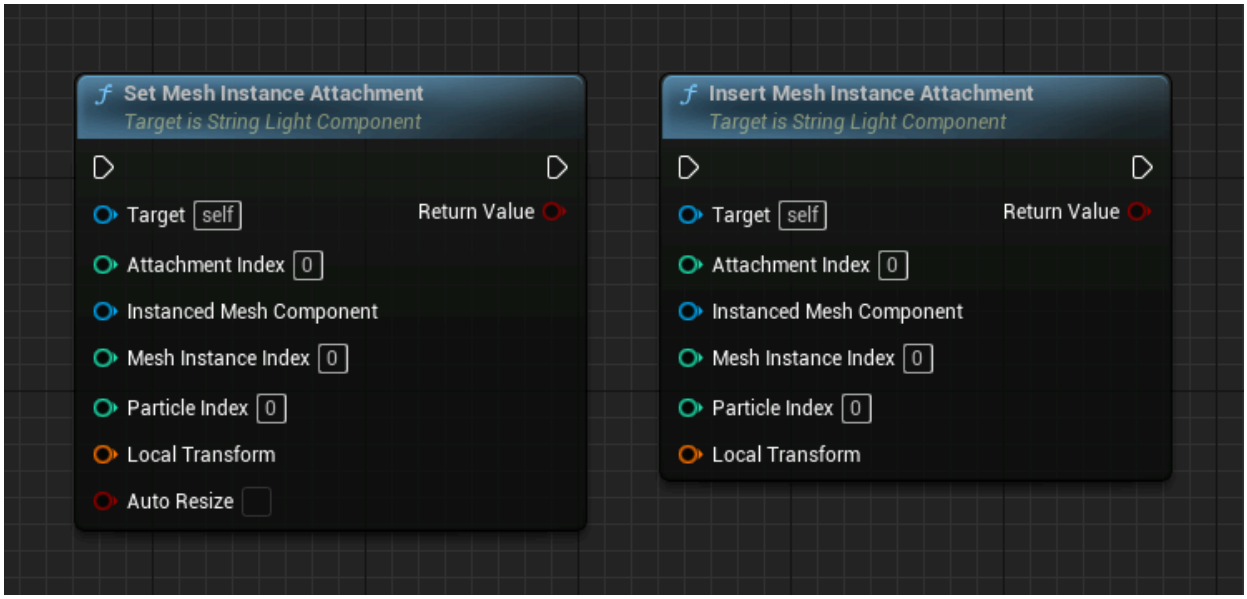
Note that this returns None if particle index is out of range.



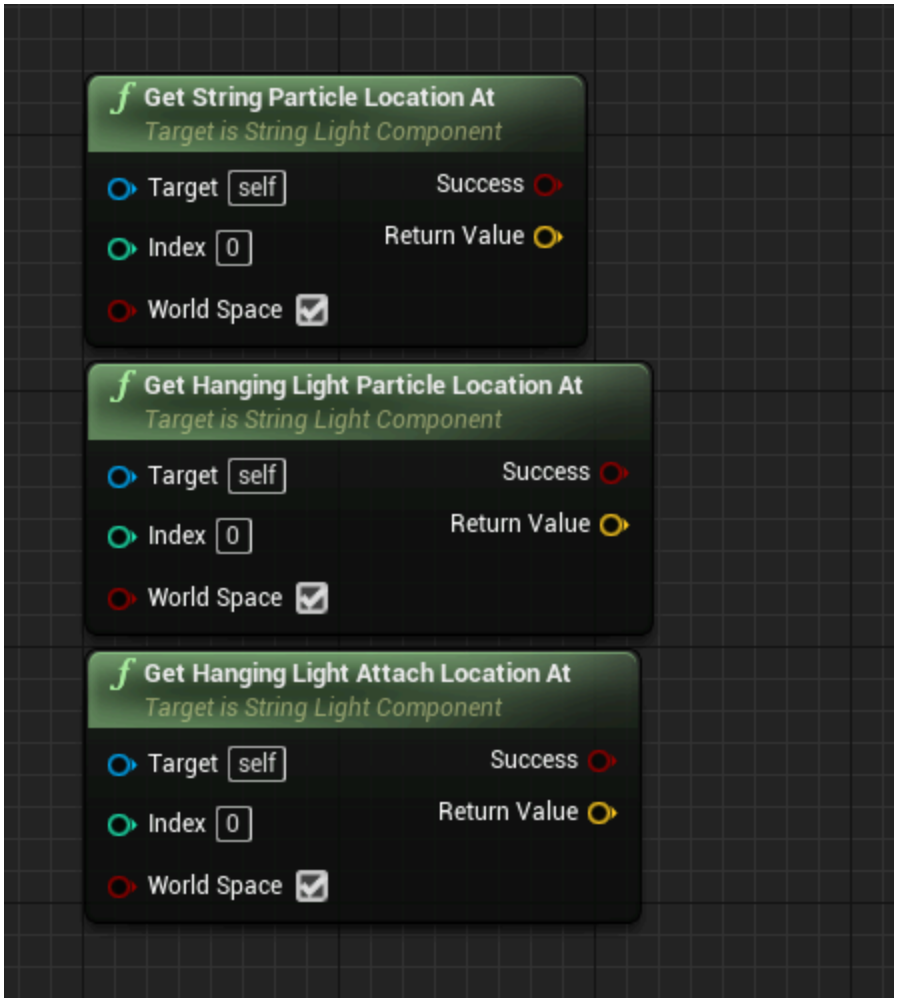
You can directly change Local Transform or Particle Index of attached Mesh Instance.



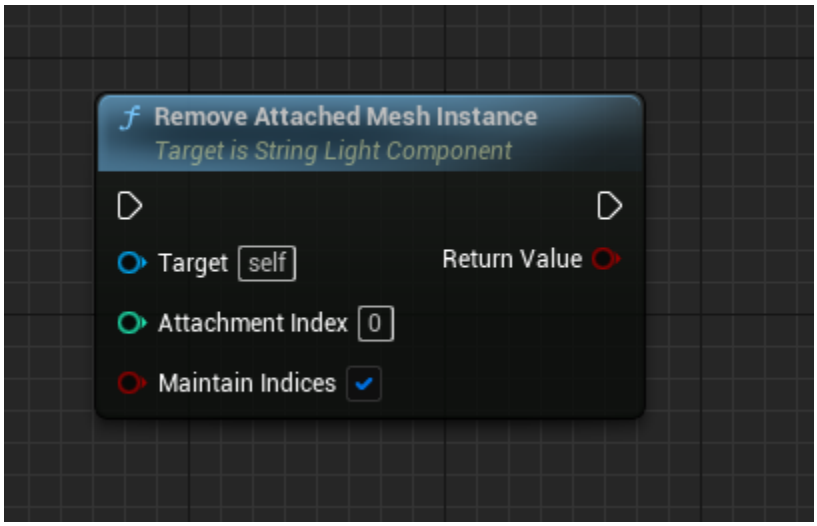
You can now Set or Insert attached Mesh Instance.



These functions return a Success boolean to indicate that the result is valid.



This function now returns a boolean to indicate that the remove operation was successful.



Stats

More Stats have been added.

- Directional Wind Simulation
- Batch Update Mesh Instance Transforms
- Distance Based Optimization
- Calculate String Bounds
- Calculate Particle Forces

Fixed Problems

Transforms

Transform calculation of attached light bulbs and other components to string lights has been fixed.

All String Light socket rotations are now consistent with each other (String Sockets, Simulated Hanging Sockets and Non-Simulated Hanging Sockets)

String Light

Light Temperature Customization: Light Temperature was not considered for creating unique HISM components and similar customizations with different light colors could override each other. This is now fixed.

Line Trace: Line traces did not work properly on Light Bulbs. This is now fixed.

Collision Profile is now Block All Dynamic for Movable Light Bulbs and Block All for Static Light Bulbs.
Collision Profile is set to No Collision when Collision Query in Advanced Mesh Customizations is disabled.

Prefer Static Lights/Meshes: Fixed the behavior of Prefer Static Lights, Prefer Static Meshes behavior. if either one is true, strings are preferred to be static.

Customization Definitions: Placement array is now cleared if manual placement is not used.

Other Changes:

bDisableCollision for HISMCs is set to true when bCollisionQuery is disabled.
bCastStaticShadow is disabled if HISMCs can not be static.
bCastDynamicShadow for HISMCs is changed with bCastShadow

String Light Component

Bounding Box: The Bounding box of String Light Components was wrong. This has been fixed which also fixed some rendering and shadow casting issues

Shadows: String Light Components can now cast shadows properly.

Skip Update When Not Visible: This option would disable component tick if string lights were hidden. But the component tick would remain disabled even after string lights become visible. This is now fixed and the component tick is automatically enabled after string lights become visible.

Number of Sides: Num Sides can no longer be set to less than 1.

Static Meshes

Some sections of Light Bulb Static Meshes located in Content/Mesh/Lights did not cast shadow. Now all sections have the cast shadow enabled.

Wind Force Component

This component and all of its options (Local Wind) will be deprecated in the future. Please use directional wind instead.

##
#####

Version 1.1

Summary

- Directional Wind Support
- Organized Properties in Details Panel
- Ability to initialize String Light Particles with arbitrary locations
- More Blueprint and C++ Improvements

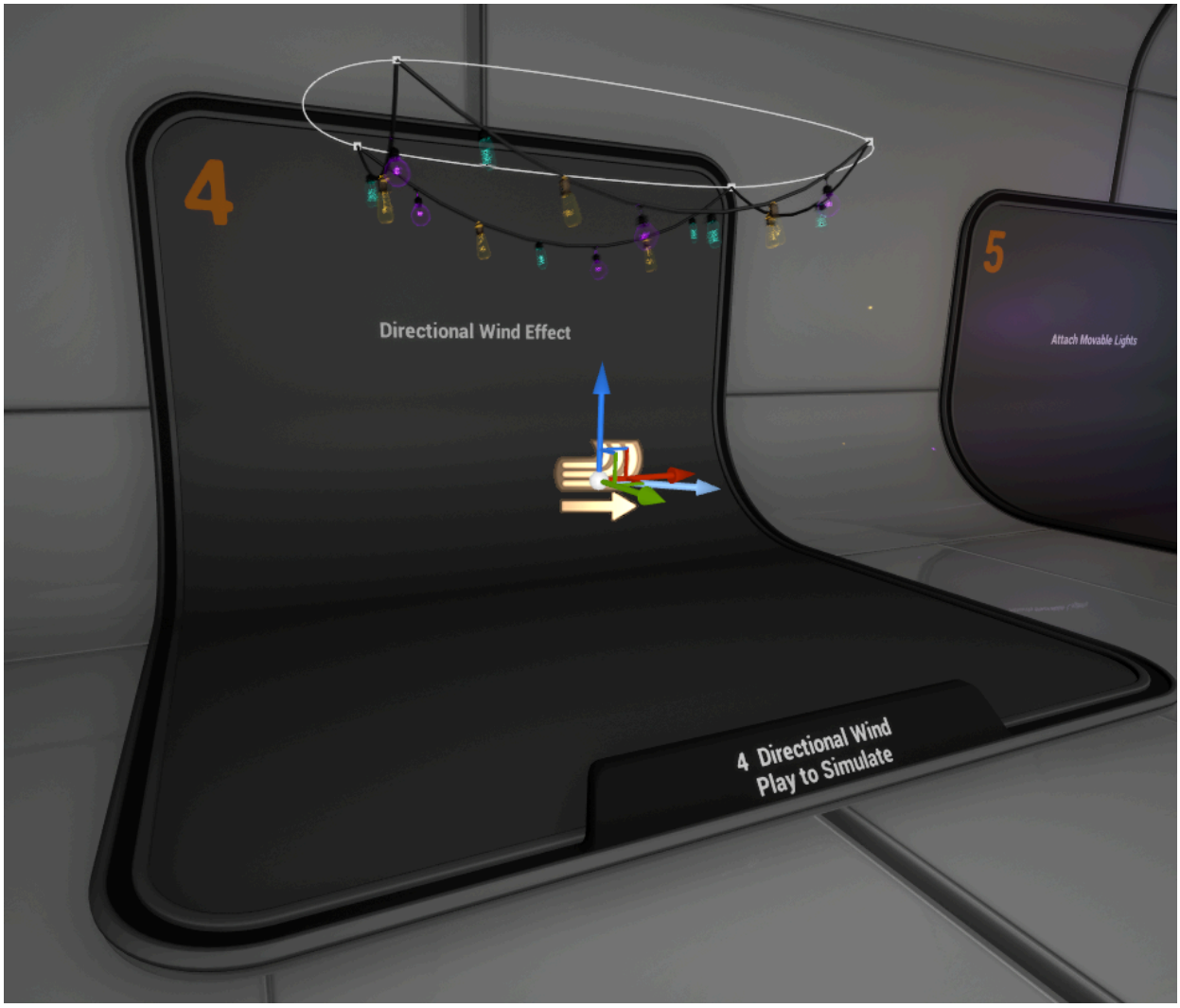
Directional Wind

- Support for “**Directional Wind”** with improved wind simulation.
  New options are added under “Cable Properties”

- “Wind Force Component” no longer changes “Particles Force” property on “String Light Component” and is simulated independently.

- Changed display name of "Wind Properties" to "Local Wind Properties".

*It is now recommended to use **“Directional Wind”** instead of “Local Wind”. We kept Local Wind for backwards compatibility.*

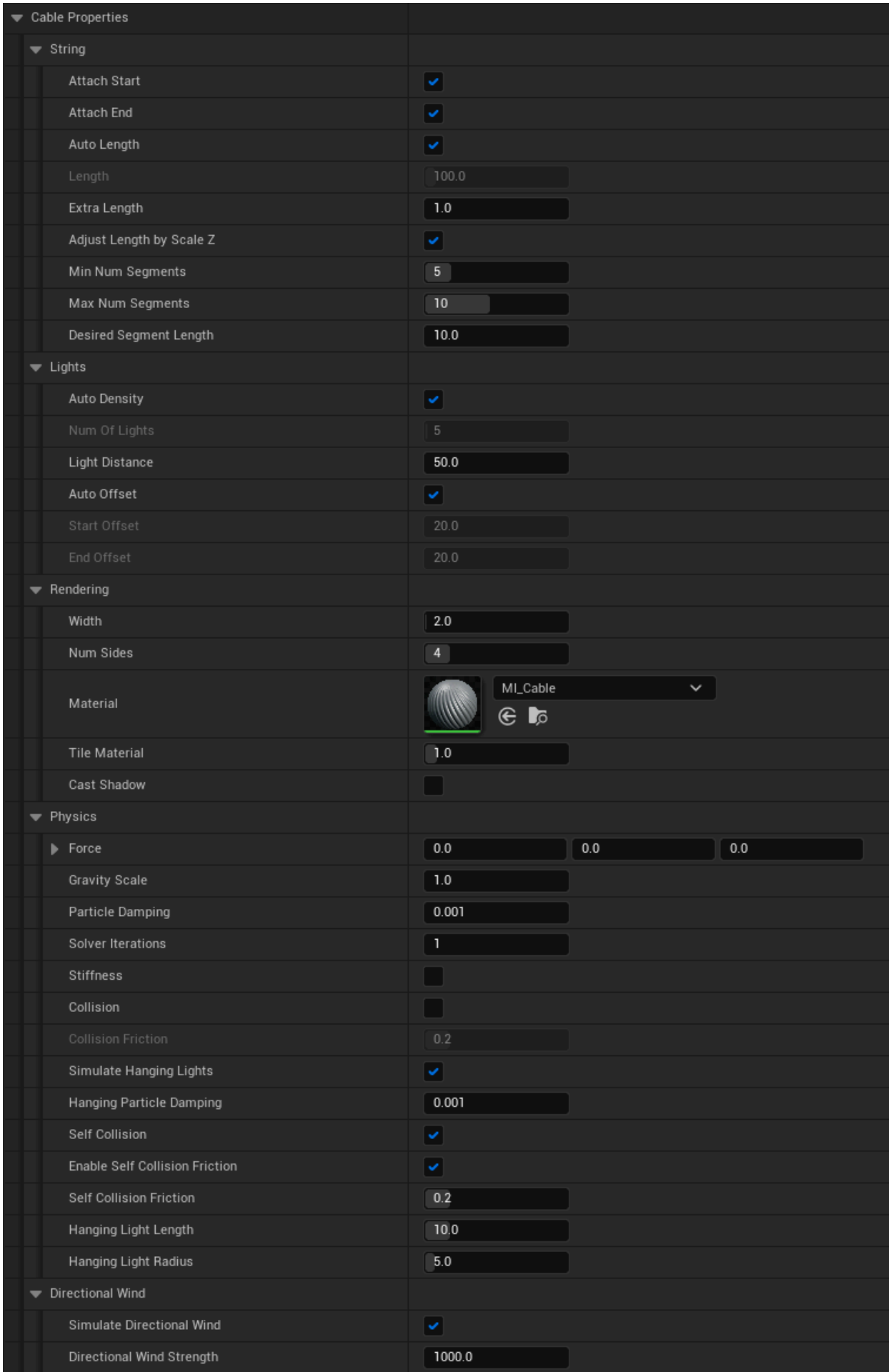


Details Panel

Organized properties in "**Cable Properties**".

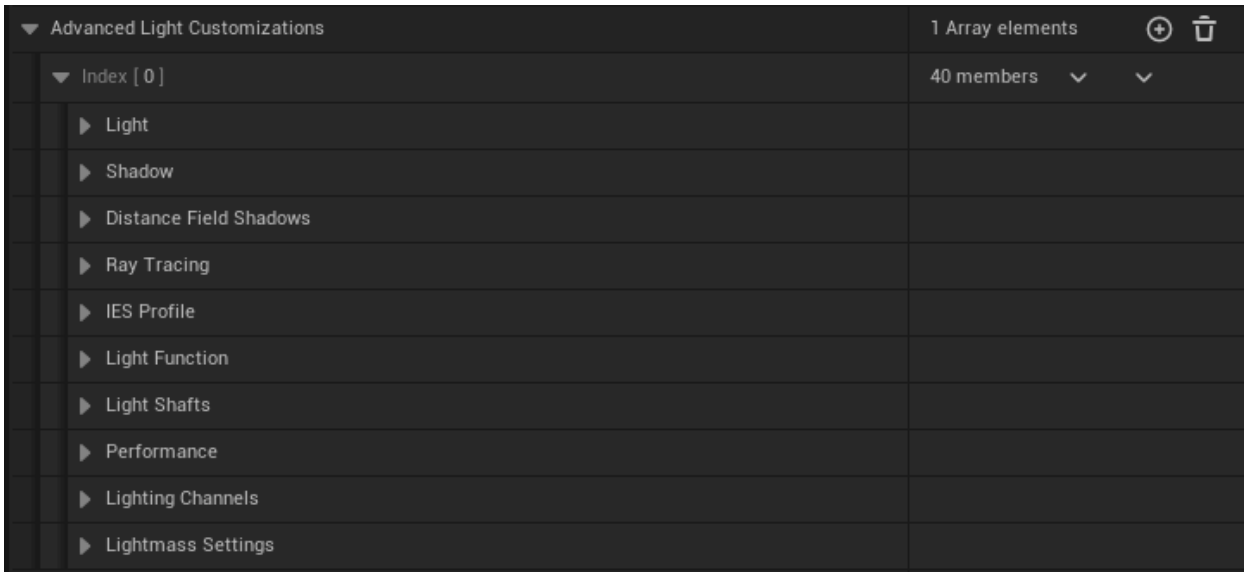
Collapsed View:

Expanded View:

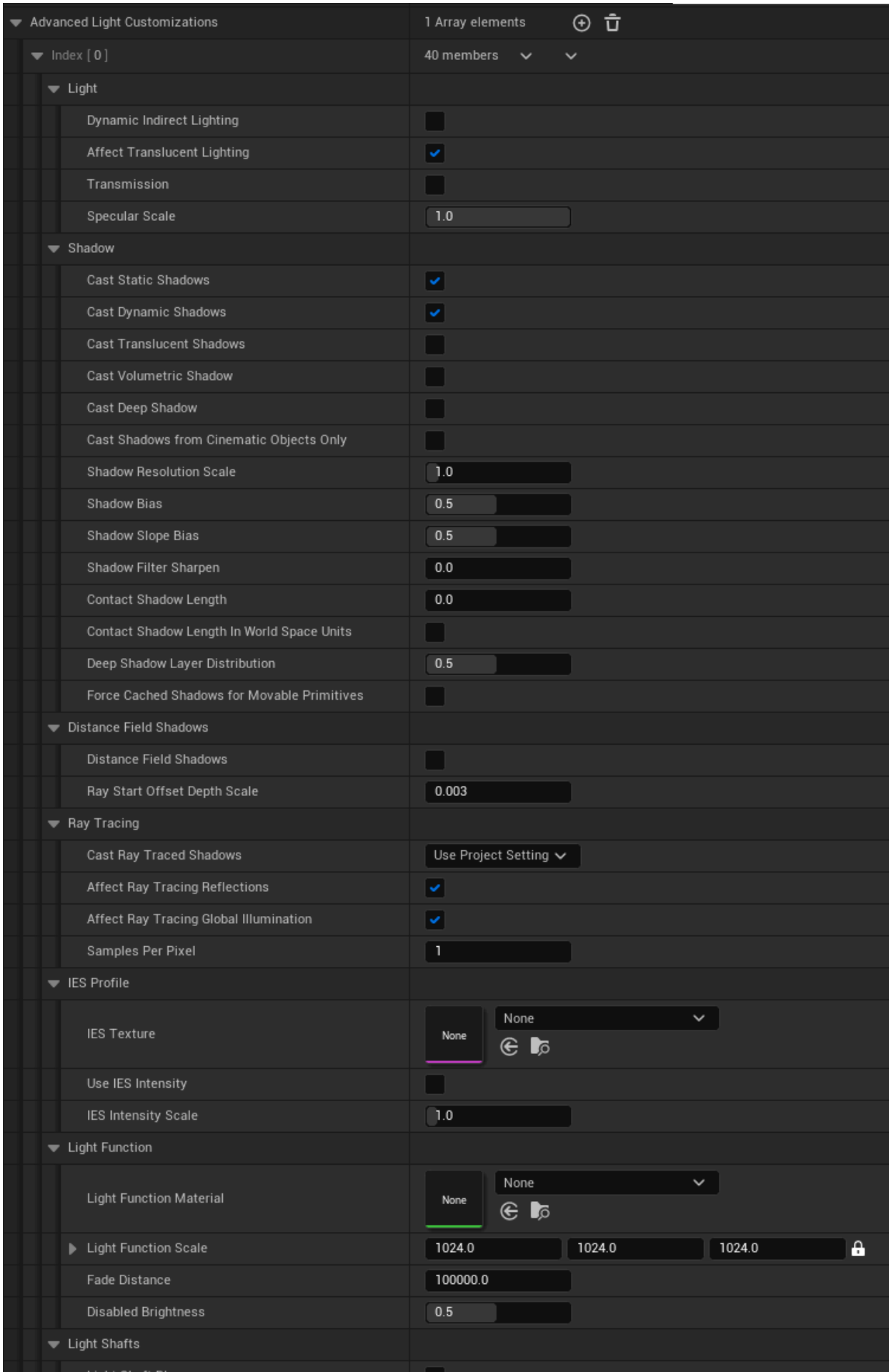

Cable Properties
String
Attach Start
Attach End
Auto Length
Length
100.0
Extra Length
1.0
Adjust Length by Scale Z
Min Num Segments
5
Max Num Segments
10
Desired Segment Length
10.0
Lights
Auto Density
Num Of Lights
5
Light Distance
50.0
Auto Offset
Start Offset
20.0
End Offset
20.0
Rendering
Width
2.0
Num Sides
4
Material
MI_Cable
Tile Material
1.0
Cast Shadow
Physics
Force
0.0
0.0
0.0
Gravity Scale
1.0
Particle Damping
0.001
Solver Iterations
1
Stiffness
Collision
Collision Friction
0.2
Simulate Hanging Lights
Hanging Particle Damping
0.001
Self Collision
Enable Self Collision Friction
Self Collision Friction
0.2
Hanging Light Length
10.0
Hanging Light Radius
5.0
Directional Wind
Simulate Directional Wind
Directional Wind Strength
1000.0

Organized properties in "**Advanced Light Properties**”.

Collapsed View:



Expanded View:

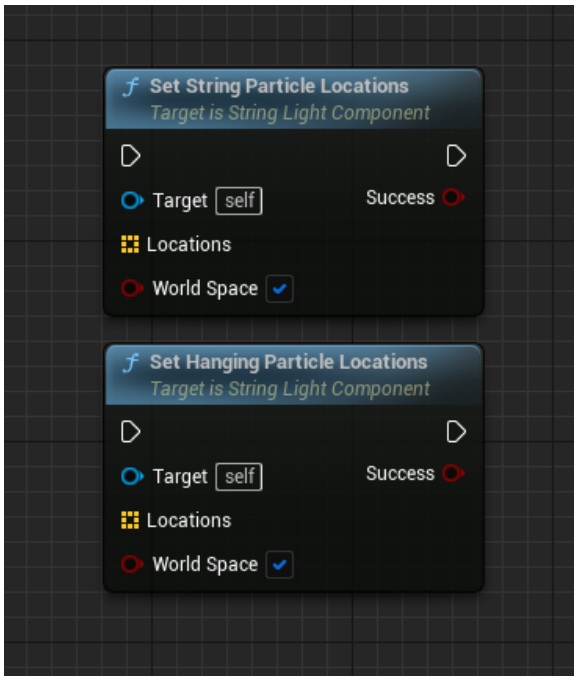

Advanced Light Customizations
1 Array elements
Index [0]
40 members
Light
Dynamic Indirect Lighting
Affect Translucent Lighting
Transmission
Specular Scale
1.0
Shadow
Cast Static Shadows
Cast Dynamic Shadows
Cast Translucent Shadows
Cast Volumetric Shadow
Cast Deep Shadow
Cast Shadows from Cinematic Objects Only
Shadow Resolution Scale
1.0
Shadow Bias
0.5
Shadow Slope Bias
0.5
Shadow Filter Sharpen
0.0
Contact Shadow Length
0.0
Contact Shadow Length In World Space Units
Deep Shadow Layer Distribution
0.5
Force Cached Shadows for Movable Primitives
Distance Field Shadows
Distance Field Shadows
Ray Start Offset Depth Scale
0.003
Ray Tracing
Cast Ray Traced Shadows
Use Project Setting
Affect Ray Tracing Reflections
Affect Ray Tracing Global Illumination
Samples Per Pixel
1
IES Profile
IES Texture
None
None
Use IES Intensity
IES Intensity Scale
1.0
Light Function
Light Function Material
None
None
Light Function Scale
1024.0
1024.0
1024.0
Fade Distance
100000.0
Disabled Brightness
0.5
Light Shafts

Blueprint

Blueprint **read** and **write** access to "**Max Draw Distance**" and "**Max Distance Fade Range**" in "Advanced Light Properties" struct.

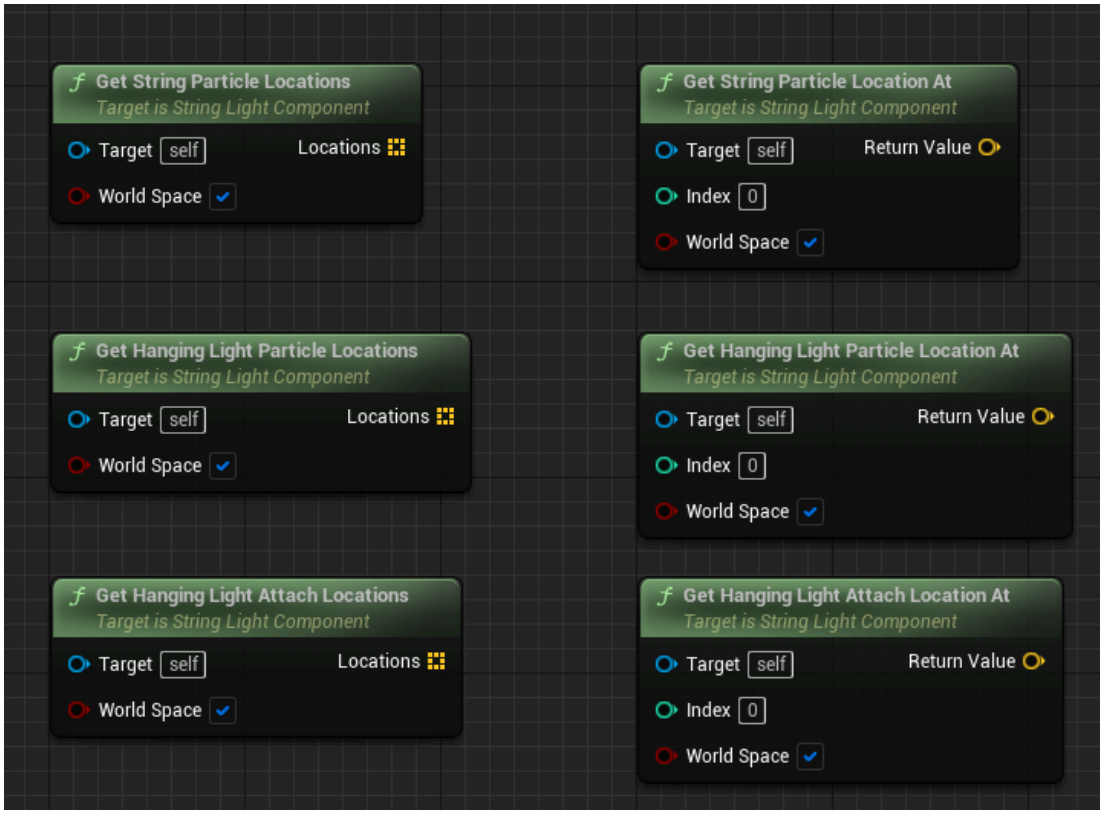
Added function "**Set String Particle Locations**" to initialize the String Light Component with arbitrary particle locations. (this can be also used to load particles from saved locations)

Added function "**Set Hanging Particle Locations**" to initialize the String Light Component with arbitrary hanging particle locations.



Added the "World Space" option to the following functions. By passing false, location is returned in component local space.

GetStringParticleLocations, GetStringParticleLocationAt, GetHangingLightParticleLocations, GetHangingLightParticleLocationAt, GetHangingLightAttachLocations, GetHangingLightAttachLocationAt



C++

Following properties in the "UStringLightComponent" are now private and should be accessed via respective public get/set functions.

StringLength, StringWidth,
NumSegments, NumSides, TileMaterial,
StartOffset, EndOffset,
NumOfHangingLights, bSimulateHangingLights