

Skip to content

Instances

	Evaluated	Instancing	
	Name	Position	
0	Geometry	-1.000 -1.000	
1	Collection	1.000 1.000	
2	Object	1.880 12.361	

The three types of instances.

In addition to storing real data like a mesh or a curve, objects can store instances, which themselves can reference more geometry, an object, or a collection. The purpose of instancing is to allow duplicating geometry and storing it in an object, without duplicating the actual data. This optimization allows render engines like [Cycles](#) to handle the same geometry data in many different locations better than when the data is duplicated.

Each instance keeps track of which geometry it corresponds to and how the instanced is transformed compared to it's source geometry. Instances can also store the `id` attribute, which is used to correct motion blur when instances move in an animation.

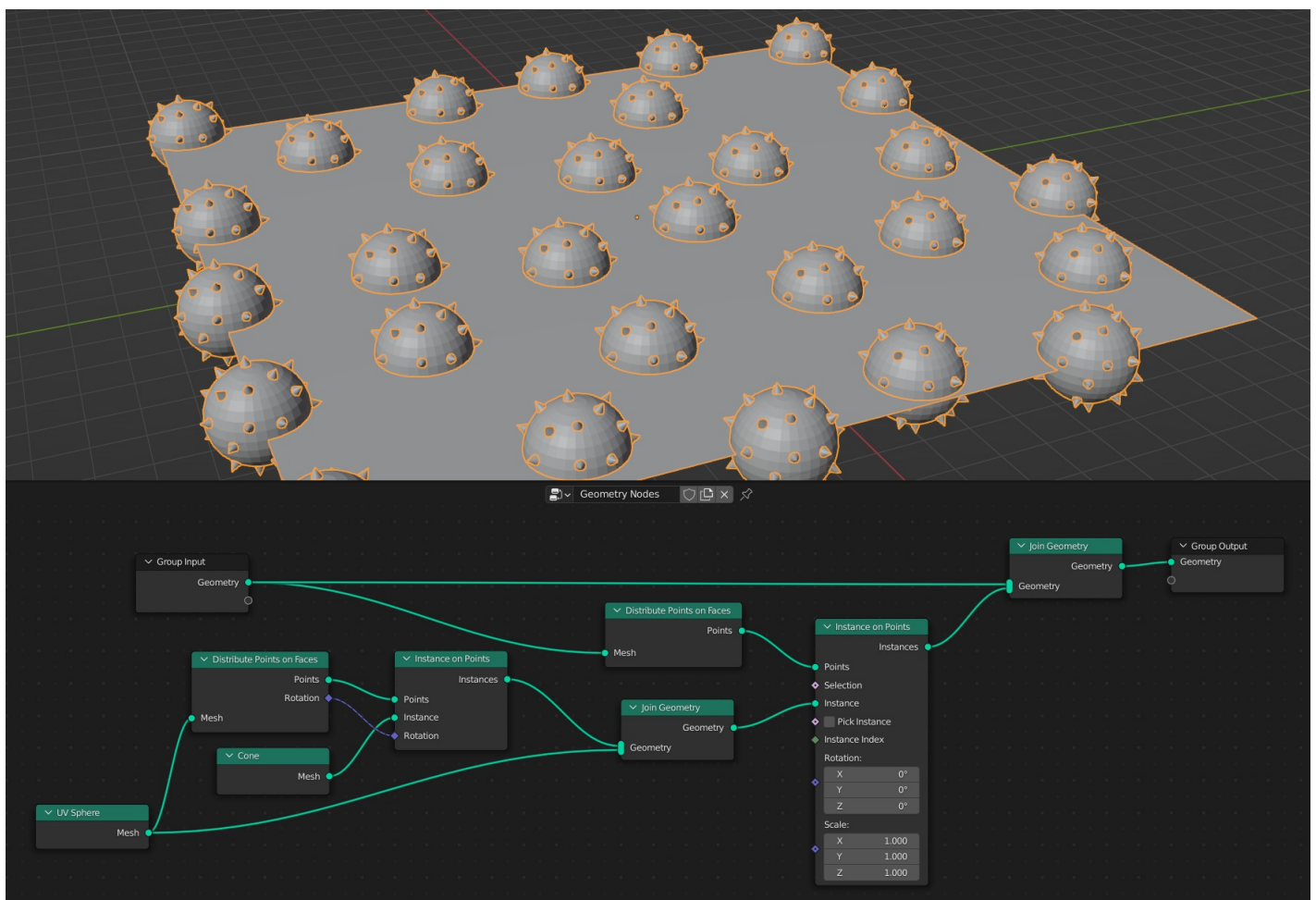
Instances can be created with geometry nodes using the [Instance on Points Node](#).

Warning

Currently instancing from geometry nodes cannot be mixed with instancing from the [Instancing](#) panel in the property editor.

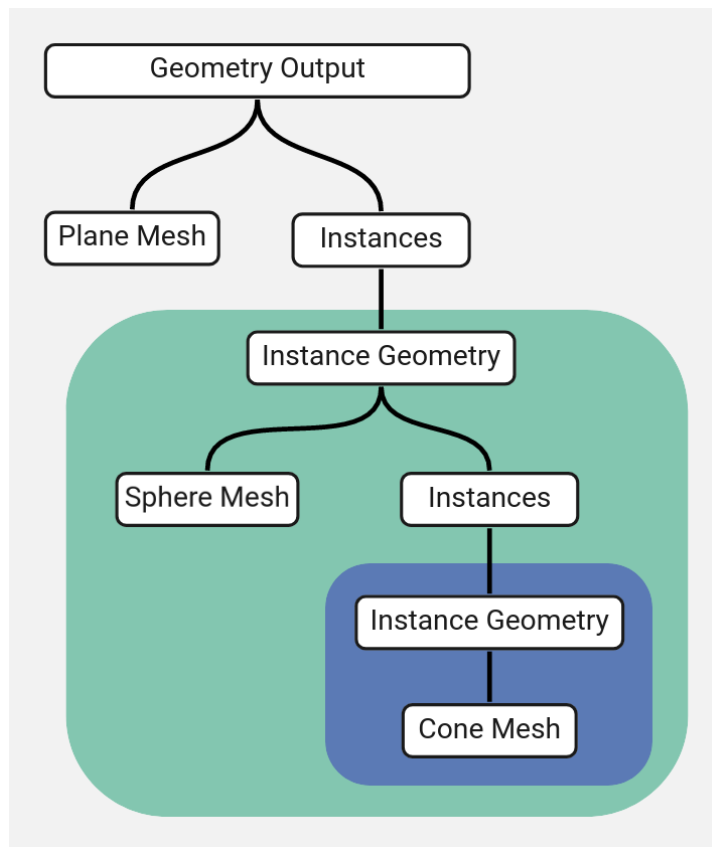
Nested Instancing

Since instances can store a geometry, and a geometry can contain instances, nested instancing is possible. In other words, it is possible to instance an instance, or even a collection of instances. For example, by default, the [Instance on Points Node](#) will create nested instances by instancing instances on the points of real geometry.



A node group that creates nested instancing by chaining [Instance on Points](#) nodes.

Here, nested instancing is used to distribute geometry that contains both a mesh and instances. The output geometry contains a ‘real’ mesh and a group of instances. Each instance contains a sphere mesh and many instances of a cone geometry.



The tree of instanced geometry for the example above.

What makes this method helpful is the output geometry only contains three unique meshes: the plane, the sphere, and the cone. This would make the performance much better if the meshes were more complicated.

Warning

Only eight levels of nested instancing are supported for rendering and viewing in the viewport. Though deeper trees of instances can be made inside geometry nodes, they must be realized at the end of the node tree.

Realizing Instances

The term “realizing” instances refers to converting the instances into unique geometry. When instances are realized they will take up more memory and manipulation to geometry will have to be processed individually rather than once per instancing geometry.

To realize instance use the [Realize Instances Node](#).

Instance Processing

Almost all nodes that process geometry do so by processing each unique geometry separately rather than realized geometry. For example, if a [Subdivide Surface Node](#) was placed at the end of the example above, it would only have to subdivide three meshes, rather than each instance of a mesh. Another important example is processing with the output of the [String to Curves Node](#), where each unique character only has to be processed once.

This method can improve performance a lot, but it means that the result of an operation will be the same for every instance of a certain geometry. In order to have unique results for every instance, the [Realize Instances Node](#) node can be used.

[View Source](#)
[View Translation](#)
[Report issue on this page](#)