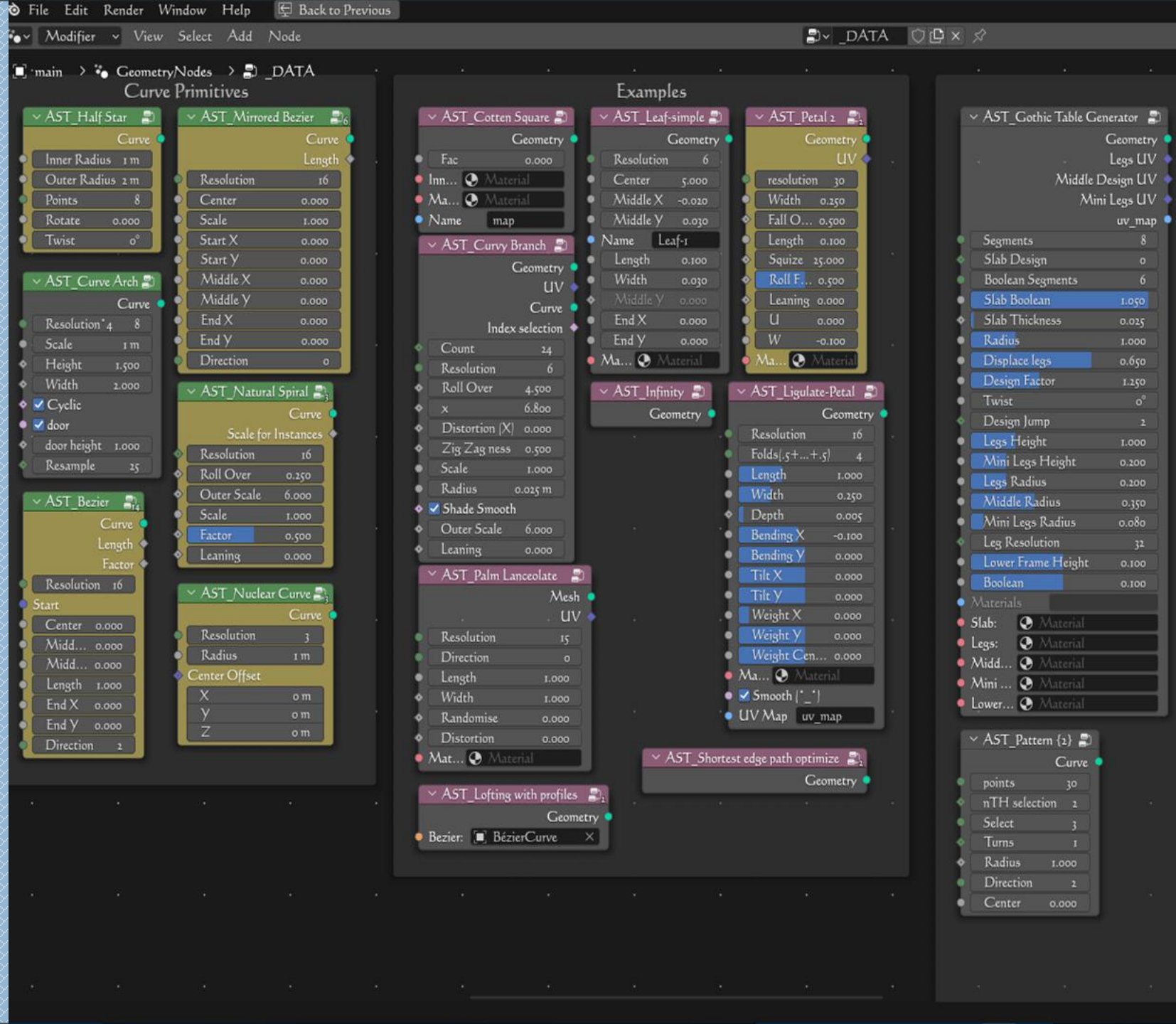
# ABHAY'S TOOL-KIT

## Guide

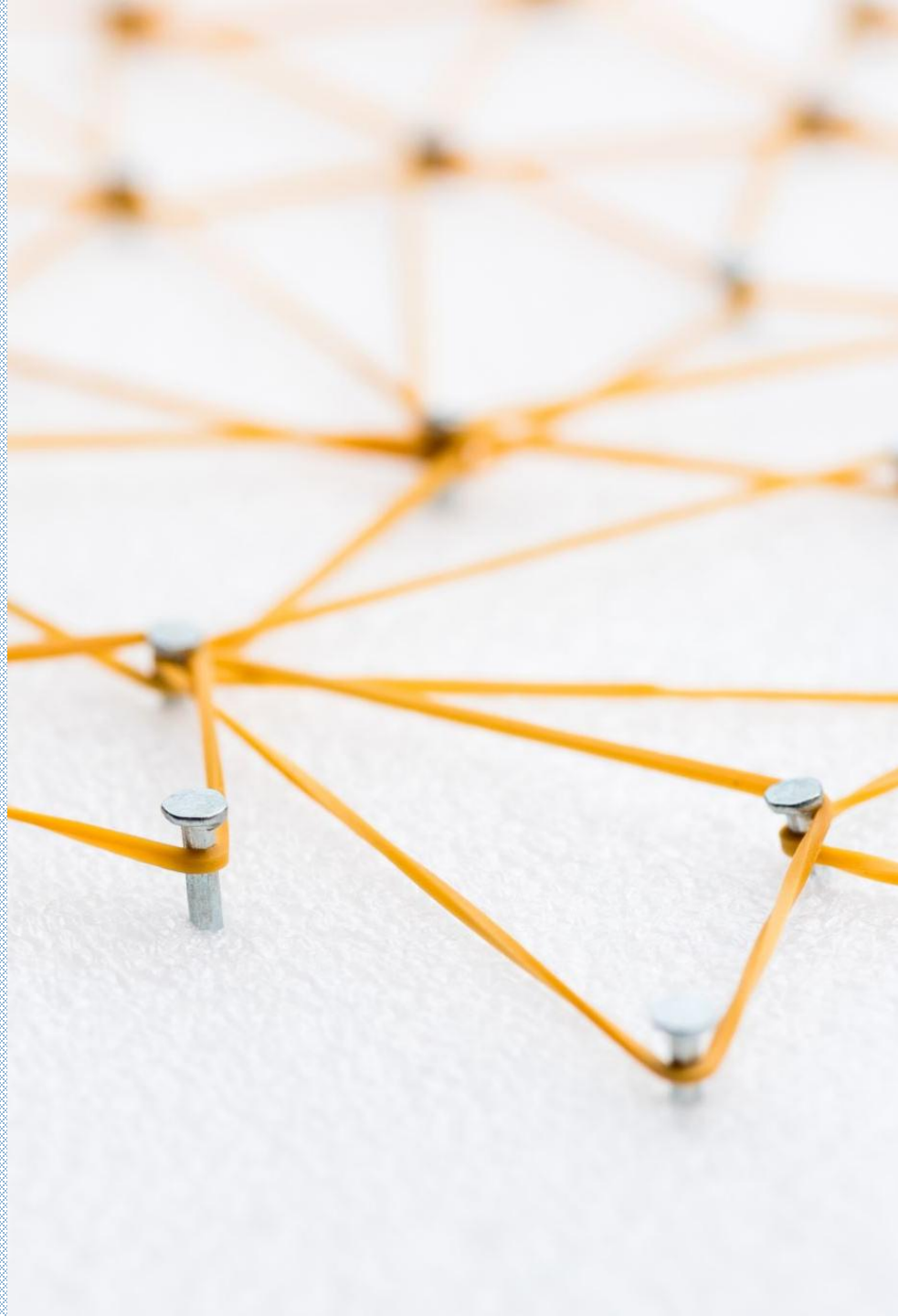


# CONTENTS

# INTRODUCTION

Blender's Geometry Nodes system is a powerful tool for procedural modeling and animation. However, users often face challenges in reusing and managing complex node setups across projects. This Tool-Kit aims to simplify and enhance the workflow for Blender users by creating a collection of reusable Geometry Node groups and an intuitive add-on for seamless integration into Blender's interface.

This is the guide to the node groups in the Tool-Kit. Here I will be explaining what these node groups do, how to use them, and their use cases.



# CURVES

List of Node Groups

i. Rebuild Curve (Spline Index Remapper)
ii. Change Spline Type
iii. Curve Deform
iv. Curve Optimize (for edge path to curves)
v. Curve Optimization Iteration
vi. Loft Curves
vii. Loop Curve
viii. Curve Rebuilder (Curve Island Index)
ix. Skin Curve
x. Spline Factor
xi. Curves Splitter

## Rebuild curves (spline index remapper)

description: this node group is specially designed for reindexing and reorganizing spline topology. It rebuilds curve topology by destroying the original spline organization and reassigning points to new splines based on the group ID attribute. It outputs reindexed curves with uniform cyclic state control.

Use cases:

spline consolidation: merging multiple splines into a single spline (assign the same group ID to all points).

Spline fragmentation: splitting a single spline into multiple segments based on group ID.

Topology reordering: changing spline rendering order.

## Change spline type

description: dynamically converts curve spline type between poly, Bezier, Catmull Rom, and NURBS with selection-based control – enabling smoothing of sharp corners, mathematical curve refinement, and automatic handle manipulation through a unified interface.

Curve select index driven switch: (0 = poly, 1 = Bezier, 2 = Catmull Rom, and 3 = NURBS)

use case: converts jagged poly curves into smooth curves for organic shapes.

## Curve deform

description: deform input geometry along a target curve. It aligns geometry to the curve's tangent while preserving local shapes with axis selection determining direction. Ideal for creating roads, pipes, vines or any geometry that needs to follow a curved path.

You can reference curve object from the scene or process curves directly from geometry nodes.
The 'curve index' input handles objects with multiple splines deform difference meshed to different splines in the same object.

It preserves original mesh topology.
You can adjust curve path after the deformation.
[Credit: implementation approach based on Erindale's curve deformation tutorial(YouTube).]

## Curve optimize (for edge path to curve)

description: optimize the output curves from 'edge path to curves'. Fixing high density of curves due to overlapping.

## Curve optimization iteration

description: optimize overlapping curves (singular iteration, you'll have to use multiple until you get the desired result). An alternate to the "AST_Curve_Optimize (edge path to curves)" captured length input takes the spline length from a capture attribute node and pass it to the next iteration, or you can use a repeat zone as well.

## Loft curves

description: generates a seamless tubular surface between input profile curves. Made this from erindale's tutorial, but mine is a little different. It connects curve profiles with adaptive geometry, creating organic shapes like pipes, tentacles, sculpted surfaces, or parametric designs. It offers precise control over mesh density and interpolation for both longitudinal and cross-sectional direction.

Converts curves into continuous surfaces.

[Credit: core algorithm based on erindale's curve lofting tutorial (youtube).]

## Loop curve

description: created this because I forgot there's a 'set spline cyclic' node. I kept this as a reminder of my stupidity (!_!). still useful for learning purposes.

## Curve rebuilder (curve island index)

description: branching curves are actually more than one curve. It converts input curves into mesh and extracts the island index. Then it converts back to curves with the same number of points, allowing you to have an index for curve islands. Even though those curve islands are actually more than one curve with overlapping endpoints, it automatically reverses spline direction when needed to match the original curve direction.

## Skin curve

description: converts input curves into tubular meshes by extruding a circular profile along the spline, with precise control over radius, resolution, and UV mapping. Generates parametric pipes, cables, or organic forms while maintaining clean topology and customizable UV mapping.

## Spline factor

description: generates a factor attribute along all the input curves. Also, outputs normal individual factor from the spline parameter as well.

## Curve splitter

description: splits the desired endpoint of all the input splines into 2, 3, or random branches. Very useful in creating foliage and trees. It has all the necessary controls for splitting large branches into multiple smaller branches with randomization, split-point control, opening of branches, as well as direction and rotation of the split.

# CURVE PRIMITIVES

List of Node Groups

I. Bezier
II. Curve Arch
III. Half Star
IV. Mirrored Bezier
V. Natural Spiral
VI. Nuclear Curve

## Bezier

Description: It is a basic Quadratic Bezier but with reorganized parameters. New parameters are simplified to expose to the modifiers panel after you are done building your project with it. The spline factor and spline length are also directly exposed for ease of convenience.

Its primary focus is foliage creation. Things you create with it are easy to modify or simple for others to use.

## Curve Arch

Description: It's an arch-shaped curve useful in creating archways and windows. All the parameters are self-explanatory.

## Half Star

Description: A curve in the shape of a star that is sliced from the middle. Smooth it with a 'change spline type' node to create organic shapes useful in making certain types of leaves.

## Mirrored Bezier

Description: Same as 'AST_Bezier', it's just a Quadratic Bezier with reorganized controls and simplified parameters, but it's mirrored. This makes it useful in profiling other curves. Play with its parameters once you're done making your model and see the usefulness.

## Natural Spiral

Description: It is a curve that mimics the way branches, stems, and plants curl up in nature, making it ideal for foliage creation, especially useful in making ferns, branches, and leaves.

## Nuclear Curve

Description: A curve in the shape of a nuclear hazard sign. Move its center to create leaves like the Livistona plant or combine it with an “AST_Change Spline Type” to create interesting patterns and mandalas.

# EXAMPLES

List of Node Groups

i. Cotton Gula
ii. Curvy Branch
iii. Infinity
iv. Leaf-simple
v. Ligulate-Petal
vi. Lofting with profiles
vii. Palm Lanceolate
viii. Petal 2
ix. Shortest edge path optimize

## Cotton Square

Description: A bloom-capable cotton square without cotton that I created using node groups from my toolkit. It showcases the techniques that I use for creating organic shapes.

## Curvy Branch

Description: It is a singular branch/twig generator that doesn't have child branches, showcasing the use case of "AST_Natural Spiral."

## Infinity

Description: A curve in the shape of infinity, useful for animating something that needs to move in the infinity shape or for creating parametric designs.

## Leaf Simple

Description: A simple leaf generator made using the nodes "AST_Bezier" and "AST_Mirrored Bezier" from this toolkit, with much control over shape and useful parameters.

## Ligulate-Petal

Description: A ligulate petal generator made using nodes from this toolkit. Very useful for creating flowers with ligulate petals like ‘Daisy’.

## Lofting with Profiles

Description: An example “AST_Loft Curve” with preloaded profiles. Just give it a Bezier Curve, and it will show you a smart technique of lofting curves with auto curve sorting. By attaching the profiles to the input Bezier, you can edit the Bezier afterwards.

## Palm Lanceolate

Description: Individual segment of the palm leaves made with “AST_Bezier” and “AST_Mirrored Bezier.” Modify the parameters a little, and it will work for a variety of trees with similar leaves.

## Petal 2

Description: A simple petal generator. The type that is usually found in vine leaves, flowers, or in the wild. Made using “AST_Natural Spiral.”

# Shortest Edge Path Optimize

Description: An example of how to optimize the geometry made using the shortest edge path technique.

# GENERATORS

List of Node Groups

i. Di-Circle Pattern
ii. Gothic Table Generator
iii. Pattern {2}
iv. Pottery
v. Rock Generator
vi. Tree – 50+

## Di-Circle Pattern

Description: A circular mandala/pattern generator. Changing each parameter randomizes the mandala. It adds two vector circles to create these patterns.

## Gothic Table Generator

Description: A round table generator in gothic style. Fully customizable with 18 parameters.

## Pattern {2}

Description: A round pattern generator useful for making parametric designs. This pattern generator provides more control over the pattern you generate.

## Pottery

Description: It is a pot generator with Bezier input. Just give it a Bezier in the shape of the desired pot profile (only one side), and it will generate the pot from that profile.

## Rock Generator

Description: A procedural rock/boulder generator. You can even generate rocks in custom shapes by giving it a rough-shaped geometry of the rock. By turning off the ICO option, it will generate without input geometry. You can also Remesh the rock within the node. Try all the parameters for a better understanding of their use.

## Tree – 50+

Description: A tree generator with more than 50 parameters to customize it. Useful for instantly creating trees for your scenes. It can be used for creating a variety of trees, like fantasy, artistic, spooky, and wild trees, etc.

# GEOMETRY

List of Node Groups

i. Align to Grid Plane
ii. Bounding Box +
iii. Bounding Box Group Index
iv. Bounding Grid
v. Bounding Lattice
vi. Rotate Geometry
vii. Self Iteration (Caution)
viii. Selective Join Geometry
ix. Simplify for Viewport
x. Transform Geometry +

## Align to Grid Plane

Description: Moves the input geometry so that its lowest point sits on the viewport grid plane. A very handy node when instancing things or sampling objects from the viewport.

## Bounding Box +

Description: A bounding box node with pre-organized outputs for faster workflow. Consider the bottom plane of the bounding box of your geometry as A, B, C, and D, and the top plane as A', B', C', and D'. It outputs all the corner positions of the bounding box along with its center, center of the bottom plane (base), scale, and bounding surface area. Where A is the "MIN" point and C' is the "MAX" point. Just hide the unnecessary outputs with 'Ctrl+H' after plugging in the needed outputs.

## Bound Box Group Index

Description: Grouped bounding boxes. "Creates bounding boxes for each loose or group-indexed geometry," or geometry with multiple meshes. Input the mesh island's island index in the group index. Output vectors are field outputs accordingly.
Learned this node from Erindale's tutorial on bounding boxes.

## Bounding Grid

Description: Creates a grid on the lowest point of the input geometry. Scales to the X and Y size of the input geometry.

## Bounding Lattice

Description: Creates a lattice around the input geometry matching the bounding box.

## Rotate Geometry

Description: Rotates geometry with field inputs, unlike transform geometry, where you can only rotate whole geometry. With this node, you can rotate the selected part of your geometry with smooth falloff. You can choose the axis, center of rotation, and angle. The factor input is where you put the mask of the selected geometry.

## Self Iterate Geometry

Description: Distributes itself on its own points with each iteration.
Caution: Only use simple geometry with a lower vertex count. Using a high-density mesh or geometry will cause your PC to freeze or crash.
Useful in creating abstract art.

## Selective Join Geometry

Description: It lets you join 5 geometries on the base geometry with an index switch. Each iteration of the index switch adds the next input geometry to the base geometry.
Use case: Imagine you made a tree. Now you want to control the visibility of each part of your tree (e.g., base branches, secondary branches, roots, twigs, leaves, etc.). You can just plug in the different parts of your tree in the geo inputs on this node and connect the index switch to the group input.

## Simplify for Viewport

Description: It lets you 'Convex hull' the heavy geometry but automatically switch to the original geometry when you render. Very useful when dealing with heavy instances. Just plug it in between the geometry that you want to instance and the Instance on Points node.

## Transform Geometry +

Description: A basic Transform Geometry node but with field inputs, meaning you can transform geometry selectively or input field values. It also has a center input, which changes the center of transformation. The default is 0,0,0.

# INSTANCES

List of Node Groups

i. Align Instance to Topology
ii. Alternate Distribution
iii. Bounding Box Instances
iv. Circular Distribution
v. Collection Info +
vi. Collection Info Custom Index
vii. Instance Packer
viii. Instance on Edges

## Align Instances to Topology

Description: Instances on input geometry with perfect alignment to the face normal and the center of the input geometry.

## Alternative Distribution

Description: Distributes instances on a curve in alternate or opposite directions, like leaves on twigs of some plants. Useful for creating foliage and plants.

## Bounding Box Instances

Description: Creates bounding boxes for instances.
Learned from Erindale's tutorial on bounding boxes.

## Circular Distribution

Description: Distributes instances in a circular manner with tilt, twist, and rotation control.

## Collection Info +

Description: Collection info with collection bounding boxes and MIN and MAX vectors.
Useful for instance packing.
Learned from Erindale's tutorial on instance packing.

## Collection Info Custom Index

Description: Collection info with an index based on position on a chosen axis or circular index. You can reverse the index afterward. You can also move around the items in your collection in the viewport, and it will automatically update the index in geometry nodes. A very useful node for making interactable models.

## Instance Packer

Description: For packing objects that are not the same size but perfectly spaced according to their size.
Needs Collection Info + to pair with it.

## Instance on Edges

Description: Distributes instances on edges. The surface normal input aligns the rotation of instances to the input geometry.

# MESH PRIMITIVES

List of Node Groups

i. Aloe vera leaf
ii. Balloon
iii. Icosahedron
iv. Julia Fractal
v. Lattice
vi. Prism
vii. Pyramid
viii. Slicing Circle
ix. Tetrahedron

## Aloe Vera Leaf

Description: An Aloe Vera leaf generator. Can be modified to make similar plants too, e.g., cacti.

## Balloon

Description: A balloon generator useful as a placeholder for instancing something.

## Icosahedron

Description: An icosahedron generator, useful as a placeholder for instancing something.

## Julia Fractal

Description: A 4D fractal called Julia.

## Lattice

Description: A lattice generator, useful for making buildings, houses, etc.

## Prism

Description: A prism generator, useful as a placeholder for instancing something.

## Pyramid

Description: A pyramid generator, useful as a placeholder for instancing something.

## Slicing circle

Description: A circle that can be bisected at any point. Useful as a placeholder for leaves in foliage creation.

## Tetrahedron

Description: A tetrahedron generator, useful as a placeholder for instancing something.

# MODIFIERS

List of Node Groups

i. Add Scales
ii. Damage Geometry
iii. Fractal Shaper
iv. Mirror geometry
v. Points to Lattice
vi. Surface Cobble
vii. Terrain Displacement
viii. Trim Geometry

## Add Scales

Description: Adds scales on the surface of the input geometry.

## Damage Geometry

Description: Damages/breaks up the input geometry to give it an old, withered look.

## Fractal Shaper

Description: Modulates the input geometry to a stepped look. Useful in landscape creation for generating the different layers of the terrain.

## Mirror Geometry

Description: Quickly mirrors the input geometry on the X or Y axis.

## Points to Lattice

Description: Inputs a bunch of points and creates a kind of lattice out of them. It has very specific use cases, like abstract art or creating a plant in a certain shape.

## Surface Cobble

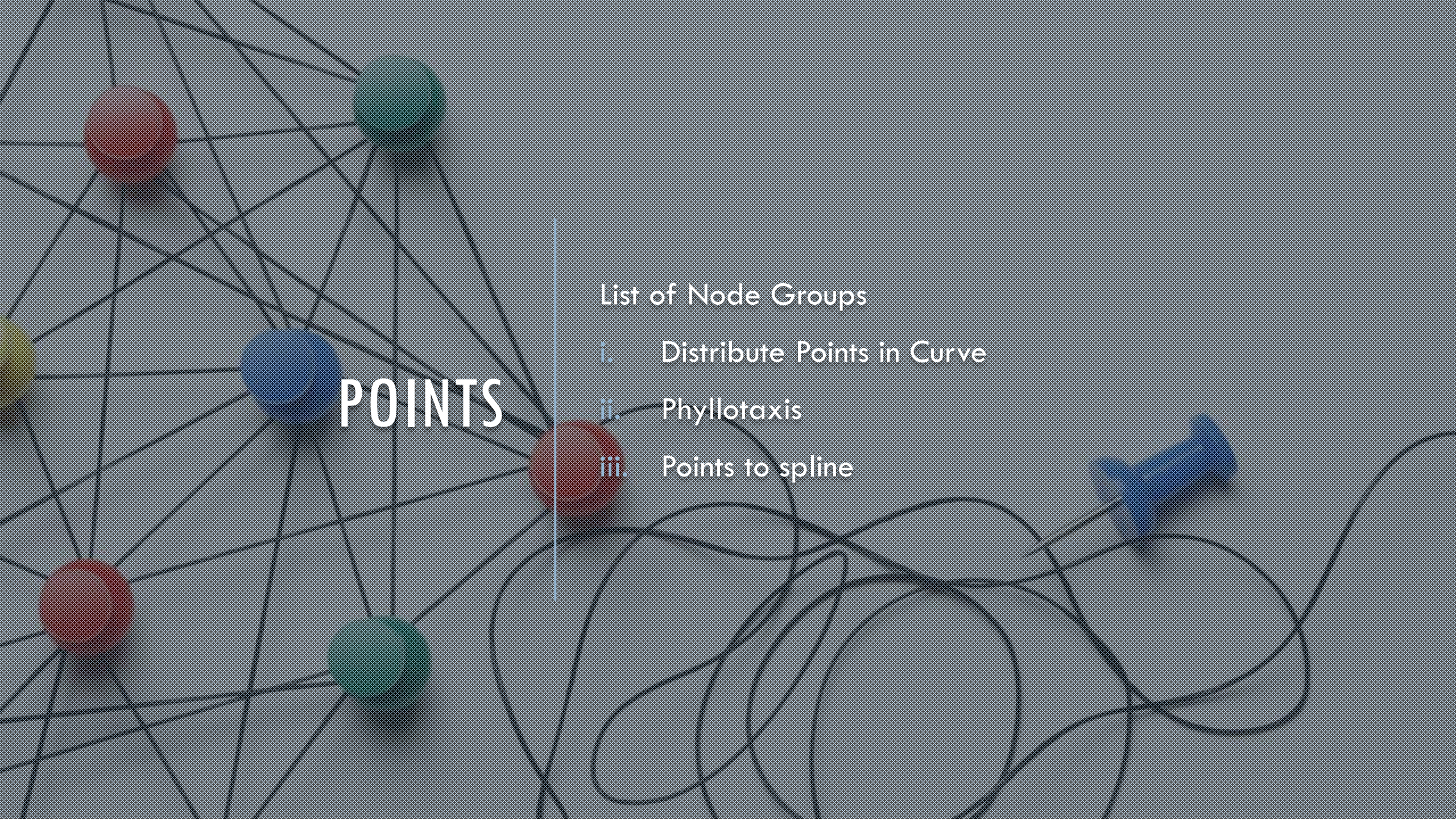
Description: Turns the surface of the input geometry into cobblestone. Great for creating cobblestone walls, paths, and other rocky assets.

## Terrain Displacement

Description: Displaces a height map into a terrain with precise control and subdivision setup. A very useful node — I use this all the time for all of my terrains.

## Trim Geometry

Description: Trims the input geometry on the X, Y, or Z axis from both sides, just like a trim curve node. It uses a Boolean node, so it’s not super fast on heavy meshes..



# POINTS

List of Node Groups

i. Distribute Points in Curve

ii. Phyllotaxis

iii. Points to spline

## Distribute Points in Curve

Description: Distributes points in the area that the input curve covers, whether it is cyclic or not. A non-cyclic curve will have points distributed in the concave area/section. You can control the density of points, randomize them, and control the probability of spawning.

## Phyllotaxis

Description: Points distributed in a phyllotaxis pattern. Useful for creating organic projects like plants and trees, especially flowers.

## Points to Spline

Description: Creates a spline from the input points. A little different from the base node. You can choose the spline type and have the option for inbuilt circular weight (basically, it creates a radial gradient from the world center and uses it as weight). You can input a position to overwrite the world position for the circular weight.

# SELECTION

List of Node Groups

i. Expand Selection
ii. Mesh Proximity
iii. Minimum Distance
iv. Nearest Point Selection
v. Select by Direction
vi. Select by Normal
vii. Select by Position
viii. Select by Index

## Expand Selection

Description: Expands your Boolean selection by one vertex in all directions.
Learned from Erindale's tutorial on geometry selection.

## Mesh Proximity

Description: Creates a mask and falloff based on the distance and clipping between target and reference geometry. Input takes the geometry from which you want to create the mask.

## Minimum Distance

Description: Outputs the distance between the input geometry and the location input or the object input, and outputs a selection based on the epsilon.

## Nearest Point Selection

Description: Selects the nearest point from the input object (or empty object). Useful for selecting specific points in the input geometry.

## Select by Direction

Description: Selects based on a certain angle (facing a certain angle).

## Select by Normal

Description: Selects based on the X, Y, and Z direction of the normals of the geometry.

## Select by Position

Description: Selects based on whether the position of the input geometry is greater or lesser than the input value on the X, Y, or Z axis.

## Select by Index

Description: Selects based on the index occurrence. For example, select every 2nd vertex (even). The selection can be inverted by changing the “Equals to” input value. Also allows a custom index.

# UTILITIES

List of Node Groups

i. Accumulate Curve
ii. Degree
iii. Flip Indices
iv. Fold Attribute
v. Grid Index
vi. Map Attribute
vii. Store Edge Angle
viii. Value++
ix. View Instance Attribute

## Accumulate Curve

Description: Accumulates the position, tangent, and normal of the input curve on any position/point on the input curve. Control that point using the length input.

## Degree

Description: A value node of degree input instead of the typical radian value.

## Flip Indices

Description: Flips the grid (2D) indices. For example, if your geometry has two indices going on different axes, you can switch them using this node.
Learned from Erindale's tutorial on Index.

## Fold Attribute

Description: Folds the input attribute at the input value. For example, if the input attribute ranges from 0 to 1 and the value input is set to 1, it will turn the range from "0 to 1" into "0 to 1 to 0." If the value input is set to 0.5, then it becomes "0 to 1 to 0 to 1 to 0."

## Grid Index

Description: Projects a grid of indices on X and Y coordinates. This custom index can be used for sampling.
Use case: Joining two ends of two pipes/cylinders with different indices. Joining them with their index will create messy geometry because of misaligned indices. Assign this custom index to both the end loops/circles, then join them to create seamless joints.

## Map Attribute

Description: Re-ranges the input attribute. Automatically clamps the minimum value to 0 and the maximum value to 1.

## Store Edge Angle

Description: Stores the edge angle, signed and unsigned, to the geometry for texturing purposes. You can use the stored data in shader nodes using an attribute node with the input attribute name set to Signed or Unsigned.

## Value ++

Description: A value node with some extra options. The positive and negative outputs are the positive and negative integers of the base input value. Random in Range {+-} will give a random value between the positive and negative integers of the base value. The %% input is a multiplier to the base value, and value to %% will give a random value between the base value and the multiplied result.

## View Instance Attribute

Description: Visualizes a float attribute of ‘Instances’ with points through the weight output. The radius controls the point radius.
Only for viewing the attribute to verify or check it.

# VECTOR

List of Node Groups

i. Add Noise
ii. Cube Projection
iii. Cylinder Projection
iv. Landscape Projection
v. Map Vector Attribute
vi. Ping Pong Vector
vii. Sphere Projection
viii. Triplanner Projection
ix. VXYZ arc

## Add Noise

Description: Noise set-up for randomly displacing geometries.

## Cube Projection

Description: Stores a UV projection from all the sides of the input geometry. Suitable for geometry with sharp corners like hard-surface modeling.
UV attribute name is “Cube_Projection.” Go to the shader editor and add an attribute node, then set the attribute name to “Cube_Projection.”

## Cylinder Projection

Description: Stores a UV projection. Unlike “AST_Cube_Projection,” this UV is three-segmented: two sides (Top and Bottom) and a third sideways loop, just like a cylinder. Suitable for cylinder-like geometry such as pillars and tree trunks.
UV attribute name is “Cylinder_Projection.” Go to the shader editor and add an attribute node, then set the attribute name to “Cylinder_Projection.”

## Landscape Projection

Description: Stores a UV projection from all the sides like “AST_Cube_Projection,” but the Top and Bottom sides have very sharp edges and only show up in nearly completely flat areas, making it ideal for texturing landscapes. Using default UV on landscapes creates angled textures that look very ugly; this fixes that. You can control at what angle the Top and Bottom UV should show up. You can also flip the X and Y coordinates with the alternate XY mask input.
UV attribute name is “Landscape_Projection.” Go to the shader editor and add an attribute node, then set the attribute name to “Landscape_Projection.

## Map Vector Attribute

Description: Re-ranges the input vector attribute. Automatically clamps the minimum value to 0 and the maximum value to 1.

## Ping Pong Vector

Description: Ping-pongs a vector input, similar to Fold Attribute.

## Sphere Projection

Description: A projection that uses two ‘Arcs,’ one going horizontally and the other going vertically, to create a UV. Useful in texturing spherical objects. It does not store the projection itself — use a “Store Named Attribute” node to access this UV in the shader editor.

## Triplanner Projection

Description: Stores a UV projection from all the sides with smoother falloff. Suitable for smoother geometry without sharp corners. On sharp geometry, the result will be unexpected. You can also control the falloff angle.
UV attribute name is “Triplanner_Projection.” Go to the shader editor and add an attribute node, then set the attribute name to “Triplanner_Projection.”

## XYZ Arc

Description: A vector of three arcs, each going on a separate axis. Each can be used to create radial weight or index on the respective axis or combine two of them to create spherical UVs. You are supposed to separate the output vector and re-combine as needed.