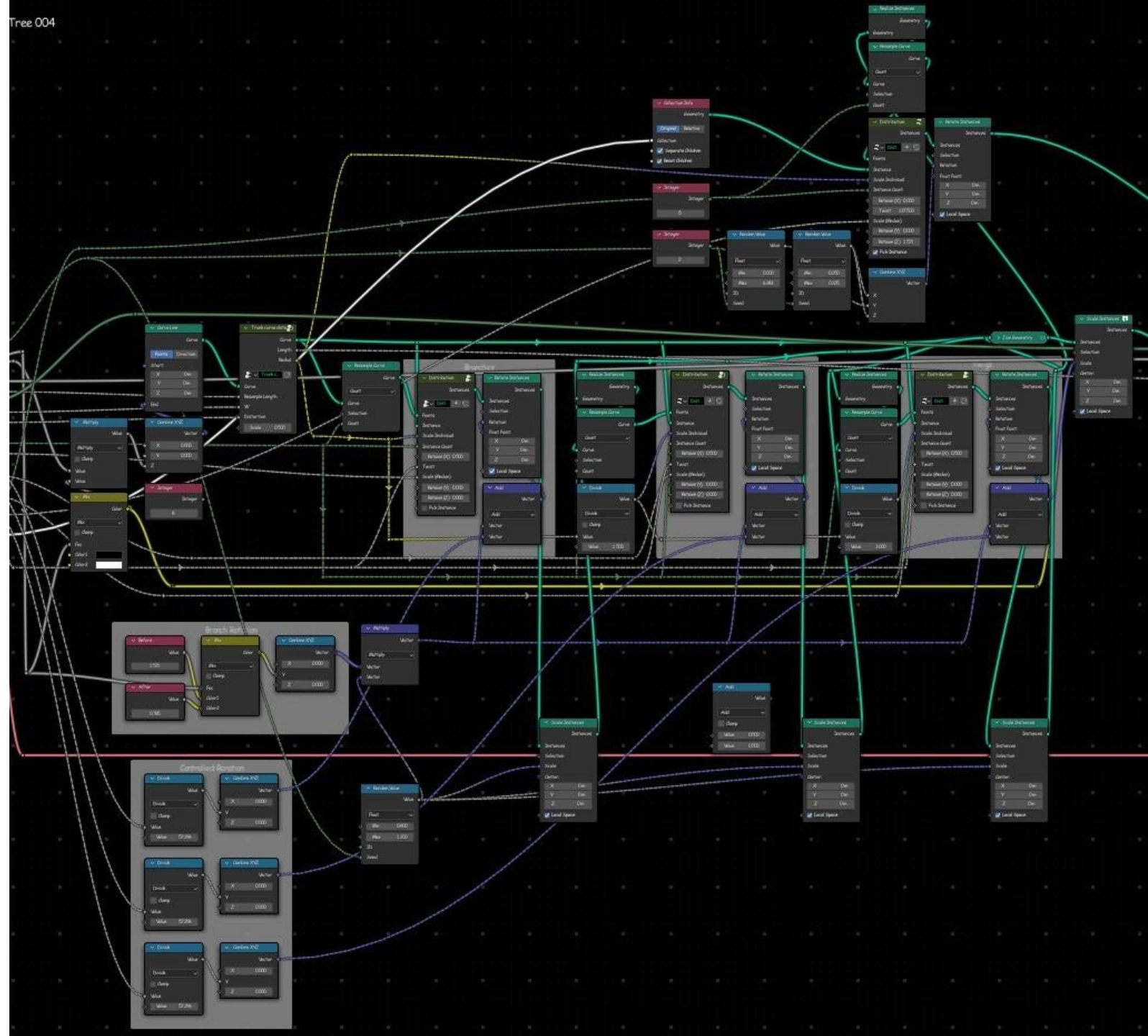


# ABHAY'S TOOL-KIT

## Guide



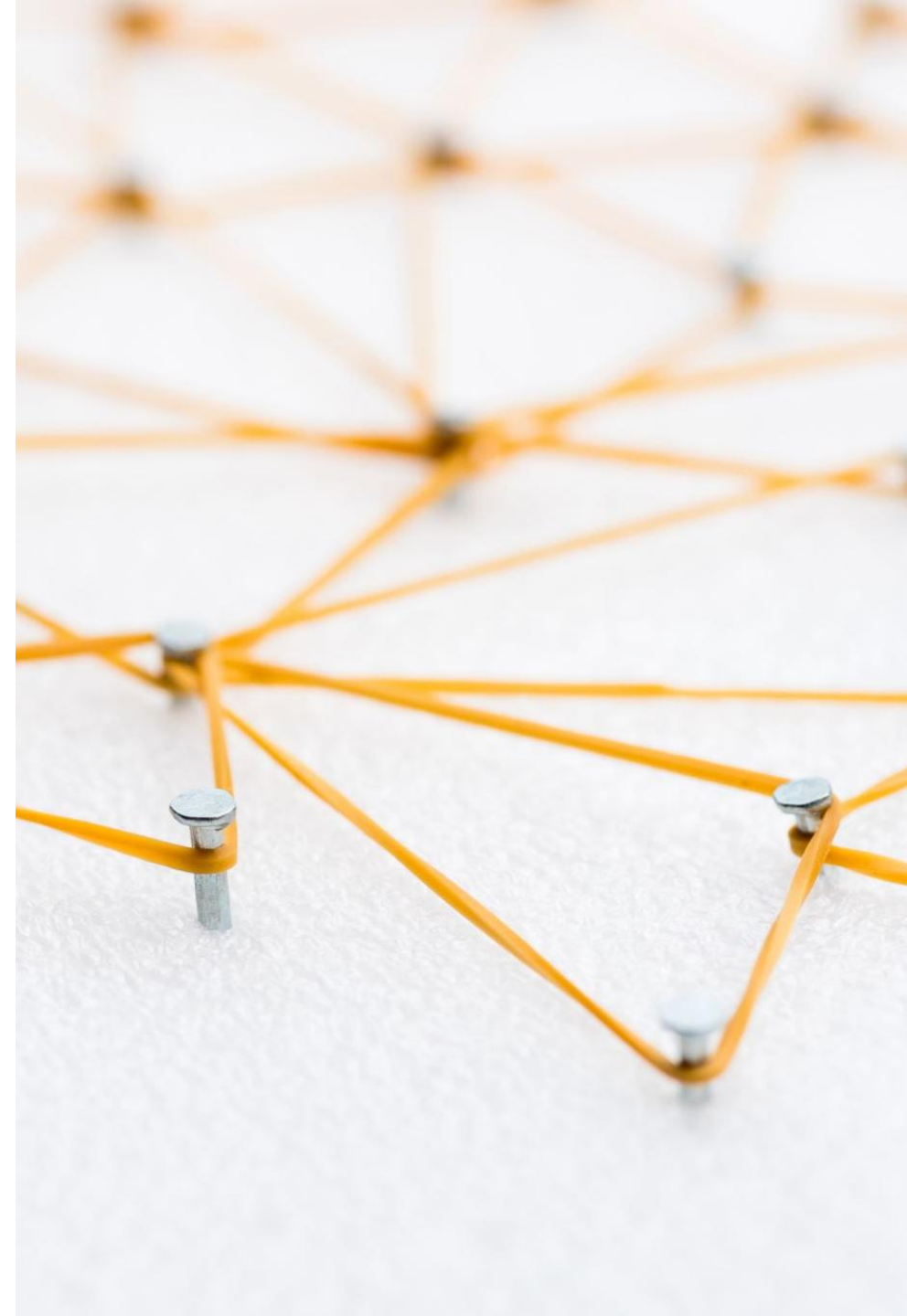
# CONTENTS

1. Introduction
2. Curves
3. Curves Primitives
4. Examples
5. Generators
6. Geometry
7. Instances
8. Mesh Primitives
9. Points
10. Selection
11. Utilities
12. Vector

# INTRODUCTION

Blender's Geometry Nodes system is a powerful tool for procedural modelling 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 addon for seamless integration into Blender's interface. And,

This is the Guide to the Nodes groups in the Tool-Kit. Here I will be explaining what these nodes groups do, how to use them, and there use case.





# 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 re organising spline topology. It rebuilds curve topology by destroying original spline organising and reassigning points to new splines based on Group ID attribute, Outputs reindexed curves with uniform cyclic state control.

Use cases:

- Spline consolidation: Merging multiple splines into single spline(assign same Group ID to all points).
- Spline fragmentation: Split single spline into multiple segments based on Group ID.
- Topology reordering: Change spline rendering order.

## Change Spline Type

Description: Dynamically converts curve spline type between poly, Bezier, catmull rom and NURBs with selection based control - enables 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 curve 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 seamless tubular surface between input profile curves. Made this from Erindale's tutorial but mine is 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-section 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 purpose.

### Curve Rebuilder (Curve Island Index)

Description: Branching curves are actually more than one curve. It convert input curves into mesh and extract the island index. Then convert back to curves with same number of points. Allowing you to have a index for curve islands. Even though those curve islands are actually more than one curve with overlapping endpoints. Automatically reversing spline direction when needed to match 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 customisable UV mapping.

### Spline Factor

Description: Generates a factor attribute along all the input curves. Also outputs normal individual factor from spline parameter as well.

### Curve Splitter

Description: Splits the desired endpoint of all the input splines into 2,3 or random branching. Very useful in creating foliage and trees has all the necessary controls for large branches into multiple small branches with randomization, split point control, opening of branches as well as direction and rotation of 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 reorganised parameters. New parameters are simplified to expose to the modifiers panel after you done building your project with it. Spline factor and spline length is also directly exposed from it for the ease of convince.

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

## Curve Arch

Description. It's a arch shaped curve useful in creating arch ways 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 some organic shapes useful in making certain types of leaves.

## Mirrored Bezier

Description: Same as 'AST\_Bezier', its just a quadratic Bezier with reorganized controls and simplified parameters. But its mirrored. Making it useful in profiling other curves. Play with its parameters once you done making your model and see the usefulness.

## Natural Spiral

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

## Nuclear Curve

Description: A curve in the shape of nuclear hazard sign. Move its center to create leaves like Livistona plant or combine it with a “AST\_Change Spline Type” to create instructing patterns and mandolas.



# 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-able cotton square without cotton that I created using nodes groups from my toolkit. It showcase 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 requires 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 so much control over shape and useful parameters.

### Ligulate-Petal

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

### Lofting with profiles

Description: A 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\_Mirrired Bezier”. Modify the parameters a little and it will go for a variety of trees with similar leaves.

### Petal 2

Description: A simple petal generator. Type that usually found in vie leaves flowers or in wild. Made using “AST\_Natural Spiral”.

### Shortest edge path optimize

Description: An example of how to optimize the geometry made using 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 randomize 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 controls over the pattern you generate.

### Pottery

Description: It is a pot Generator with Bezier input. Just give it a Bezier in the shape of 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. Otherwise it will generate without input geometry. You can also Ramesh the rock within the node. Try all the parameters for 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, 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. Very handy node while instancing things or sampling objects from 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 top plane as A', B', C' and D'. It output all the corner position of the bounding box along with its center, center of 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 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 output accordingly.

Learned this node from Erindale's Tutorial on bounding boxes.

### Bounding Grid

Description: Create 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: Rotate 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 Axis, center of rotation and angle. Factor input is where you put the mask of selected geometry.

### Self Iterate Geometry

Description: Distribute itself of its own points with each iteration

Caution: only use simple geometry with lesser vertex count. Using 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 up 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: Lets you convex Hull the heavy geometry but automatically switch to original geometry when you render. Very useful when dealing with Heavy instances. Just plug it in between your geometry that you want to instance and Instance on Points node.



### Transform Geometry +

Description: Basic Transform Geometry node but with field inputs, meaning you can transform geometry selectively or input field values. Also have a center input which change the center of transformation. Default is 0,0,0.

# INSTANCES

The background features a dark gray grid with thin white lines. Scattered across the grid are numerous 3D cubes of varying sizes and orientations. Some cubes are simple, while others are more complex, resembling L-shapes or T-shapes. A vertical white line is positioned to the left of the list of node groups.

## 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: Instance on input geometry with perfect alignment to face normal and center of the input geometry.

### Alternative Distribution

Description: Distribute instances on a curve in alternate direction or in opposite direction like leaves on twigs of some plants, useful for creating foliage and plants.

### Bounding Box Instances

Description: Creates bounding boxes for instances.  
Learned in from Erindales's tutorial on Bounding Boxes.

### Circular Distribution

Description: Distribute 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 it from Erindale's tutorial on Instance packing.

### Collection Info Custom Index

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

### Instance Packer

Description: For packing objects that are not same size perfectly spaced. According to their size. Need collection info + to pair with it.

### Instance on Edges

Description: Distribute instances on edges. 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 place holder for instancing something.

### Icosahedron

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

### Pyramid

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

### Slicing Circle

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

### Tetrahedron

Description: A tetrahedron generator, useful as 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: Add scales on the surface of the input geometry.

### Damage Geometry

Description: Damage/Break-ups the input geometry to a old withered look.

### Fractal Shaper

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

### Mirror geometry

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

### Points to Lattice

Description: Input a bunch of points and create a kind of lattice out of them. It has very specific use-case like abstract art or creating a plant in certain shape.

### Surface Cobble

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



## Terrain Displacement

Description: Displace a height map into a terrain with precise control and subdivisions setup. Very useful node. I use this all the time for all of my Terrains.

## Trim Geometry

Description: Trim the input geometry on X, Y or Z axis. From both a sides just like a trim curve node. It uses a Boolean node, so its not super fast on heavy meshes.

The background of the slide features a dark, textured surface. On the left, there is a network diagram consisting of several circular nodes connected by thin, dark lines. On the right, there is a tangled, dark wire or string. A single pushpin is pinned to the wire on the right side. The word "POINTS" is written in large, white, sans-serif capital letters in the center-left area.

# POINTS

## List of Node Groups

- i. Distribute Points in Curve
- ii. Phyllotaxis
- iii. Points to spline

### Distribute Points in Curve

Description: Distribute points in the area that input Curve covers, Whether it is cyclic or not. Non-cyclic curve will have points distributed in the concave area/section. You can control the density of points, randomize them and control probability to spawn.

### Phyllotaxis

Description: Points distributed in Phyllotaxis pattern. Useful for creating organic projects like plants and trees specially flowers.

### Points to spline

Description: Creates a spline from the input points. Little different from the base node. You can choose spline type and have option for inbuilt circular weight (Basically it creates a radial gradient from the world center and use 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: Expand your Boolean selection by one vertex in all directions. Learned it from Erindale's Tutorial on geometry selection.

### Mesh Proximity

Description: Creates 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: Output 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: Select the nearest point from the input object (or empty object). Useful in selecting specific points in the input geometry.

### Select by Direction

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

### Select by Normal

Description: Select based on X,Y and Z Direction of the Normals of the geometry.

### Select by Position

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

### Select by Index

Description: Select based on the Index occurrence. For e.g. select every 2<sup>nd</sup> vertex (even). Selection can be inverted by changing the “Equals to” input value. Also allows 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: Accumulate the position, tangent and normal of input curve on any position/point on input curve. Control that point using length input.

### Degree

Description: A value node of degree input instead of typical radiant value.

### Flip Indices

Description: Flip the Grid(2D) Index. For e.g. if your geometry has two indices going on different axis. You can switch them using this node.

Learned it from Erindale's tutorial on Index.

### Fold Attribute

Description: Fold the input attribute at the input value. For e.g. if input attribute ranges from 0 to 1. and the value input set to 1. it will turn the range from "0 to 1" to "0 to 1 to 0". If value input set to .5 then "0 to 1 to 0 to 1 to 0".

### Grid Index

Description: Project a grid of index on X and Y coordinates. This custom index can be used to sampling.

Use-case joining two ends of two pipes/cylinder with different index. Joining them with their index will create a messy geometry because of the misalignment of their indices. Assign this custom index to both the end loops/circles then join them to create seamless joints.

### Map Attribute.

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

### Store Edge angle

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

### Value ++

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

### View Instance Attribute

Description: Visualize a float attribute of 'Instances' with points through weight output. Radius controls the point radius. Only for viewing the attribute to verify or check it.




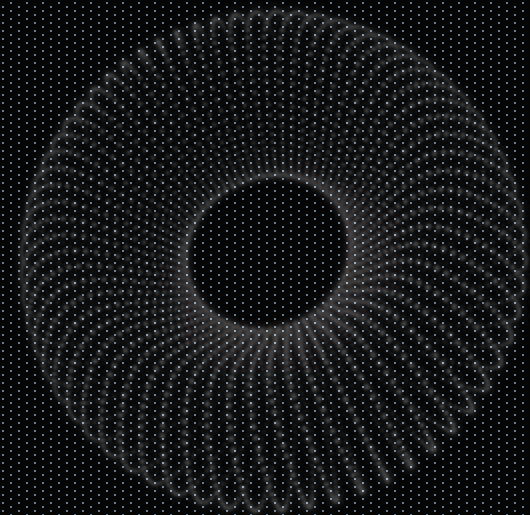
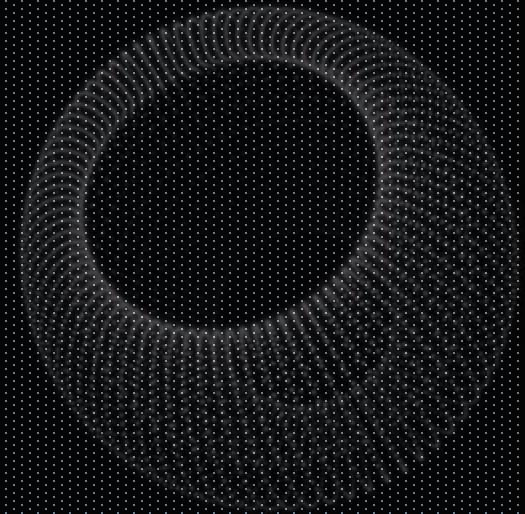


# VECTOR



## 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: Store a UV projection from all the sides of input geometry. Suitable for geometry with sharp corners like hard surface modelling.

UV attribute name is "Cube\_Projection". Go to shader editor and add a attribute node and set attribute name to "Cube\_Projection".

### Cylinder Projection

Description: Stores a UV projection, unlike "AST\_Cube\_Projection" this UV is three segmented. Two side Top and Bottom and a third sideways loop, just like a cylinder. Suitable for cylinder like geometry like pillars and tree trunks.

UV attribute name is "Cylinder\_Projection". Go to shader editor and add an attribute node and set attribute name to "Cylinder\_Projection".

### Landscape Projection

Description: Stores a UV projection from all the sides like "AST\_Cube\_Projection" , but Top and Bottom sides have very sharp edges and only show up in near completely flat areas. Making it ideal for texturing landscapes. Using default UV on landscapes creates angled textures which looks very ugly, It fixes that. You can control on what angle, top and bottom UV should show up. You can flip X and Y coordinates with alternate XY mask input.

UV attribute name is "Landscape\_Projection". Go to shader editor and add a attribute node and set attribute name to "Landscape\_Projection".

### Map Vector Attribute

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

### Ping Pong Vector

Description: Ping Pong a Vector input similar to Fold Attribute.

### Sphere Projection

Description: A projection that uses two 'Arcs' one going horizontally and 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 shader editor.

### Triplanner Projection

Description: Stores a UV Projection from all the sides with smoother fall off. Suitable for smoother geometry without sharp corners. On sharp geometry result will be unexpected. You can also control the falloff angle.

UV attribute name is "Triplanner\_Projection". Go to shader editor and add a attribute node and set attribute name to "Triplanner\_Projection".

### XYZ arc

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