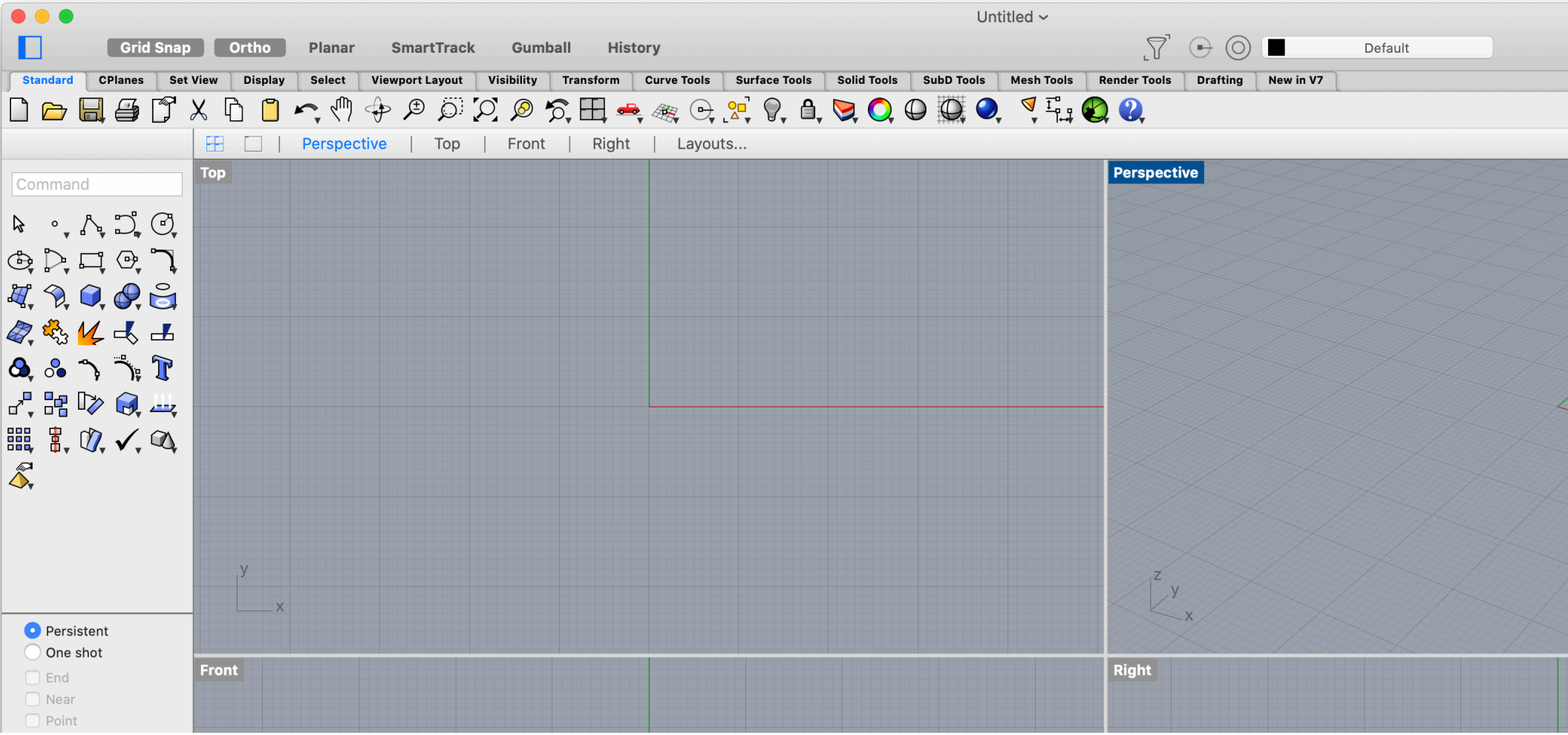
Computational Fabrication

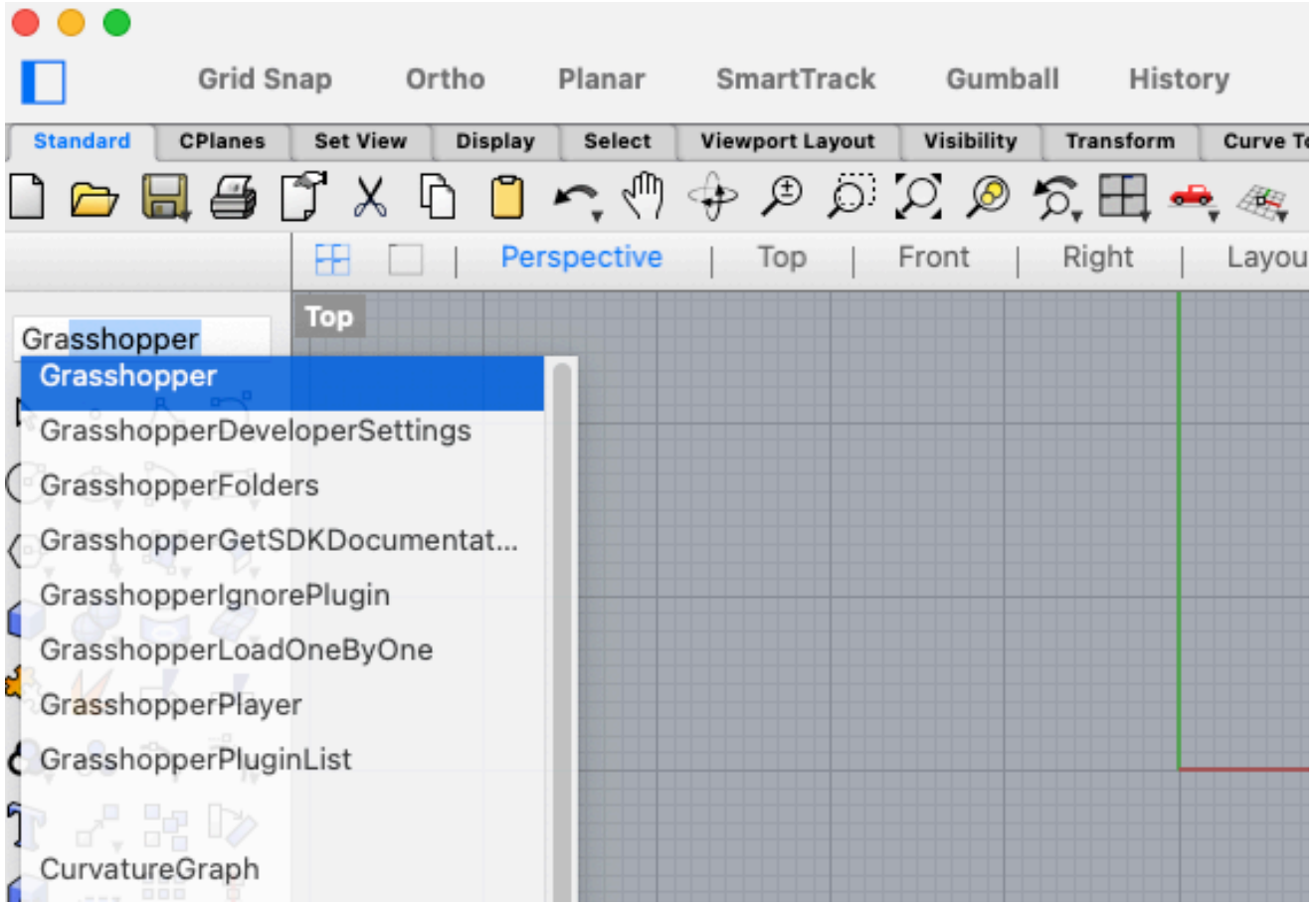
CS 491 and 591
Professor: Leah Buechley
https://handandmachine.org/classes/computational_fabrication/

Rhino +
Grashopper

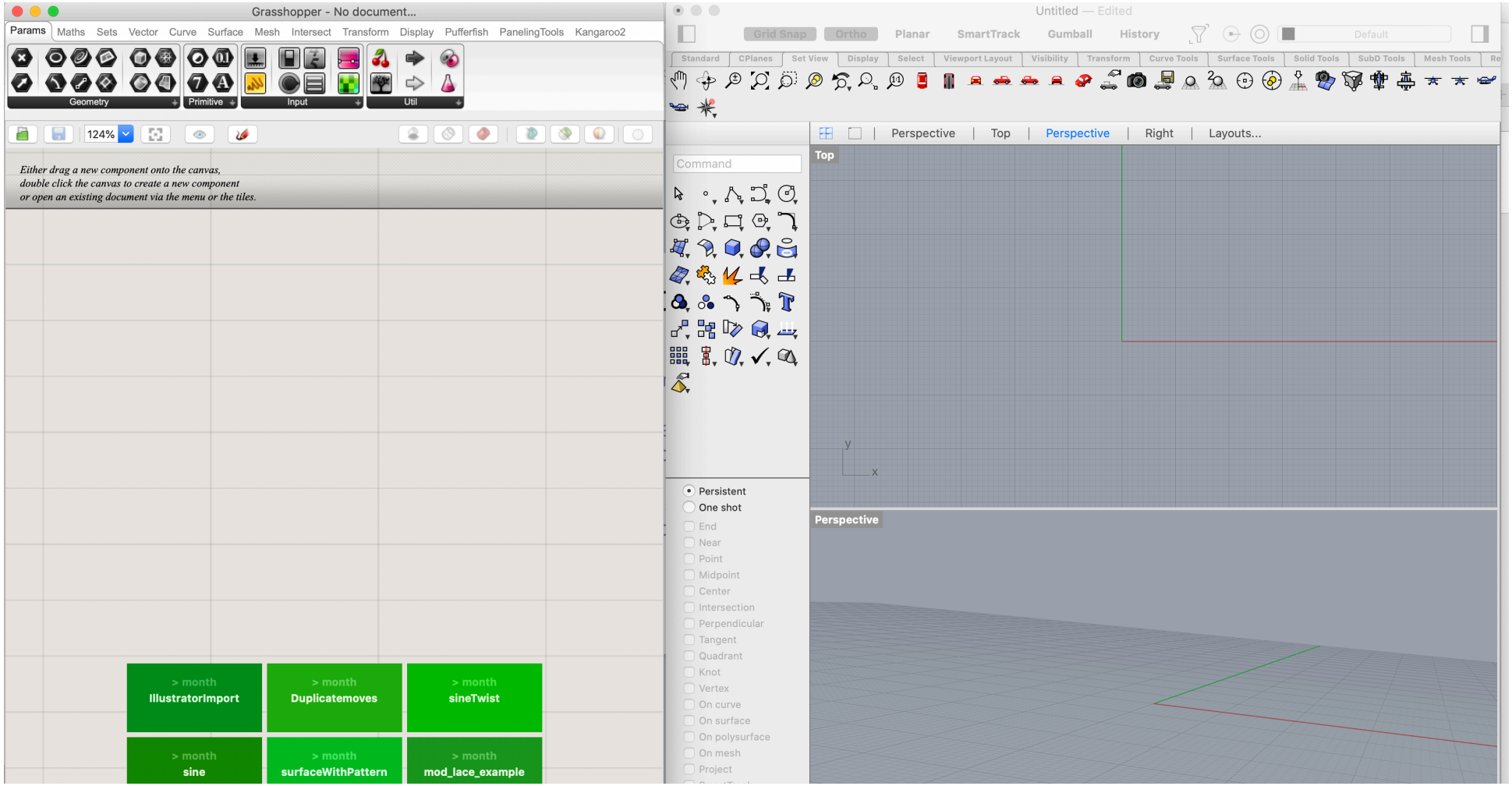
open up Rhino



Open Grasshopper



Set up so you can see both applications



Rhino

NURBS based 3D modeling software

NURBS: mathematically precise way of defining shapes (as opposed to meshes of polygons, which are approximations)

Cheaper than many software alternatives (ie: AutoCAD)

Powerful SDKs and API: https://developer.rhino3d.com/

Many plugins and an active developer community

Reminders and Tips

Extremely helpful to use a mouse instead of a trackpad! Bring a mouse to class.

Very useful to have two screens,
one for scripting (Grasshopper) and one for visualizing (Rhino).

If you don't have two screens, use two side-by side windows,
one for scripting (Grasshopper) and one for visualizing (Rhino).

Grasshopper

A visual programming language for Rhino

A popular Rhino Plugin. Developed independently. Now part of the official Rhino Installation.

Widely adopted by architects and designers.

Can be integrated with text-based scripting. We'll use Python

Has its own SDK, plugins, and developer community

Grasshopper (GH) Overview cont.

- A visual data flow programming language
- Code blocks generate and manipulate Rhino geometry
- Grasshopper files (.gh) are separate from Rhino files.
- A grasshopper file always needs a Rhino file to execute. Geometry is generated in the Rhino file. This Rhino file can be temporary. Think of it as your GH console.

Grasshopper Documentation

Open and bookmark:

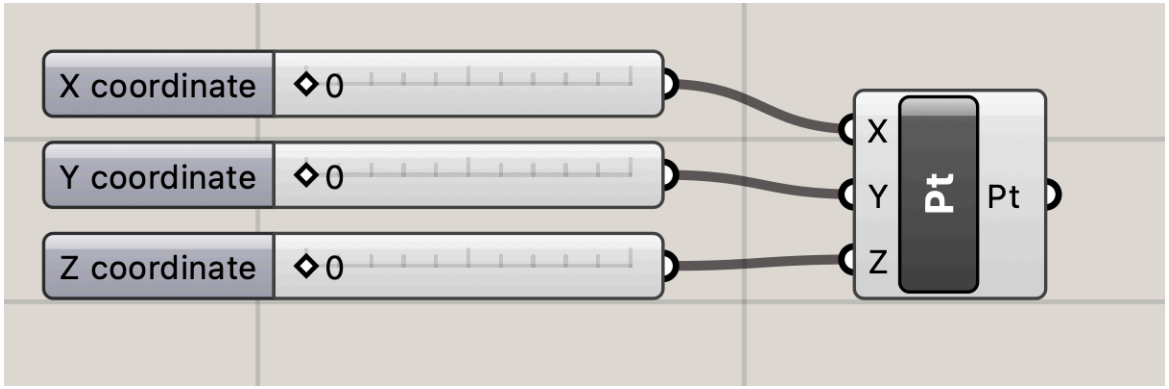
https://grasshopperdocs.com/categories/core-grasshopper.html

Today: 3D Modeling in Grasshopper

Camila's modeling tutorial, part 1:

- Points
- Lines and Polylines
- Curves
- Circles and Ellipses
- Text

Points and Numbers

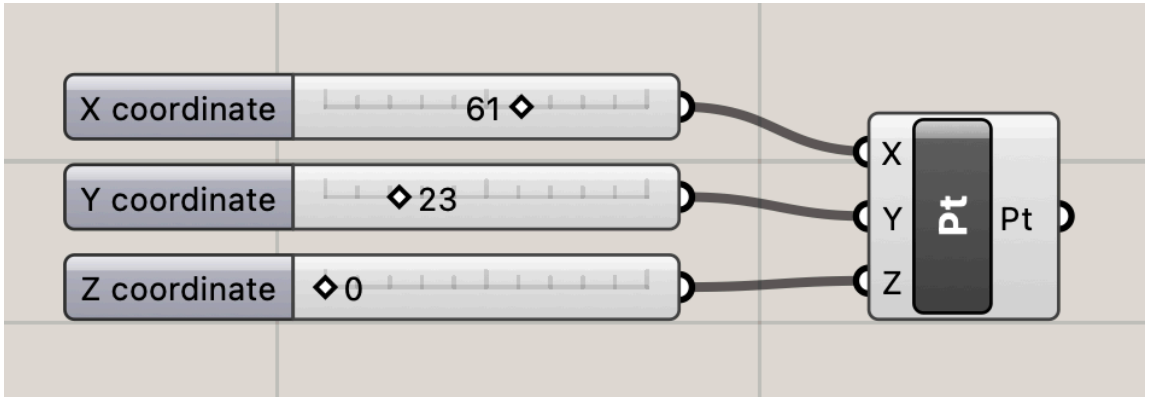


Number sliders ConstructPoint

GH code Rhino Output

https://grasshopperdocs.com/components/grasshoppervector/constructPoint.html

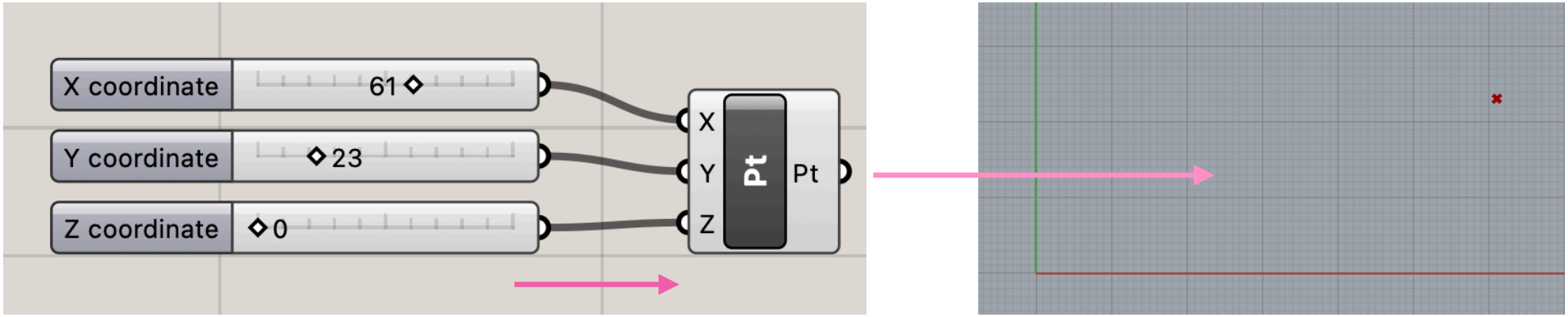
Grasshopper = Code + UI



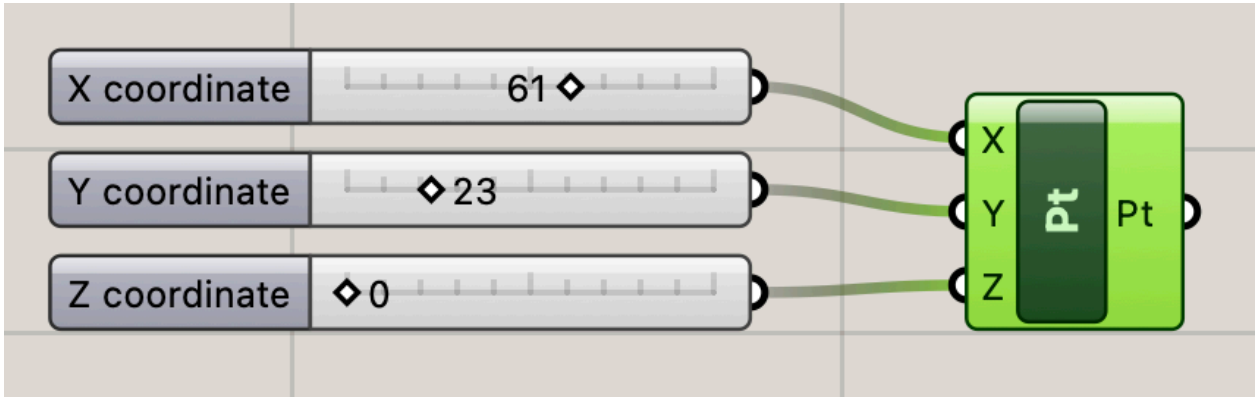
Change GH

Dynamic Rhino Output

Data Flow

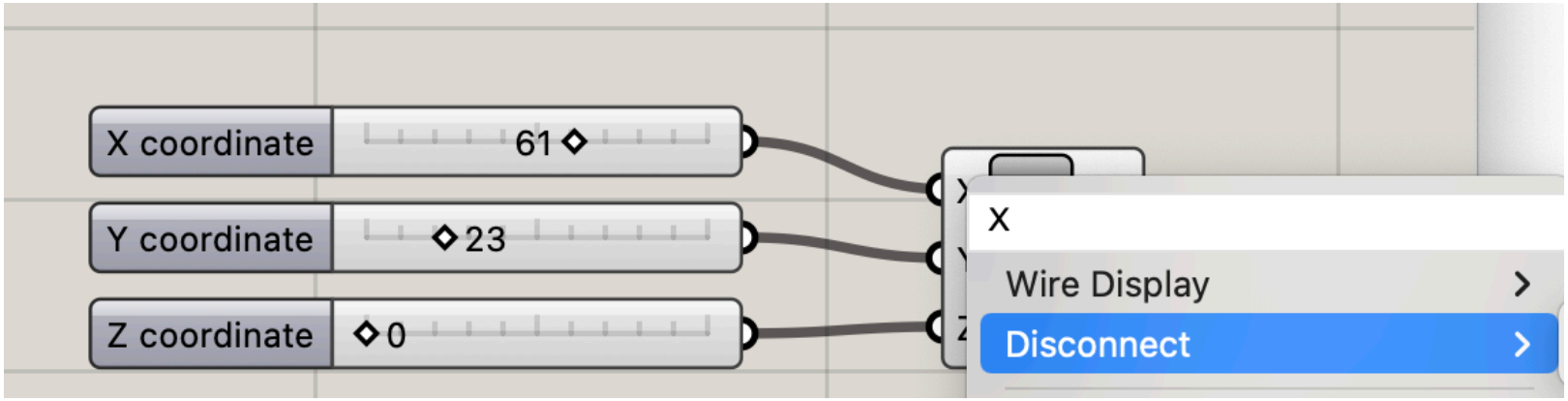


Selecting



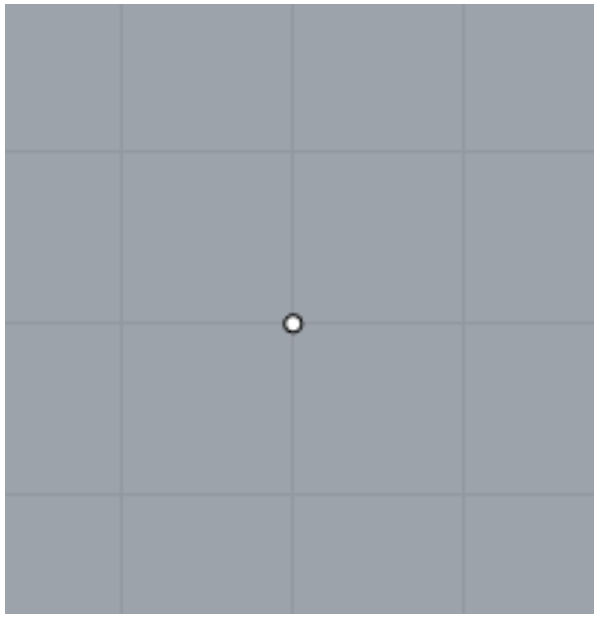
Selecting GH block, turns Rhino geometry green.

Disconnecting

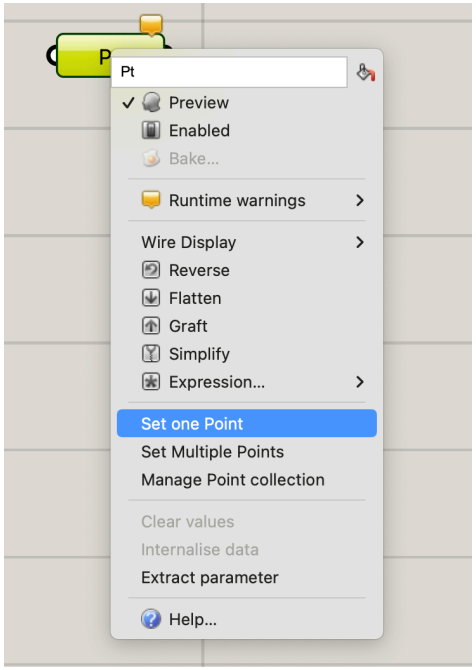


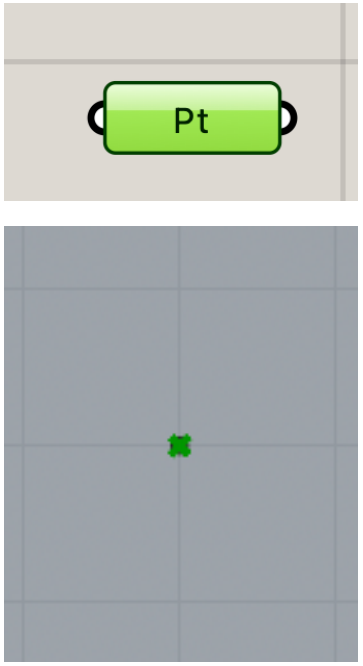
Right click on input to disconnect.

Points cont.



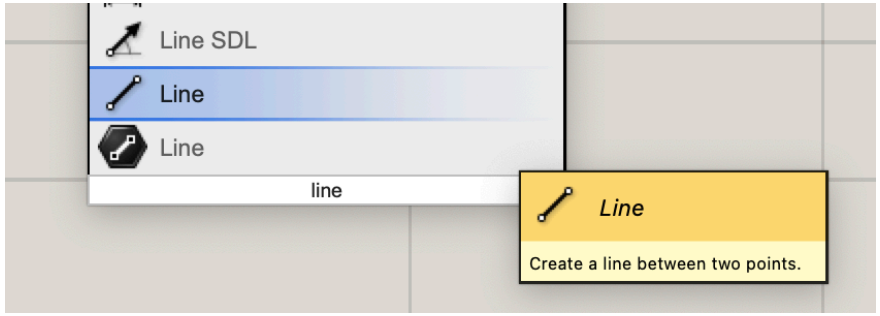
Create Point in Rhino

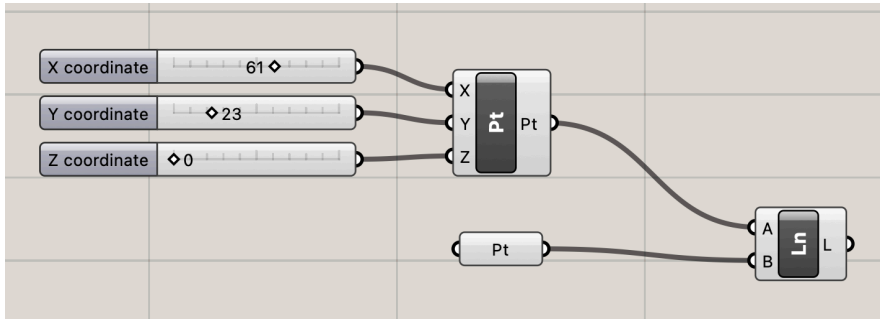




Associate with GH block

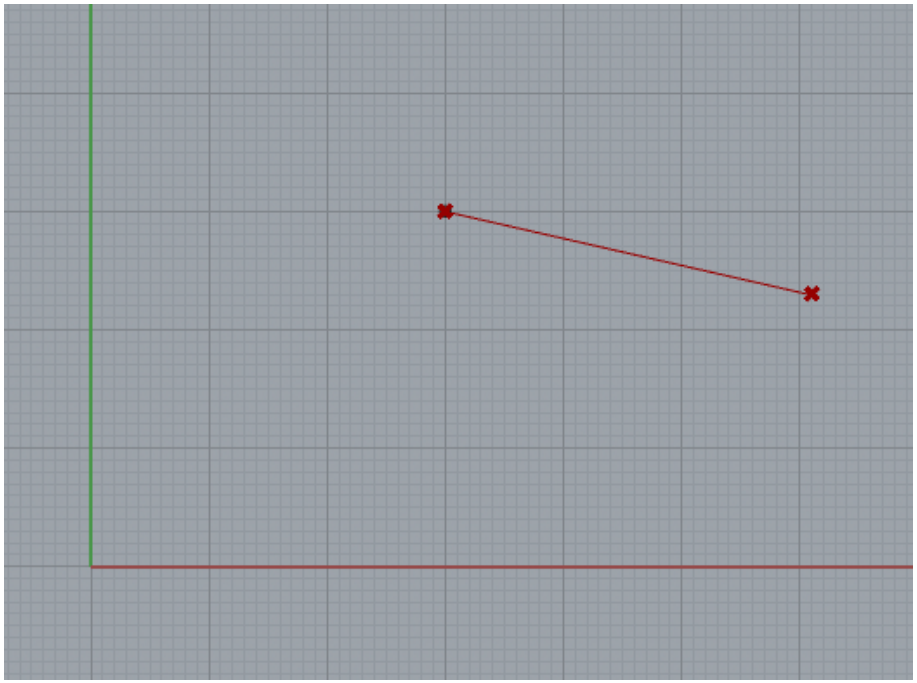
Lines





Line

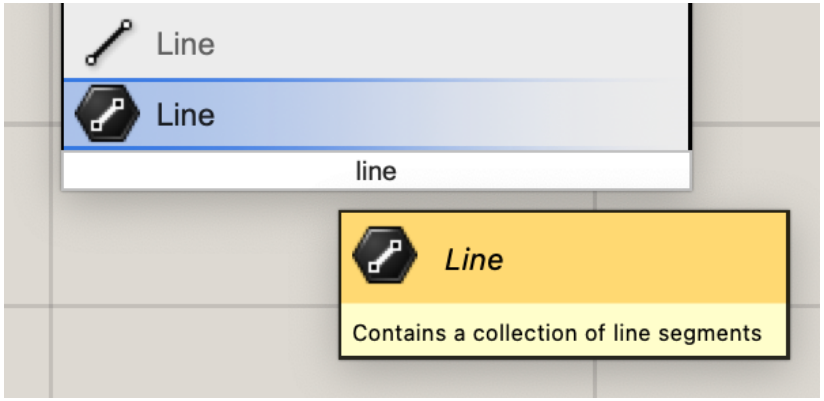
GH code

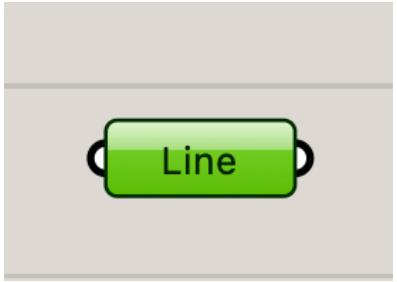


Rhino Output

Lines

Create Line in Rhino



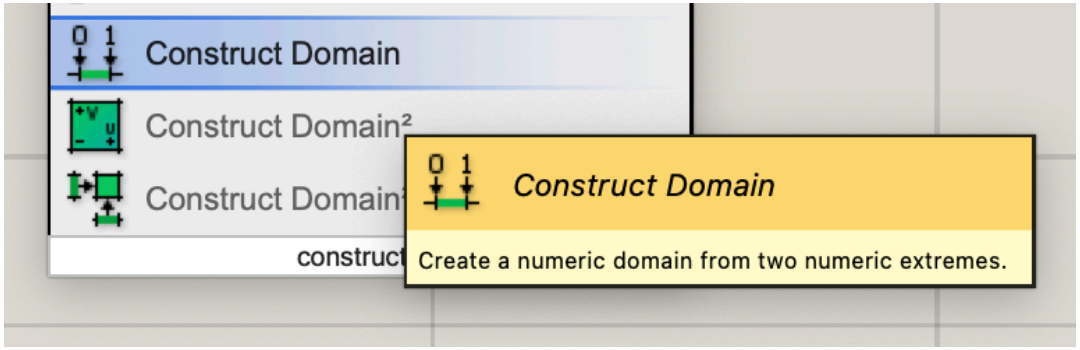


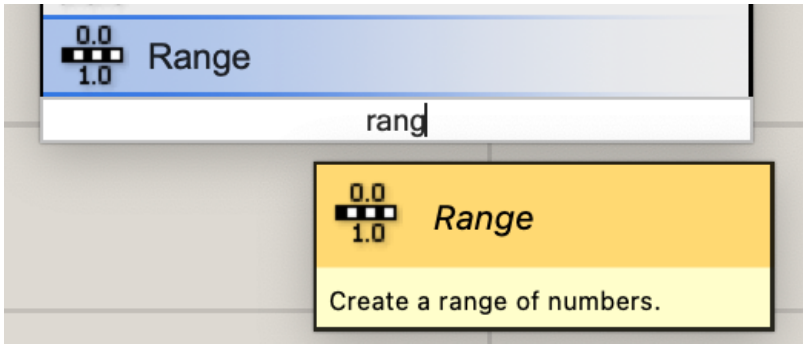
Associate with GH block

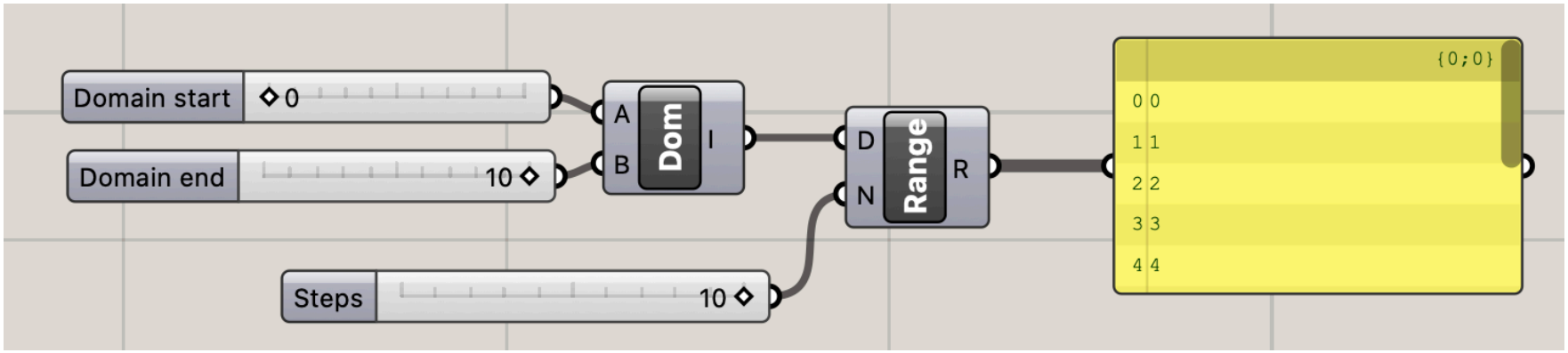
questions?

Polylines

Domain & Range







Construct Domain

outputs a domain object
domain has start of A and end of B

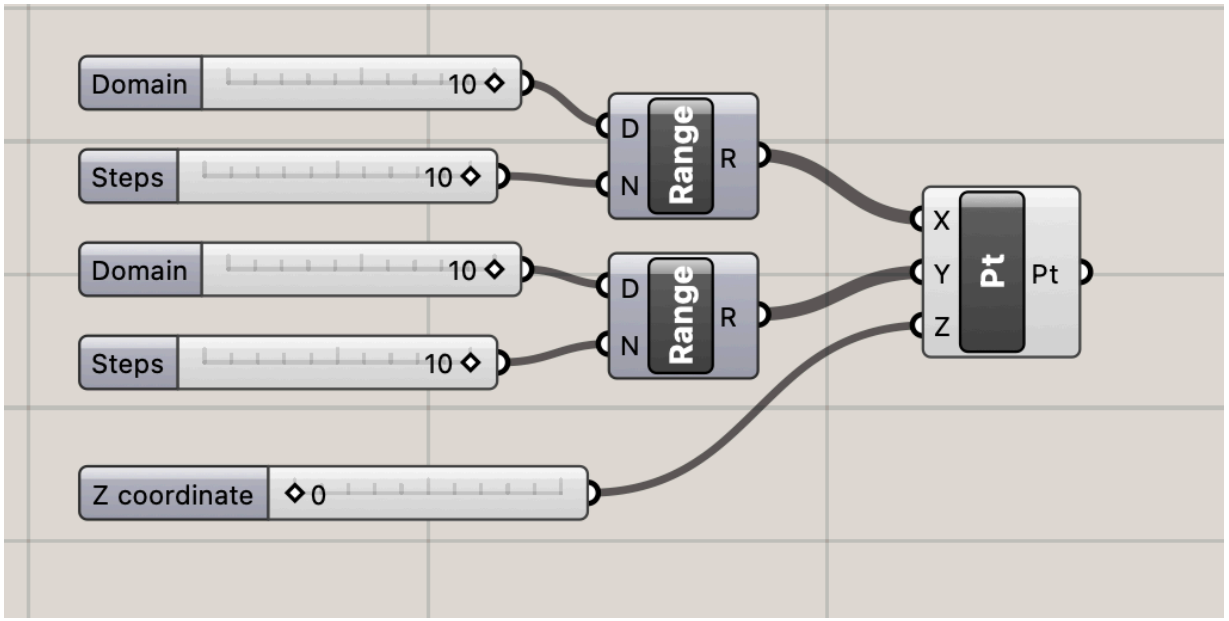
Range

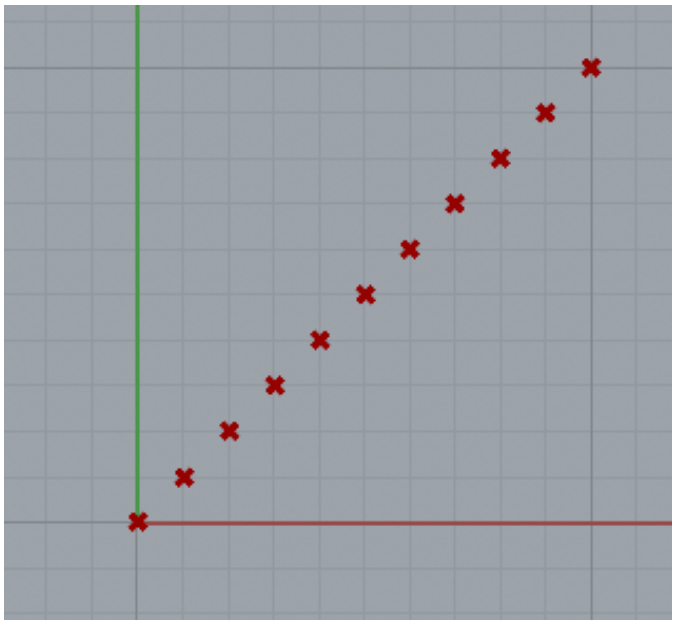
outputs a list of numbers
in domain D
N = number of items

Panel

lets you see output

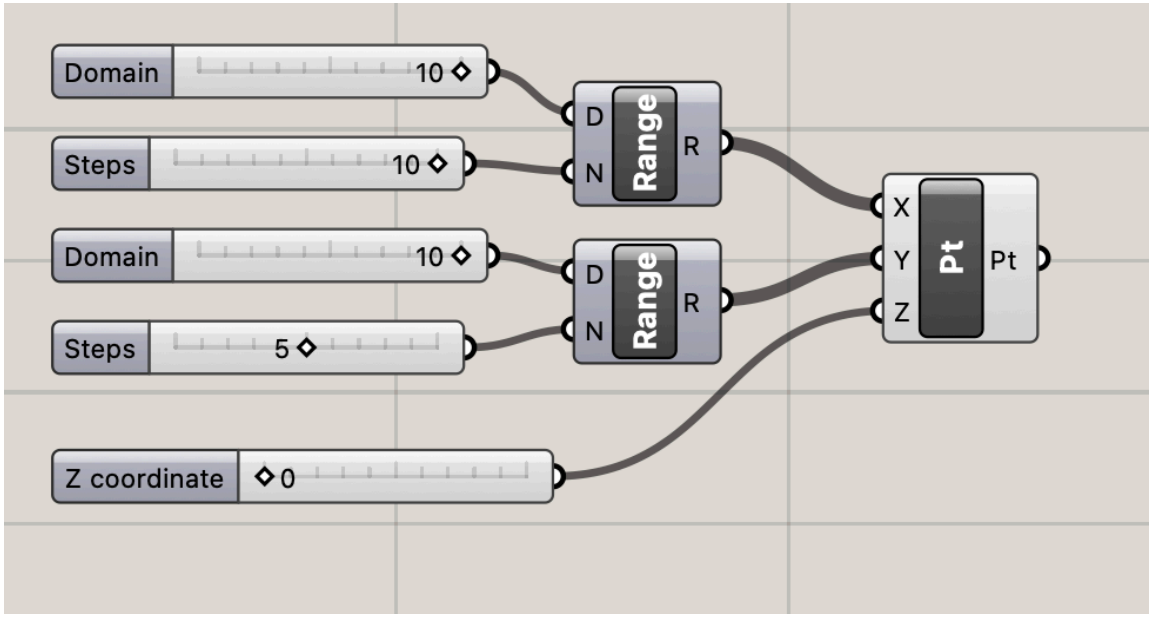
Range + Point

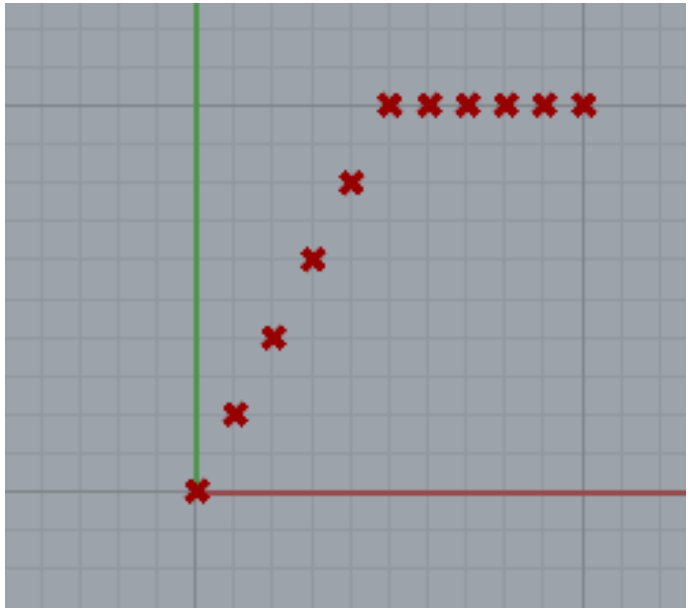




Note: Grasshopper does parallel execution for all inputs!

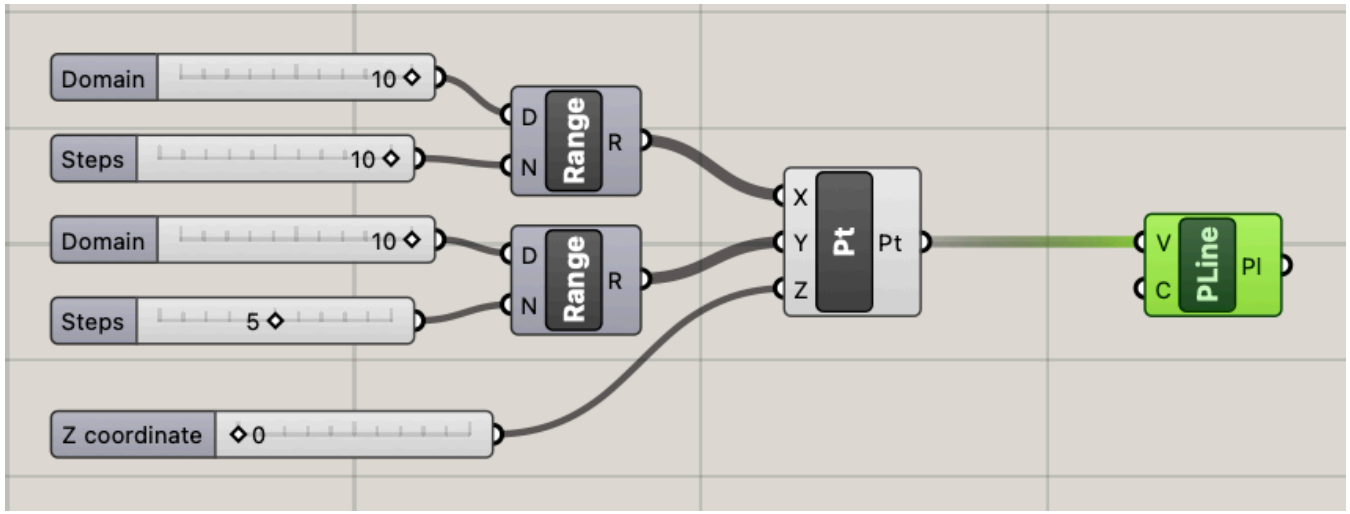
Range + Point





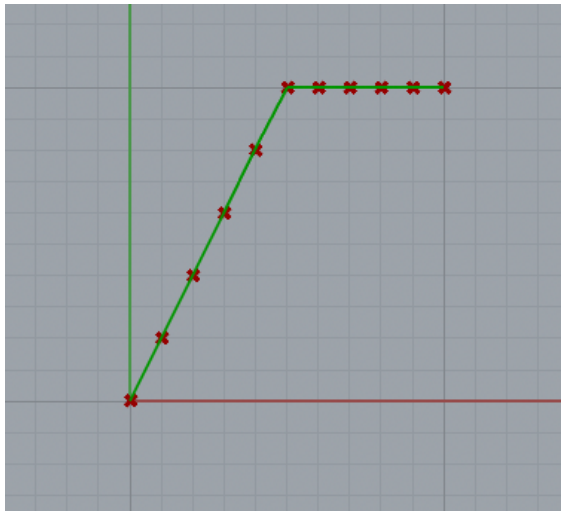
Dynamic

Polyline



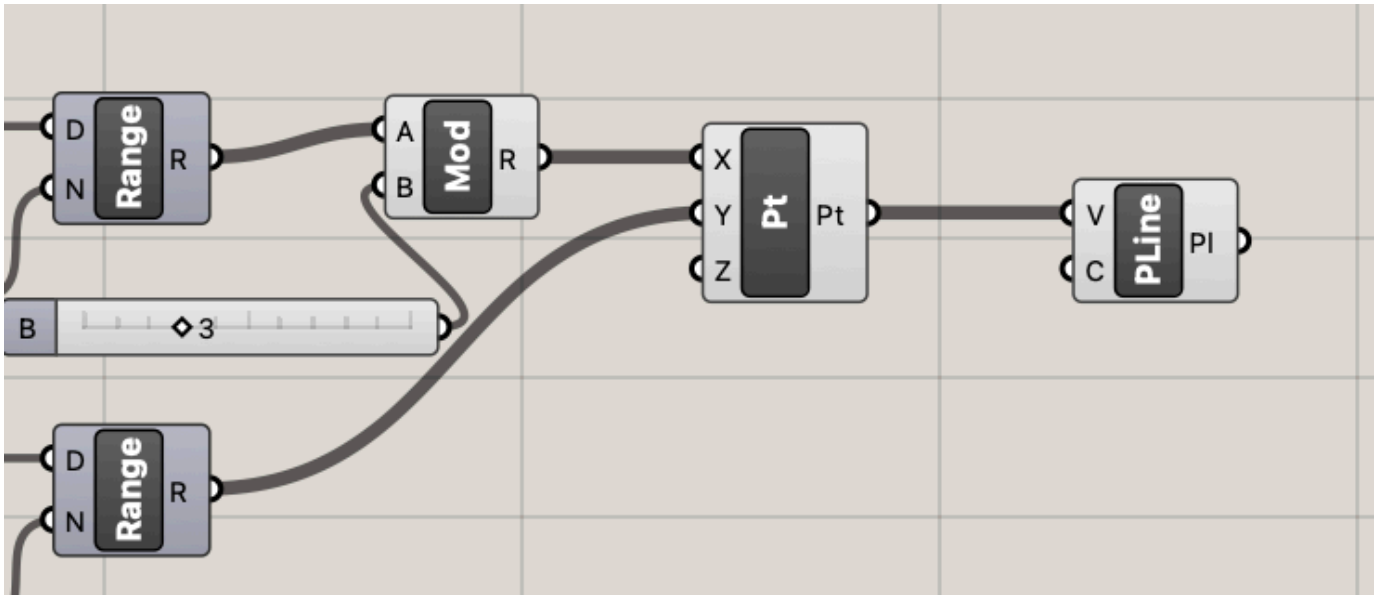
Range Points PolyLine

GH code

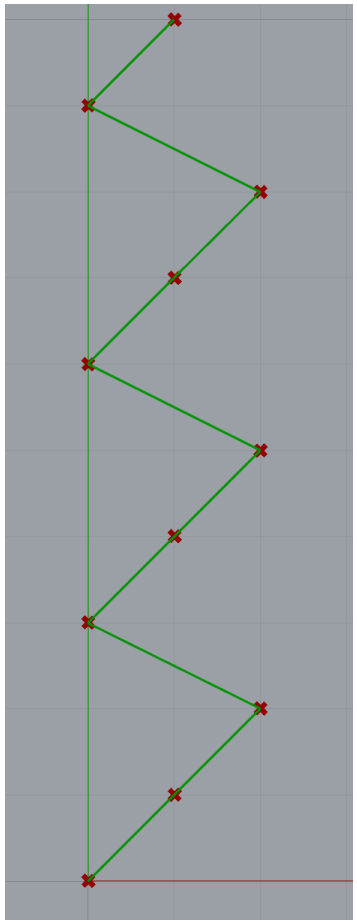


Rhino Output

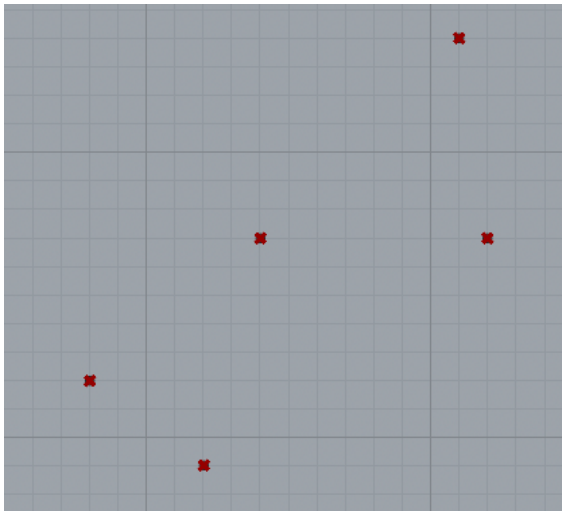
Play with % (Mod) Operator



Range Mod Points PolyLine

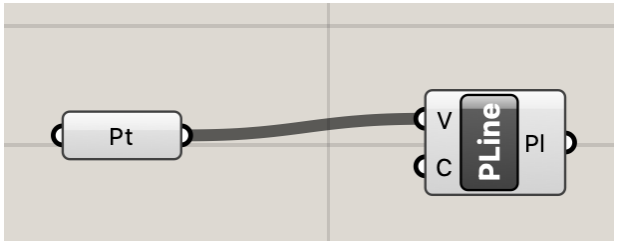


PolyLines

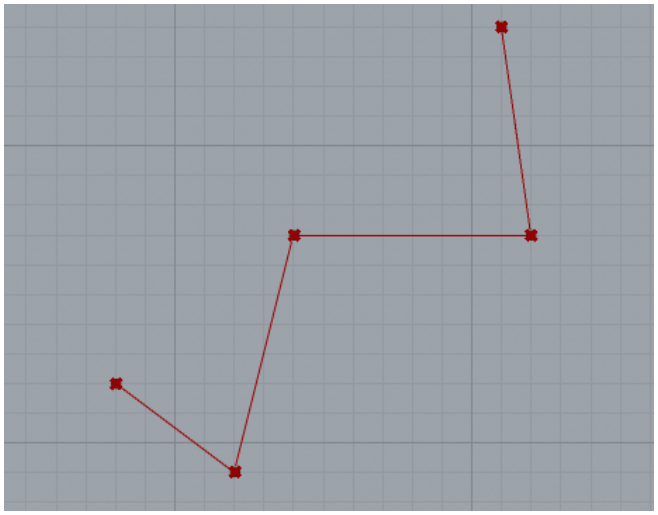


Create Points in Rhino

Associate with GH Point block



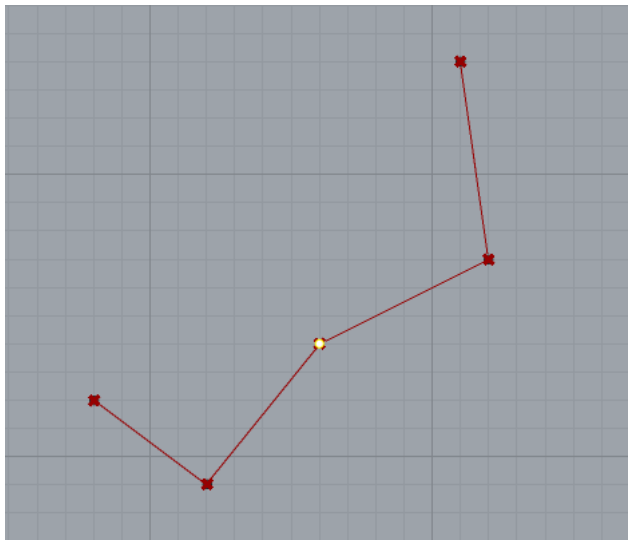
Use as PolyLine Input



Rhino Output

GH Responds to Rhino Edits

Move Point in Rhino &
PolyLine in GH Updates

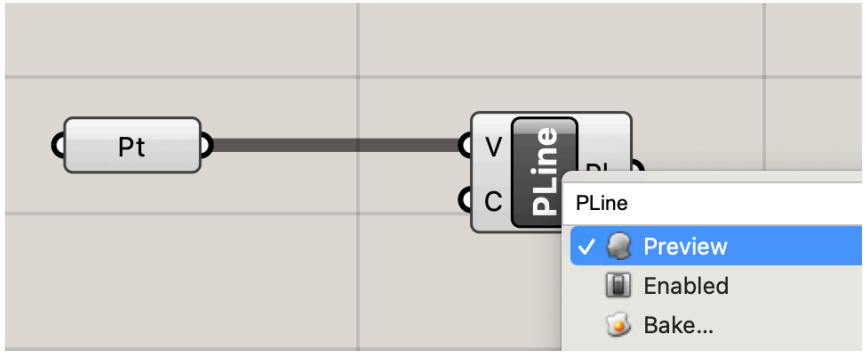


Rhino Output

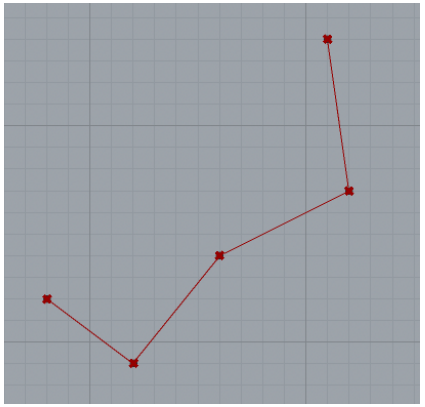
questions?

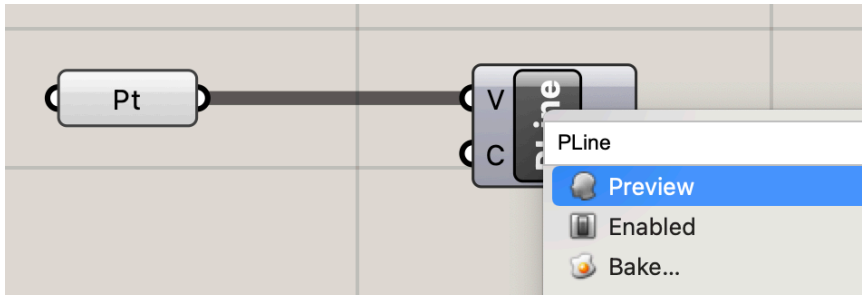
Curves

Hide GH Outputs in Rhino

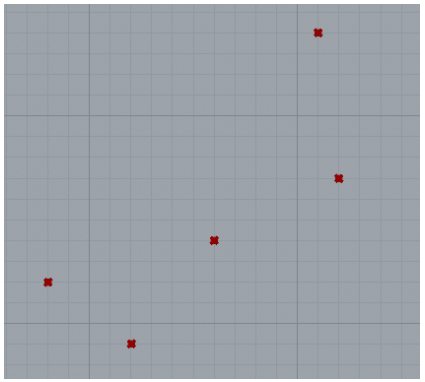


Preview On

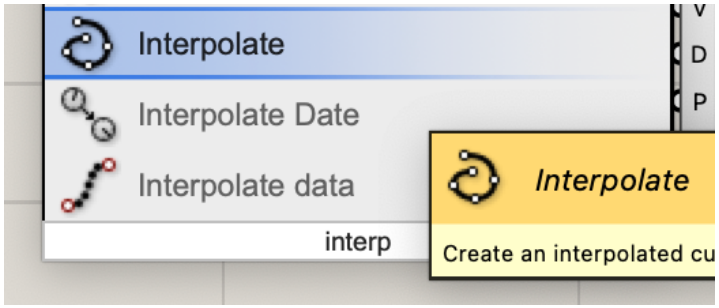


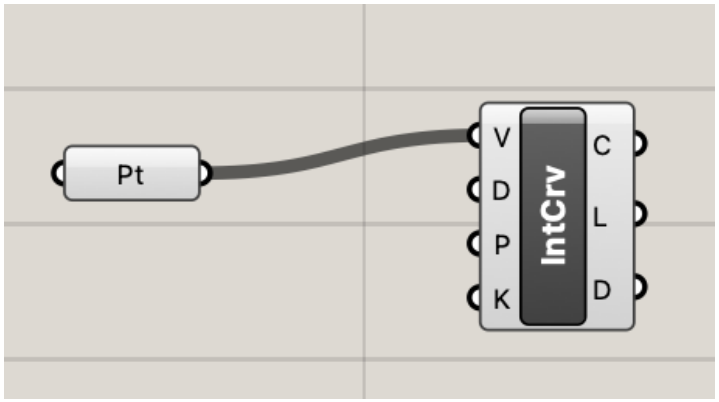


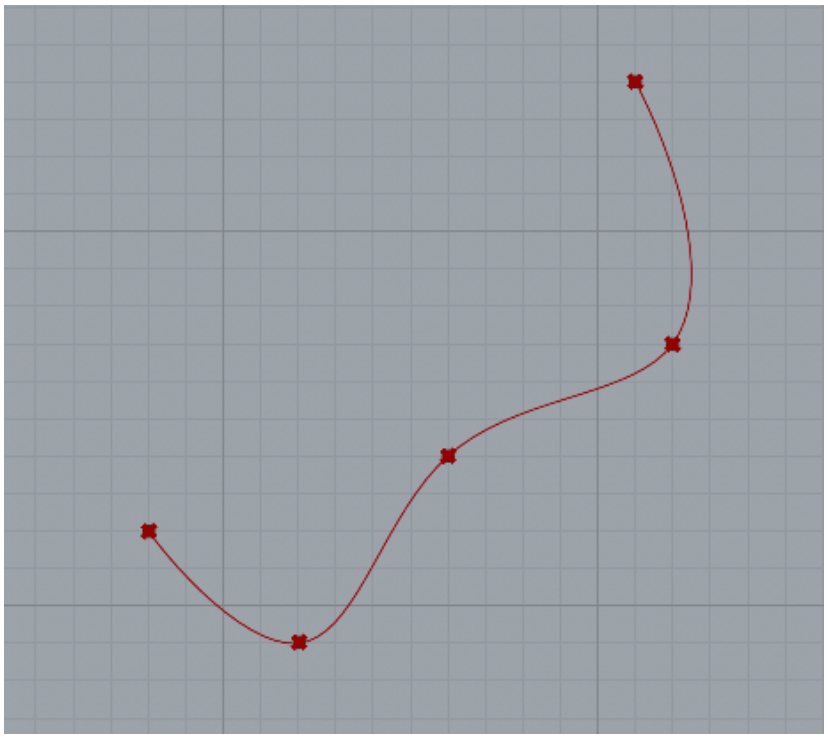
Preview Off



Interpolate Curve





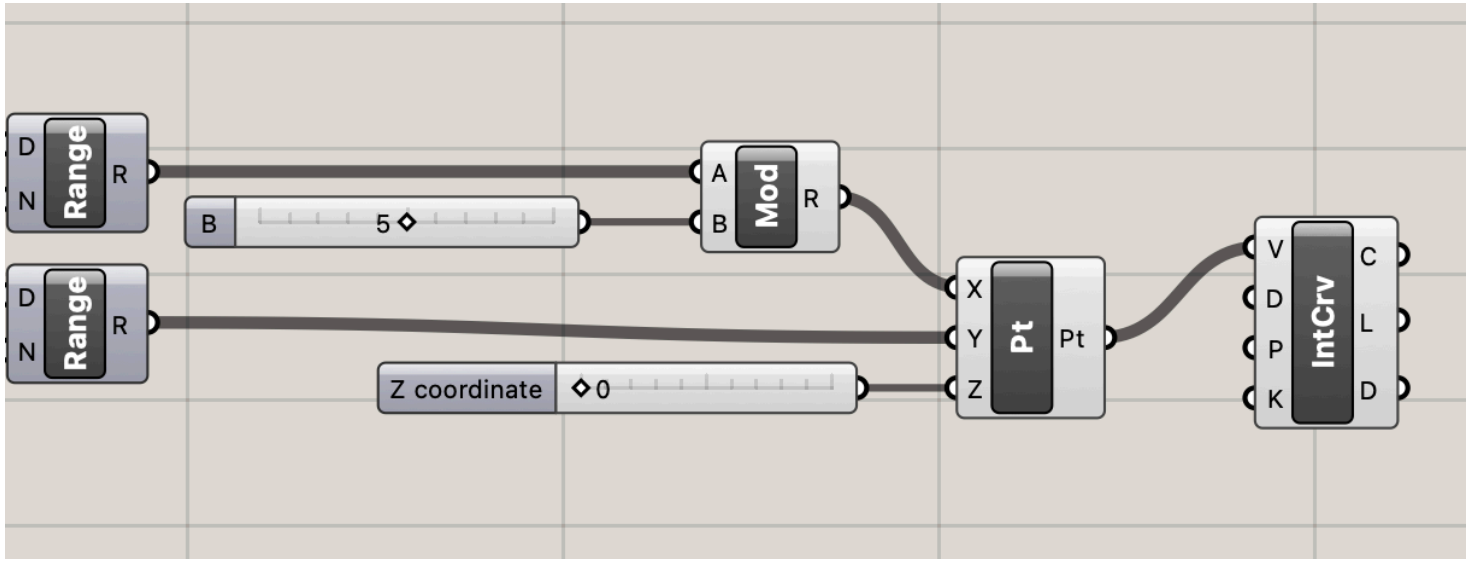


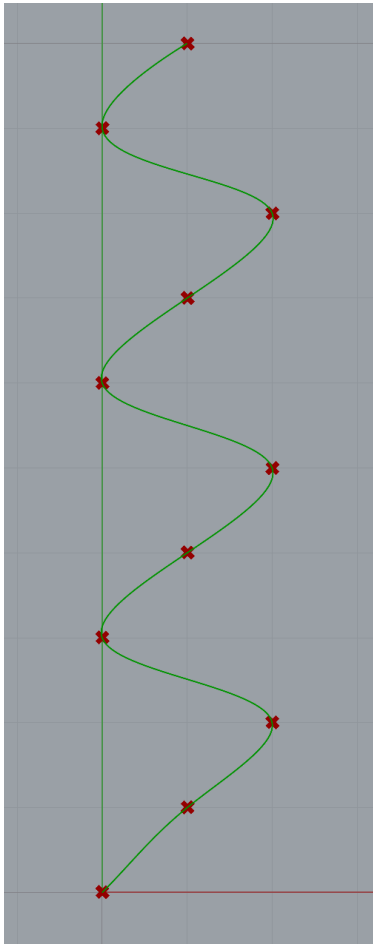
Points

Interpolate

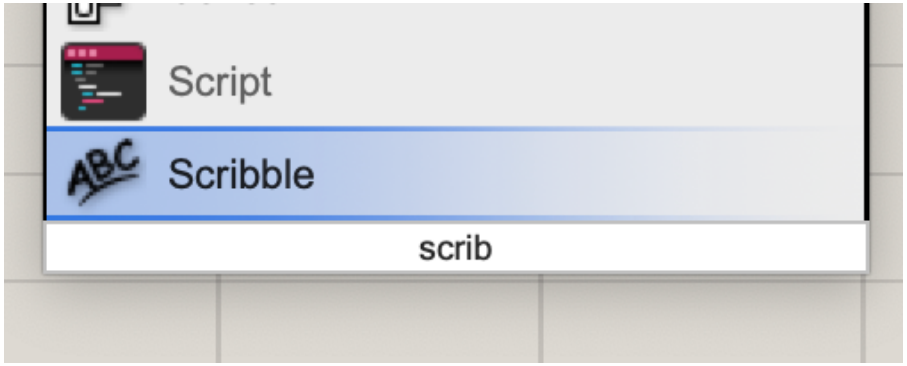
Rhino Output

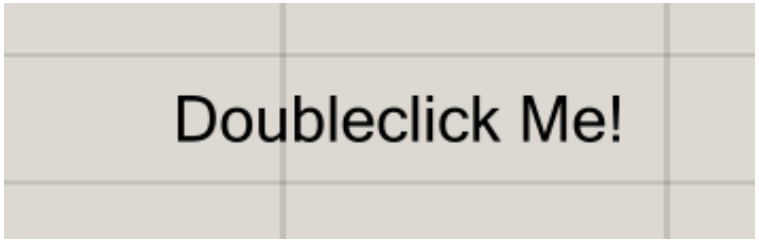
Play with % (Mod) Operator



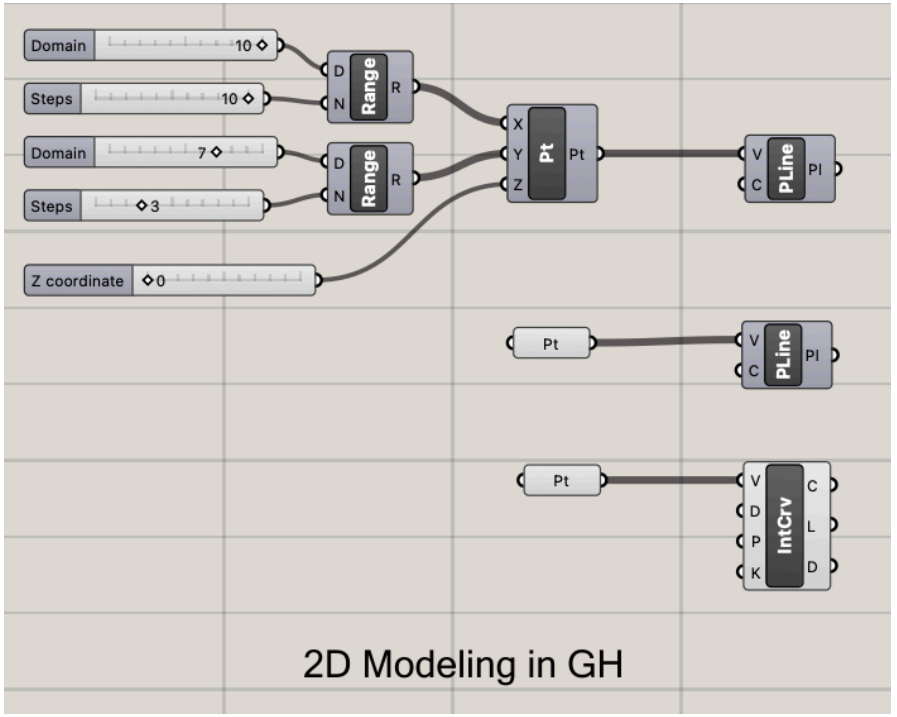


Grasshopper Comments

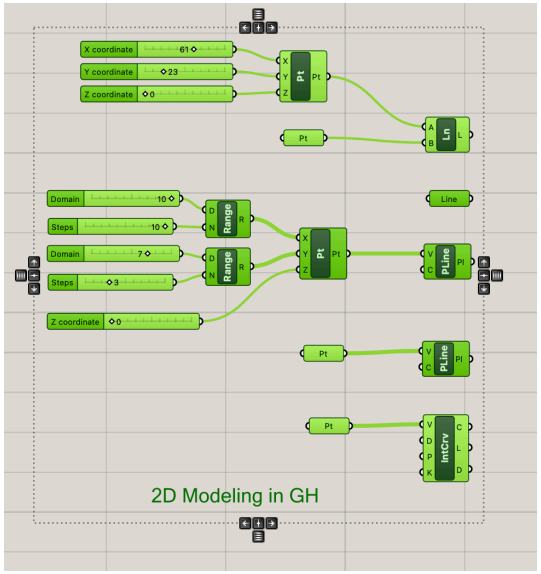




Scribble

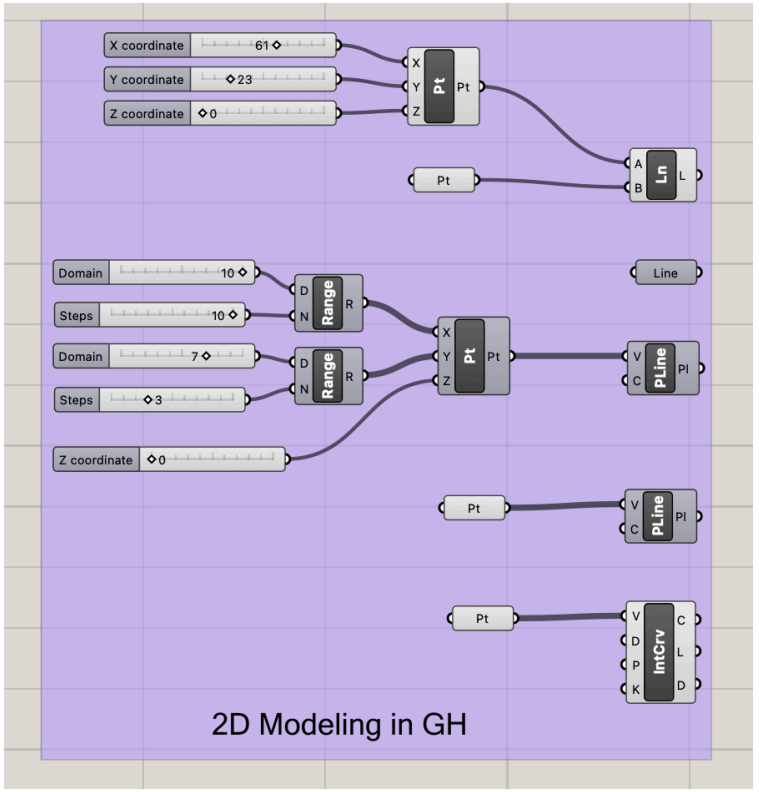


Grouping & Organizing



Select

Group
Edit—> Group



Result
No logical impact, just visual

2D Modeling in Grasshopper

Rhino command:

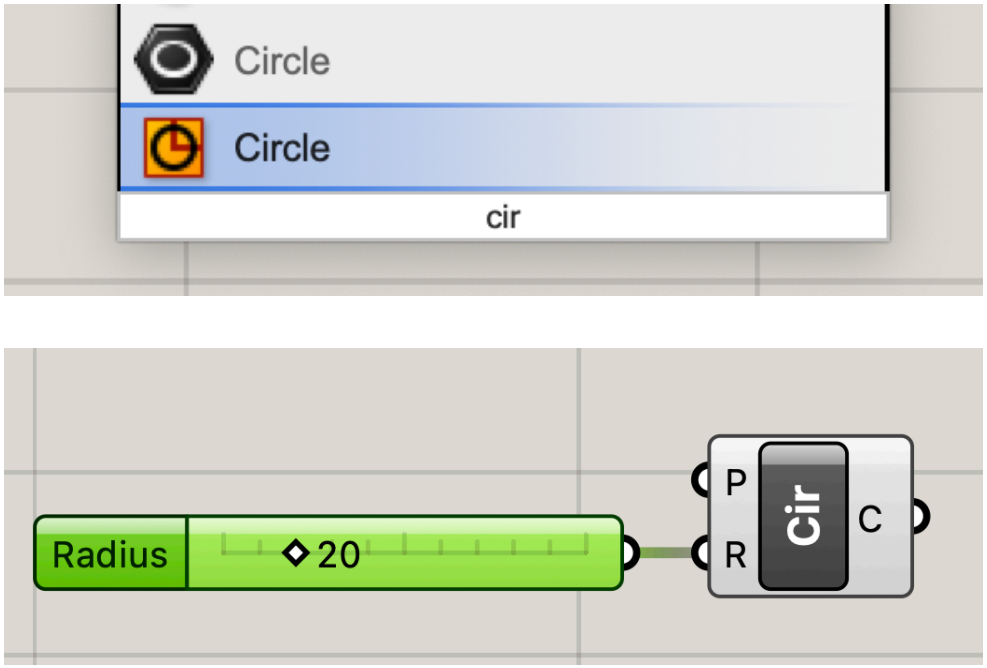
- Point
- Line and Polyline
- Curves
- Text

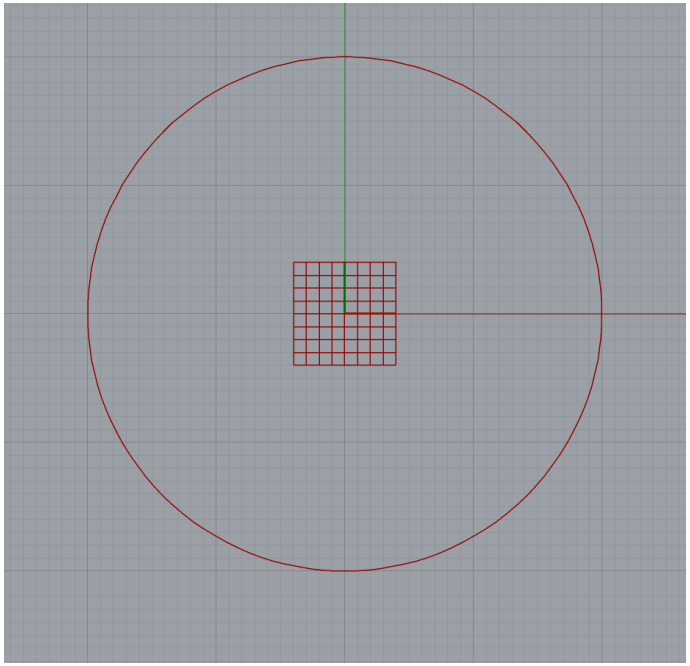
GH command:

- ConstructPoint
- Line and Polyline
- Interpolate
- Scribble

Moving into 3D

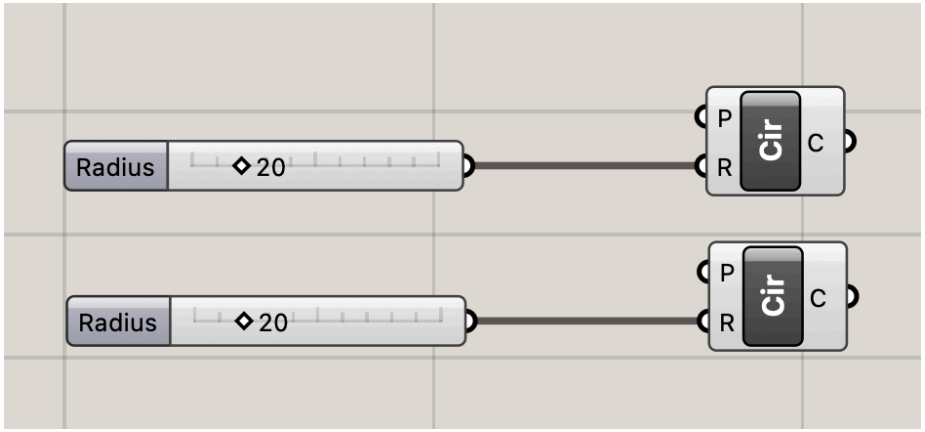
Circle

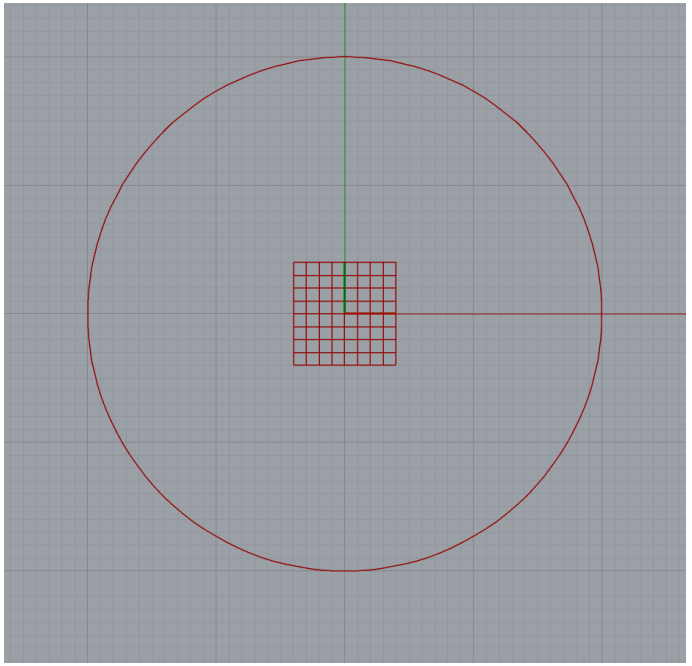




Circle

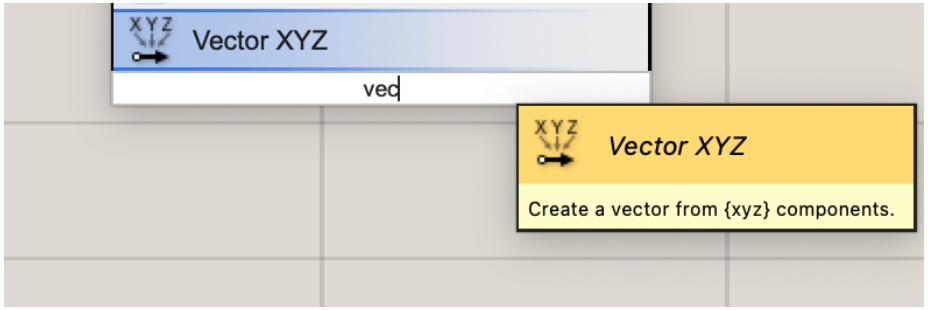
Copy & Paste Blocks

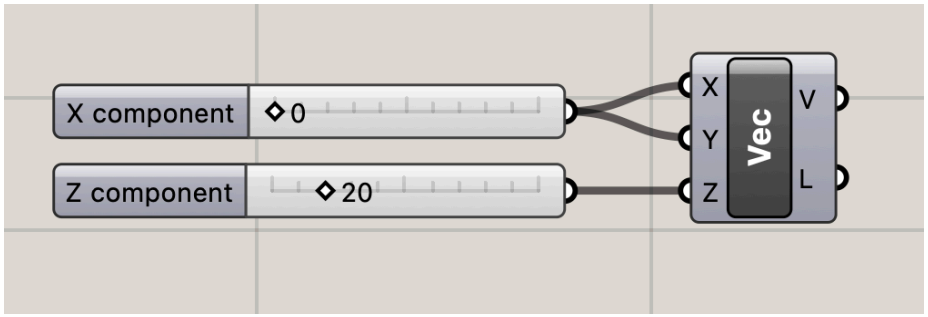




2 circles in the same position
Now, let's move one of them up in Z

Vectors

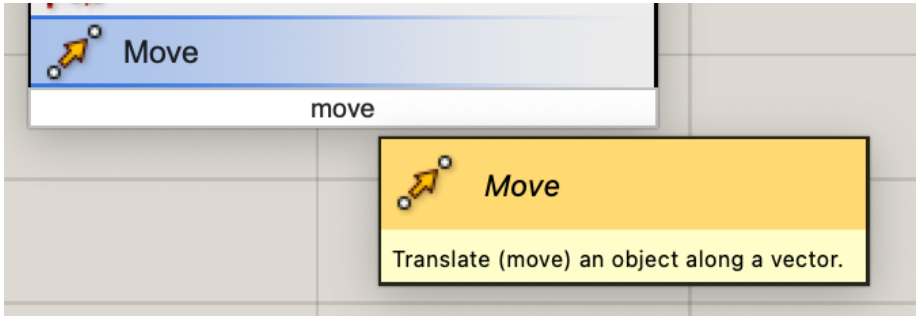


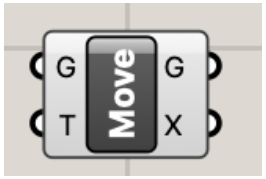


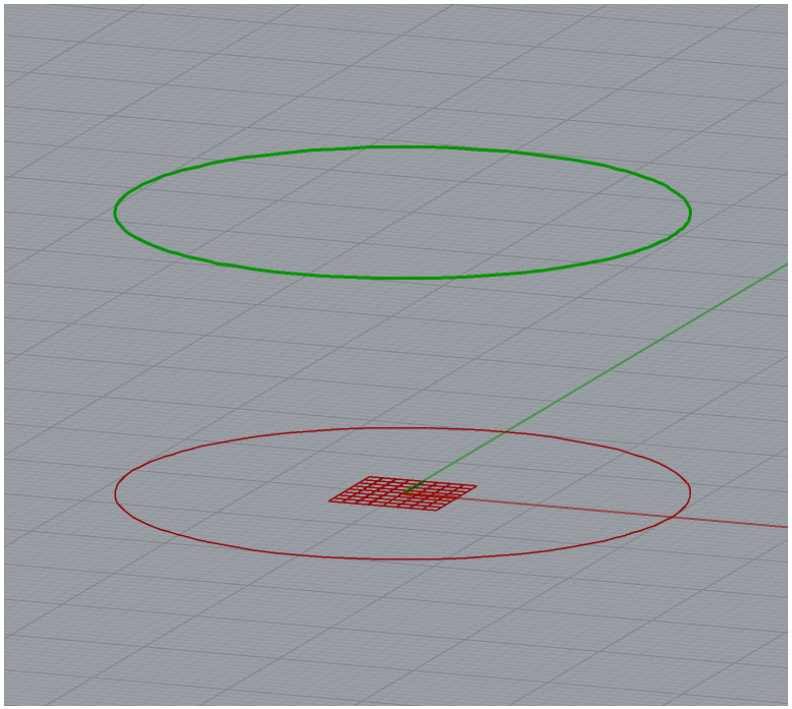
We'll use a vector to move one of the circles up in Z direction.

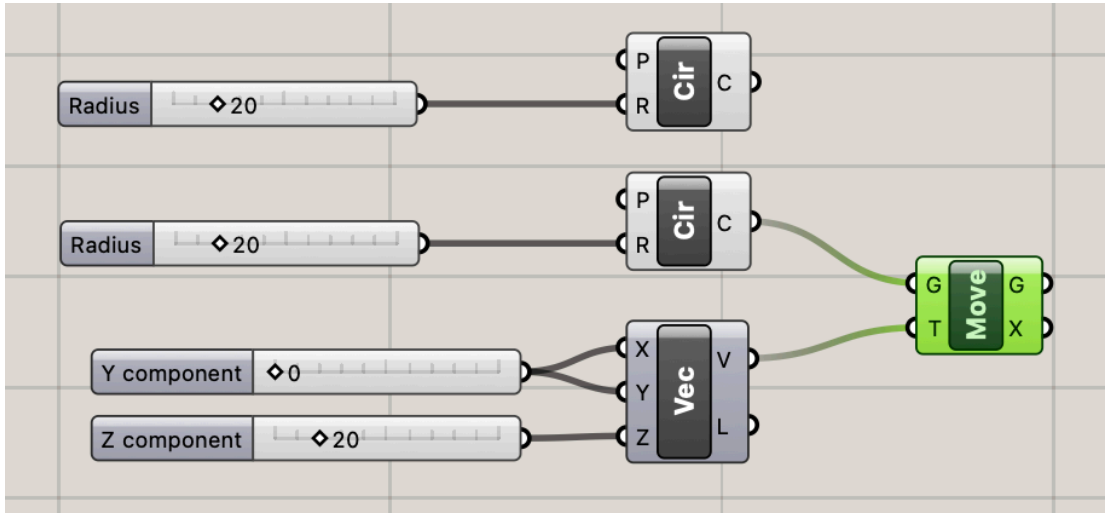
No Rhino output

Transformations: Move



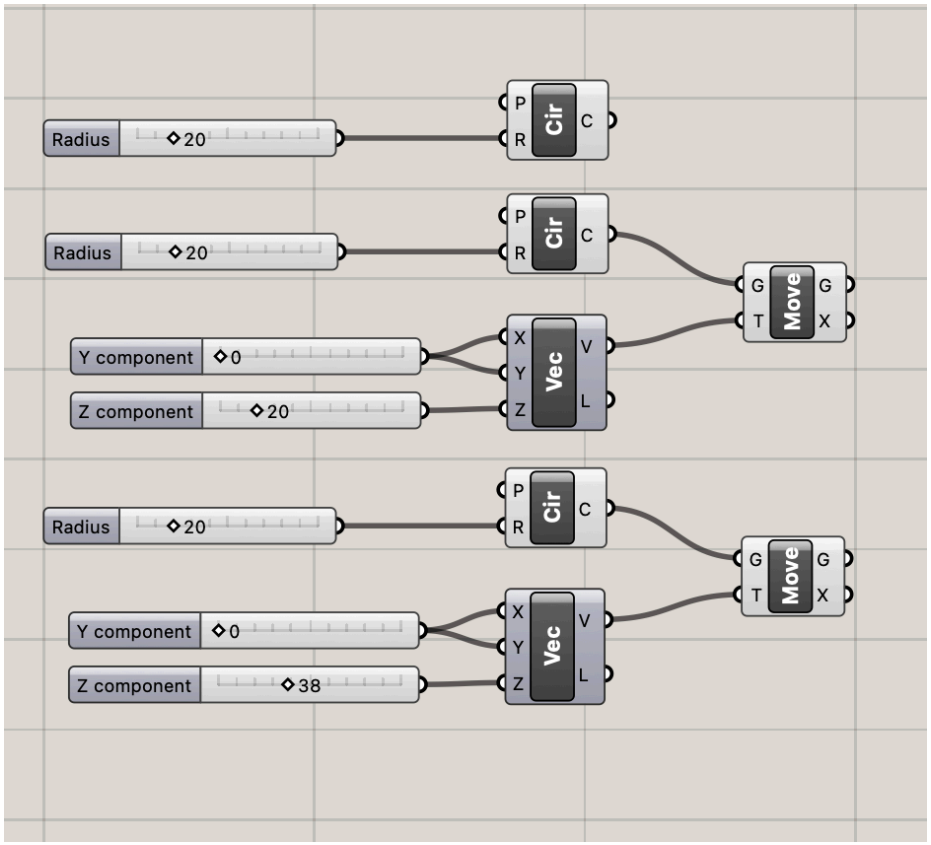


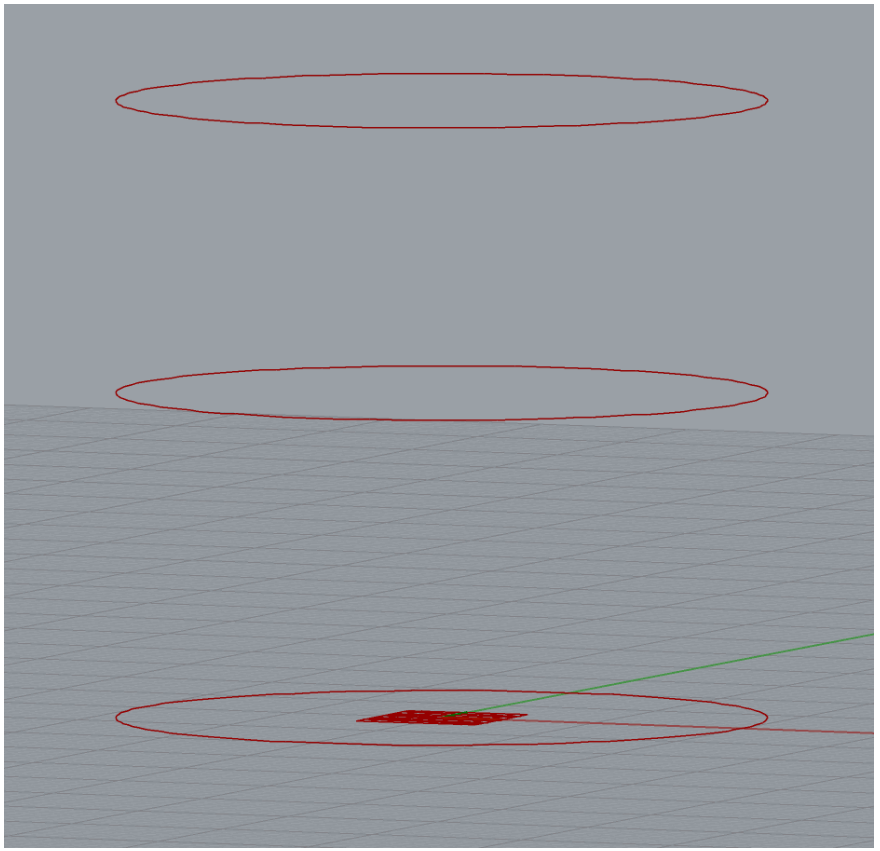




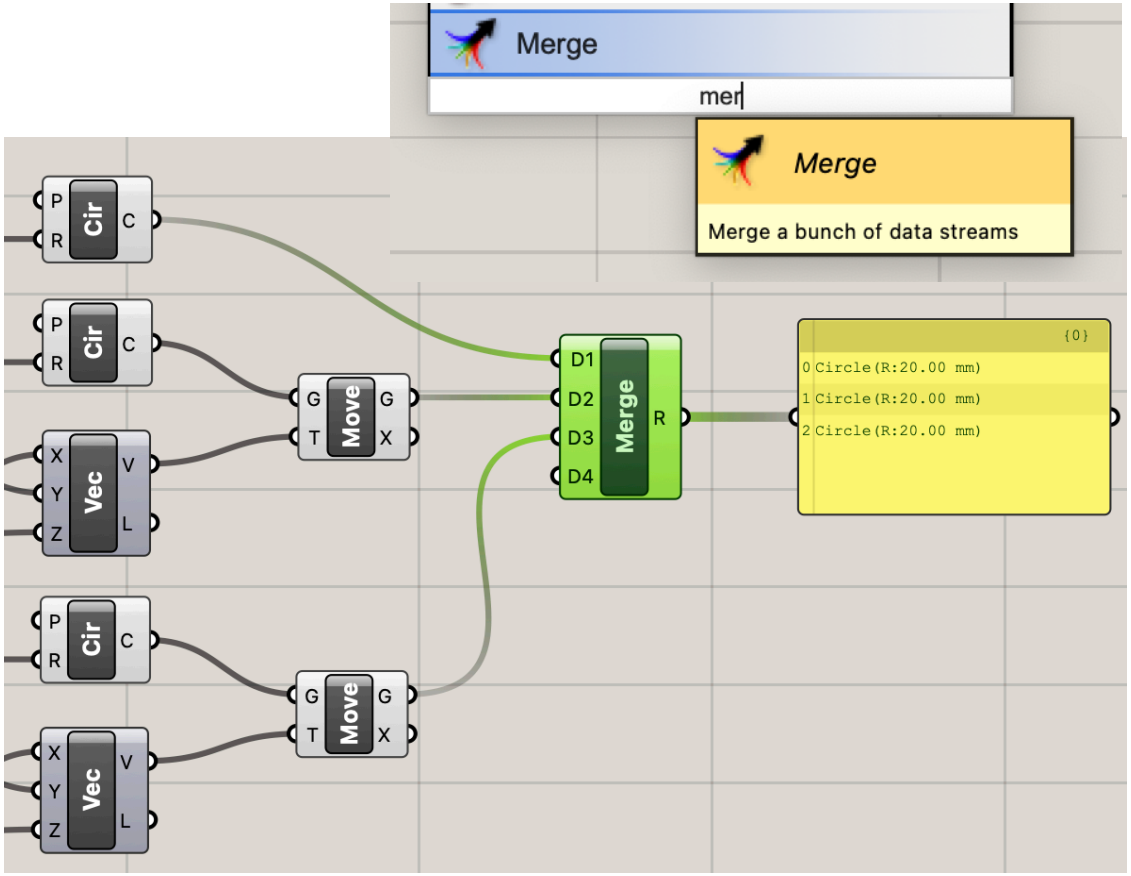
Move

Make a 3rd Circle & Move it up more

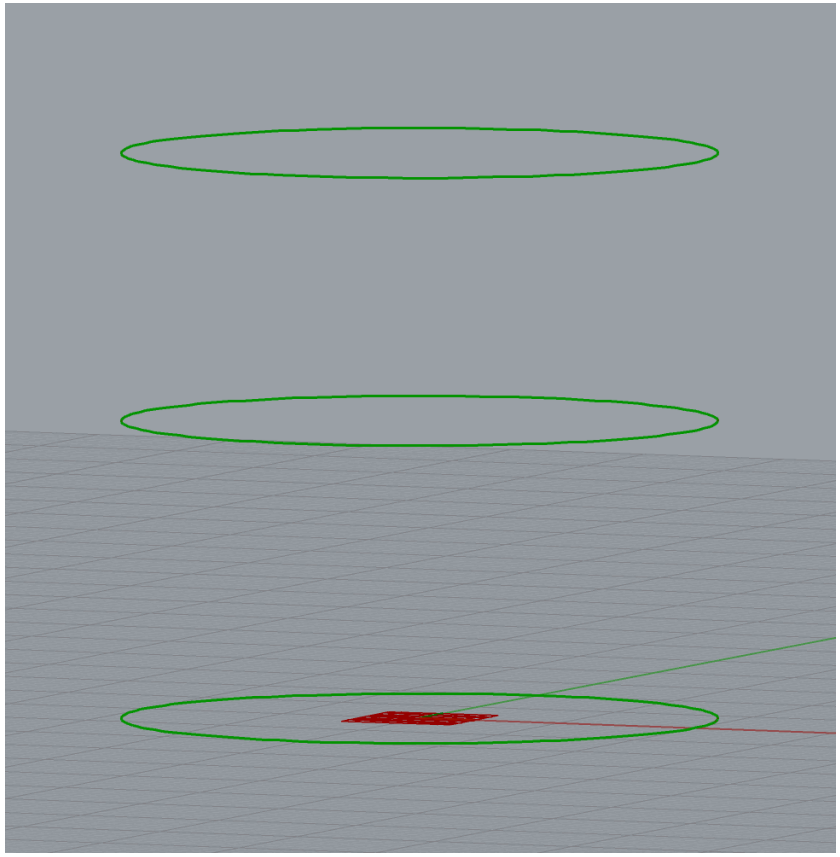




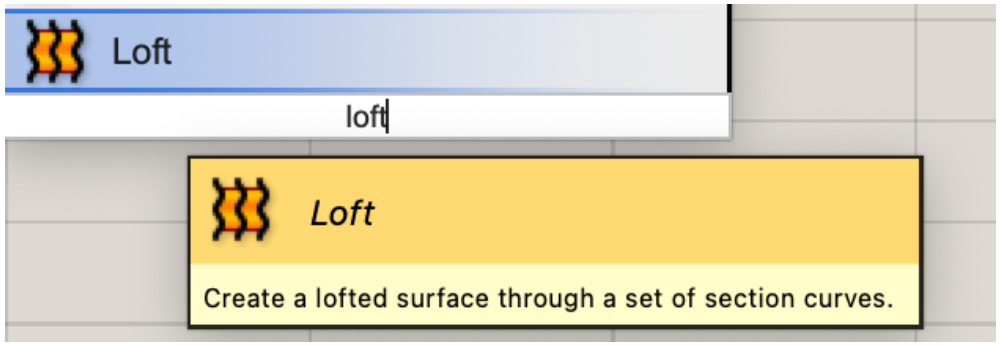
Merge: Create a List

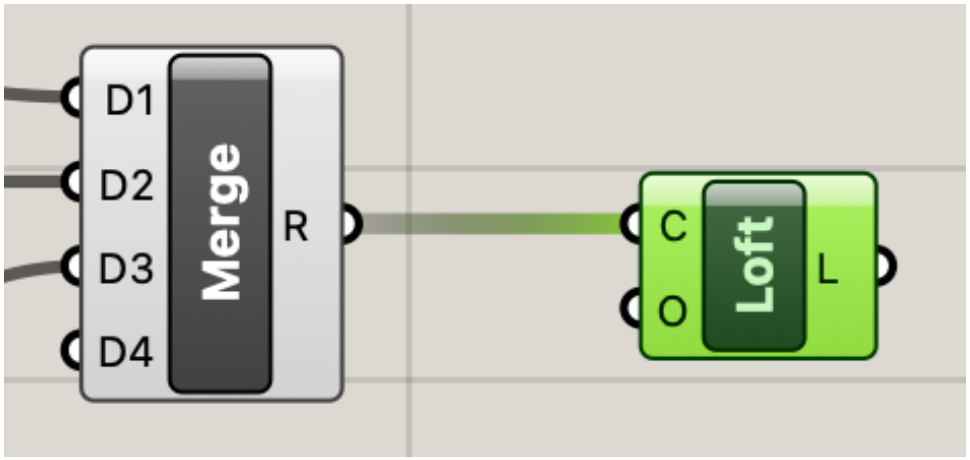


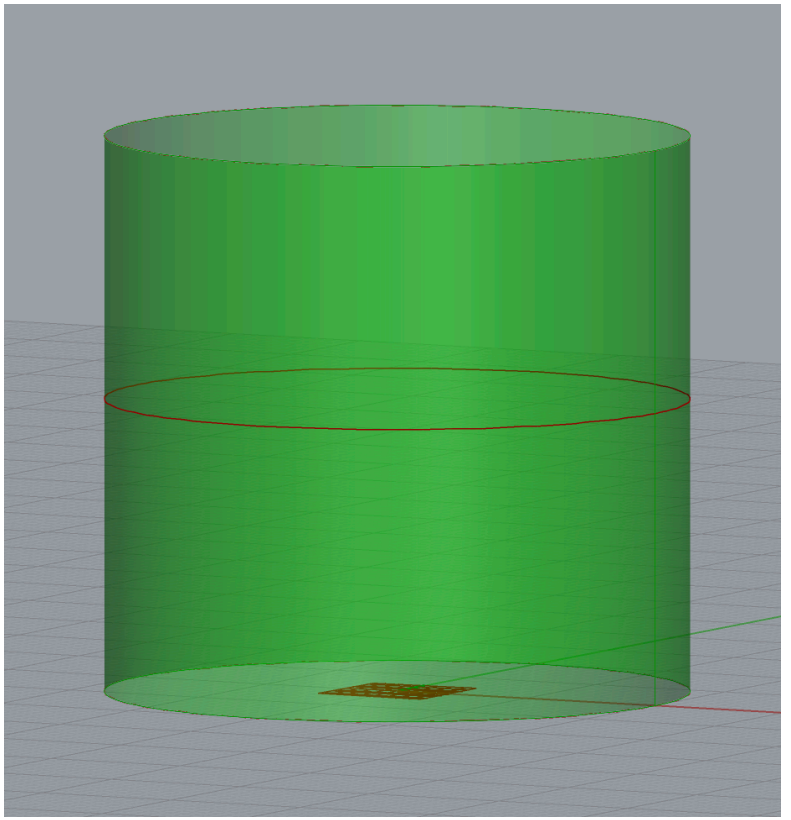
Merge



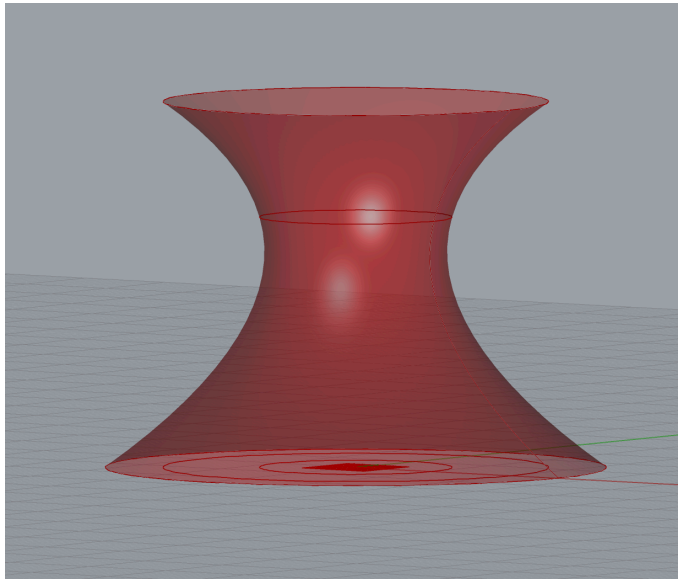
Surfaces: Loft

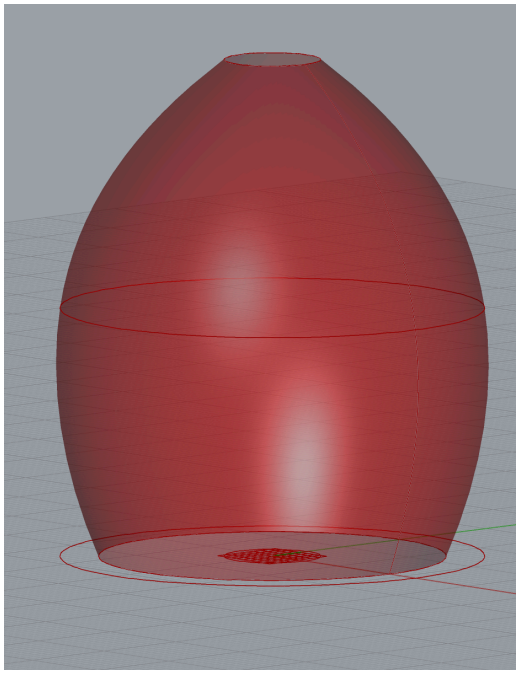


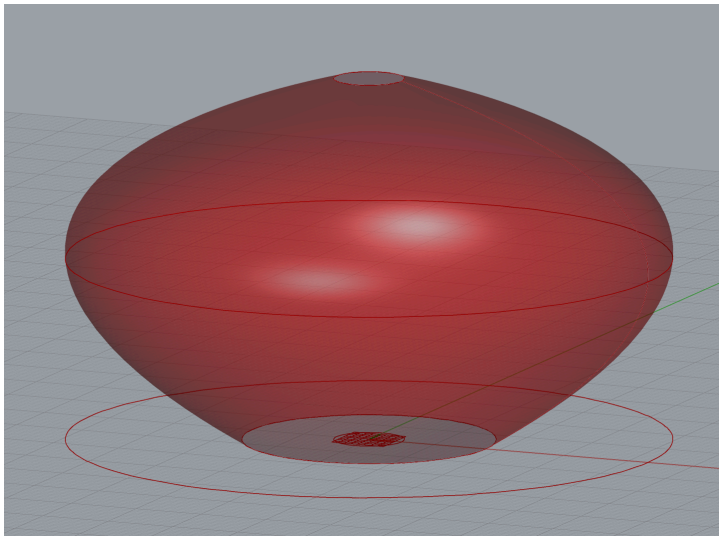




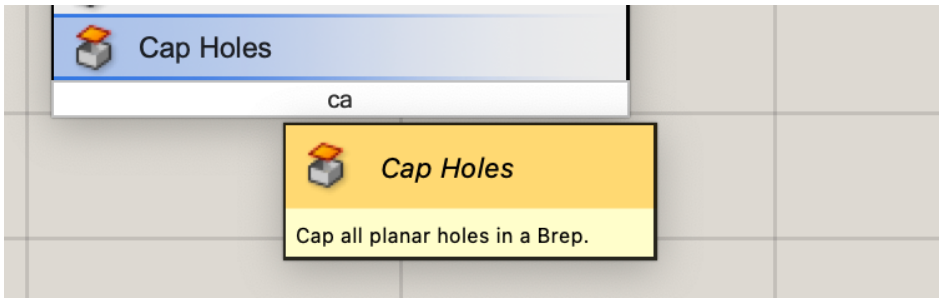
Play with Move & Radius Values

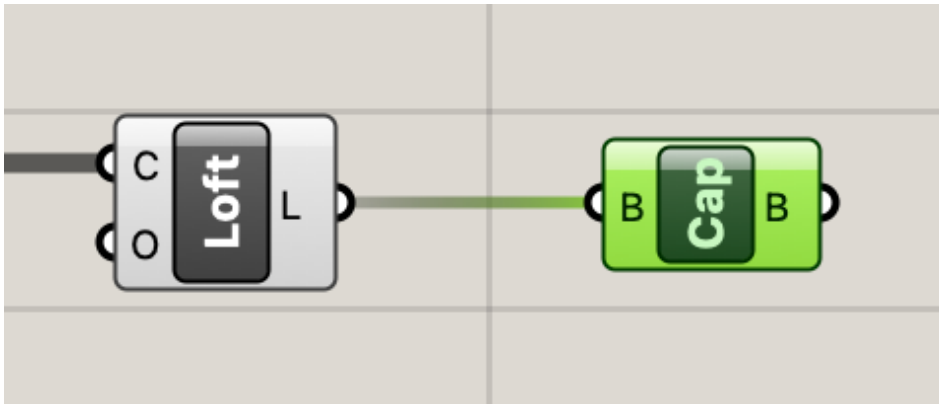


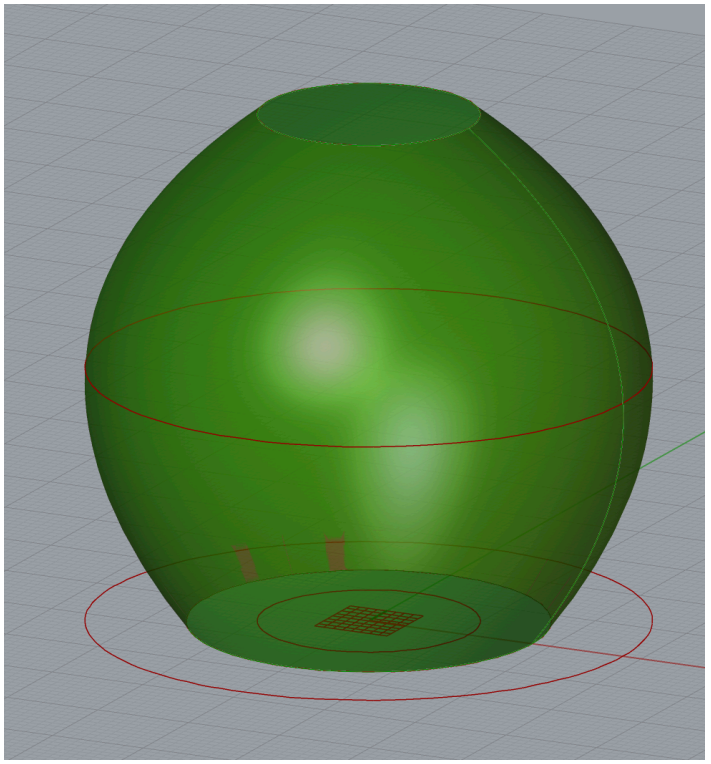




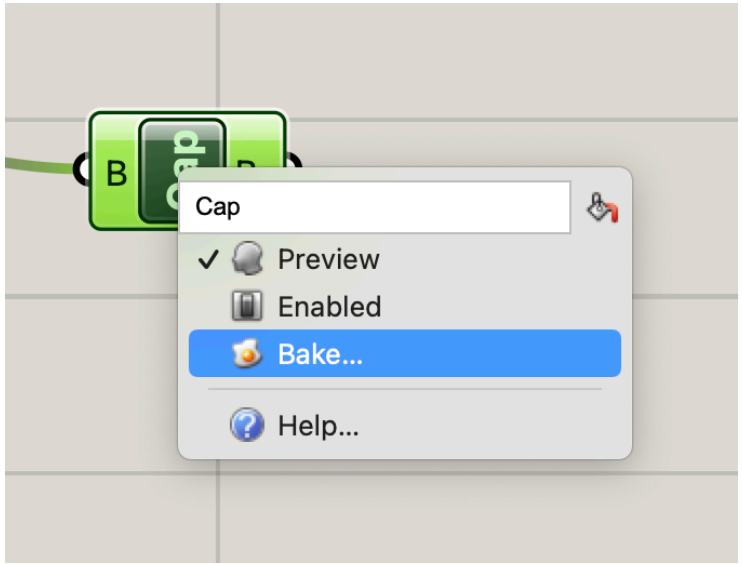
Close the Surface: Cap Planar Holes



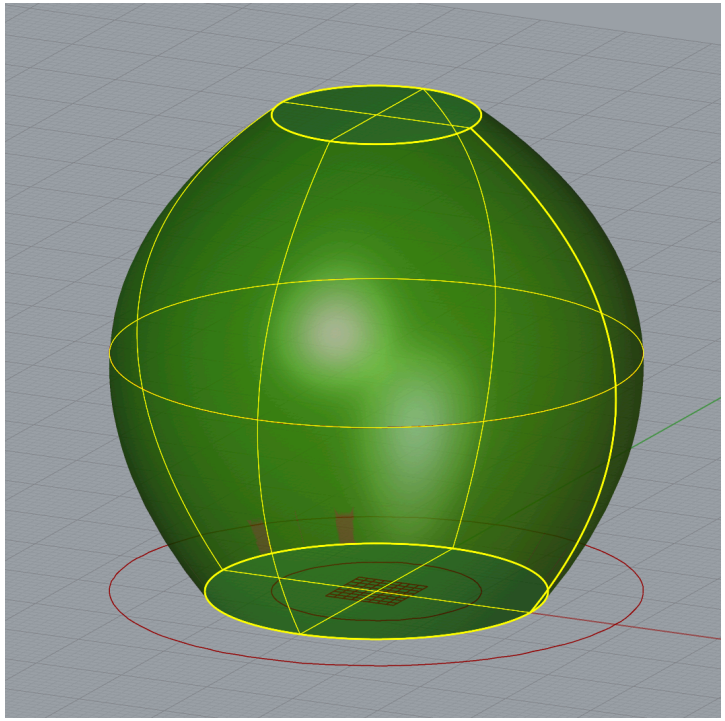




Bake your shape



right click on Cap



now exists as permanent, fixed Rhino Geometry

Baking

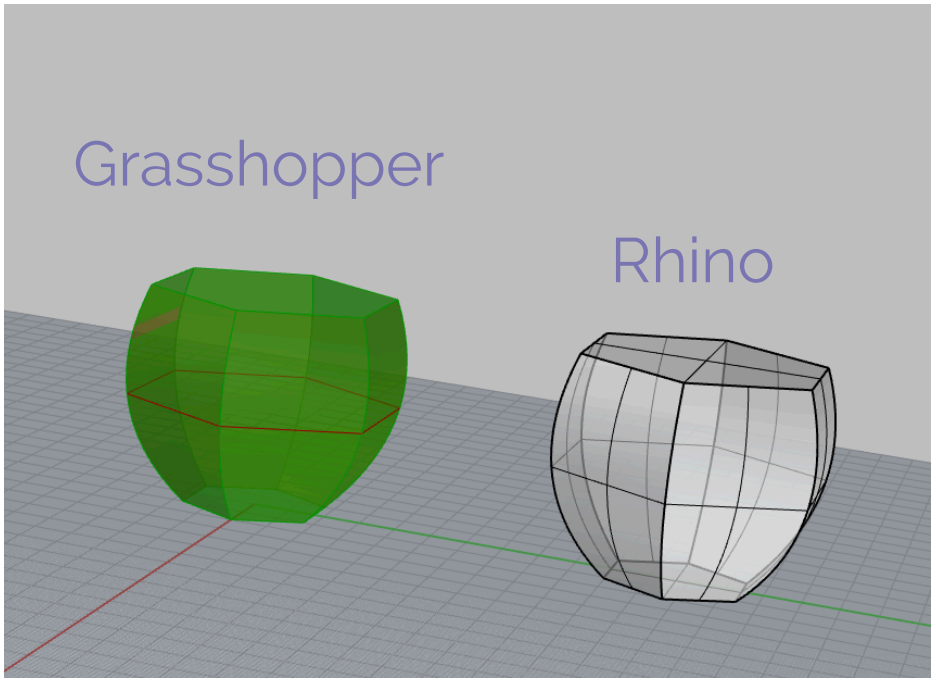
Transforms Grasshopper geometry, which is dynamic and parametric

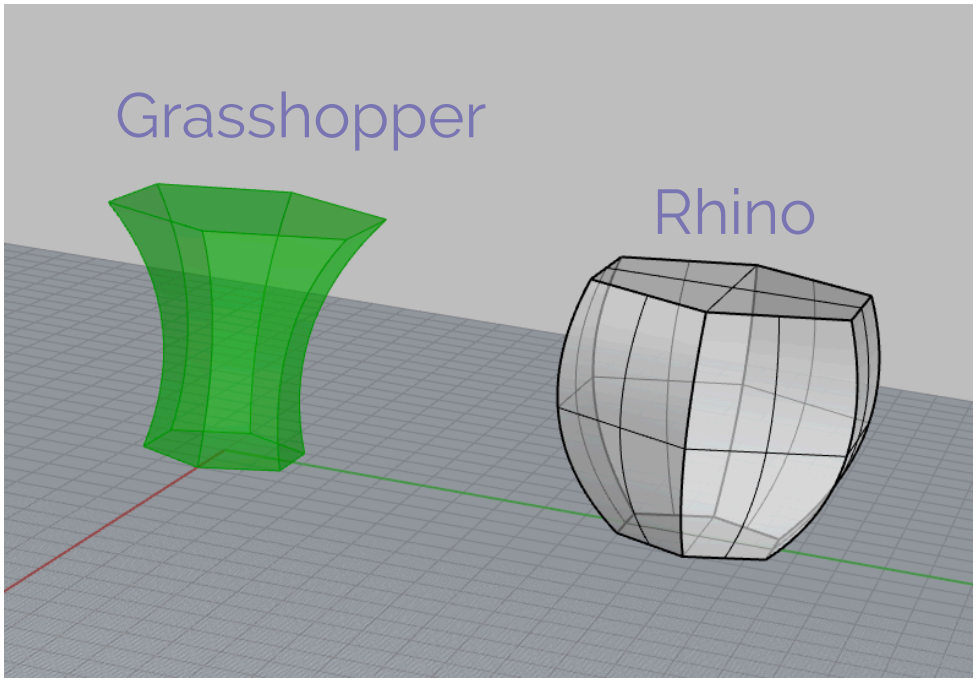
Into Rhino geometry, which is fixed. Rhino geometry cannot be changed by Grasshopper programs.

One you bake something you are done. The baked form remains fixed in Rhino as you continue to work in Grasshopper.

Geometry must be baked into Rhino before 3D printing.

Baking in the workflow: save a shape and keep working





continuing to edit in Grasshopper

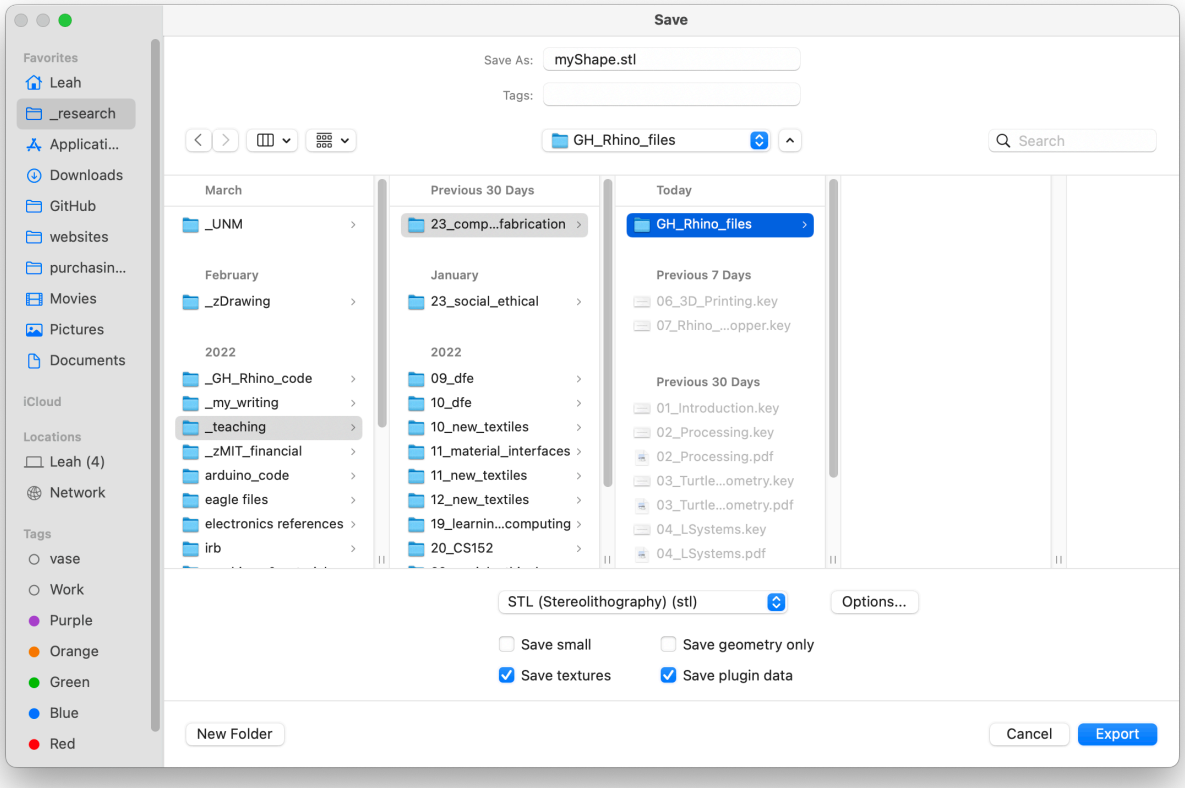
questions?

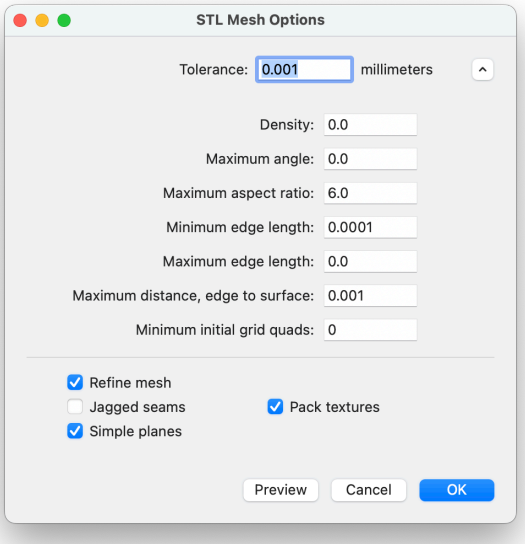
Export baked geometry as a .stl file in Rhino

- File —> Export
- Choose .STL (Stereolithography) as the file format
- Choose .01 as the resolution for your export
- Make a note of where you saved the file.

Export baked geometry as a .stl file in Rhino

File—>Export—>STL

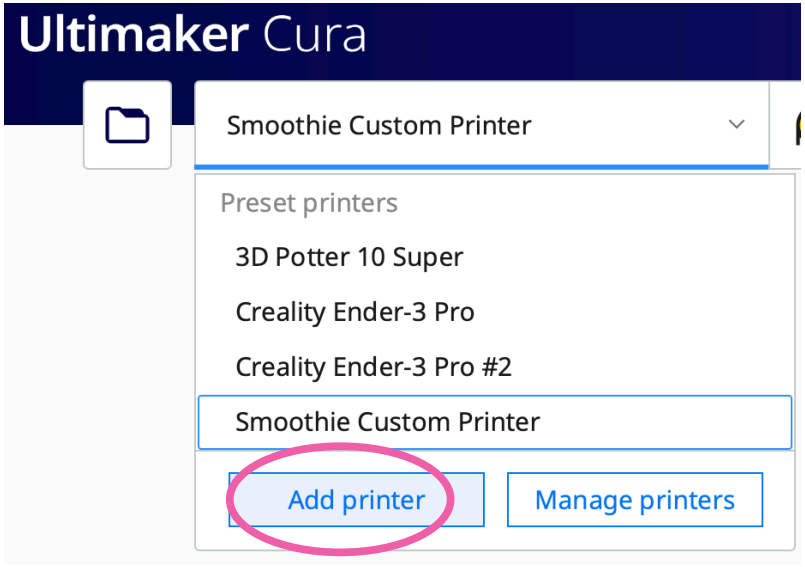


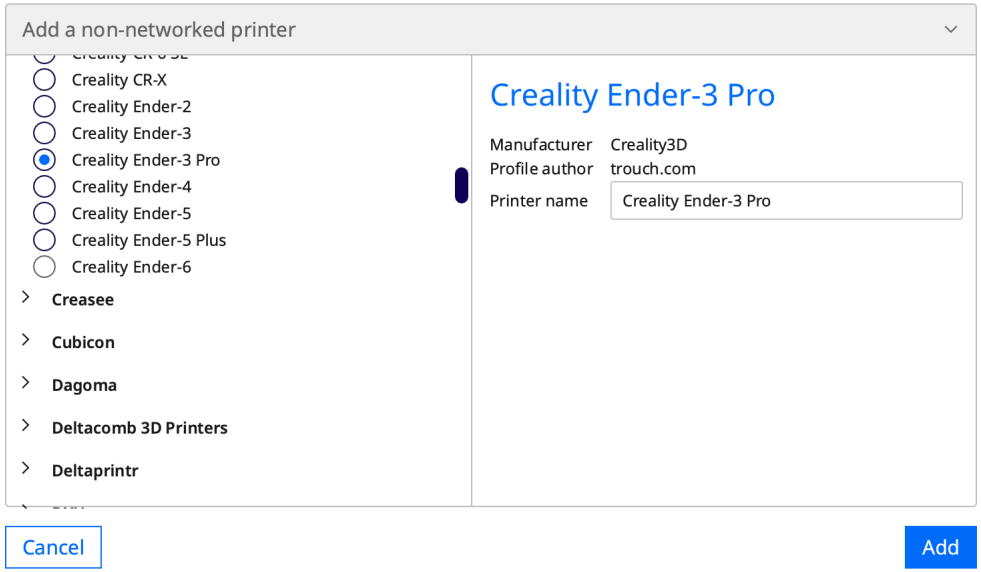


choose appropriate resolution.
.01 mm is good

Open up Cura

Add Your Printer in Cura

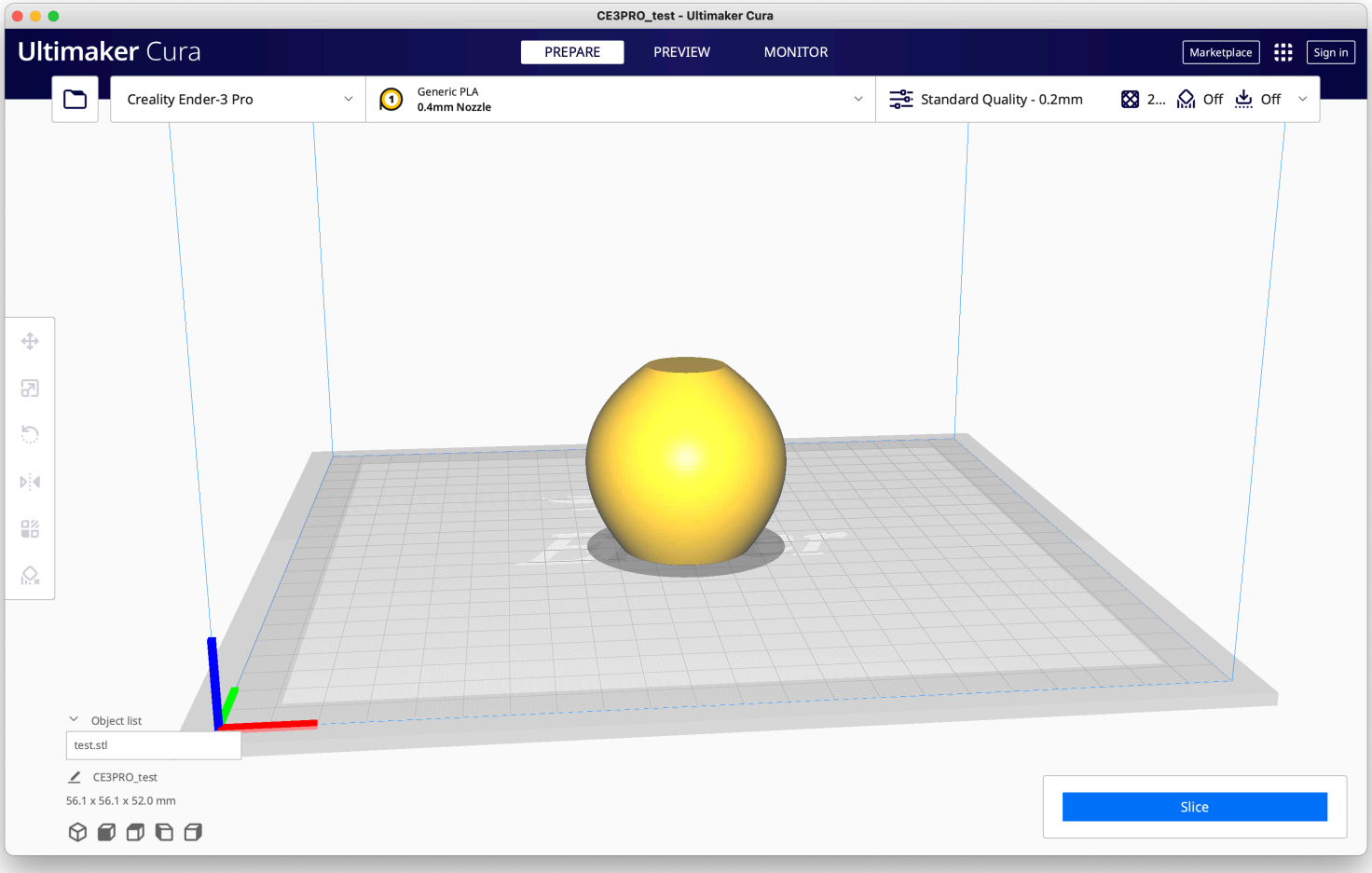




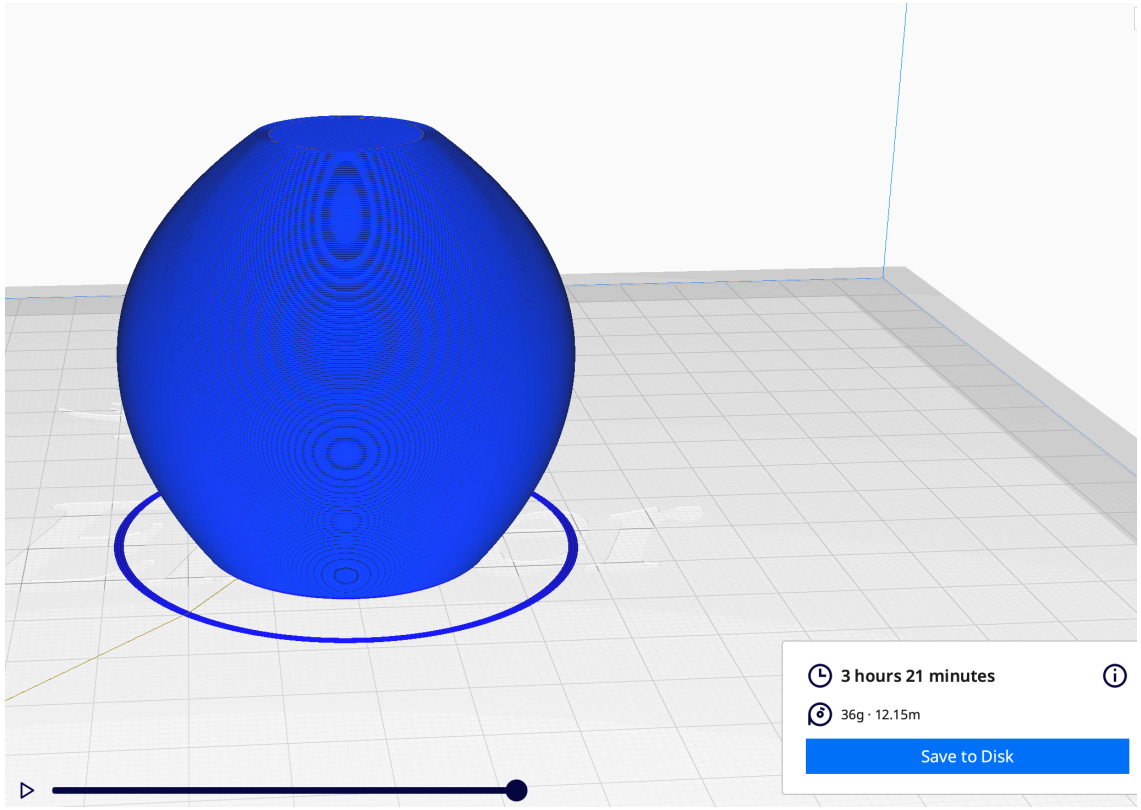
Non-networked printer

Choose the (default) Ender 3D Pro settings for Generic PLA material

Import .stl into Cura



Slice and Preview



Save .gcode file & print!

Thank you!

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.org/classes/computational_fabrication/