Rhino Grasshopper
Beginner's Tutorial

Presented by Digital Research Hub, University of Auckland
Muqi Chen

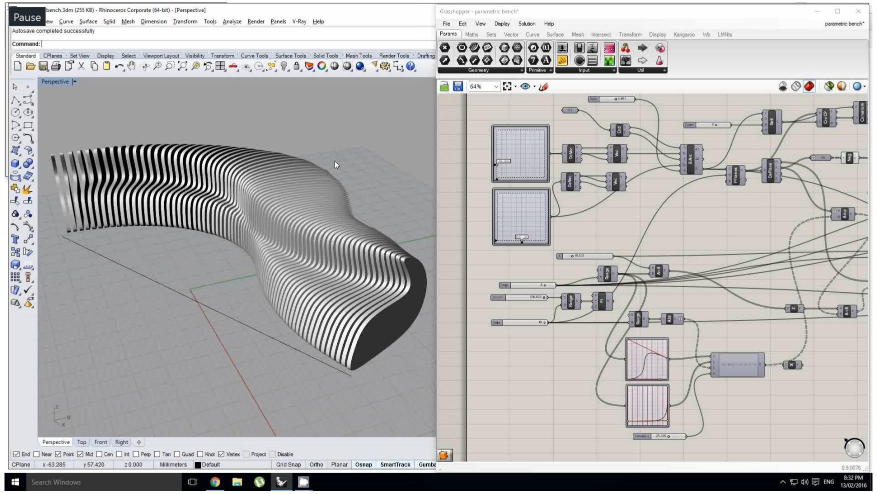
DIGITAL RESEARCH HUB

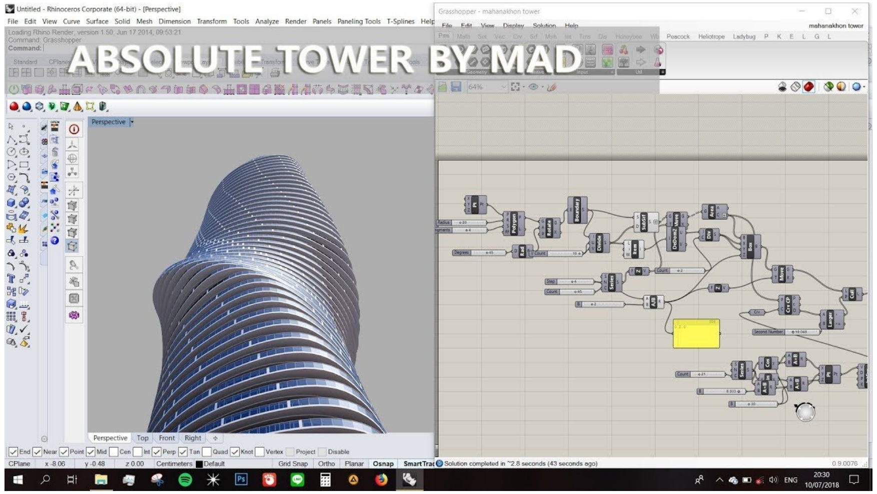
Introduction

Grasshopper is a visual programming language and environment that runs within the Rhinoceros 3D computer-aided design (CAD) application.

Grasshopper is primarily used to build generative algorithms, such as for generative art.Many of Grasshopper's components create 3D geometry.Programs may also contain other types of algorithms including numeric, textual,audio-visual and haptic applications.

— Wikipedia





Advanced uses of Grasshopper include parametric modelling for structural engineering, parametric modelling for architecture and fabrication, lighting performance analysis for eco-friendly architecture and building energy consumption.

Interface

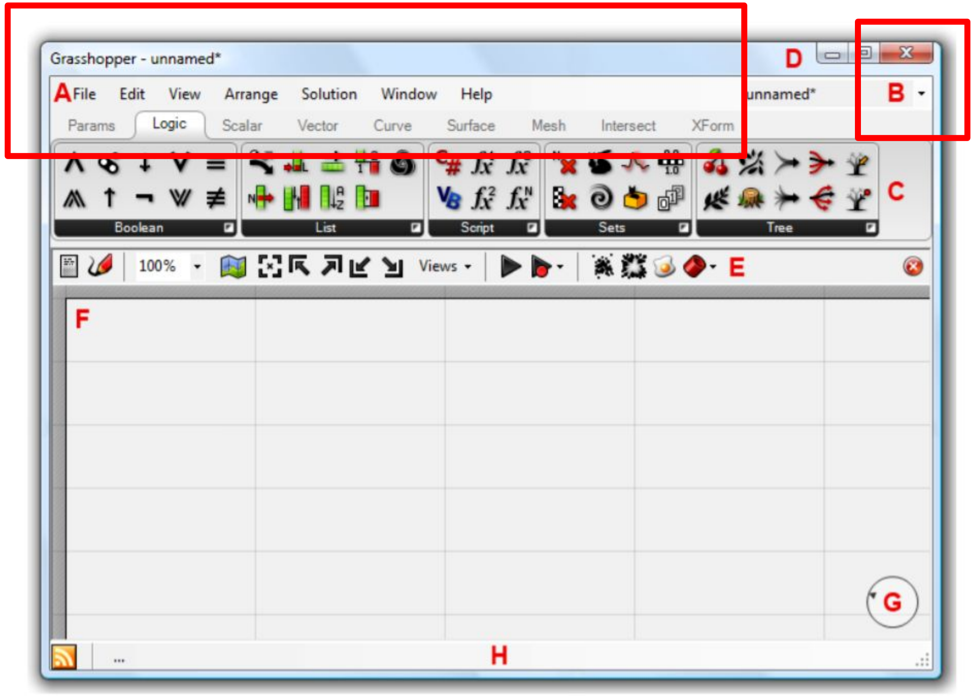
Interface Introduction

The Main Dialog:

Once you have loaded the plug-in, type "Grasshopper" in the Rhino command prompt to display the main Grasshopper window:

A. The Main Menu Bar: The menu is similar to typical Windows menus, except for the file-browser control on the right

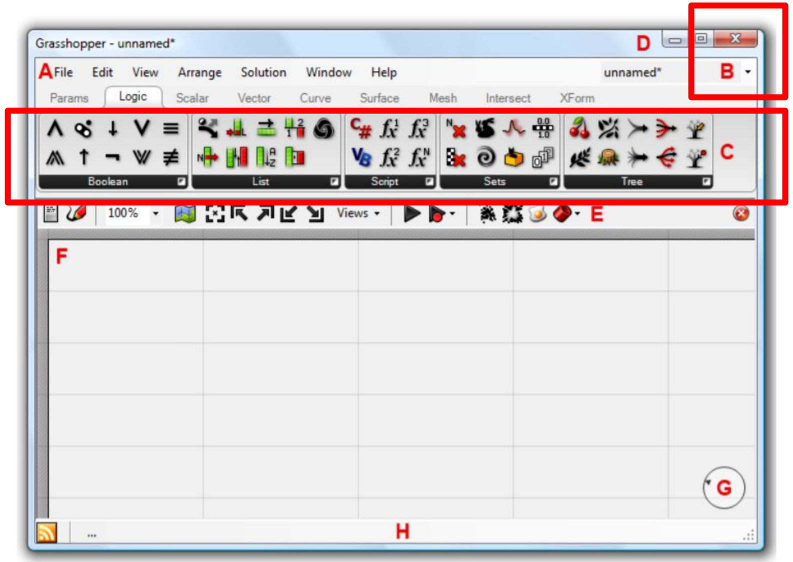
B. You can quickly switch between different loaded files by selecting them through this drop-down box. Be careful when using shortcuts since they are handled by the active window. This could either be Rhino, the Grasshopper plug-in or any other window inside Rhino. Since there is no undo available yet you should be cautious with the Ctrl-X, Ctrl-S and Del shortcuts.

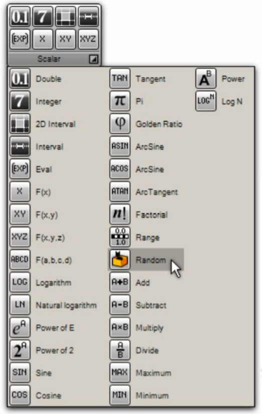


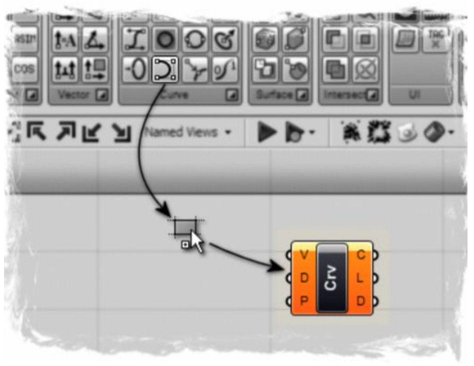
Interface Introduction

B. File Browser Control As discussed in the previous section, this drop down menu can be used to switch between different loaded files.

C. Component Panels This area exposes all component categories. All components belong to a certain category (such as "Params" for all primitive data types or "Curves" for all curve related tools) and all categories are available as unique toolbar panels. The height and width of the toolbars can be adjusted, allowing for more or fewer on-screen buttons per category.



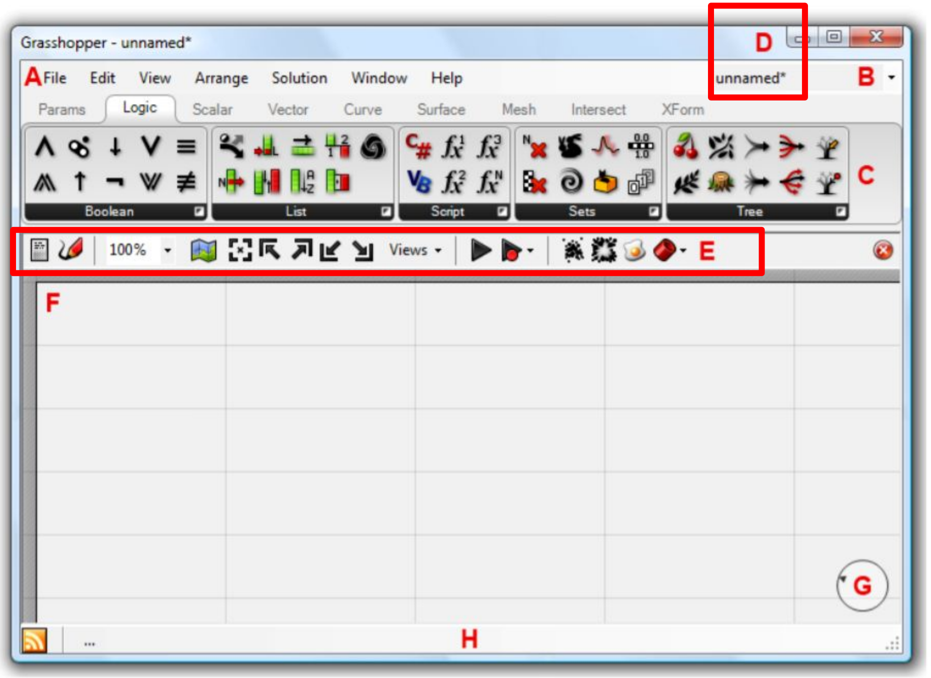




Interface Introduction

D. The Window Title Bar: The Editor Window title bar behaves different from most other dialogs in Microsoft Windows. If the window is not minimized or maximized, double clicking the title bar will fold or unfold the dialog.

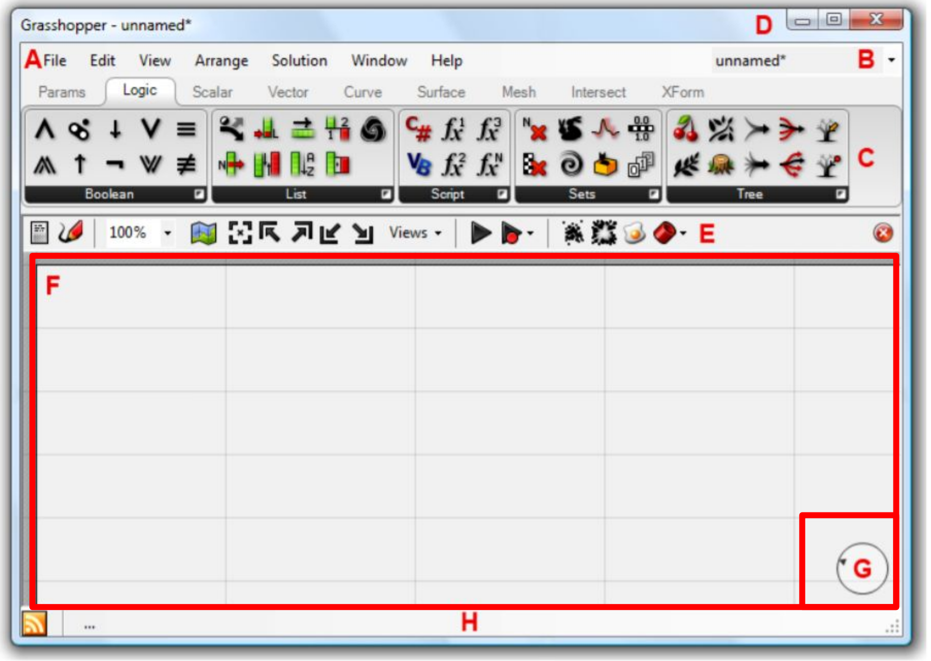
E. The Canvas Toolbar: The canvas toolbar provides quick access to a number of frequently used features. All the tools are available through the menu as well, and you can hide the toolbar if you like. (It can be re-enabled from the View menu).



Interface Introduction

F: The Canvas This is the actual editor where you define and edit the history network. The Canvas hosts both the objects that make up the definition and some UI widgets.

G: UI Widgets: Currently, the only UI widget available is the Compass, shown in the bottom right corner of the Canvas. The Compass widget gives a graphic navigation device to show where your current viewport is in relation to the extents of the entire definition. The Widgets can be enabled/disabled through the View menu.



Interface Introduction

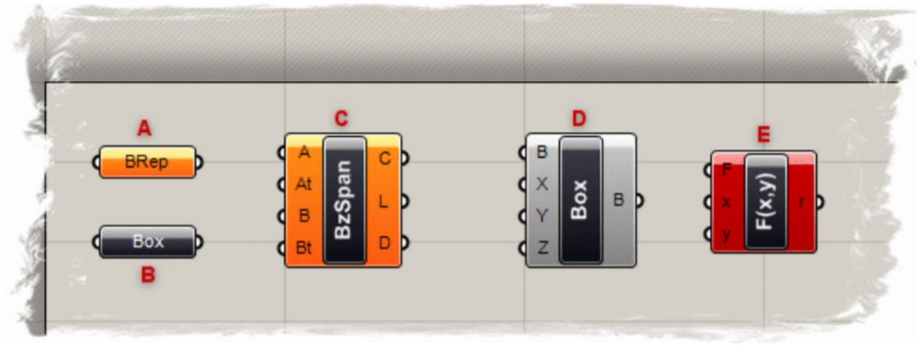
A) Parameter. A parameter which contains warnings is displayed as an orange box. Most parameters are orange when you drop them onto the canvas since the lack of data is considered to be a warning.

B) Parameter. A parameter which contains neither warnings nor errors.

C) Component. A component is always a more involved object, since it contains input and output parameters. This particular component has at least 1 warning associated with it. You can find warning and errors through the context menu of objects.

D) Component. A component which contains neither warnings nor errors.

E) Component. A component which contains at least 1 error. The error can come either from the component itself or from one of its input/output parameters.



Interface Introduction

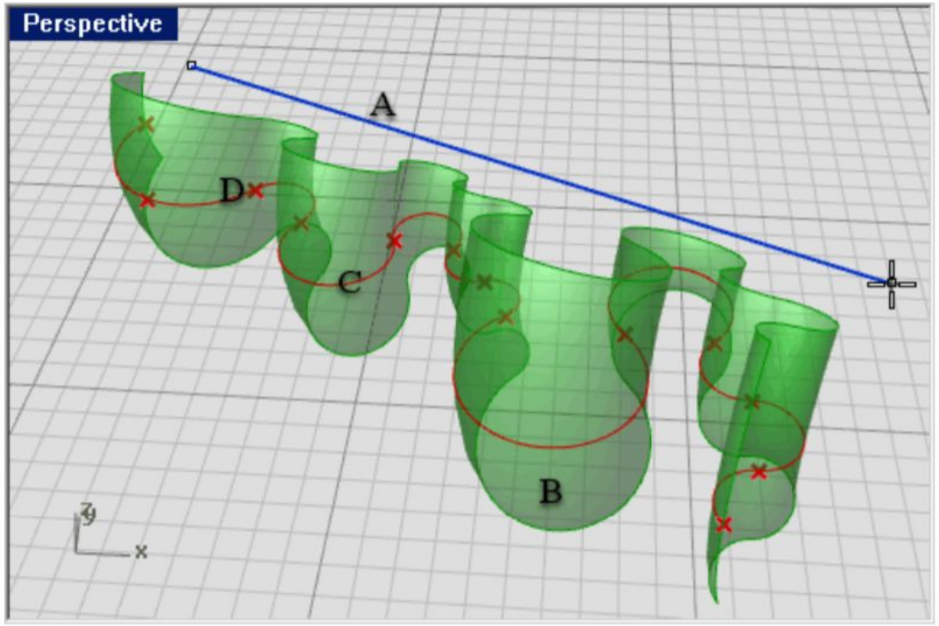
Viewport Preview Feedback:

A) Blue feedback geometry means you are currently picking it with the mouse.

B) Green geometry in the viewport belongs to a component which is currently selected.

C) Red geometry in the viewport belongs to a component which is currently un-selected.

D) Point geometry is drawn as a cross rather than a rectangle to distinguish it from Rhino point objects.



Grasshopper Objects

Grasshopper Objects

Grasshopper Definition Objects:

A) Grasshopper definition can consist of many different kinds of objects, but in order to get started you only need to familiarize yourself with two of them:

- **Parameters**

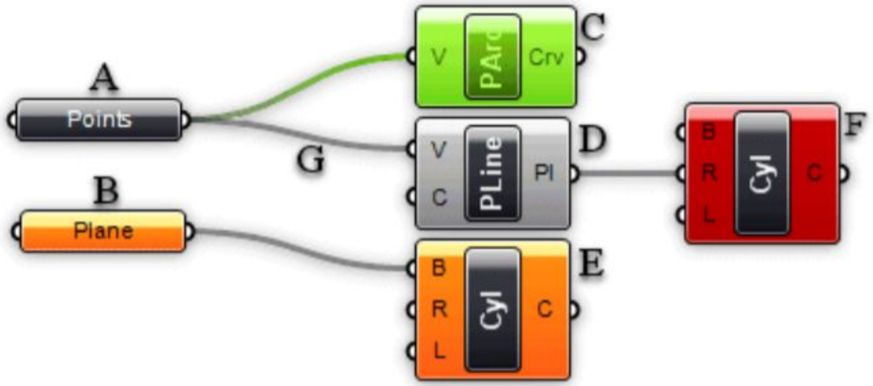
- **Components**

Parameters contain data, meaning that they store stuff. Components contain actions, meaning that they do stuff. The following image shows some of the possible objects you are likely to encounter in a Grasshopper definition:

Grasshopper Objects

A) A parameter which contains data. Since there is no wire coming out the left side of the object, it does not inherit its data from elsewhere. Parameters which do not contain errors or warnings are thin, black blocks with horizontal text.

B) A parameter which contains no data. Any object which fails to collect data is considered suspect in an Explicit History Definition . Therefore, all parameters (when freshly added) are **orange**, to indicate they do not contain any data and have thus no functional effect on the outcome of the History Solution. Once a parameter inherits or defines data, it will become **black**.



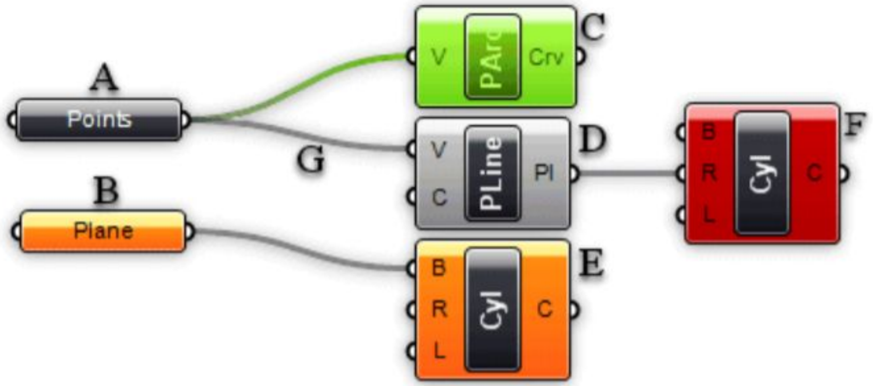
Grasshopper Objects

C) A selected component. All selected objects have a **green sheen** to them.

D) A regular **component.**

E) A component containing **warnings**. Since a component is likely to contain a number of input and output parameters, it is never clear which particular object generated the warning by just looking at the component. There may even be multiple sources of warnings.

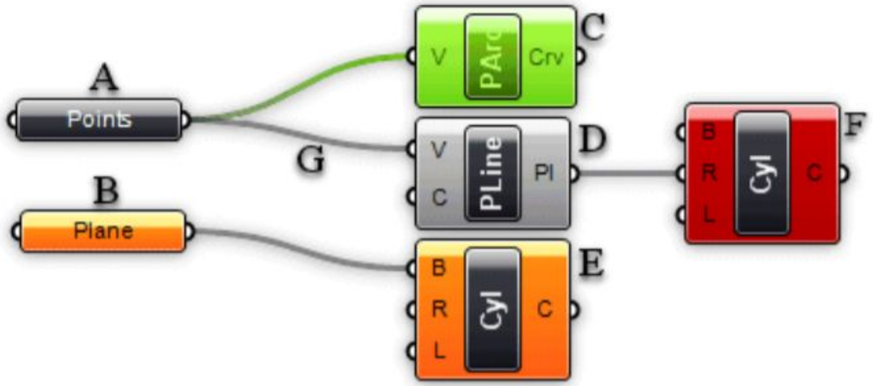
Note that warnings do not necessarily have to be fixed. They may be completely legit.



Grasshopper Objects

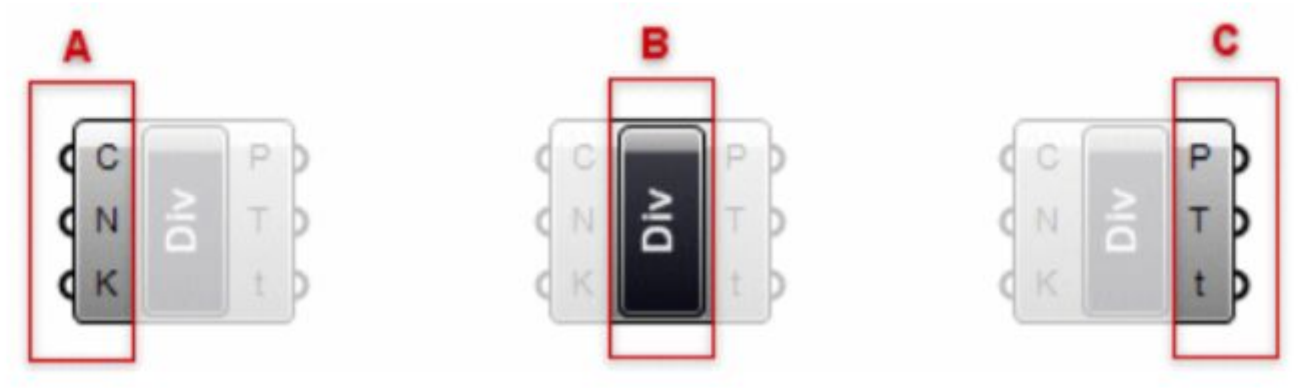
F) A component containing **errors**. Similar to warnings, it is not possible to see where the error was generated in a component. You'll need to use the context menu.

G) A connection. Connections always appear between an output and an input parameter. There is no limit to how many connections any particular parameter may contain, but it is not allowed to create a setup with cyclical/recursive connections.



Component Parts

A component usually requires data in order to perform its actions, and it usually comes up with a result. That is why most components have a set of nested parameters, referred to as Input and Output parameters respectively. Input parameters are positioned along the left side, output parameters along the right side:

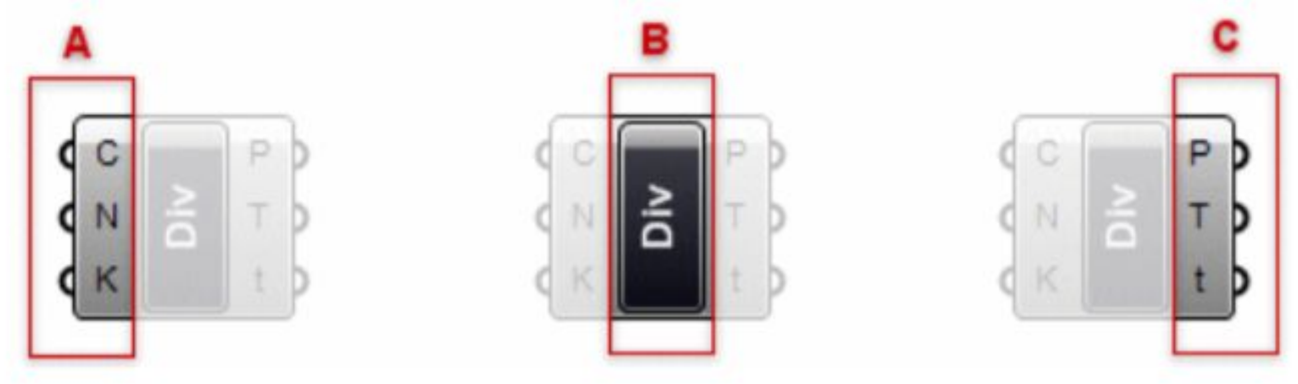


Component Parts

A) The three input parameters of the Division component. By default, parameter names are always extremely short. You can rename each parameter as you please.

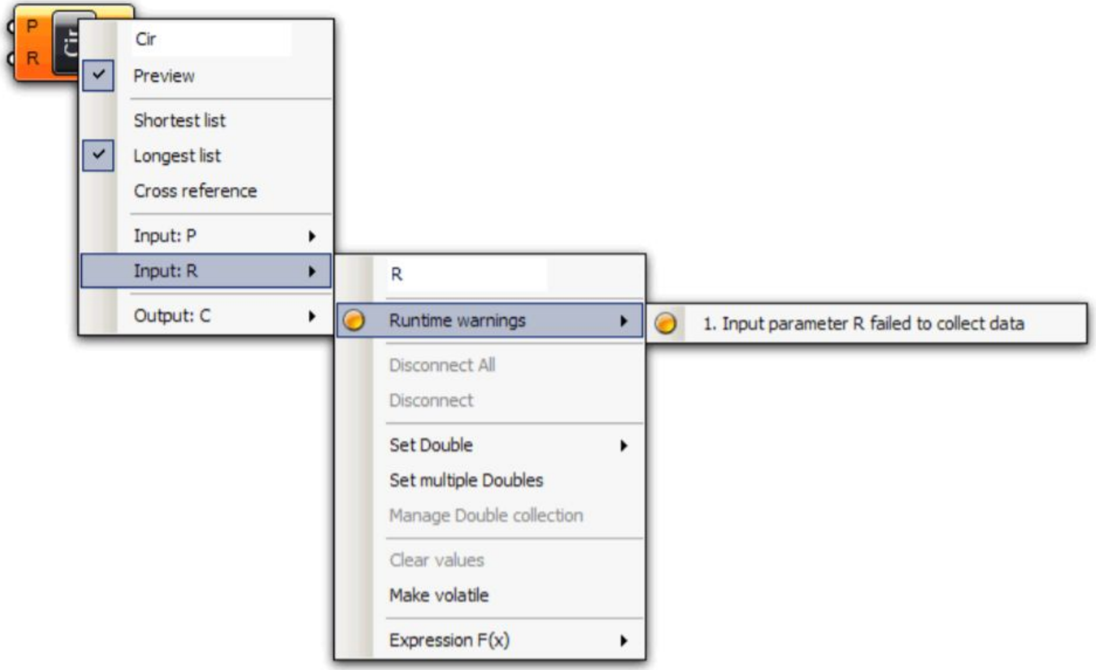
B) The Division component area (usually contains the name of the component)

C) The three output parameters of the Division component.



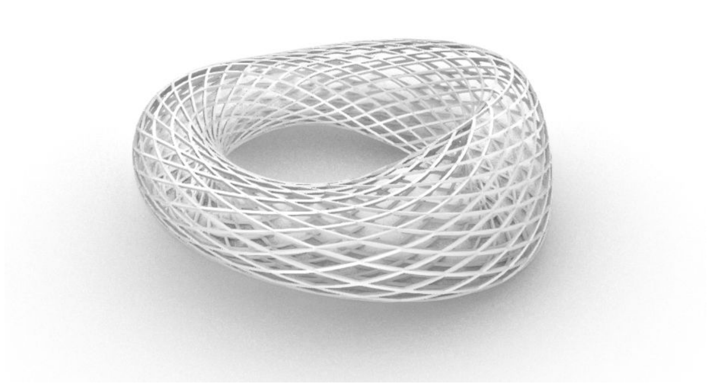
Component Parts

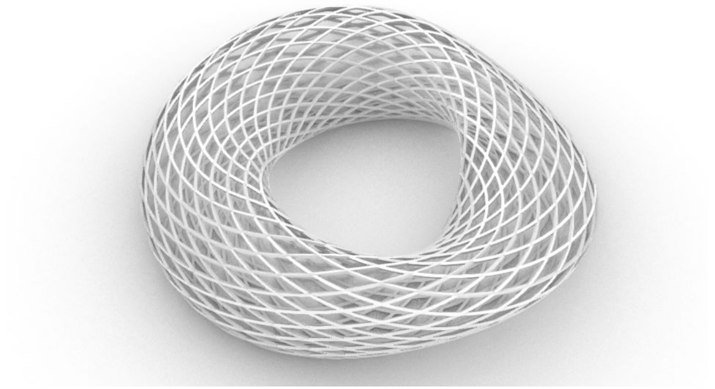
Using Context Popup Menus All objects on the Canvas have their own context menus that expose most of the features for that particular component. Components are a bit trickier, since they also expose (in a cascading style) all the menus of the sub-objects they contain. For example, if a component turns orange it means that it, or some parameter affiliated with the component, generated a warning. If you want to find out what went wrong, you need to use the component context menu:

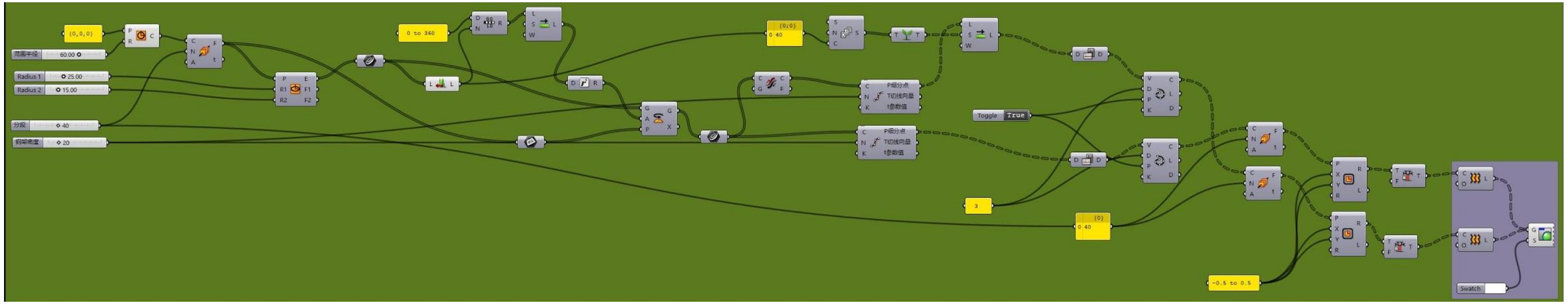


Demonstration

Let’s create a simple parametric structure using grasshopper:







{0,0,0}
范围半径 60.00
Radius 1 25.00
Radius 2 15.00
分段 40
钢架高度 20
0 to 360
{0;0} 0 40
P细分点
T切线向量
t参数值
Toggle True
3
{0} 0 40
-0.5 to 0.5
Swatch