PCPA_GH Manual

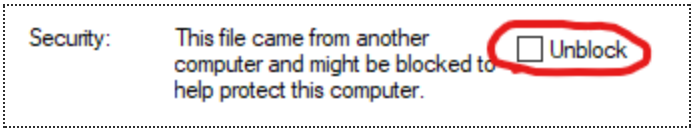
Current release 1.8

PCPA_GH is a set of bespoke components in Grasshopper that facilitates design by providing task automation. Many functions are central to PCPA's design approach and target typology. If you are a developer, there are a few APIs in the assembly that can be called externally. Try `using PCPA_GH;` and type `API.` and let intellisense guide your curiosity. Toolbar created and maintained by Max Mensching. Contact author (mmensching@pcparch.com) or visit page on Food4Rhino for license keys and download link. Version 1.8 is built against Rhino 7R6, so keep yours updated.

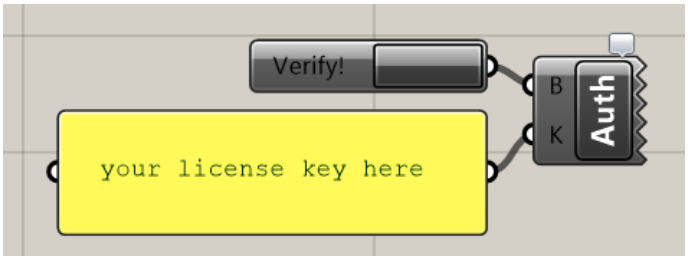
Contents

Installation

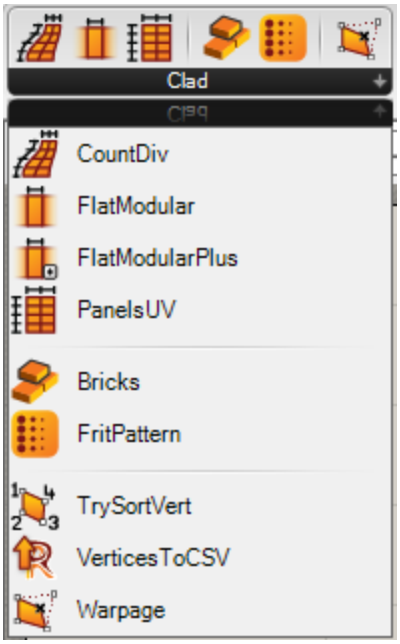
Drop the PCPA_Gh.gha file into Grasshopper's component folder. Make sure the file is unlocked by right clicking -> properties -> unblock. *Always a good idea to keep your Rhino up to date!*



Next time Grasshopper is launched, the tools will be loaded under the REACH tab. Each time this plugin is updated, please *overwrite the old* one in the folder. To activate the components, a license key must be provided. Use the Authentication component (Auth) under the Util group to validate your key. Note, in order to preserve cache and not flush the entire active script, components might still say "License Invalid" after Auth validates PCPA_GH. Re-compute affected components will resolve the error. *You will need an internet connection to authenticate PCPA_GH*.



Cladding tools



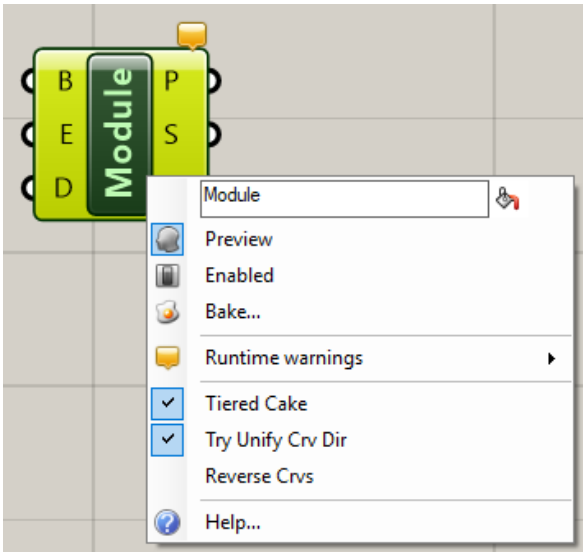
Cladding tools or “Clad” are components related to modeling building enclosures.

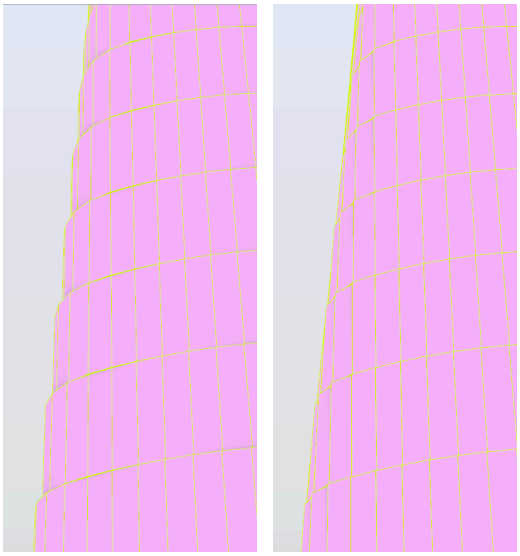
NDiv - CountDiv

This component divides facades into rows of panels of equal numbers, so vertical mullions will always be continuous from bottom up. Input B takes in a Brep as the base surface. E takes in a list of elevations where stack joints happen. C takes in the number of divisions. Output P gives the divided panels while S returns the stack joint curves.

Modular - FlatModular

This component automates panelization of building facades into a unitized system, which has constant width panels. It takes in one Brep (B) as the base shape, a list of stack joint elevations (E), and a modular distance (D). It returns flat panels (P) that best approximate the facade shape, as well as all the intersection curves at stack joint elevations (S) and module division points (L) on each stack joint curve. An “X” output records the panel corner points for deviation analysis. Right click on the component to access context menu options (below left). Toggle between “Tiered Cake” to see panels that are purely vertical (tiered cake, above middle) or leaning to follow facade geometry (above right). In the leaning panels, a fourth input “C” will be created. It takes in a number by which the panel widths shrink or extend. This is for compensating the overlaps of leaning flat panels and leaving gaps for visual clarity. If rows of panels on each level seem to begin at different places on the facade, toggle “Try Unify Crv Dir” on and off to test the marching direction. “Reverse Crvs” toggle will flip every marching direction. See https://youtu.be/oJ-KmtAfEzI for a tutorial on this component.





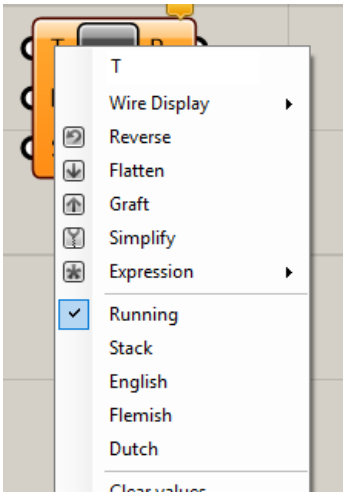
Modular+ - FlatModularPlus

Utility component in conjunction with FlatModular. Input P inherits result data from P output on FlatModular as panels, S inherits S as stack joint curves, L inherits L as panel division points as parameters on stack joint curves. X inherits X as deviation points. Input A and B take horizontal and vertical mullion sections. Output H returns the swept form of horizontal

mullions, V for vertical mullions. C returns center lines of vertical mullions. D returns the maximum deviation number. M gives the pair of points where max deviation happens.

PUV - PanelsUV

Divides surfaces into UV patches. This component also works on Polysurface as it divides each surface of a Polysurface into UV patches. It takes in an object (O), and the U parameter space division number (U) along with the V division number (V). It returns the patches (P) on success. Trims of Brep will be preserved.



Bricks - Bricks

Arrays typical brick patterns on a surface. Input T takes an integer representing the bond type. 0 = running bond, 1 = stack bond, 2 = English, 3 = Flemish, 4 = Dutch. Right click on input to select manually. Input B takes in a text description of the brick dimensions. Usually this should be left alone. It has a default value of a standard masonry brick. If it has to be changed, be sure to follow the relationship that (brick length+bond thickness) : (brick width+bond) : (height+bond) = 6:3:2

Frits - FritPattern

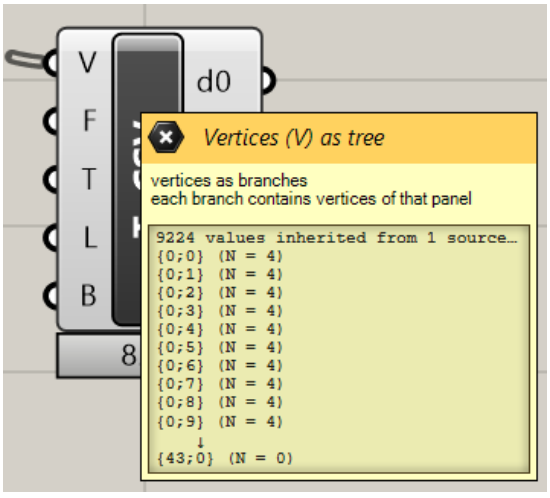
A component that generates frit pattern/perforations on a rectangle. Input R takes a rectangle as the substrate. P is percentage of opening in terms of area. Note that this is a goal parameter rather than a driver so results cannot be perfectly matched. Input D takes a division number on the short dimension. Output C returns all the circles of perforation. A returns the actual opening percentage. S returns the spacing between perforations. Right click on component and access context menu. There is a toggle between straight stack pattern and staggered pattern.

VSort - TrySortVert

Attempts to sort quad panel vertices. Useful if panels have bad vertices order (criss-cross) that may botch adaptive component deployment. Input P takes quad panels. S takes a number as shift to the order. Component returns a tree structure with each four vertices of a panel on a branch (V). Right click to access context menu where user can toggle whether to reverse the order.

ToCSV - ExportCSV

Exports quad panel vertices to file, mostly for use in Revit as adaptive component nodes. It takes panel vertices as tree (V, pictured right), optional family name to use in Revit (F), optional type name to use in Revit (T), export location folder path (L, this value defaults to user folder on the C drive), and a boolean to activate writing to file (B). Right click to access context menu, where user can input point precision. F and T parameter either take a single input or a tree structure that matches the V input. A .csv file will be written to disk upon success. This file will have the x, y, z coordinates of vertices, family name and type name on each row. Every four rows define a facade panel.



L/d - Warpage

Analyze panel warpage with division of diagonal distance through a vertex by that vertex's off-plane distance. Component takes quad panels (P) as input and outputs warpage numbers (W) and panel corner angles (A) for reference.

Rhino Commands

Exposing Rhino commands that are not in the Grasshopper native components. These will NOT require a valid license to run.

AMP - AreaVolume

Queries geometry area or volume information with unit in output. G input takes in geometry. T input takes in a boolean value indicating whether area(true) or volume(false) is computed. R returns result with unit. N output is the raw number.

ArcPPR - Arc PPR

Exposes the arc command that makes an arc with start and end points given a radius. Input A and B are the start and end point of the arc. Input R takes in the radius. Input P takes in the plane in which the arc is drawn. Note that if the two endpoints provided are not in the plane provided, the points will be projected onto the plane. Input G is a boolean value to determine whether to draw the major arc or the minor arc. Out R gives the resultant arc. Output C returns the center point of this arc. "S" outputs the sweep angle of the arc. "L" outputs the length of the arc curve.

CCrv - CloseCrv

Exposes the close curve command in Rhino. Input C takes the open curve. J output returns the closed curve. Warnings will be displayed if self-intersecting curves are detected or any curve is skipped.

Spiral - CurveSpiral

Exposes the spiral command that is absent from built-in grasshopper components. Input A takes a curve as axis for the spiral (the spiral spin around this curve). P is input for "pitch". A positive pitch produces counter-clockwise spinning. C stands for turn count. S and E are radii of the spiral at the start and end of the axis. Output C is the spiral as curve. Right click for context menu where the point count can be specified (the more points, the closer the spiral approximates circular turning).

Offset - OffsetIn

Exposes the offset command in Rhino with predictable direction. Input C takes in a curve and Input D takes in the distance of offset. Negative number of D means offset inwards. Output R returns the offset curve. This component is useful in offsetting building footprints from site boundaries.

Shrink - ShrinkFace

Exposes the shrink face command of Rhino. F takes in a Brep. S returns the Brep with all faces shrunk. A shrunken face has UV parameter domains as tightly around the boundary/edges of the face as possible.

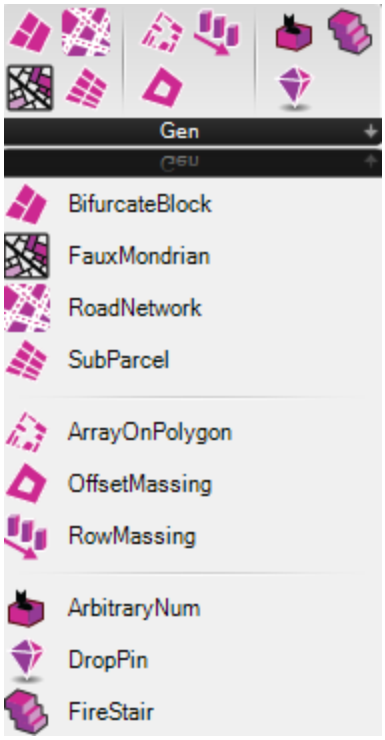
Star - StarPolygon

Exposes the star command in Rhino. Input L takes in a point as the center for the polygon. "R" is the first radius and "r" is the second radius. Between the two radii inputs, the larger one will determine the circle

within which the star is inscribed. The smaller radius determines the ring inscribed inside the star. Input N takes in the number of points/corners of the star. Output S returns the star polygon curve.

Generative tools

Generative tools or “Gen” quickly create geometries. Most of them are useful for planning projects. Some utility generators are included as well. Many components of Gen tools are discussed in this clip https://youtu.be/Q1uwLkTByao



BiDiv - BifurcateBlock

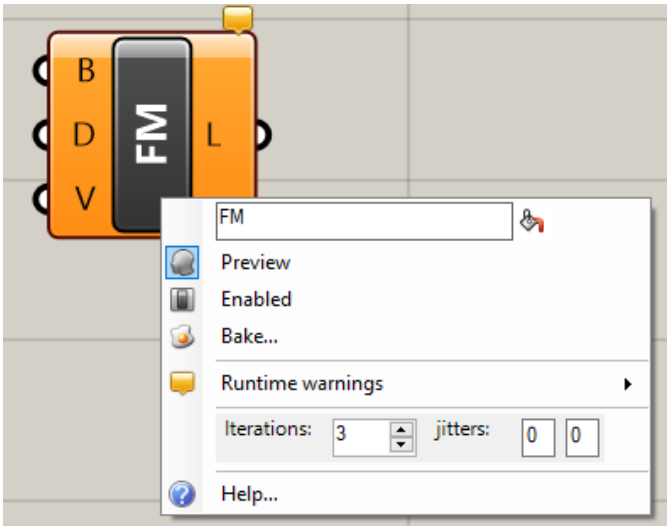
Strikes a division line in the middle of a block if the block area is too great. This component is recursive. Input C takes in area boundary curve(s) and A the desired area limit. L outputs the subdivisions as closed curves. This component only works on planar curves

ArrOnPgon - ArrayOnPolygon

Fills a city block with box building footprints hugging the street front. Currently only work on polygons. Please convert splines into polylines. Input B takes in the base curve as a city block. W takes in the guide width of each building footprint, D the depth factor of the building depth (depth in respect to the size of the city block), and J a jitter factor to randomize footprint sizes. Component returns the building footprints (F). Use footprint to extrude mock context.

FM - FauxMondrian

A component to make pseudo-Mondrian patterns on a plot of land. Input B takes in a boundary curve within which pattern is created. D takes in the minimum dimension for each cell. V takes in a decimal number between 0 and 1 to vary the minimal dimension (only growth). Output L returns the lines that define the pattern. Right click on the component to access context menu, where the iteration can be altered (although not recommend) and angular jitter assigned. The first angular jitter



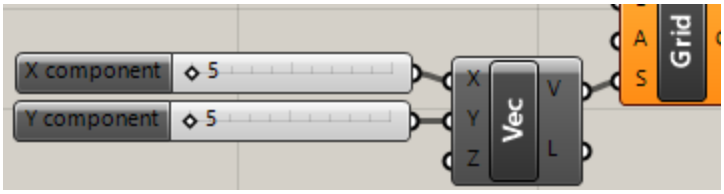
is for major divisions (iteration 1) and the second for all subsequent divisions.

OSMass - OffsetMassing

Offsets city block boundary to create courtyard-like building footprints. This component will arbitrarily determine which segment of boundary is offset. Results may be full courtyards or open C, L shapes or singular bars. Input C takes the boundary curve. D takes in a guide offset distance. J takes in a jitter factor to randomize offset depth. Component returns the offset region curves (R)

Grid - RoadNetwork

This component fills a region with a grid sized to different precedent cities or user inputs. Used to rapidly divide a plot of land into an urban road network. Input B takes a boundary curve. A (optional) takes in angle of rotation of the street grid. S (optional) takes in a shift amount indicated by a vector. Pictured here is to shift the grid by 5 units in each direction. Z value of the vector is ignored. This component returns the grid lines (G). Right click to access context menu. Several city grid sizes have been built-in. User can however input desired size in the pair of text boxes and hit enter.

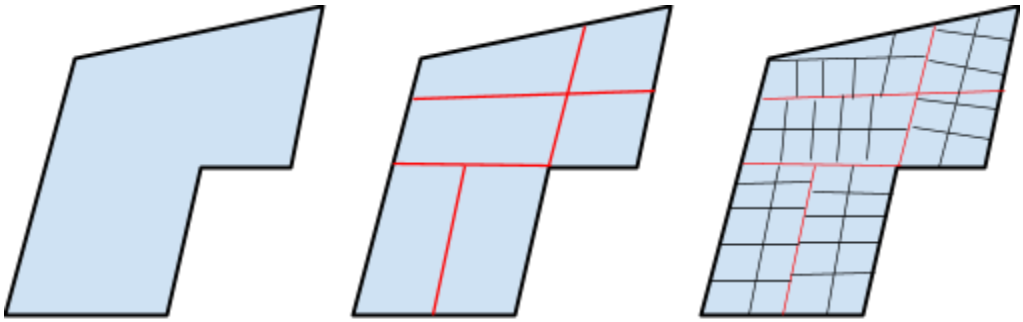


Row - RowMassing

Arrays massing boxes along a path curve. Input P takes a path curve. W is the guide width for massing boxes, D the guide depth, G the gap between boxes. The component returns a list of result footprints (R). Right click to access context menu. A slider between value 0 to 1 will apply randomization to the sizes of footprints. The component uses a proto-binary-search algorithm to guarantee spacing, and the tolerance is input by the context menu. In general keep this number larger than 0.1 units. If results don't occupy the entire length of the path curve, try recomputing a few times.

SubPrcl - SubParcel

Divides a plot of land with cutting road curves, into city blocks, and turn them into parcels hugging the longer sides. Input B takes in the boundary of the plot of land. C takes in curve cutters (must extend to boundary). W takes in a width of the parcels. Component returns parcels (P), the city block curve loops (L), and the property line splitting the block into parcels (S). This component can be used to only cut a plot into blocks. Right click to access context menu, and uncheck "parcel-ize!".



ANum - ArbitraryNumber

Generates a series of arbitrary numbers. Essentially the native random component of GH, but this gives random numbers independent of a seed control. “C” input takes the number of these arbitrary numbers. “R” takes the range within which the arbitrary numbers fall. “N” outputs the arbitrary numbers.

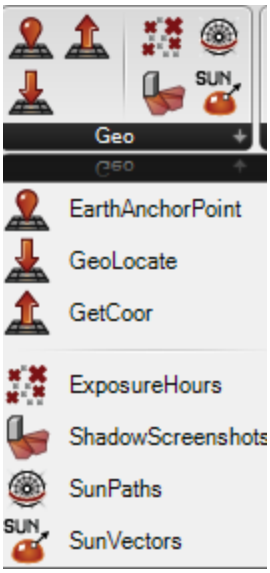
Pin - DropPin

Generates a mesh preview at a particular location. Used to quickly display any location-centric point data. Component takes a location parameter (L) as points, a height (H) and girth (G). It returns a mesh object as the pin (P). Right click to access context menu to choose from a few preview geometries.

Stairs - FireStairs

Makes a few geometric elements representing stairs going up a unit floor within the given shaft opening boundary. R input is the rectangular footprint of the shaft opening. H input takes a number representing floor to floor height. T is the transverse boolean that toggles which direction the stairs run. C outputs the profile of the steps as polylines. S outputs steps as polysurfaces. P outputs the stair steps as lines in plan.

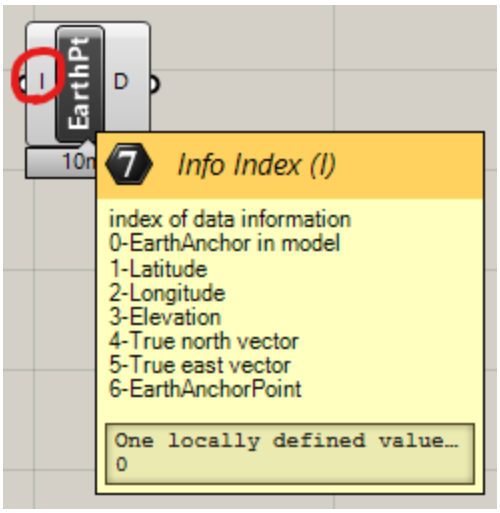
Geographic tools



Geography related tools (Geo) positions the model in the world and gets location specific information such as sunlight vectors. Many components of Geo tools are discussed in this clip https://youtu.be/9BCc5J4TPuo

EarthPt - EarthAnchorPoint

Extracts the EarthAnchorPoint object in the Rhino model. If it isn’t set up, no valid data will be returned. To set up EarthAnchorPoint properly, please refer to the official doc. Input index is the data to be extracted, output at D. See picture right for the list of indices and their corresponding info.



GeoLoc - GeoLocate

Locates a world coordinate in the Rhino model. Component takes in the latitude (LA) and longitude (LO) of a location, as well as the Rhino model EarthAnchorPoint (P) as reference. Returns the location point (L).

Coor - GetCoor

Translates model location back into world latitude and longitude. Input L takes the location point in the Rhino model and P takes the EarthAnchorPoint as reference. Outputs latitude at LA and longitude at LO.

SunPaths - SunPaths

Generate solar paths. Used in tandem with the “SunVec” component. Input P takes in an anchor point around which the sun revolves. S takes in the scale factor of the path geometries. T is a list of timestamps, ideally acquired from “SunVec” as it formats them nicely. Output L returns the location points of the sun. M the sun path across the sky. R the trails of sun migration across months.

ExpHrs - ExposureHours

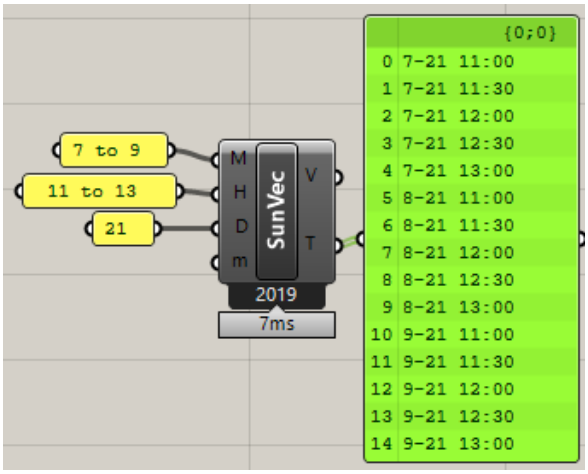
Computes the percentage of hours exposed to sunlight. Mostly used in tandem with SunVec and PolySrfSampling components. See https://youtu.be/df8ewQliouY for tutorial. V input takes in sun vectors. S is the sampling patches of a polysurface. O input is obstacles (often include subject itself too). Output P is the percentage in decimals, in the same order as input S patches.

SSS - ShadowScreenshot

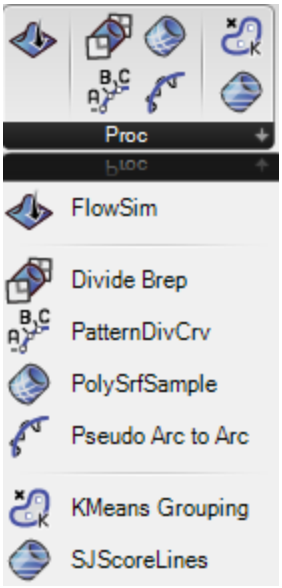
Outputs screenshots of current viewport (please set to a rendered display mode for shadows to show), each frame having the sun set to a particular timestamp. Input "T" takes a list of timestamps, usually from the "T" output of the SunVec component. Valid formats are "03/28/2011 09:16:59" or "12/02/1999 17:00:00", etc. "P" takes in the folder directory as text. "S" takes in a point object that has x value as frame width and y value as frame height. Right click on component and click "Dump!" to execute export. Windows explorer pointing to your directory should pop up afterwards. File names are the timestamps. An example of how to use this can be found here https://youtu.be/9BCc5J4TPuo

SunVec - SunVectors

Gets sun vectors given time parameters. This component queries the Rhino document sun for location accuracy so make sure the sun is set up correctly in the Rhino model. Input M takes a range representing months of the year (1 - 12). Input H takes a range representing hours of the day (0 - 23). This input can parse decimal texts. For example, "11.25 to 13.667" translates to 11:15am to 1:40pm. D input is the day of the month. Input m is the minute interval between each moment. Right click to access context menu when user can input a year parameter. *Beware that different years do produce slightly different results* as the motion of Earth around the Sun is not a perfectly regular cycle. Component returns vectors (V) and the timestamp associated with them (T).



Processing tools



Components that process (Proc) information such data manipulation or simulation

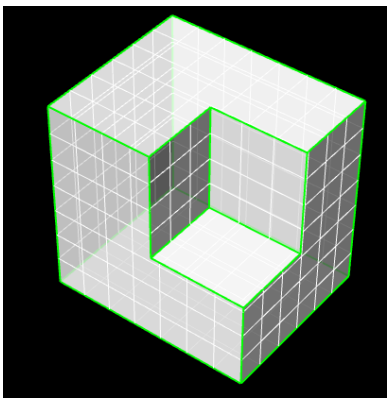
FlowSim - FlowSim

Agent-based simulation of fluid flow on a topography by solving geometric low points. NO physics engine built in. "T" input takes in a Brep geometry as topography. Should be "thin" piece. Closed volumes may not be handled correctly. "L" takes in a collection of points to be "dropped" onto the topography. "D" input specifies distance of drop, in effect the resolution of calculation. Smaller "D" value will increase run time and decrease noticeable change between iterations. However lower "D" produces better results at cliff drops. Output "P" returns the current positions of all point agents. "C" outputs the trace curves. "N" indicates how many iterations have been executed. Right click component to access context menu. Recompute of this component does NOT reset agents. Manual reset is required.

BrepDiv - Divide Brep
This component takes in a Brep (B) and cutters (P) as Planes. It returns the cut pieces of the Brep by the planes.

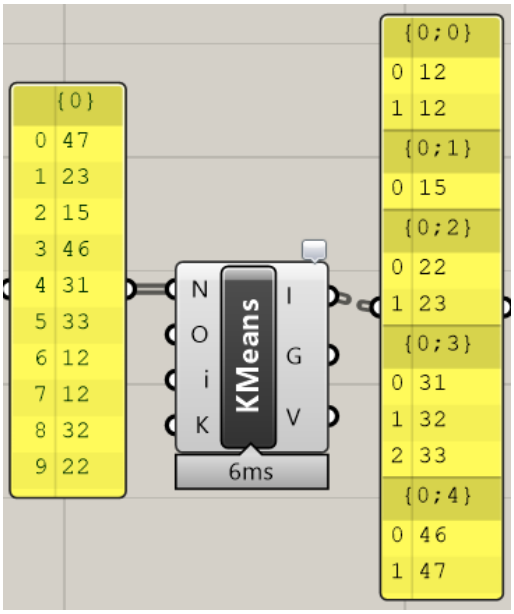
CrvDivP - PatternDivCrv
A component that divides curve by a length pattern. Input C takes the curve to divide. P takes in a string indicating the length patterns such as "5,6,12" meaning dividing the curve every 5 units and then 6, 12 and repeating the pattern. Output X returns division points while P returns curve parameters at those points. Right click on component to toggle between multithreaded or single thread computing.

PSSample - PolySrfSample
Generates sub-patches on Brep faces as a sampling mechanism. Input S takes a number as sample size. Beware that a small sample size produces a large number of patches that will drastically slow down interface. Investigate how large the object to be divided is first before inputting a test value. B input takes in a Brep. P output returns a list of all sub patches. The result patches will NOT be the exact square with the given size but very close estimates to evenly divide the polysurface.



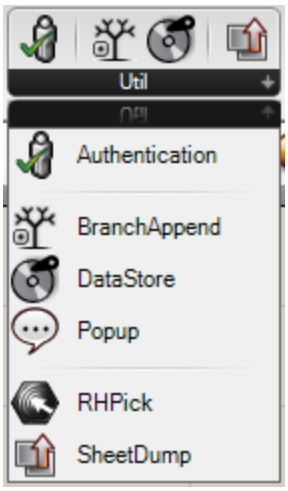
ToArc - PseudoArc to Arc
Turns messy curves into clean radii arcs. Component takes in the curve to clean up (C), the nearest unit to round to (N), and a boolean (F) to flip arc direction. It returns the clean arc (A) and the resultant clean radius (R).

KMeans - KMeans Grouping
Sort a list of objects into groups by the proximity of a metric. The foundation of the algorithm is discussed here. Component takes a list of number keys (N), optional list of objects (O) to group by the keys, number of iterations (i), and the number of groupings (K). First return value is grouped keys (I), second grouped objects (G), third the sum of variance of current grouping (V). This component works in a similar fashion as the native sort component. Higher i number means longer computing time but better grouping results.



SJSL- SJScoreLines
A component to etch score lines on a model in preparation for 3D printing. Input M takes in the Brep geometry. Shape should be closed! Input E takes in the elevations where the scoreline should happen. D takes in the dimension string of the groove such as "5,10" meaning 5 units high and 10 units deep. Output O returns the objects with grooves to be 3D printed and F returns the "floor lines" for troubleshoot. Right click to access the toggle between single thread and multithreaded computing.

Utility tools



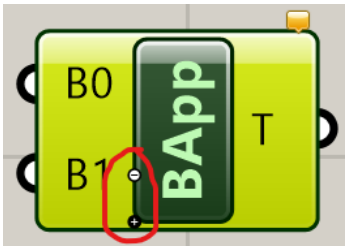
Suite of tools to improve user experience

Auth - Authentication

Use this component to authenticate the copy of PCPA_GH.gha and a free license key can be acquired by signing up at https://forms.gle/zDKhMZ1yJE6Nt4pC7. Use a button component for input "B" and feed license key into "K"

BApp - BranchAppend

Bunch up different data streams into branches of a single tree. Similar to the "merge" component but keeps each input on its own branch. Zoom in to activate expandable parameters. Each input is arbitrary data. "T" outputs the tree structure with each "BX" on a branch.



Save - DataStore

Tracks data by a key. T input is the tag associated with data and D input is the data stream itself. Output data at R. When flag is true, the data into D will be remembered with a tag into T. When flag is false, the component tries to retrieve data tagged. K output shows all keys present in storage. This component is commonly used in tandem with some of the components that have built-in randomization processes that are not driven by a retrievable "seed" parameter. Beware that large, complex geometries saved in this component will significantly increase Grasshopper file size.

Pop - PopUp

Calls up a dialog that requires user attention to dismiss. Input B is boolean to launch dialog. Input T collects texts to be displayed. The popup window requires action before the user can proceed to any other task.

Pick - RHPick

Pick an object in the Rhino model. Essentially the same as setting geometries in the parameter components. However this is geared towards user interfaces. Input x takes in the boolean value to initiate the pick. Link a button to "x" and not a toggle. Input T takes in the type of object to be picked. Right click on the parameter to select preset values. Output I returns the object GUID. This can be passed to any parameter component and GH will attempt converting it back to actual geometry linked.

Sheets - SheetDump

Exports either named views of layouts in Rhino model. "P" input takes in the directory path as text. "F" input takes in a list of names of the views or layout that should be skipped in export. "S" takes in a point object whose x and y value will be used as the dimensions of the export images. "N" outputs the list of names that have been found. Access context menu by right clicking the component, where the component can toggle between exporting named views (PNGs) or layouts (PDF).