Grasshopper Training

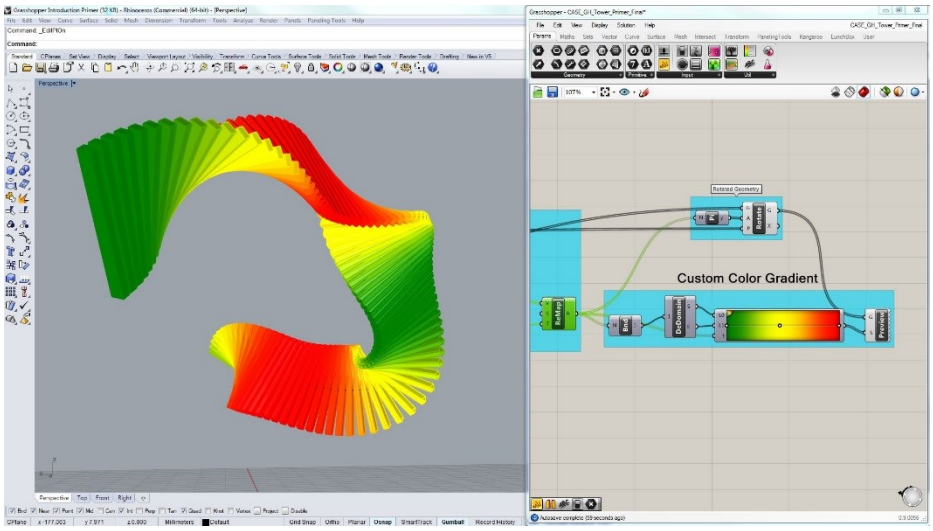
Summary

The Grasshopper plug-in training will demonstrate how the Grasshopper interface paired with Rhinoceros can be used as a tool to quickly conceive, modify, and develop parametric building forms with iterative controls. This training will also explain how this software can generate essential non-visual building data for use in analysis as well as 3D geometry.

Essential Knowledge Before You Begin Modeling

Creating in Grasshopper is different than modeling in Rhino. Rather than forming static shapes, Grasshopper allows users to create visually linked relationships of commands that can be modified parametrically. The following primer will introduce Grasshopper modeling procedures and essential interface tools. Understanding these processes will help you effectively navigate Grasshopper as you begin your first project.

Part 1 - Grasshopper Primer



Part 1 Learning Objectives

- Introduce Grasshopper and its user interface
- Demonstrate Grasshopper tools and navigation
- Show how Grasshopper procedurally creates form/data and how it differs from Rhino
- **Deliverable:** A series of numbers organized into lists
    - Organized data trees
    - Boxes scaled using an attractor script
    - Lines shifted around a radius
    - A simple definition with an arrayed series of boxes along a curve

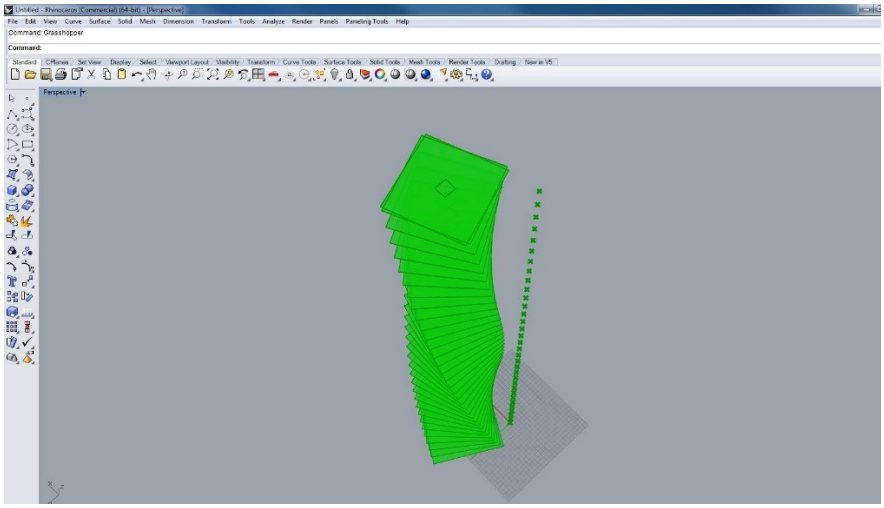
Part 1 Itinerary

- Launch Grasshopper from Rhinoceros
- The Grasshopper interface will show the basic canvas and a series of toolbars and components tabs

    - Top of interface is a familiar set of menus with options for displays, opening saving files, etc
    - Basic components that you will use to create the grasshopper definition
    - Components organized into tabs each representing a specific type or attribute
- Open menu options for components
- Open input and output menu for components
- Center clicking the scroll wheel shows expanded resources including several options for navigating the canvas
    - Search command finds components matching the search query and highlights them in yellow on the screen
    - Navigate command shows a miniature version of the entire work canvas where you can quickly pan through and zoom to areas of the definition
    - Access preferences menu and change look of interface
- Linking components together
    - Each component has inputs and outputs. Data is fed into the inputs on the left side, then the processed data is organized into outputs on the right side
    - Wiring multiple components together creates a procedural chain of commands
    - Click and hold down on an output and drag to an input
    - Move linked components on the canvas
    - Detach one component from another; hold the control key down and drag the wire between the two linked components, this will sever the wire and separate the commands
- Although many commands in Grasshopper are relatively intuitive, some of the commands we will be using in the next exercises require extra explanation
    - Range is a component that divides a set amount by a number of steps. This component is used to create a list of numbers that can be plugged into other values
    - Series operates similarly to range, a list of numbers with an established step size and step count is created
    - List is a series of related components that are used to organize and extract data
    - Remap is a command that will be used frequently in the next exercises. It takes a set of values and redefines them in a new numeric domain. For instance, you might use this tool if you had a set of evenly distributed values from 1 to 100 and you wanted the series changed to a domain of evenly distributed values from 1 to 40
- Organizing lists
    - Create a series of numbers with an adjustable step size and count
    - Create list components including length, item, reverse, split
    - Link the series component to all the list components
    - Wire adjustable slider panels
    - Attach a panel to all the list components and see how the outputs differ for each list type
- Using trees to organize data
    - Create a square grid
    - Create a box centered at each grid corner
    - Create a parameter viewer component and view the data structure of the boxes
    - Explode each box into its component parts
    - Link the vertices to a another parameter viewer
    - Find the number of boxes and place a tag labelling each box at its center point
    - Extract the edges of each box in the grid
    - Extract a single box from the grid
    - Find the corner points of the box and add a tag to each box corner point
    - Double click on the viewer and change the way the data is visualized
    - How is the data from the grid of boxes organized differently in each tree and panel, how are the data types similar
- Create an attractor system
    - Create a square grid with adjustable rows, columns, and grid size
    - Create a point in rhino and reference it in Grasshopper
    - At each grid intersection point create a box
    - Create a mathematic equation to calculate the distance between each box and the referenced point
    - These referenced distances dictate the size of each box in the grid system
- Offsetting items in a list
    - Create a circle with an adjustable radius

    - Offset the circle with an adjustable spacing between the two circles
    - Divide the top and bottom circle into a series of equally spaced points
    - Connect the top and bottom points with a line
    - Create a shift component and link to the points on the upper circle
    - Use an adjustable slider component to make the amount of offset adjustable
    - Wrap and unwrap the shift
- Use graph mapper to modify geometry
    - Create a curve in Rhino and reference it in Grasshopper
    - Create perpendicular frames along the curve
    - Set a slider to control the number of frames
    - Create a box centered on each frame with adjustable dimensions
    - Use domain to create the lower and upper bounds of each object
    - The domain start values need to be multiplied by negative one to center correctly on the line. Do this by writing a custom expression(A*-1)
    - Use a set of components with graph mapper and rotate the boxes around their center axis
    - Create a range of numbers the same length as the number of boxes on the line
    - Link the range to the graph mapper component
    - Take the values generated from the graph mapper and remap them with an adjustable domain for more control over rotation
    - Convert the output into a factor of pi
    - Create a rotate component and link the boxes on the line with the perpendicular rotation frames and the remapped rotation angles
    - Link the rotated objects to a set of components giving it a custom color gradient
    - Created a preview shader component and link it to the rotated geometry
    - Find the bounds of the remapped numbers
    - Deconstruct domain into upper and lower bounds
    - Wire the upper and lower bounds into a gradient component
    - Wire the remapped numbers to the gradient range parameters
    - Connect the color gradient output preview shader override
- By default Grasshopper computes components as they are placed in the definition, however the solving mechanism can be turned off in two ways. Turning off definition computation becomes very helpful as the definition become more complex and processing times go up
    - Disabled individual components
    - Disable solver in the solutions tab
- After you have created a functioning series of components the canvas may begin to look cluttered. To fix this there are several organizational tools and commands
    - Align and distribute options when selecting multiple components
    - Scribble component
    - Grouping highlights and color codes portions of the definition
    - Components can also be relabeled directly to show what their function is.
    - Panel to display values and additional notes
- Normally previewed materials are green when selected and red when unselected, but you can change visualization options as well as add custom colors to previewed materials
    - Turn preview on or off
    - Disabling a component
    - Preview geometry toolbar that has different options for changing type of preview
    - Disable preview on multiple components
- Bake processed components
- Saving definition
    - .ghx file is a larger but more stable file size
    - .gh is a smaller but more fragile file type
- Saving states within definition
    - State manager inside toolbar
- These steps outline the basic functions needed to collect and process data within Grasshopper. As your abilities with Grasshopper expand you will discover new tips and tricks to streamline the design process

PART 2 – The Fundamental Parametric System



Part 2 Learning Objectives

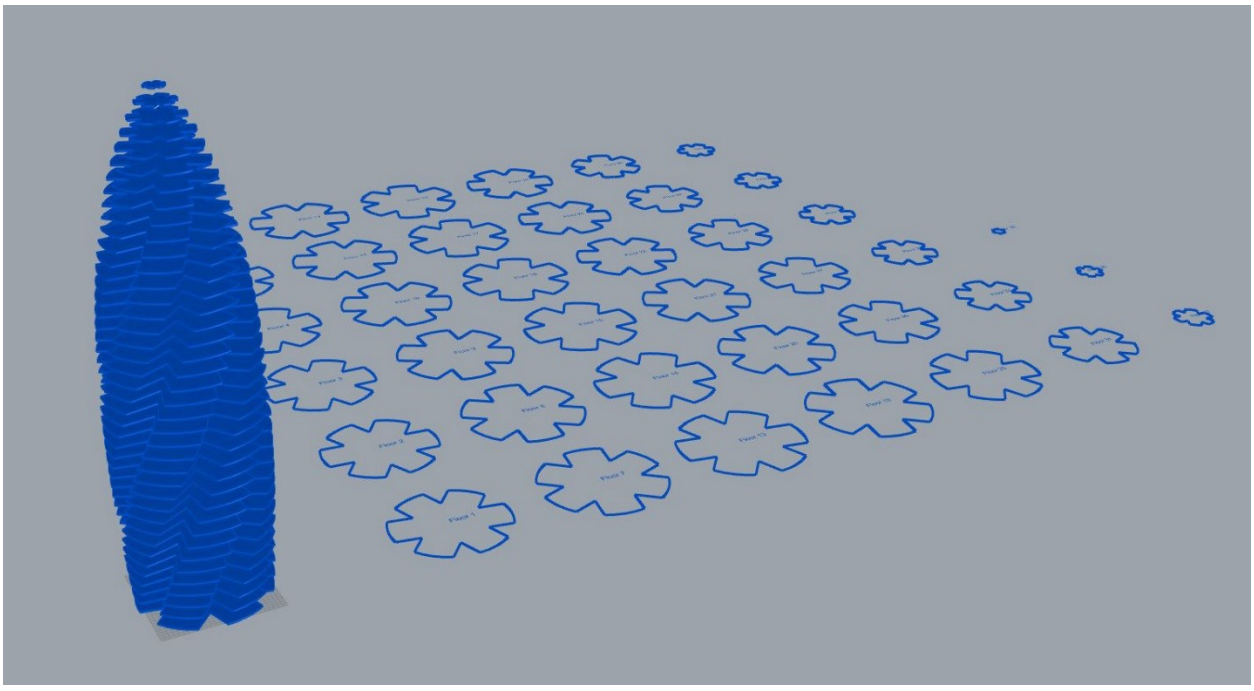
- Create a simple model with dynamic controls
- Introduce modeling concepts with procedural options for building shapes and controls for transforming them parametrically
- Saving states, definitions, and baking geometry into the parent Rhino model
- Extract useful non-visual data from the modeled geometry
- **Deliverable:** Rectangular tower iterations demonstrating parametric form
    - Components for controlling shape, floor height, and building rotation
    - Screen capture to clipboard, file, etc.
    - Capturing these iterations to different states within the file

Part 2 Itinerary

- Open Grasshopper plug-in
- Create series component
- Number slider component
    - Label
    - Set domain
    - Connect to series
- Copy/paste components
    - Disconnect copied components
- Panel component
    - The panel can be attached to any command to check the data coming out of it
- Construct points along the z axis in a straight line
- PlaneXY component
    - Connect to the point component
    - This function will place a horizontal plane at each point location
- Connect the horizontal planes to the rectangle component
    - set a slider for its x and y dimensions
- Use the boundary component and create a surface from each rectangle
- Using area centroid, find the center point of each surface

- Using the transformation Parameters
    - Connect the “floor number” slider to the range component
    - Set the range domain from zero to one
    - Connect the range to a graph mapper component
    - Drag a scale component on to the canvas and connect it to the floor planes
    - Connect the center point input to the area centroid output
    - Connect the scale factor to the graph mapper output you just created
    - The floor size should scale as you move the curve in the graph mapper command
    - Connect the geometry created with the scale command to a rotate component
    - Center of the rotation plane should be input from the area centroid
    - The rotation angle is determined by linking a range component to a curve mapper
    - Fix the domain of each at 0 to 360
    - Convert values to radians and wire into the rotate command
- Definition now has adjustable scale, rotation, total height, and floor to floor height
- Link floor geometry to an area command
- Connect area output to a panel
- Deconstruct floor surface center points into x, y, and z components
- Extrude floor surfaces in the z vector to produce solid floors
- Build elevator shaft
    - Use list to extract a start point for the elevator shaft
    - Create a rectangle centered on this point
    - Extrude the rectangle in the z direction. Make sure that the elevator shaft is taller than the highest floor height.
- Boolean intersection creates the final form
- Bake geometry
- Use the state manager to save different definition states
    - Move back and forth between different saved states

PART 3 – Morphing and Documenting Geometry



Part 3 Learning Objectives

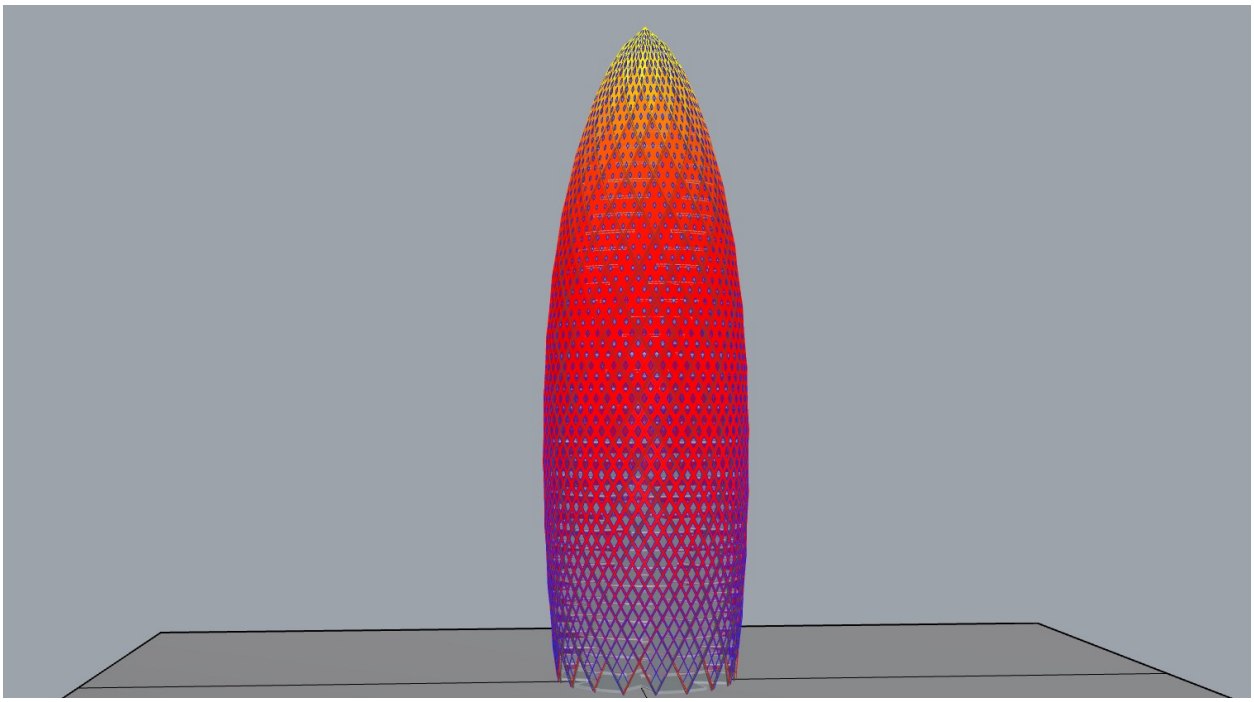
- Create a model that references geometry created in Rhinoceros
- Introduce more controls for analyzing geometry and transforming it parametrically
- Extract useful non-visual data from the modeled geometry
- Extract useful building plans from the geometry
- **Deliverable:** Advanced Gherkin tower iterations demonstrating parametric form and referenced geometry
    - Sliders and controls for changing building profile, height, and rotation
    - Updating building data, exporting to Excel
    - Organized, labeled floors exported to different file formats

Part 3 Itinerary

- In Rhino import reference drawings of the Gherkin
- In Grasshopper create a referenced point at the origin
- Create a line segment using the referenced point as a start point
    - Extrude the line in the z direction
    - Give the line an adjustable length
- Divide the line into an adjustable series of segments defined by their division points
- Draw an adjustable horizontal line at each division point
    - Create a range of numbers linked to division points
    - Link the range of numbers to a graph mapper component to produce a range of values between zero and one
    - Remap the values
    - Set the remapped domain between 0 and the maximum building width of 120 feet
- Extract the end points of each horizontal curve
- Create an interpolated curve from the line endpoints
- Revolve a surface from the interpolated curve
- Contour the surface horizontally with an adjustable floor to floor height
- Divide each contour in half
- Create a line across the diameter of each contour connecting the endpoints
- In Rhino import a referenced floor surface and diameter line
    - Link this geometry to a referenced brep and curve in Grasshopper respectively
    - Evaluate the floor diameter and contour diameter at their center points
- Orient the floor geometry to each contour diameter center point
- Create a scale component
    - Link the oriented floor planes to the scale geometry component
    - The center of scaling is the center point of each floor diameter
    - Remap the floor diameter to match the contour diameter
    - Use these remapped numbers as the scaling factor for the floor planes
- Create a rotate component
    - Link the scaled geometry to the rotate component
    - Create a series of numbers equal to the number of floors
    - Have each number in the series increase by five for each successive number
    - Convert the numbers in the series to radians and link to the rotation angle
    - Link the rotation plane to the diameter center point
- Extrude the scaled rotated planes into solid floors
- Extract the area and center point from each floor plane
- Link the area output to a panel component
- Deconstruct each center point into its x, y, and z components
- Link the z component to a panel to show the elevation of each floor
- Create a panel component and create a heading with area and elevation
- Link the heading, area, and elevation to an excelwrite component from the Lunchbox plugin
- Create an toggle switch and link it to the update excel input
    - This toggle will export the building data to an open excel document when the turned to “true”

- Open a new Microsoft excel document
- In Grasshopper turn the toggle to “true”. The excel document should fill with the data exported from the Grasshopper panels
- Save the excel document
- Create a floor layout
- Create a square grid
    - Create adjustable grid cells
    - Create adjustable rows and columns for the grid layout
- Deconstruct the floors into their constituent parts
    - Join the edges together
    - Create the floor layouts
- Find the center point of each layout
- Orient each floor layout to a point on the square grid
- On the oriented layout create a custom tag labeling each floor number
    - Find the center point of each arrayed floor layout
    - Create a series of numbers
    - Link the series of numbers to the concatenate command
    - Plug a panel component reading “floor” into the a input
    - Link the combined fragments to a tag component
    - Place a tag at each floor center location
- Bake the tagged floor layouts into Rhino
- In Rhino select the floor layouts and export the curves as an Adobe Illustrator file
- Export the 2D layout as an AutoCAD file

PART 4 – Defining the Tower Facade



Part 4 Learning Objectives

- Elaborate on the model produced in the last session with a parametric façade
- Introduce the Paneling Tools plugin and use its menu of advanced components for façade paneling
- Create an attraction based component and create a variable façade surface
- Evaluate the panelized surfaces and export the façade data to Excel
- Introduce custom color preview options for analyzing data
- **Deliverable:** Advanced Gherkin tower iterations demonstrating parametric form and referenced geometry
    - Paneled façade surface
    - Custom previewed components
    - Rendered model image
    - Exported building data

Part 4 Itinerary

- This exercise focuses on building skin/façade
- Overall building form, floor orientation, and spacing has been developed in the previous exercise
- Link a brep to the revolved surface
- Offset the brep with another surface that will be used for the paneling façade
- Create a diagrid surface on the brep
    - Create adjustable UV divisions
- Divide the diamond panels into quad panels
- Create a diagrid structure on the original brep for the structural grid
- Use area centroid to find the center point of each quad panel
- Create an attractor wave component
    - Link the targets to each quad panel center point
    - Set the attractor wave domain from zero to one

    - Apply an adjustable slider to the frequency input
    - Create points in Rhino and reference them to point components in Grasshopper
    - Link these to the attractor point inputs
- Create a frame component on each quad panel
    - Link the scale component to the values generated from the attractor wave
    - Link the frames to a brep
- Deconstruct the values from the attractor wave into start and end domains
- Create a gradient component and link the attractor wave values to the gradient component inputs
- Create a custom preview shader And link it to the façade frames and the gradient component
- Link the diagrid lines on the original façade to a pipe component
    - The radius of the pipes is generated by elevation
- Create a custom preview component where the color is dictates by the pipe elevation
- Format the data generated from the structural grid
    - Extract the start and end points of each structural segment
    - Measure the length of each segment
    - Use concatenate and series to create a name for each segment
- Load the Lunchbox component 'Excel Write' and format the data for export to an excel document
- Open Microsoft excel and export the Grasshopper data into a blank excel spreadsheet
- Save the file