Digital Built Week NA 2019
BILT NA
Hyatt Regency Seattle
18 – 20 July,2019

## Session 1.4

# Generative Design with Project Refinery

**Lilli Smith, Autodesk, Inc.**

AUTODESK

### Class Description

Project Refinery is an Autodesk generative design tool for the architecture, engineering and construction industry that gives users the ability to explore, analyze, and optimize their Dynamo and Revit designs. This lecture and demo will feature pragmatic end-to-end workflows for architectural design option generation and evaluation using Project Refinery in Dynamo for Revit. The Dynamo and Refinery project team will show how to frame a design problem in terms of goals and constraints in Dynamo, how to automate design option creation, evaluation, and optimization in Refinery, and how to convince stakeholders about the value of this approach.

We will focus on these workflows:

1. Massing Study Simple – a simple workflow to explain concepts.
2. Massing Study Complex - retail and office distribution and configuration for a building on an urban site.
3. Tiling – tile layout on a given surface from Revit so that the least number are cut off and the least amount of tile waste goes to the landfill.
4. Office furniture layout – desk layouts in a Revit room that provide the most desks but also the most private desks while maximizing desk area per person and minimizing unutilized space.

### About the Speaker:

Lilli Smith, AIA Sr. Product Manager, AEC Generative Design, is an architect with a passion for re-envisioning the way that buildings are designed. After working for several years as an architect, she joined Revit Technology as a fledgling start up and helped grow it to where it is today in almost every architect's tool box. She has gone on to work at Autodesk on many different tools including Vasari, FormIt, Dynamo, and Project Fractal. Her most recent focus is on Project Refinery: Multi Objective Optimization and Optioneering for Dynamo.

## Introduction

This class is about how to use generative design workflows to advance architectural processes.

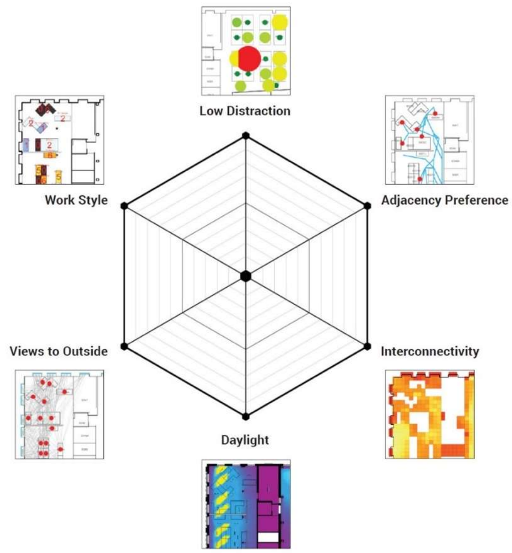
### Why learn about generative design workflows?

By 2050 there will be 10 billion people on earth. The global middle class is growing faster than at any time in our history. In 1990, only 23% of the global population had sufficient income to be classified as middle class. Today, 45% of us do. All of these people are going to drastically increase the population of our cities and this is going to require a lot of work: a lot of new buildings and infrastructure to support them. Recently Autodesk partnered with a firm called Statista to calculate just how much work. We're going to have to build 13,000 buildings every day and enough roads and rail to wrap around the earth six times every year. That's over 1.2 million km of roads and rail needed for our global infrastructure each and every year between now and 2050. I'm going to assert that there is a problem: *we need to produce more work at a higher quality with less resources*. In today's design processes, most people just evaluate a few designs and make decisions based on rules or thumb or instinct, generative design can let you study much more data driven designs to achieve better outcomes.

The traditional way of delivering buildings has been recording designs on paper and delivering the drawings to others who execute to build it. We've evolved to computerizing drawings making them more data rich and easier to update. But what we really want to do now is to combine intelligence of building professional with algorithmic intelligence of a computer to produce better buildings, faster. Autodesk Research has been experimenting with some ways to do this. The following studies are some examples of their work.

### Research Example 1: Autodesk MaRs Office Design in Toronto

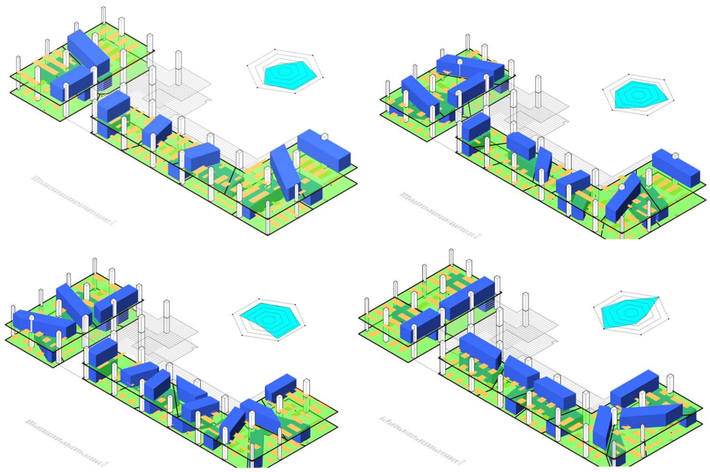
A few years ago, Autodesk moved our Toronto office to a new office in the MaRS innovation district. The Living, an architecture firm that Autodesk has incorporated into Autodesk Research to test out ideas, used a generative approach to design the new office's space layout. They began by surveying existing employees and asking them questions like “Who do you need to sit by?” “How much daylighting do you need to do your job?” “How much distraction can you handle?” When they had all of this data they identified 6 goals that they felt would make a “good” office.




***Autodesk MaRs office design goals***

~~They developed A~~algorithms were developed to measure these goals and tested the measurements on their existing office space. Here is an example of a way to measure low visual distraction. Note that the scoring equation is critical to the evaluation of the goals.

A ~~Next they built a~~ flexible model was built to generate designs that ~~they~~ could be tested against the goals ~~that they had developed~~.

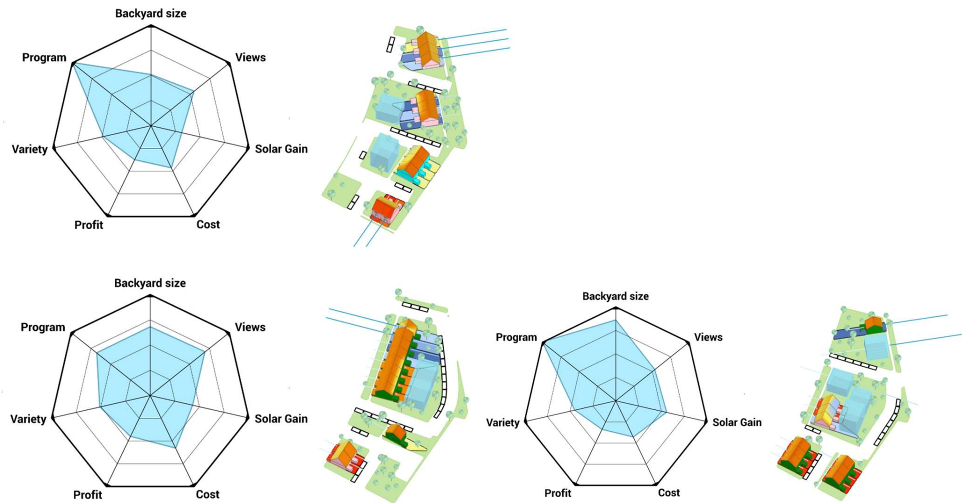


***Autodesk Mars Office Design Project***

Different floor plan layouts ~~are~~ were generated from the model including blue amenity bars of meeting rooms and yellow locations of desks. Each design option also ~~involves~~ feeds data into a "dashboard" ~~of~~ showing performance ~~according to its score on~~across the six goals. ~~They set up~~ V~~v~~ariables were set up so that designers could flex ~~them to~~the model and see what the effects of the variables are. ~~They generated~~ 10s of 1000s of options were created and ~~then developed~~ a methodology was developed to sort ~~through them~~ and help ~~them~~ make decisions. ~~They also used the computer and~~ O~~o~~ptimization algorithms were used to help ~~them~~ evolve the designs towards identified goals. By ~~choosing~~ clearly defining goals before the ~~designs~~ options were generated, the computer ~~could~~ was able to assist ~~help~~ the design team in ~~them~~ producing not just~~e~~ more designs, but designs that better ~~that~~ met ~~their~~ the stated goals.

### Research Example 2: Van Wijnen Neighborhood Design

Autodesk Research and The Living also worked on a project with the Dutch construction company Van Wijnen. In this project they followed a similar workflow where they identified certain goals, developed ways to measure those goals, developed a flexible model, and used the computer to help generate, evolve, and rank their designs.




***Van Wijnen Neighborhood design project***

*"We want to be able to learn from more designs than it is physically possible to generate or evaluate 'by hand'."* – Danil Nagy, The Living

### How can I use generative design workflows?

The tools that The Living used to do these studies were custom tools that were not available ~~for others to use~~to the public at the time. Autodesk is interested in democratizing ~~some of these tools~~this methodology and in making them accessible to more people. Project Refinery is a beta ~~tool~~ application ~~that~~ Autodesk is developing to help preform generative design workflows like these. Project Refinery:

- Automatically runs Dynamo created logic to create options
- Works on Dynamo for Revit or Dynamo Sandbox
- Created for one customer Van Wijnen in 2/2018
- Public Beta as of 11/2018
- Available at **autodesk.com/solutions/refinery-beta**

The outcome of an automatic design search is completely dependent on the ~~design~~ process used to create ~~of~~ the design space, as well as the specification of various measures and search parameters. So how do you ~~design~~ create a good design space? The traditional architectural design process is similar to the generative design process in many ways. In the traditional architectural design process, you ask for requirements from your client – how big a building do you want? What kinds of spaces to you need in the building? What goals do you have for the design?

This traditional process of identifying goals and requirements is still important when incorporating generative design techniques. With generative design you usually can't design all aspects of a building in one automated process, you have to identify what kinds of problems might be good to look at in ways that the computer can help you to generate options and evaluate them, but you are still defining goals and measuring success. It's critical to design how the problem with be generated by the computer and how you will measure success. How are you going to make decisions on the results? The first step is good preparation:

- Understand what you want to achieve
- Define your problem
- Decide on ways to measure success
- Think about how to review the results

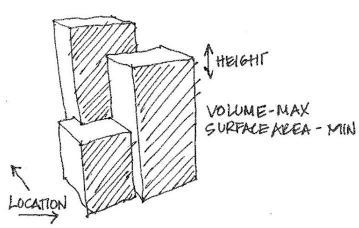
Traditional Design process is a little bit like a game of battleship. Ideas are tested one at a time. "Does this work?" "What about this?" You might get a hit, you might get a miss. Often times only a few designs can really be evaluated because you run out of time. Solutions may be presented based on cost only when you really want to show the client more options. With generative design techniques, people and computers work together to created and evaluate more designs.

*"The goal of generative design is not to automate the design process, or to replace human designers with artificial ones"* – Danil Nagy, The Living

### Example 1: Simple Massing Study

Let's see how some examples of design logic can be created in Dynamo and run through Refinery to learn about generative design workflows. If you are unfamiliar with Dynamo, you can learn more in this Dynamo Primer: https://primer.dynamobim.org/

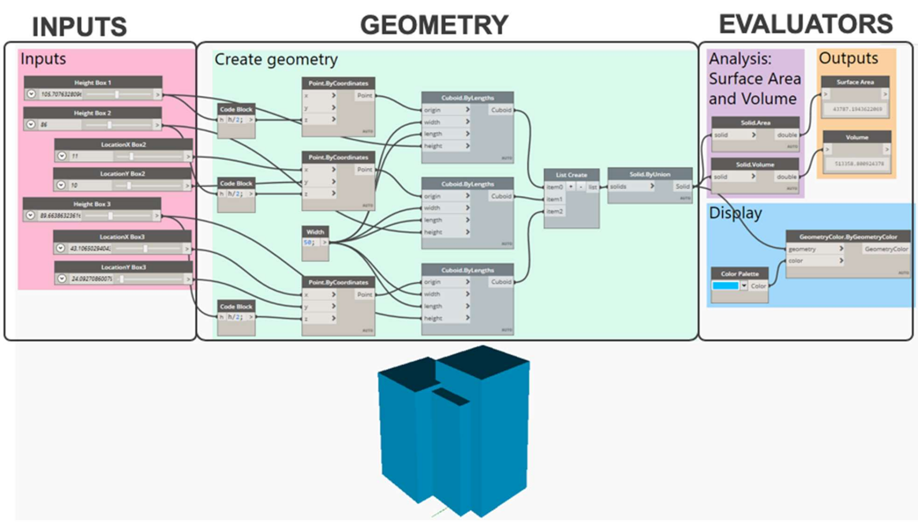
The first step with any generative design workflow is to identify what problem we are solving for. In this case we want to study a building mass to look at trade-offs between volume and surface area.



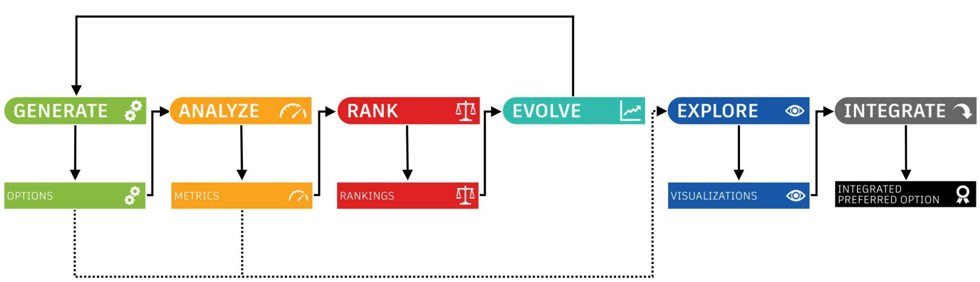

- Variable Inputs:
    1. Locations of 2 of the masses
    2. Heights of all 3 masses
- Goals:
    1. Maximum volume (rentable area)

2. Minimum surface area (expensive façade)

The Dynamo graph for this study is broken down into 3 sections:




Open the 3Box.dyn sample file in Dynamo and move the sliders in the Inputs section to understand how they are controlling the model. Next, look at the Geometry section to see how the geometry is being created with points and cuboid nodes. Finally, the Evaluator section contains nodes that measure surface area and volume, watches the outputs, and controls the display color of the geometry. There are several stages to the generative design workflow:




***Generative design workflow***

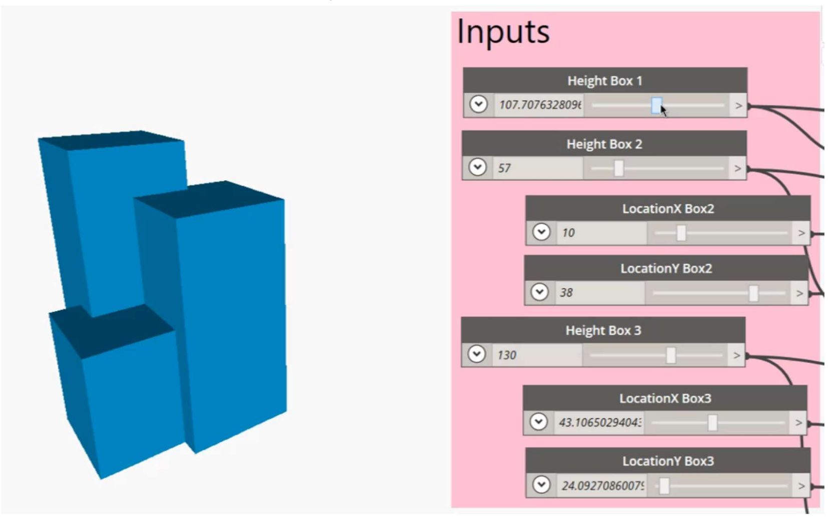
Let's look at some key concepts in each stage.

### Generate

Create a flexible design system in Dynamo to generate designs.
3 key concepts:

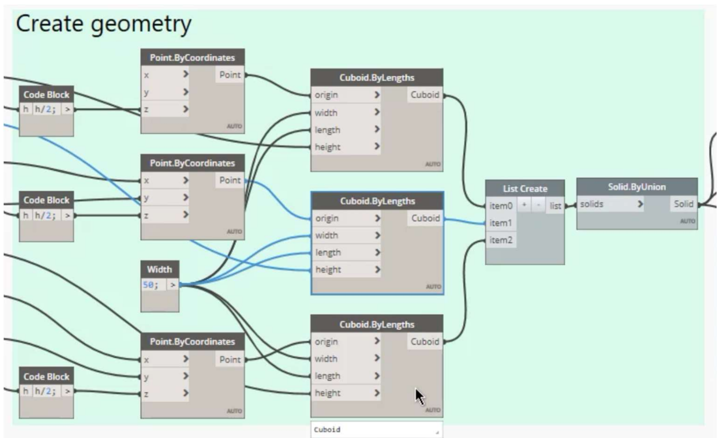
- How you parametrize the model is critical
- Too many inputs will lead to a design space that is too big to explore
- Too few inputs won't yield a big enough design space

The flexible system in the 3Box.dyn model is very simple. The Inputs in the pink group control point locations for the cuboids in the geometry section of green groups. As well as the cuboid locations, the inputs also control the height of the cuboids.



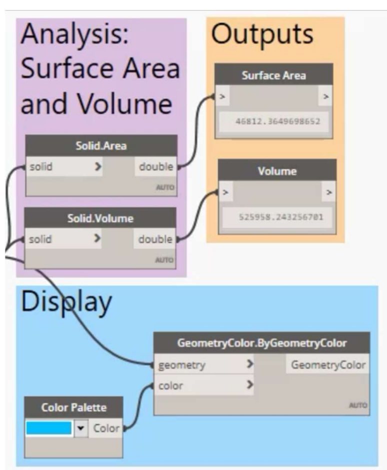

***3Box.dyn Inputs***

In the green geometry creation section, there are 3 points, that place 3 cuboids. Once placed, the 3 cuboids are unioned together to form a single piece of geometry.




***Geometry creation section of 3Box.dyn***

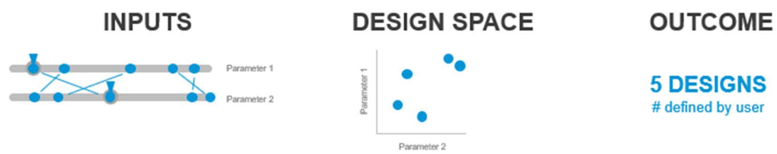
The next sections involve analyzing the resultant geometry for surface area and volume, watching the outputs, and controlling the display color of the geometry.



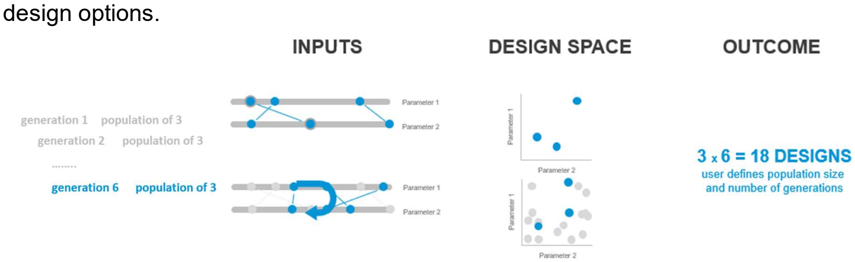

***Analysis, Display and Outputs of 3Box.dyn***

Now that we have a design system defined, we can use Refinery to automate or generate the creation of design alternatives that exist in the design system. Generation methods in Refinery include:

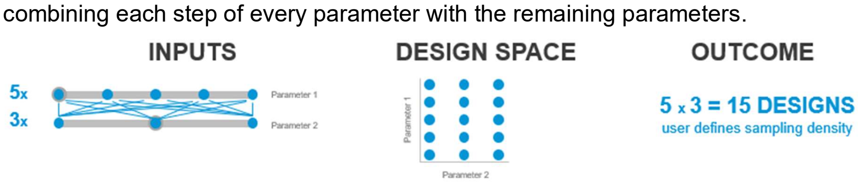
1. **Randomize** When Refinery uses the Randomize option it will generate a specified number of design options, by randomly assigning a value to each of the input parameters. This option is used when facilitating an optioneering process.



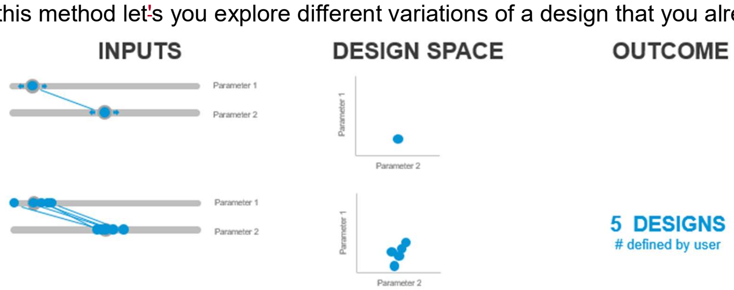

2. **Optimize** is the method for doing an optimization run with Refinery. During an optimization run Refinery will evolve the design based on the evaluators outputs. The optimization process works by running multiple generations of a design, each generation will use the input configuration from previous generations and from that optimize the new design options.




3. **Cross Product** method let's you explore the entire design space of your design, by combining each step of every parameter with the remaining parameters.



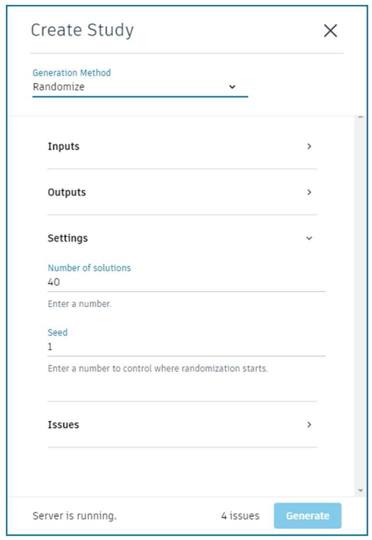

4. **Like This** will make Refinery apply slight variations to your current input configuration. Using this method let's you explore different variations of a design that you already like.




Refinery can be run using the different methods above. In the Refinery window you can choose from the four different methods.

**How to run an optioneering process using Refinery**

An optioneering process lets you explore all possible solutions that the graph can produce. This is helpful if you want to test the flexibility of the graph or you're unsure about what to optimize for. Refinery will generate the solutions based on the constraints that were defined in the Dynamo graph. To run an optioneering process in Refinery, follow these steps:

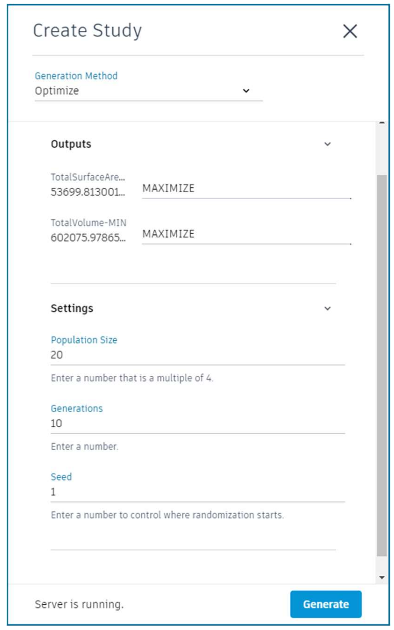



1. Launch Refinery from the Refinery menu in Dynamo
2. Create a new study and select Randomize, Cross Product or Like This as the generation method
3. Under Inputs make sure that all the desired inputs are present
4. For inputs that should not change on each run, set the desired value and uncheck the box alongside it
5. Under Outputs ensure each output defined in the graph is listed
6. Under Settings choose how many options Refinery should create
7. Under Settings select a random seed (number) to initialize the randomization
8. Under Issues resolve any items
9. Click Generate

**How to run an optimization process using Refinery**

An optimization process lets Refinery evolve your design to find the most suitable options based on the constraints and goals provided. Refinery will run multiple generations of options and each time it will take the fittest (best) options of the generation and use them to create a new generation. Refinery is using NSGA-II, an 'elitist' multi objective genetic algorithm to optimize results. To run an optimization process in Refinery, follow these steps:

1. Open Refinery from the Refinery menu in Dynamo
2. Create a new study and select Optimize as the generation method
3. Under Inputs make sure that all the desired inputs are present
4. For inputs that should not change on each run, set the desired value and uncheck the box alongside it
5. Under Outputs go through each objective and set the optimization goal you want to achieve - Maximize, Minimize, or Ignore
6. Under Settings set a population size, which represents the number of options that will be created in each generation.
7. Under Settings set the amount of generations you want to create. Each new generation is a range of options that falls between the two best designs of the previous generation
8. Click Generate

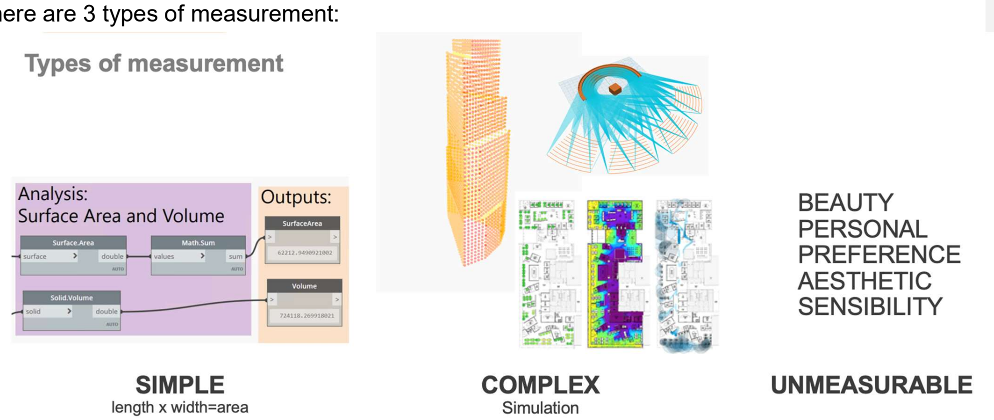



### Analyze

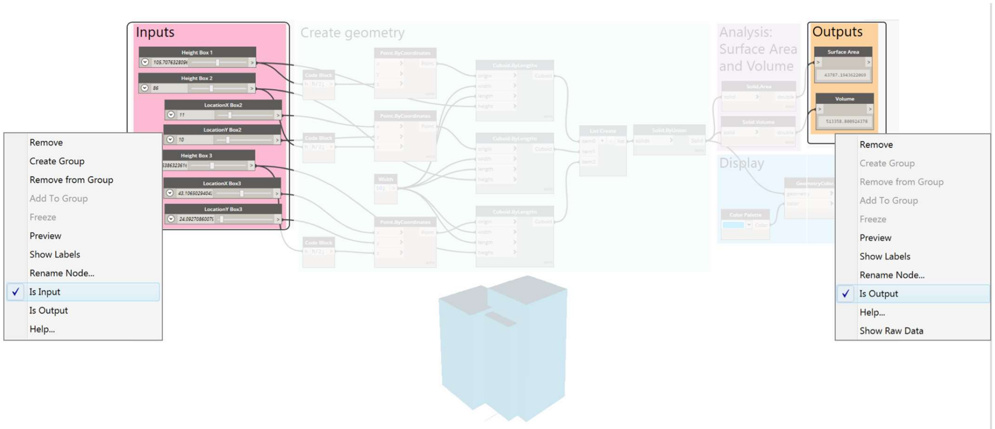
Judging outcomes is the critical 2nd half of the problem. To do this you can use a set of measures that determine how the building is performing. Why?

- Compare designs objectively (apples to apples)
- Evaluate designs based on non-intuitive measures
- Explore more designs than is possible manually

There are 3 types of measurement:

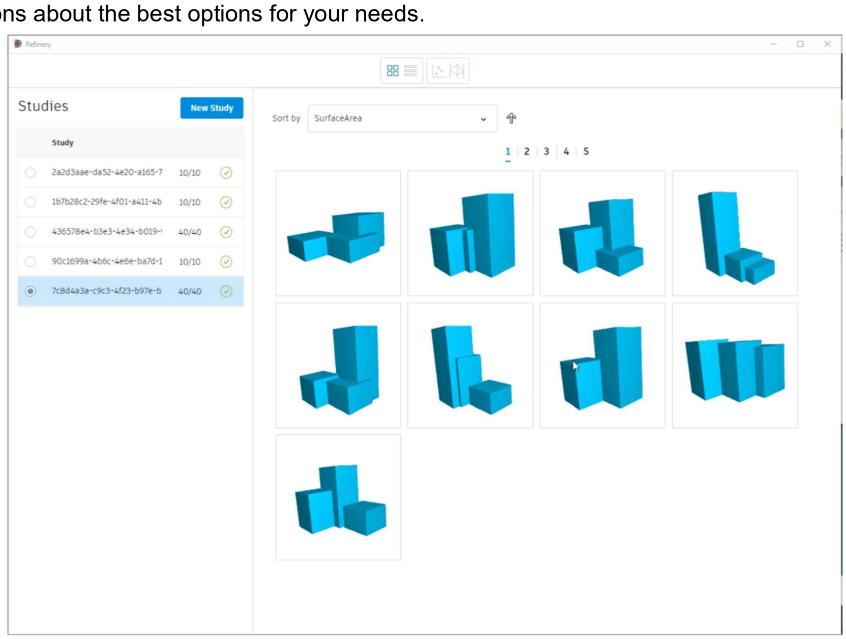


Designating inputs and outputs in Dynamo for use in Refinery:




## Rank

In Refinery, you can use sorting and filtering to rank resultant design options and make decisions about the best options for your needs.



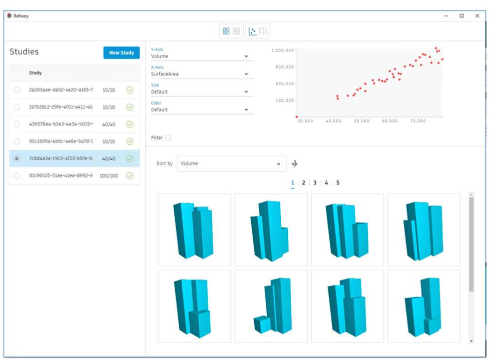

***Refinery designs sorted by Surface Area***

### Evolve

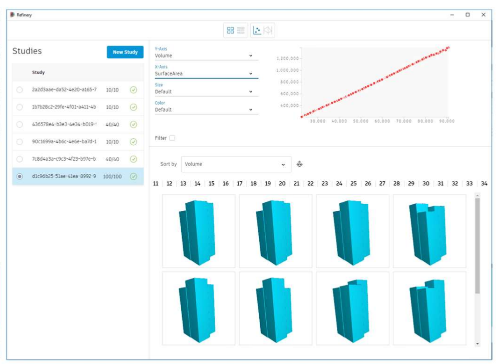
After you have explored some possibilities in the design space that you have set up in Dynamo, you may want to set some goals for the design to guide the design generation and produce options that meet your goals. To do this, you can use the Optimize generation method. There are several parameters that effect the optimization

- Output goals (maximized or minimized)
- Population size (how many designs should be tested in each generation?)
- Generation size (how many generations should be tested?)
- Seed (initialization value)

Optimization runs can take longer, but will return only results that meet your goals. The runs below show the difference between the Random method and the Optimization method. The scatter plot charts below are both set to Volume on the Y-axis and Surface Area on the X-axis. Note that in the optimization run on the right, the designs return look much more similar and form a tighter line comparing volume and surface area.




40 random runs – 10 sec



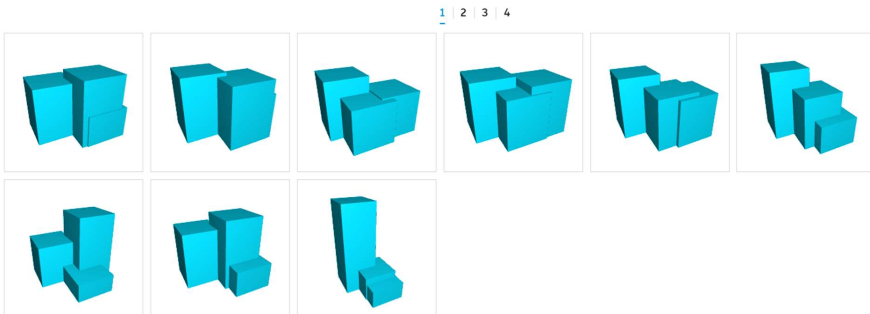

100x100 optimization – 4 min

### Explore

After Refinery has run the generative process, the results are displayed in both geometric form and through a series of charts or tables. All of the resulting views are interlinked and selecting an option in one view will highlight it in the other views currently displayed. If Dynamo is running in Automatic mode in the background, selecting an option will also update the graph to show this design. There are several features of the explore interface:

1. **Design Grid** The design grid shows each option as a 3d geometrical thumbnail that can be individually rotated, zoomed and panned to explore the design in more detail. The

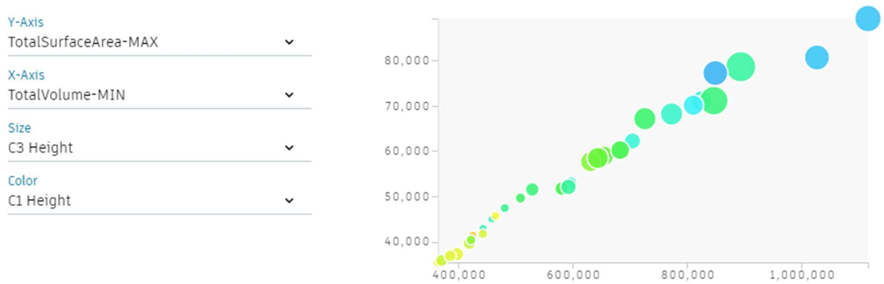
order of the thumbnails can be sorted based on the inputs or outputs of the Dynamo script, with a toggle for both ascending and descending values.



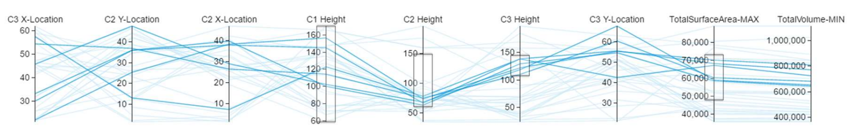

2. **Design Table T**he design table replaces the design grid, if chosen, and lists each option in a table form with each column representing the values for the inputs and outputs.

| TotalSurfaceArea-MAX | TotalVolume-MIN | C3 X-Location | C2 Y-Location | C2 X-Location | C1 Height | C2 Height | C3 Height | C3 Y-Location |
|---|---|---|---|---|---|---|---|---|
| 41456.712 | 454217.178 | 41.954 | 24.791 | 11.654 | 61.331 | 59.381 | 102.729 | 34.832 |
| 41578.589 | 424163.121 | 47.119 | 46.957 | 19.060 | 58.759 | 95.981 | 25.227 | 31.460 |
| 48569.928 | 529109.489 | 26.810 | 42.372 | 38.189 | 65.695 | 109.178 | 100.908 | 52.748 |
| 57784.215 | 630028.715 | 21.829 | 12.137 | 39.870 | 94.405 | 51.141 | 118.784 | 55.295 |
| 65351.505 | 805919.903 | 53.759 | 18.735 | 21.948 | 111.373 | 160.120 | 113.769 | 39.955 |
| 69265.414 | 876697.721 | 42.328 | 36.077 | 11.438 | 190.133 | 182.257 | 25.506 | 21.742 |
| 76891.400 | 986952.474 | 53.550 | 15.168 | 29.379 | 178.812 | 172.316 | 110.951 | 49.648 |
| 77314.139 | 848258.692 | 59.459 | 4.693 | 1.417 | 170.390 | 97.898 | 157.210 | 20.587 |
| 77508.007 | 993015.152 | 26.138 | 16.635 | 36.074 | 147.930 | 188.559 | 95.979 | 61.584 |
| 78710.577 | 892655.056 | 21.179 | 41.879 | 27.823 | 135.508 | 53.463 | 198.658 | 63.065 |

3. **Scatterplot** The first chart Refinery uses to visualize data is a Scatterplot : a type of mathematical diagram that uses cartesian coordinates to display values across a set of data. Refinery allows you to select what values are displayed along both the X and Y axis as well as which ones drive the size and color of the circles in this 4-dimensional view. The values can be chosen from the inputs or outputs you defined in the Dynamo graph in the previous steps. Selecting a circle from the graph space will also highlight the chosen option in the design grid or design table.



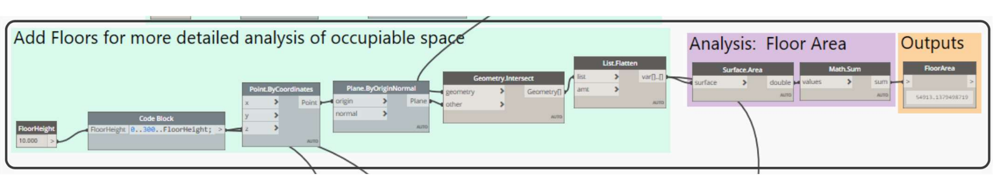

4. **Parallel Coordinates T**he other chart available in Refinery is a Parallel Coordinates graph. This chart shows a set of vertical parallel lines, equally spaced, that represent the inputs and outputs. Each design option is represented as a polyline whose vertices sit on each parallel axis. The position of the polylines vertices on the axis corresponds to the value of the input or output. The graph can be filtered by dragging the selection on each vertical axis.



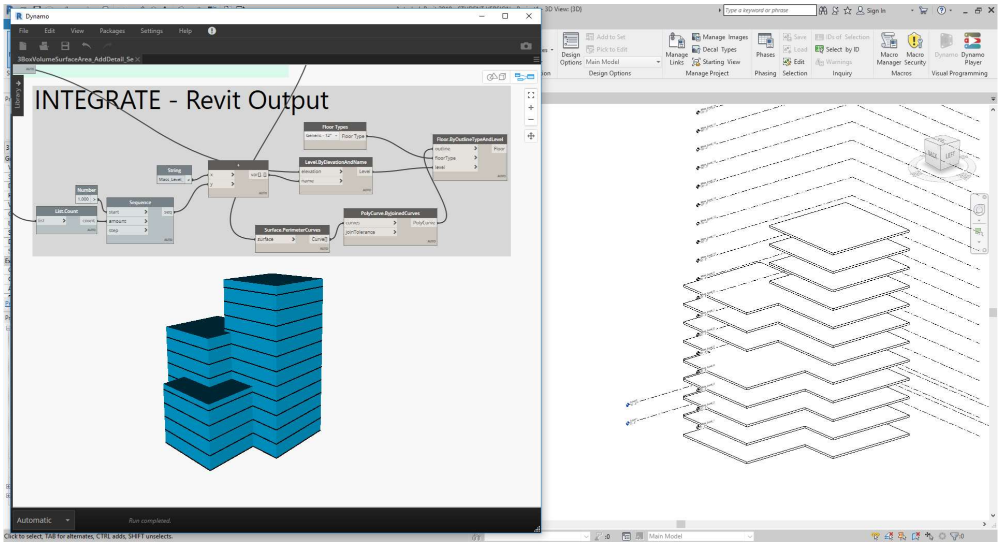

The kind of visualization you choose for your project may vary depending on what kind of process you are running. If you are running an optioneering process, it can be beneficial to visualize it using a parallel coordinates chart as it will be easier to filter options, while visualizing a multi-objective optimization using a scatterplot chart will make it easier to find the best trade-off between two objectives.

### Integrate

The final step in the generative design workflow is to integrate your chosen design into a context where you can develop it further. One way to do this is to add a branch to your Dynamo graph which places Revit elements. In this example we're first adding floors to the Dynamo model for a more detailed analysis of the occupiable space.



And then we can add those same floors to the Revit model.

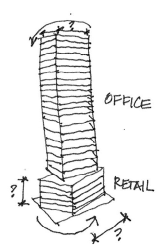



***Integrate design data into Revit as floors.***

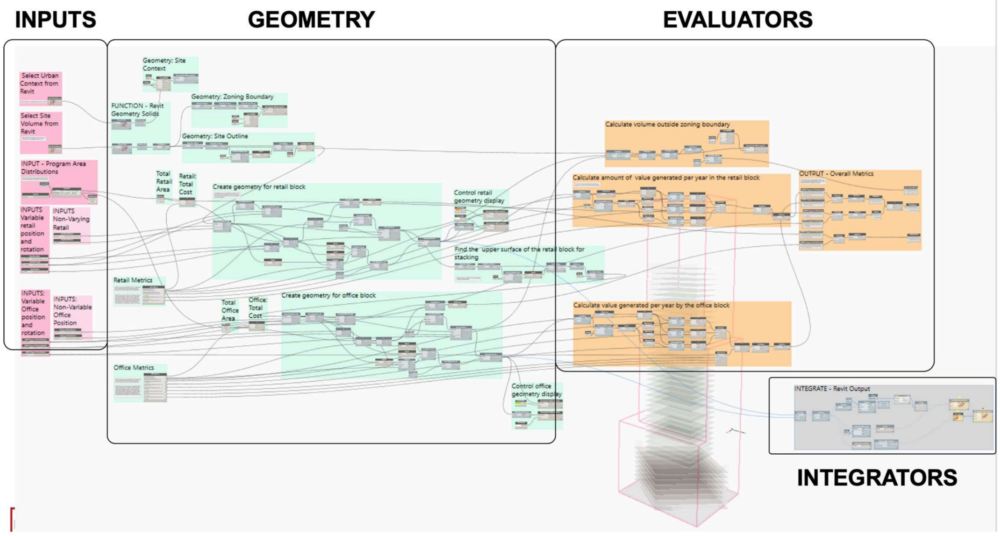
### Example 2: Building Massing – Complex

This study looks at the retail and office distribution and configuration for a building on an urban site.

- Variable Inputs:
    1. Ratio retail to office
    2. Program block size
    3. Program block rotation
- Goals:
    1. Minimize zoning envelop overlap
    2. Minimize cost
    3. Maximize total value per year

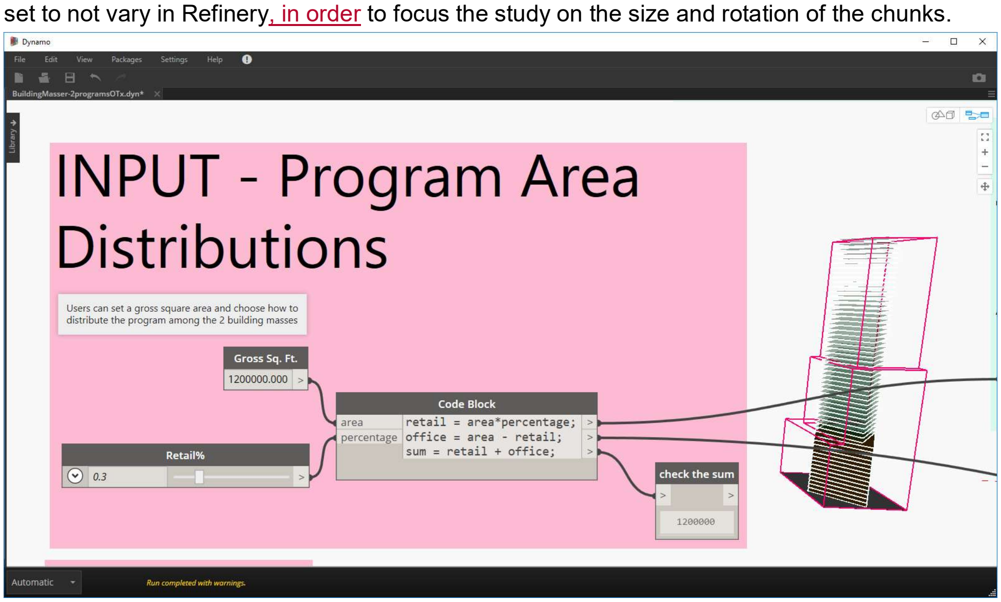


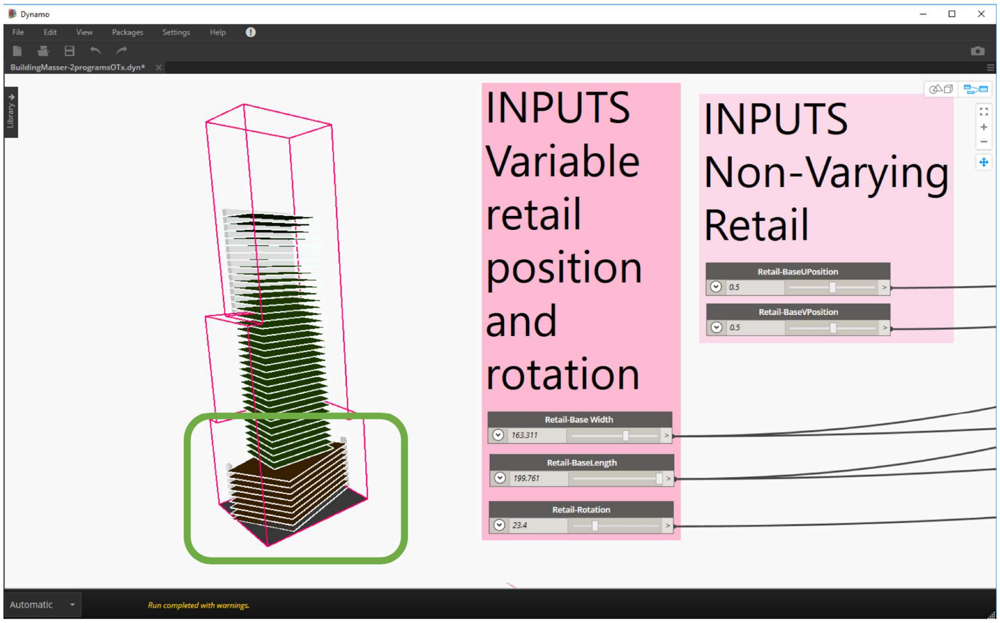

The graph contains the same groupings as our previous example, just with a bit more complexity.

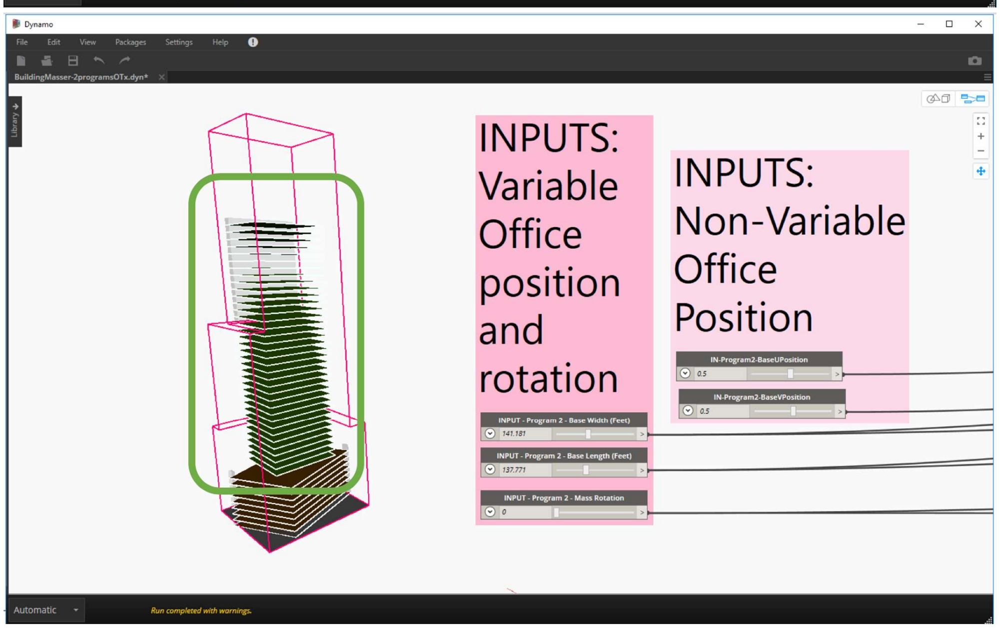



***BuildingMasser-2programsBiLT_NA2019.dyn***

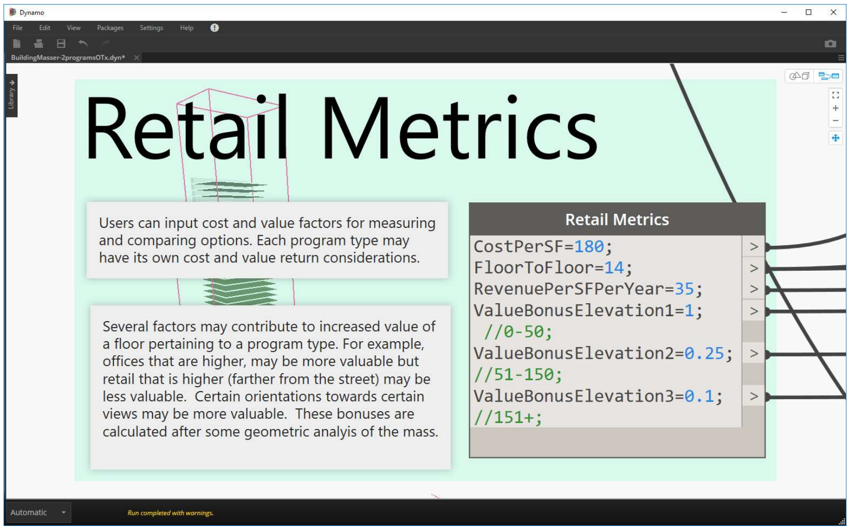
First let's look at the inputs. Users can set a gross square area and choose how~~t~~ to distribute the program among the 2 building parts of the building mass. Each building mass chunk can be controlled via length, width, and rotation with optional controls for position. These controls are set to not vary in Refinery, in order to focus the study on the size and rotation of the chunks.

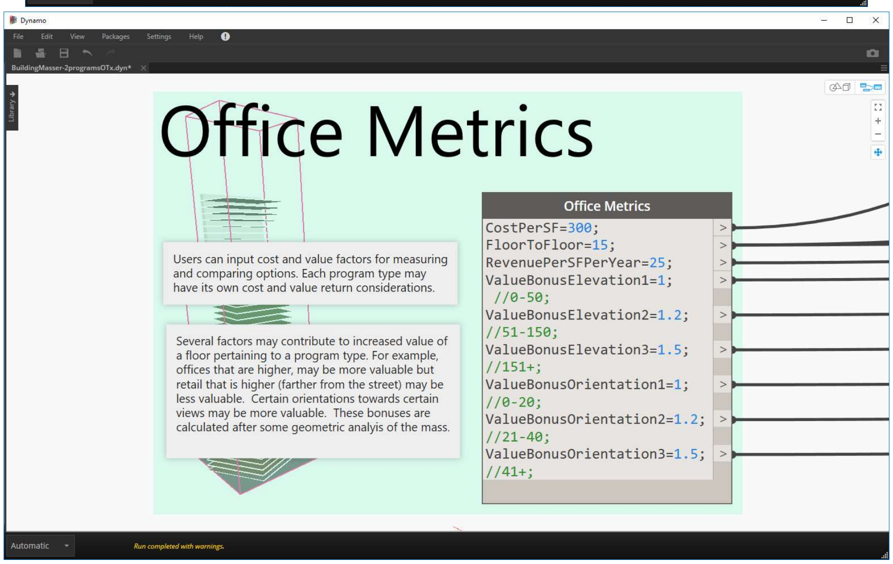





Next let’s look at the metrics that are used to guide the sizes and rotation of the building mass.




Summarizing and comparing the metrics:

| Metric | Retail | Office |
|---|---|---|
| Cost Per SF | $180 | $300 |
| Floor to Floor Height | 14 | 15 |
| Revenue/SF/YR | $35 | $25 |
| Elevation Bonus (0-50’) | 1 | 1 |
| Elevation Bonus 51-150’ | 0.25 | 1.2 |
| Elevation Bonus 151’+ | 0.1 | 1.5 |
| Orientation Bonus 0-20 | n/a | 1 |
| Orientation Bonus 21-40 | n/a | 1.2 |
| Orientation Bonus 41+ | n/a | 1.5 |

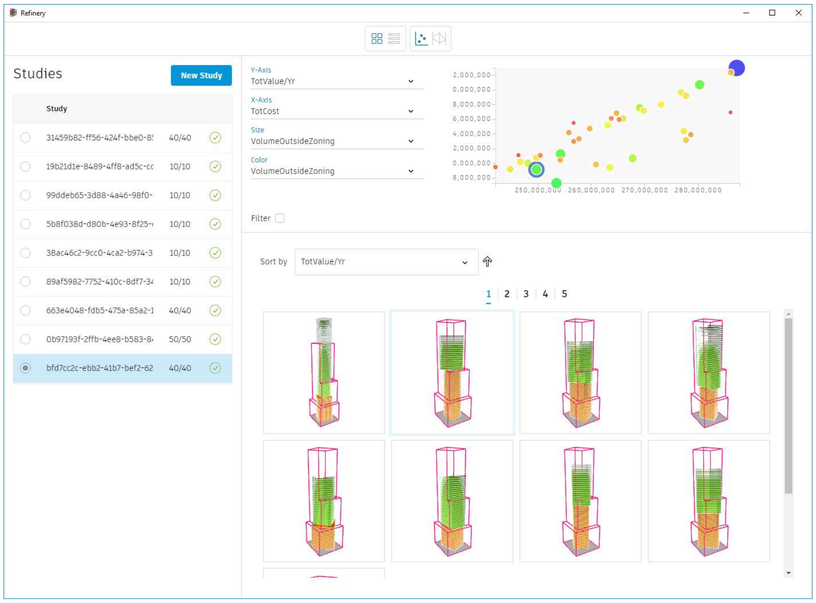
The next steps is to use the inputs to create geometry that can be flexed. Open the script to explore in more detail, but the primary functions of geometry creation in the graph are grouped as follows:

1. Create geometry for the retail block
2. Find the upper surface of the retail block for stacking
3. Create geometry for the office block (stacked on retail)

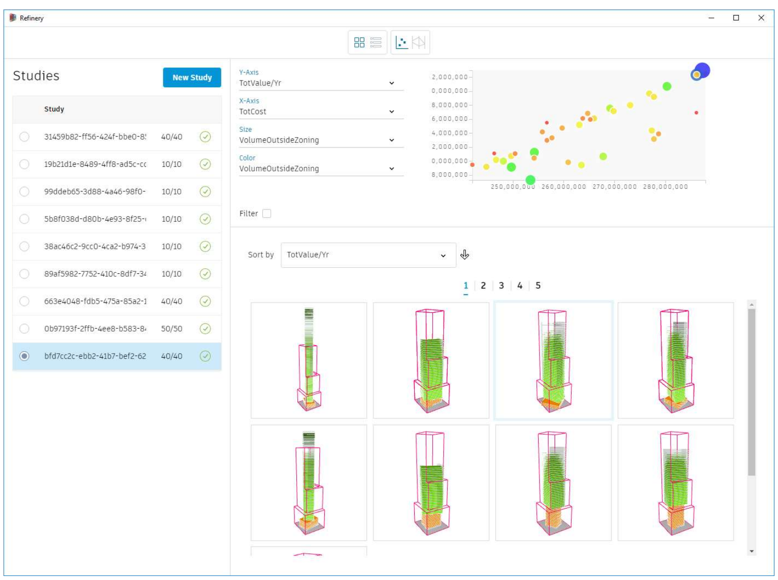
Once you have the geometry created and performance metrics designated, you can measure the outcomes via pretty basic mathematical calculations for:

1. Volume Outside Zoning
2. Total Value / Year
3. Years to Payback
4. Total Cost
5. Total Exterior Surface Area

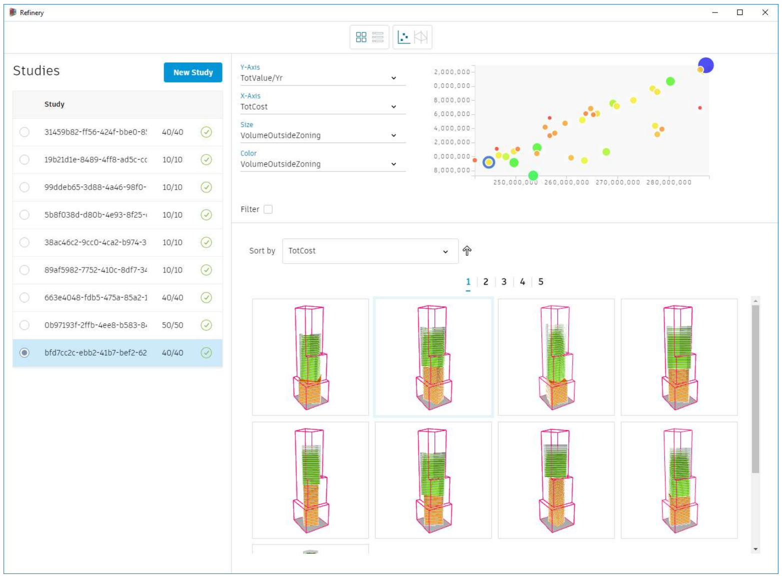
Once the logic is described in Dynamo, you can then automate the generation of design alternatives that you can measure and rank against each other. Start by using the “Randomize” generation method to get an idea of the design space.



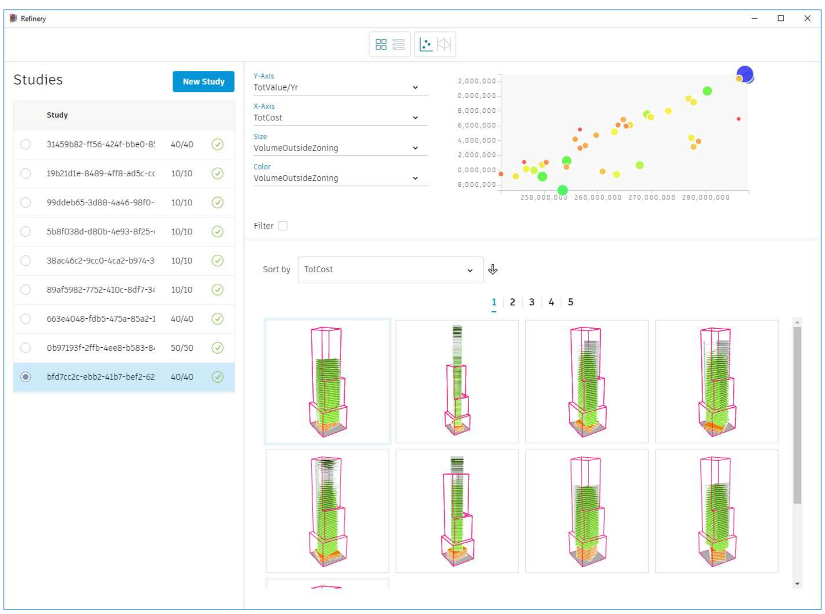

***40 Randomized runs showing least TotalValue/Year***



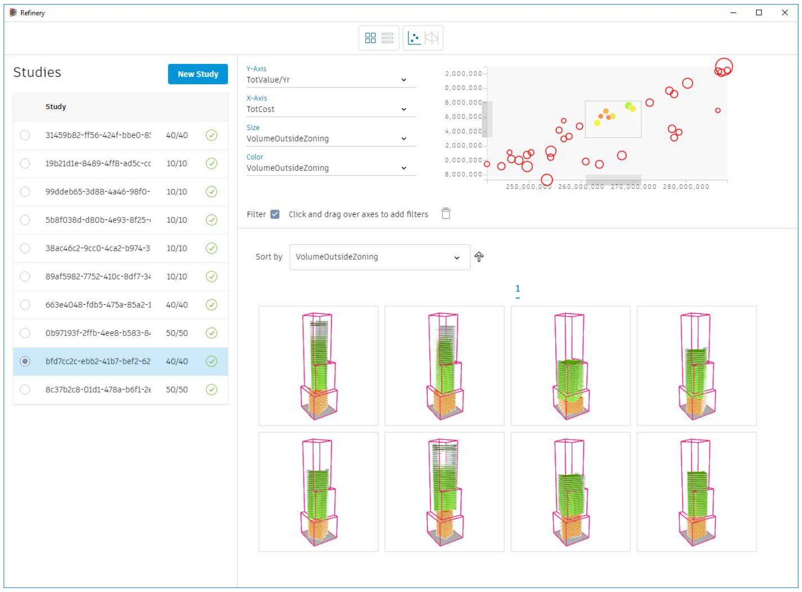

***40 Randomized runs showing most TotalValue/Year***




*40 Randomized runs showing lowest cost*

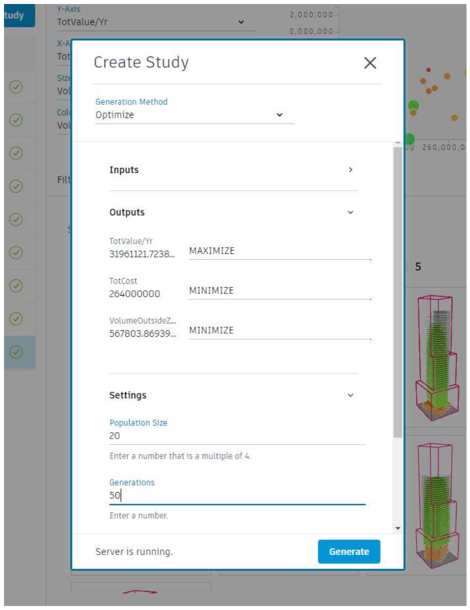



*40 Randomized runs showing highest cost*

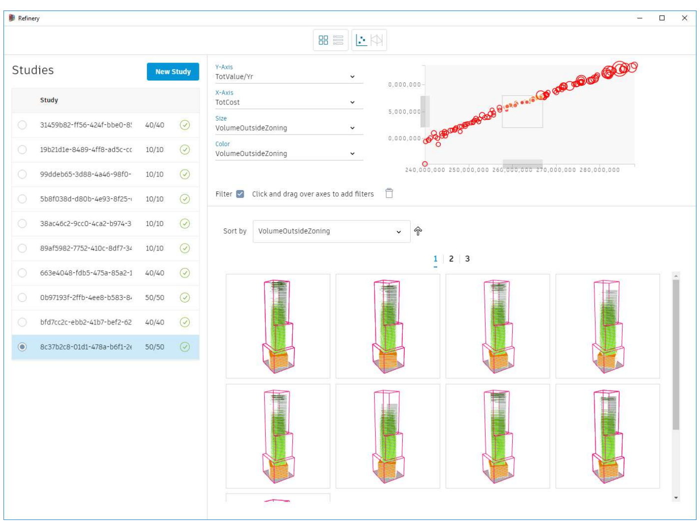



***Use filtering to select options with median cost and value, sort by lowest Volume Outside Zoning***

Finally, you can use the Optimize method to set goals before creating runs. If you set the Outputs to maximize Total Value/Yr, minimize Total Cost and minimize VolumeOutsideZoning, the computer will evolve the solution set towards designs that meet your stated goals.




*Setting up for an optimization run in Refinery*



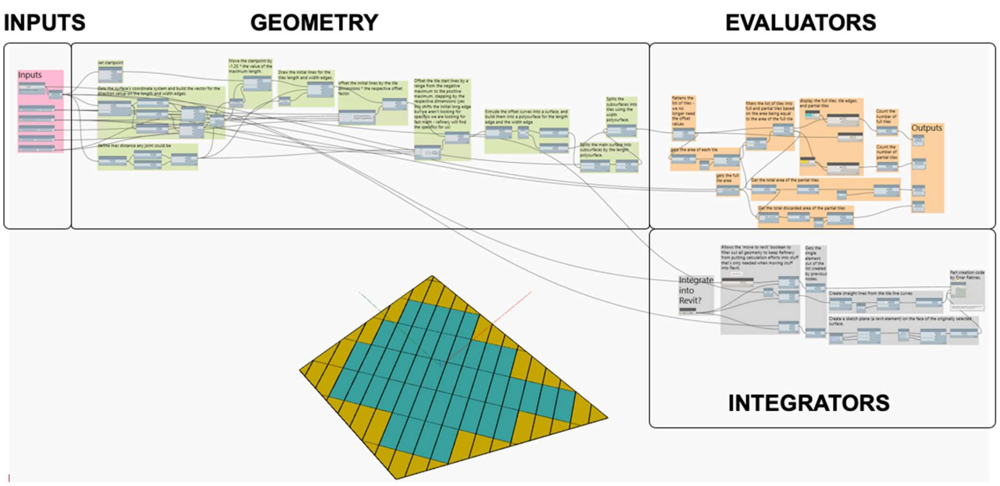

***20x50 Optimization run filtered for medium Cost, medium Value, and sorted for lowest VolumeOutside Zoning***

### Example 3: Tiling

This study looks at how to lay out tiles on a surface coming from Revit.

- Variable Inputs:
    1. Revit surface
    2. Tile Width
    3. Tile Length
    4. Width Offset
    5. Length Offset
    6. Rotation
- Goals:
    1. Maximize Full Tile Count
    2. Minimize Partial Tile Count
    3. Minimize Partial Tile Area
    4. Minimize Discarded Area

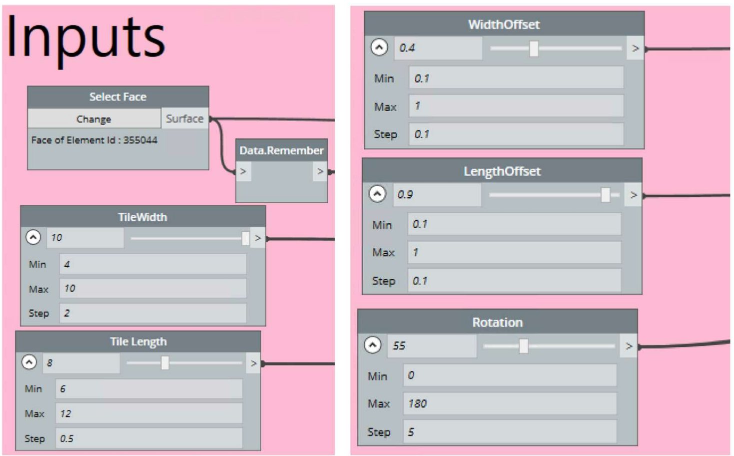
Once again, the graph contains the same groupings as our previous example:



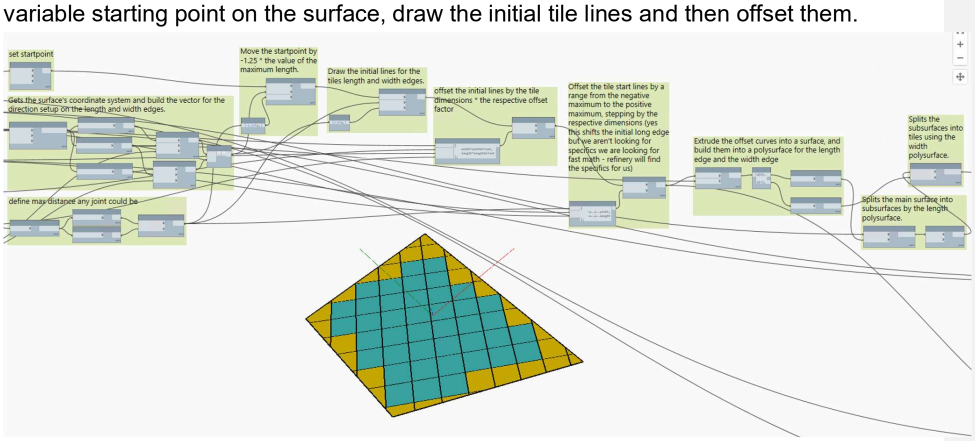

Let's look at each section.

1. Inputs – the minimum and maximum range settings in Dynamo are critical to define the design space for variable inputs. Refinery will also take into account the Step value as set in Dynamo. Notice the settings for the Tile Width and Tile Length. Tile widths can only be 4,6,8,or 10. Tile lengths can only be 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5 etc…to a maximum of 12. Setting these ranges to reflect available tile sizes is necessary to guide

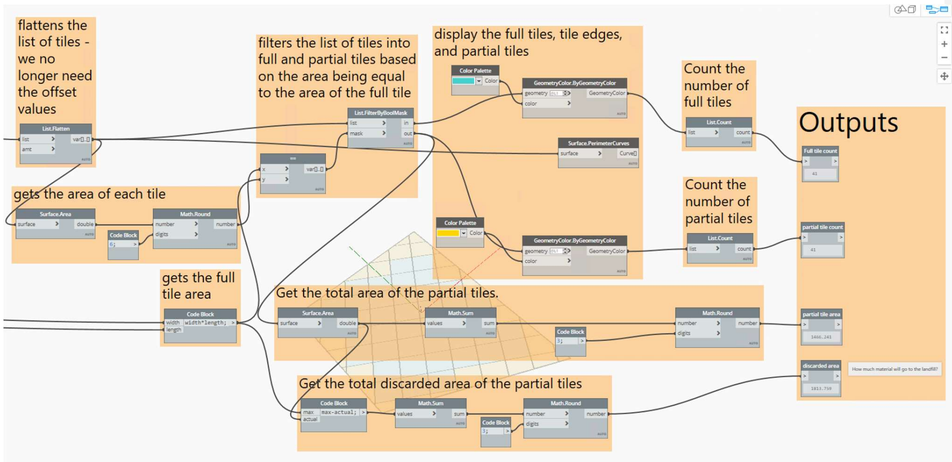
the design system to produce useful design alternatives.



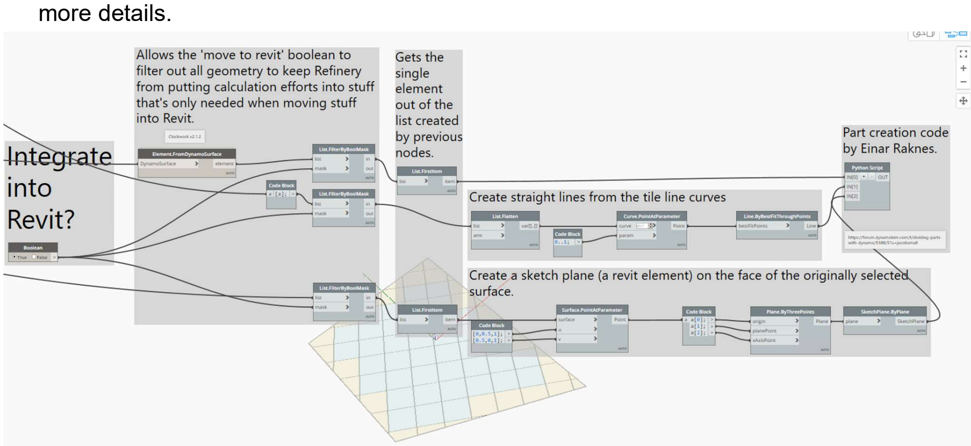

2. Geometry – The green section of the graph contains nodes which create the tile geometry. Open the graph to explore in more detail, but the basic idea is to understand the incoming surface for boundary conditions and maximum distances. Establish a variable starting point on the surface, draw the initial tile lines and then offset them.



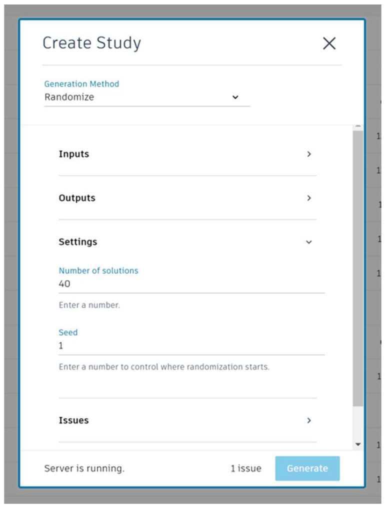

3. Evaluators – In the orange section of the script, the resultant tile layout is measured by :
    a. Counting the number of full tiles
    b. Counting the number of partial tiles
    c. Computing the total area of the partial tiles included in the layout
    d. Computing the total area of discarded partial tiles (amount of waste tiles)



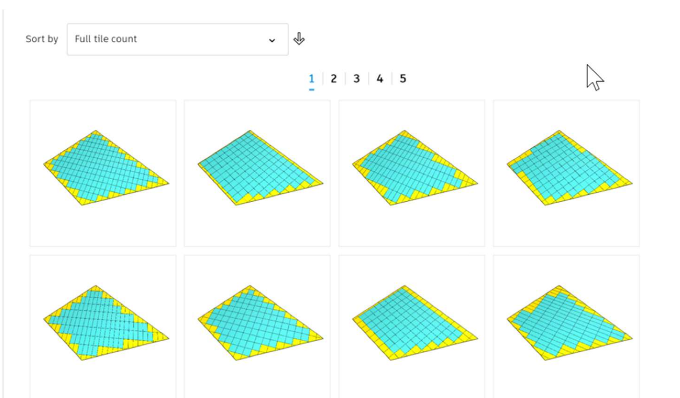

4. Integrators – the final grey section of the script contains nodes which can write a tile pattern back to Revit as parts. There is a Boolean node toggle at the start of this section which controls whether or not this section of the code is run. It can be turned off by toggling to "False" when running the script in Refinery and turned back on when you have selected a design and are ready to return the results to Revit. This section gets the Revit element that the original Revit surface was selected from and then uses the set tile pattern to create parts out of that element in Revit. The Part creation code happens in a Python node and was created by Einar Raknes. See [https://forum.dynamobim.com/t/dividing-parts-with-dynamo/5388/5?u=jacobsmall](https://forum.dynamobim.com/t/dividing-parts-with-dynamo/5388/5?u=jacobsmall) for more details.




Now that we have our script ready we can look at how we can do automated design explorations using Refinery. First we'll look at a run using the Randomize method to test the script.



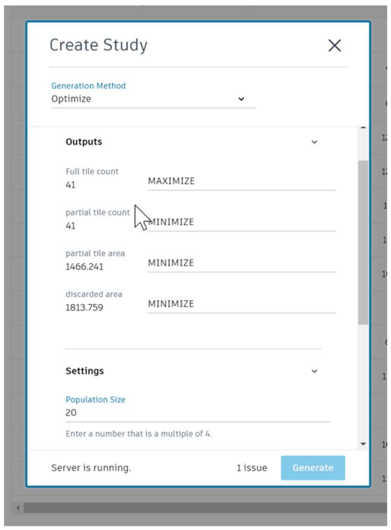

Generating 40 random options quickly gives us some results:

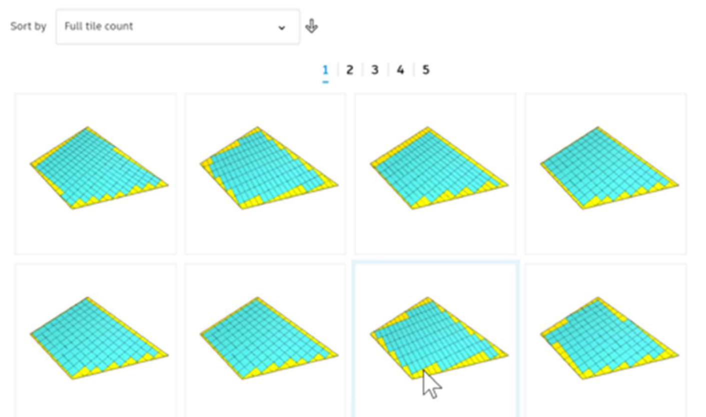



| Full tile count | partial tile count | partial tile area | discarded area |
|---|---|---|---|
| 159.0 | 81.0 | 930.241 | 1013.759 |
| 114.0 | 53.0 | 642.241 | 1265.759 |
| 106.0 | 71.0 | 1142.241 | 1271.759 |

***40 runs using Randomize method in Refinery***

Now that we can see what the design system is capable of, we can make any adjustments we need to make to our input ranges or how the script is working. Or if the Random run yields decent results we can use the Optimize method to guide the results to meet our goals of maximum tile count, minimum partial tile count, partial tile area and discarded area. Performing an optimization run will give us design alternatives that are closer to our stated goals.




| Full tile count | partial tile count | partial tile area | discarded area |
|---|---|---|---|
| 166.0 | 65.0 | 762.241 | 797.759 |
| 125.0 | 64.0 | 996.241 | 923.759 |
| 116.0 | 54.0 | 1034.241 | 693.759 |

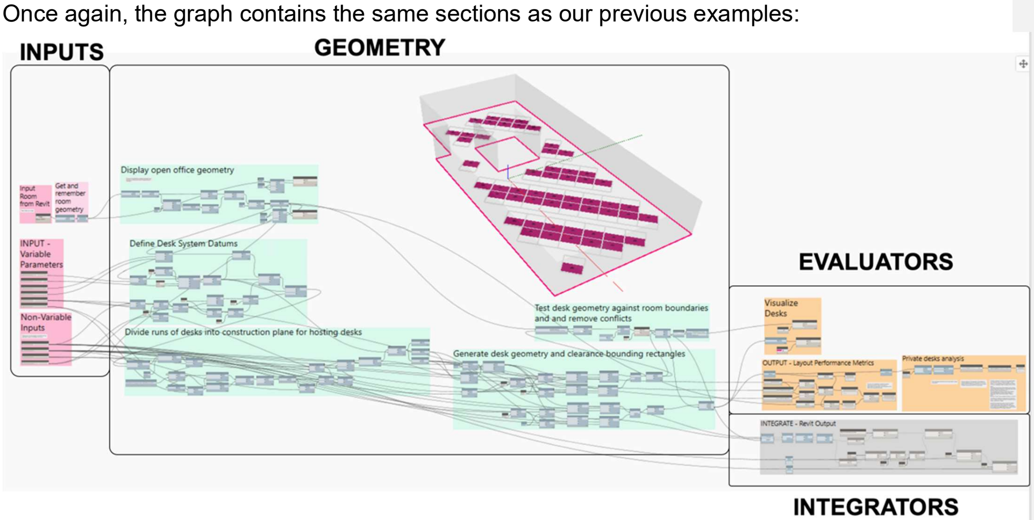
***20x10 Optimization Run in Refinery***

### Example 4: Open Office Layout

This script lays out desks given a Room element coming from Revit. In this study we will look at desk layouts that gives us the most desks but also the most private desks while maximizing desk area per person and maximizing space utilization.

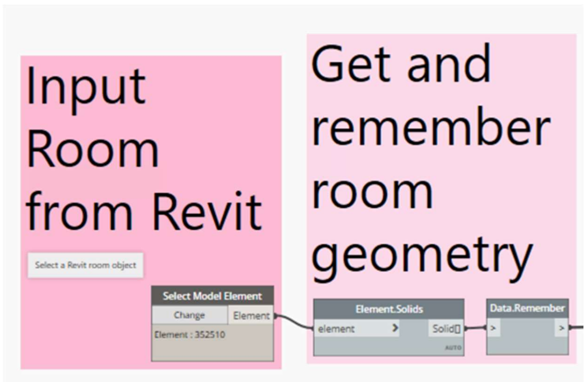
- Variable Inputs:
  1. Edge to start layout
  2. Desk rotation
  3. Position on room surface
  4. Desk clearance
  5. End offset
- Goals:
  1. Maximize number of desks
  2. Maximize desk area per person
  3. Minimize unutilized space
  4. Maximize number of private desks

Once again, the graph contains the same sections as our previous examples:

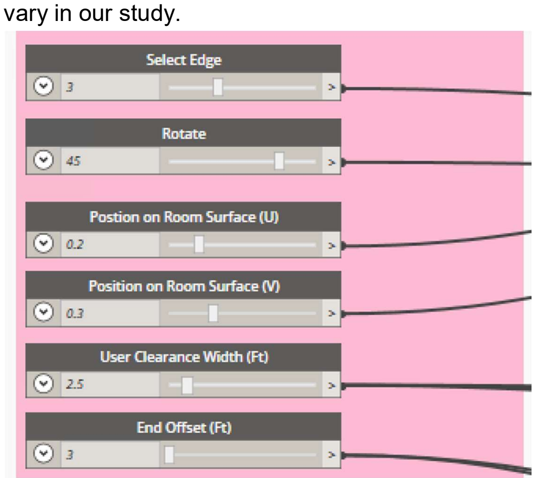



Let's look at each section:

1. Inputs: The first input is a Room element coming from Revit. This gets remembered for use in Refinery.

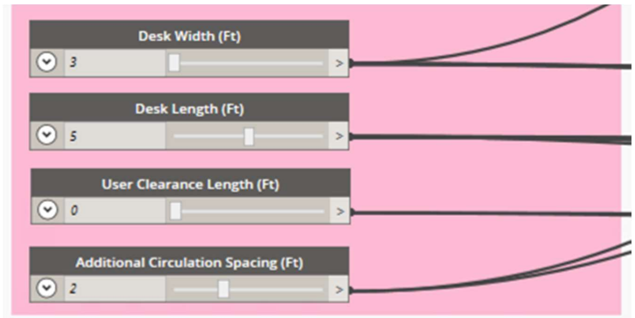



The next 2 groups of inputs are divided into variable and no variable inputs for clarity. The sliders in the first group are set to "Is Input" **on** via the context menu. These are the inputs that we want to vary in our study.




***Variable Input Parameters***

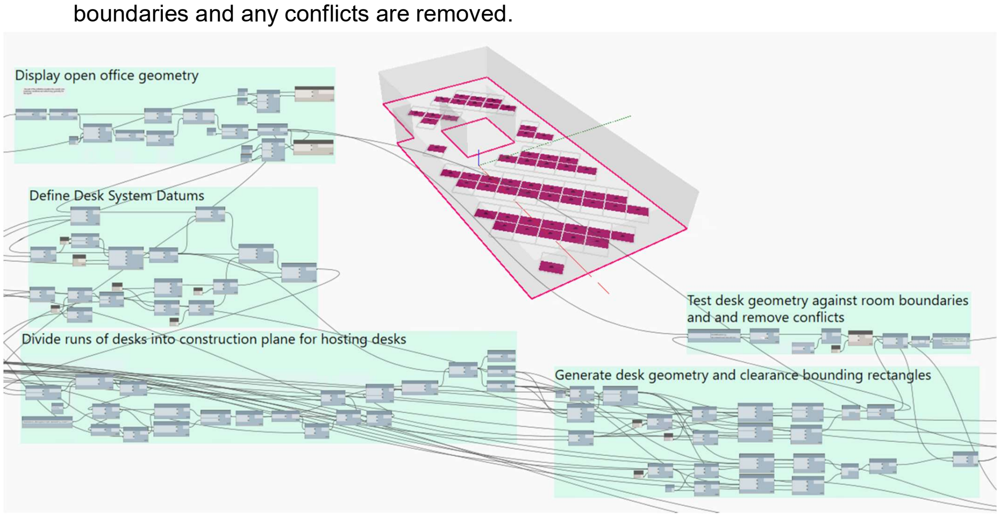
The sliders in the second group are set to "Is Input" **off** via the context menu. These are inputs that we want to remain constant during the study. Inputs can be changed to vary or stay constant by setting the "Is Input" flag in Dynamo or changing the variation option in Refinery.




***Non-variable Input Parameters***

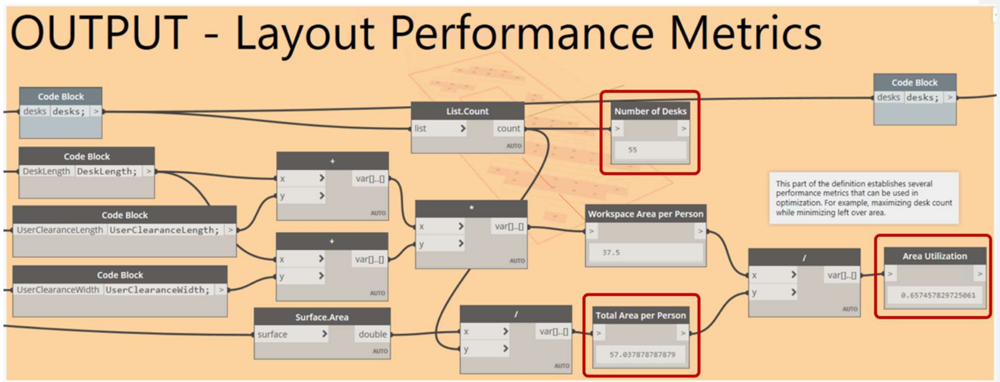
Be sure to consider how the maximum, minimum and step values are set for the range of your sliders so that you can limit the results to desirable values. For example, here, the rotation step value is set to 15 with a range of -90 to 90 so that the system won't return many options with tiny rotation changes.

2. Geometry – The green section of the graph contains nodes which create the desk layouts based on the room geometry coming from Revit. Open the graph to explore in more detail, but the basic idea is to process and display the room geometry coming from Revit. Then define a system of datums based off of this geometry. Next, runs of desks can be divided into construction planes for hosting each desk. Then the desk geometry and clearance rectangles can be drawn. Finally, the geometry is compared against the room boundaries and any conflicts are removed.




***Geometry section of OpenOfficeLayout.dyn***

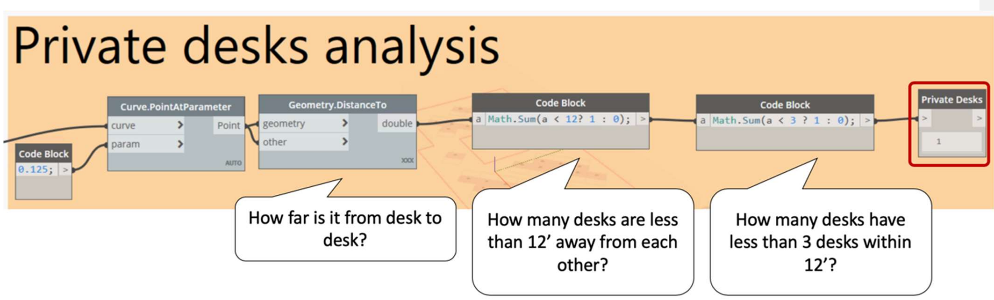
3. Evaluators. In the orange section of the scripts, the designs are evaluated against our established goals. The first orange section counts the number of desks, calculates the Total Area per Person and the Area Utilization. These nicknamed watch nodes are all set as “Is Output” so that Refinery will recognize them as outputs.




***Desk Layout metrics***

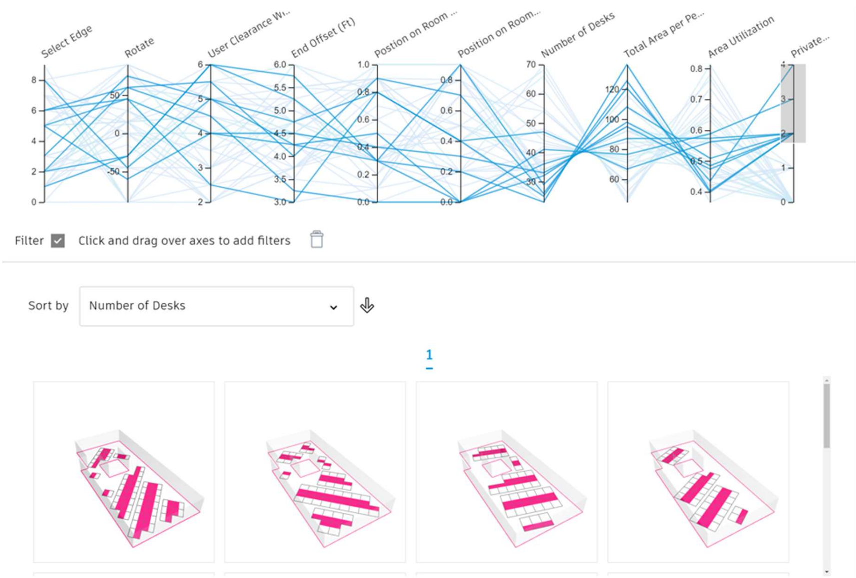
In the second orange section, we perform some calculations to count the number of “private desks”. This calculation does the following:

1. Measure how far each desk is from the other.
2. Count the number of desks that are less than 12' from each other
3. Count the number of desks that have less than 3 desks that are less than 12' away.




Note that both the “12’ distance” and “3 desks within 12’ “ evaluator rules are somewhat arbitrary. I noticed I could hear my neighbours if they were at a desk less than 12' away, but I was oblivious to whoever was at a desk more than 12' away. But….privacy at work is relative. Is there white noise in your office? What kind of work are you doing? Are you workers doing loud things or talking at their desks? Someone who only cares about what shows on screen might be fine sitting within 12’ of 3 or more people, but someone who needs audio privacy might not be ok with being within 12’ of even 1 person. For this study, we have defined “Private Desks” as described above. Let’s see how we can use it to guide design decisions.

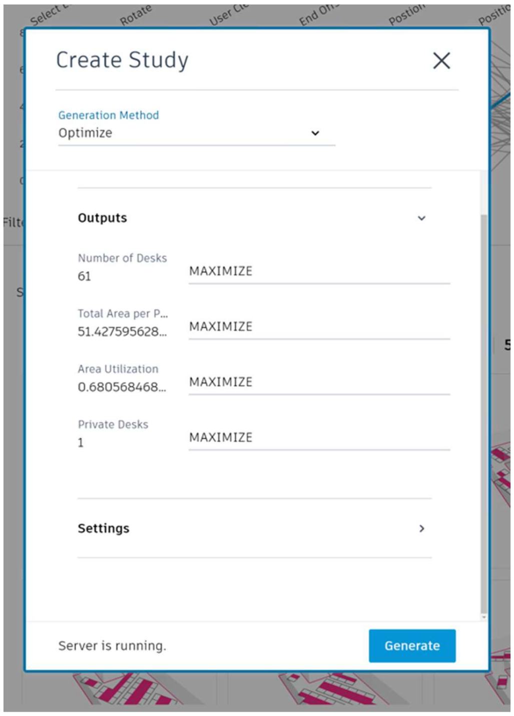
The first step is to do a run using the “Randomize” method. After generating 40 designs, we can use the Parallel Coordinates chart view to filter for alternatives that have the most number of “private desks” and then rank that result by alternatives that have the most desks. Design alternatives that rotate the desks seem to give us some interesting options along the lines of what we’re looking for.




| Number of Desks↓ | Total Area per Person | Area Utilization | Private Desks |
|---|---|---|---|
| 47.0 | 66.746 | 0.562 | 2.0 |
| 41.0 | 76.514 | 0.588 | 3.0 |
| 36.0 | 87.141 | 0.574 | 2.0 |
| 33.0 | 95.063 | 0.473 | 2.0 |

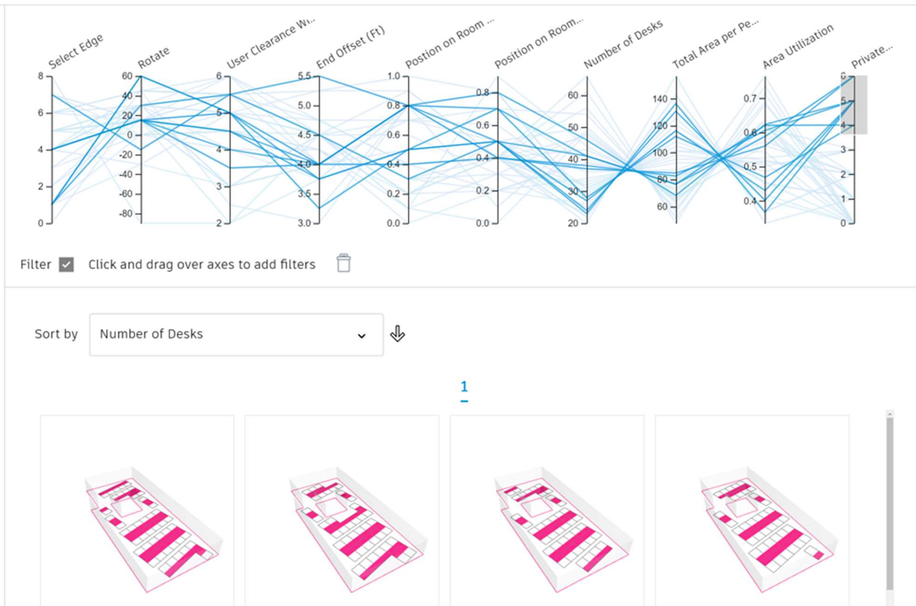
***40 Random runs, filtered by most Private Desks and sorted by most Desks***

Let’s see if an optimization run will get us even closer to our goals. Using Refinery, create a new Study using the Optimize generation method and the following output settings. The most private desks we get are 3 with 41 total desks.




***Optimization settings in Refinery***

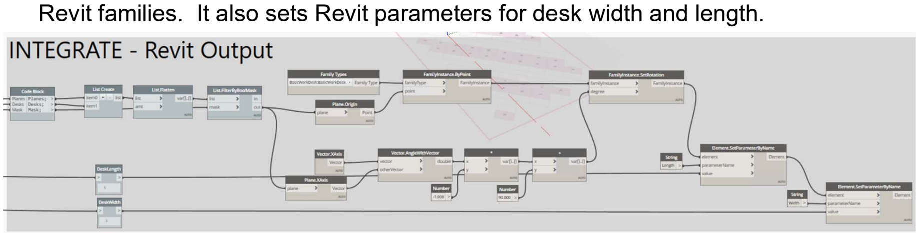
The optimization run does give us some even better results. Now we see options with 4, 5, and 6 private desks and total desk counts of 41, 46, and 41 respectively. We can continue to sort, filter and rank the results to evaluate the different designs quickly. If the underlying room geometry changes, we can simply run the study again.




| Number of Desks↓ | Total Area per Person | Area Utilization | Private Desks |
|---|---|---|---|
| 46.0 | 68.197 | 0.623 | 5.0 |
| 41.0 | 76.514 | 0.621 | 4.0 |
| 41.0 | 76.514 | 0.588 | 6.0 |
| 38.0 | 82.555 | 0.606 | 5.0 |

***10x20 Optimization run filtered by most Private Desks and sorted by most desks***

4. Integrators. The final step is to take the chosen design and return it to Revit for further development. The Integrate group takes desk location and rotation and place Revit families. It also sets Revit parameters for desk width and length.



### Conclusion

We hope that these examples will give you some ideas about how you can start to use generative design workflows with Dynamo and Revit. Keep in mind that there are many different scales of workflows where you can use generative techniques from very complex examples like the Mars Toronto Autodesk offices (see https://www.keanw.com/2019/06/project-rediscover-is-now-available-for-download.html for more details and the Dynamo graph) to very simple examples like 3Box which can be extended and adapted for specific workflows – see https://www.youtube.com/watch?v=uqFgg2Dorec for example. We encourage you to share your work with #ProjectRefinery, ask question on the Dynamo and Refinery forums or contact us at refineryfeedback@autodesk.com.

# Additional Resources

1. Getting Started with Dynamo: https://primer.dynamobim.org/
2. Dynamo Questions, inspiration: https://forum.dynamobim.com/
3. Design Script: https://dynamobim.org/wp-content/uploads/forum-assets/colin-mccroneautodesk-com/07/10/Dynamo_language_guide_version_1.pdf
   Or http://designscript.io/DesignScript_user_manual_0.1.pdf
   Or https://dynamobim.org/wp-content/links/DesignScriptDocumentation.pdf
   Or https://github.com/Amoursol/dynamoDesignScript
4. Refinery: https://www.autodesk.com/solutions/refinery-beta
5. Refinery Primer:
   a. https://refineryprimer.dynamobim.org/
   b. https://github.com/DynamoDS/RefineryPrimer
6. Refinery Toolkit Dynamo Package and Examples Files:
   ~~5.~~a. https://github.com/DynamoDS/RefineryToolkits
~~6.~~7. Generative Design education: https://medium.com/generative-design

Formatted: English (United States)
Formatted
Formatted: No underline, Font color: Auto, English (United States)
Formatted: English (United States)
Formatted: English (United States)
Formatted