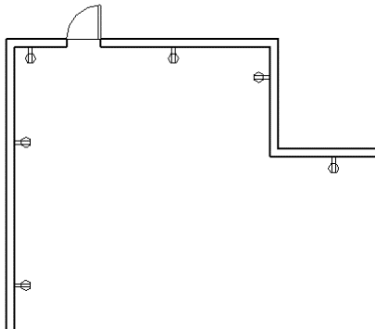
PLACING ELECTRICAL RECEPTACLES WITH DYNAMO

Placing hosted elements with Dynamo can be a bit tricky. This exercise will demonstrate how to do this by placing receptacles along the perimeter of the space. No need to redesign your firms Revit family that are required to be hosted. Learn how Computational design tool can save time in MEP workflows.



REQUIRED FILE

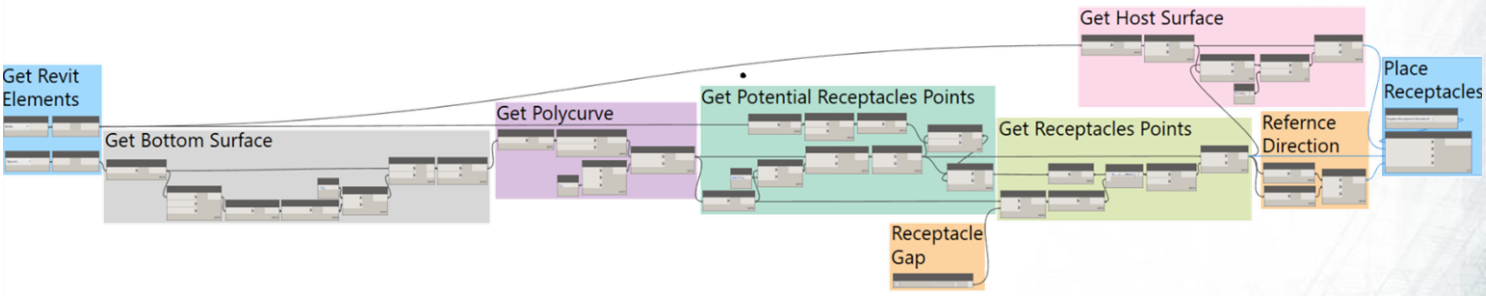
Revit 2019: Place Receptacles rvt
Dynamo 2.0 File: Place Grid of Diffusers.dyn

WORKFLOW OVERVIEW

In this exercise, we will place electrical receptacles around the perimeter of a space's at a given gap between each receptacle. Dynamo will be used to get Revit space geometry. From the space geometry, the boundary curve of the space can be gotten as a polycurve. Then points will be obtained along curves. To find the best receptacle location points, other geometry will be analyzed with logic and math. To finish the new hosted Revit elements will be placed.

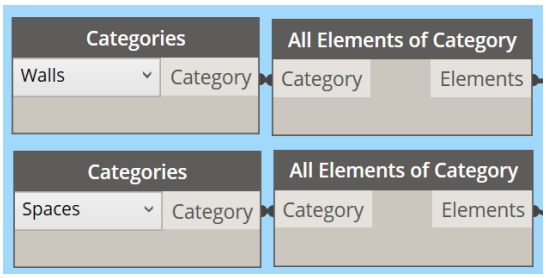
STEPS

1. Get Revit Elements
2. Get Bottom Surface of Space
3. Get Space's Bounding Curve
4. Get Potential Receptacle Points
5. Get Receptacle Points
6. Get Host Surface
7. Get Reference Vector
8. Place Receptacles

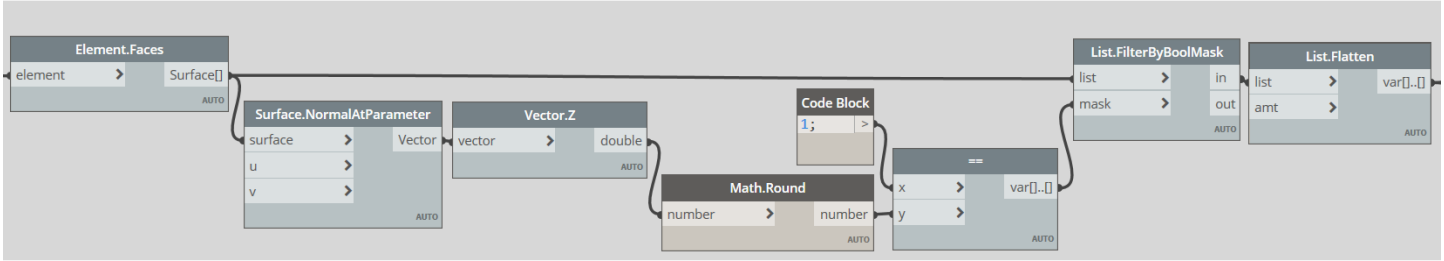


STEP 1- GET REVIT ELEMENTS

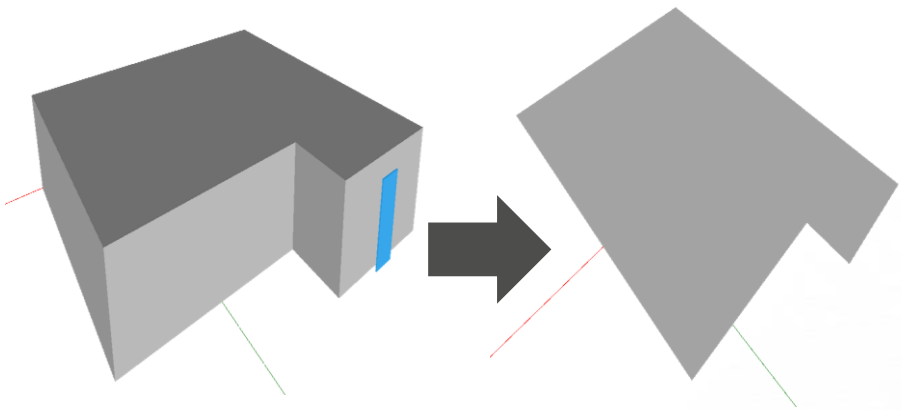
First, we need to collect all the Revit elements. The first step is to use the **Categories** and the **All Elements of Category** nodes to generate a list of the walls and spaces.



STEP 2-GET BOTTOM SURFACE OF SPACES

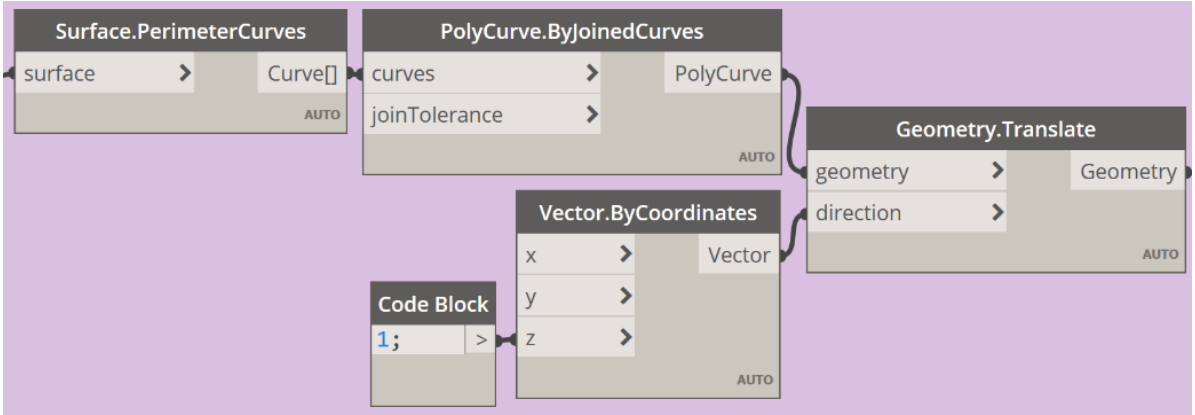


The objective of this section is to get the bottom surface of the space. This section is identical to the "*Get Top surface"* section in the first exercise with two exceptions—first, the vector *z* component needs to be set to equal negative one rather than one. The other exception is that the code will be collapsed into one *Code Block* using Design Script. Design Script offers a hybrid approach of combing visual programing with textual programming. I tend to use this to help keep graphs clean and organized. This is why I often use Design Script rather than the traditional node. No spaghetti messes.

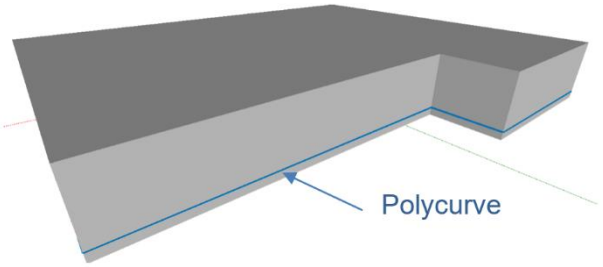


STEP 3 - GET SPACE'S BOUNDING CURVE

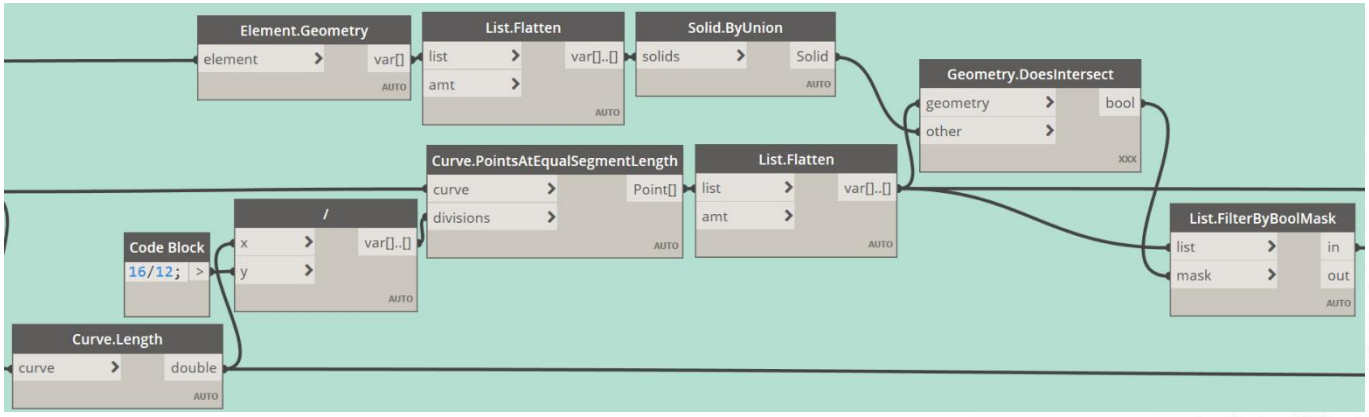
The goal of this section is to get a polycurve that goes around the space's boundary and at the receptacles required elevation. The curves that make up the bottom surface from step two are recovered using the



Surface.PerimeterCurves node. This list of curves are then combined into one polycurve using the **PolyCurve.ByJoinedCurves** node. Next, we will make a vector that points up in the z-axis using the **Vector.ByCoordinates** node. We can then shift the polycurve in the direction of that vector using the **Geometry.Translate** node.



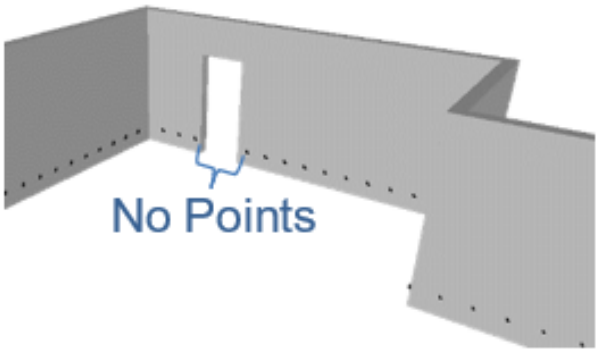
STEP 4 - GET POTENTIAL RECEPTACLE POINTS



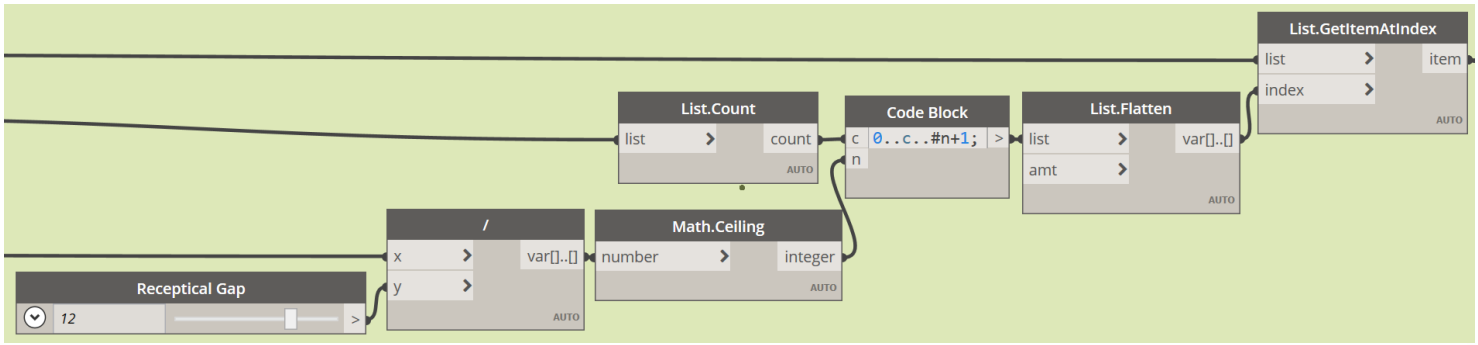
This next step is used to make sure our receptacles do not end up in the void in the walls where our doors are. This is achieved by making a bunch of points along the space's boundary curve and then checking to see if they intersect the wall's geometry.

First, the length of the polycurve is measured with the **Curve.Length** node. This length is then divided by a spacing amount. In this case, 16/12 was used since this is the spacing between 2X4s behind the wall.

Next, the Curve.PointsAtEqualSegmentLength node is used to get the points along the polycurve. This list of points is then flattened to remove the sub-list and the Revit walls are fed into the Element.Geometry node. The output are the wall's solids. This list of solids is then flattened and connected to the Sold.ByUnion node to get one larger solid. This will make the output for the next step, to check the intersection of the points and solid using the Geometry.DoseIntersect node, easy. The output of true and false is then connected to the mask input of the **List.FilterByBoolMask** node. The list input is the list of points and we now have a list of potential points for the receptacles**.**

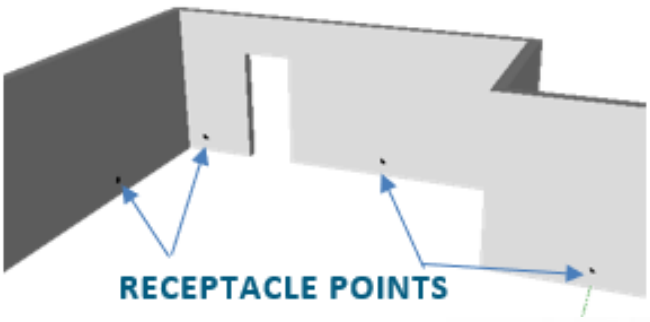


STEP 5 - GET RECEPTACLE POINTS

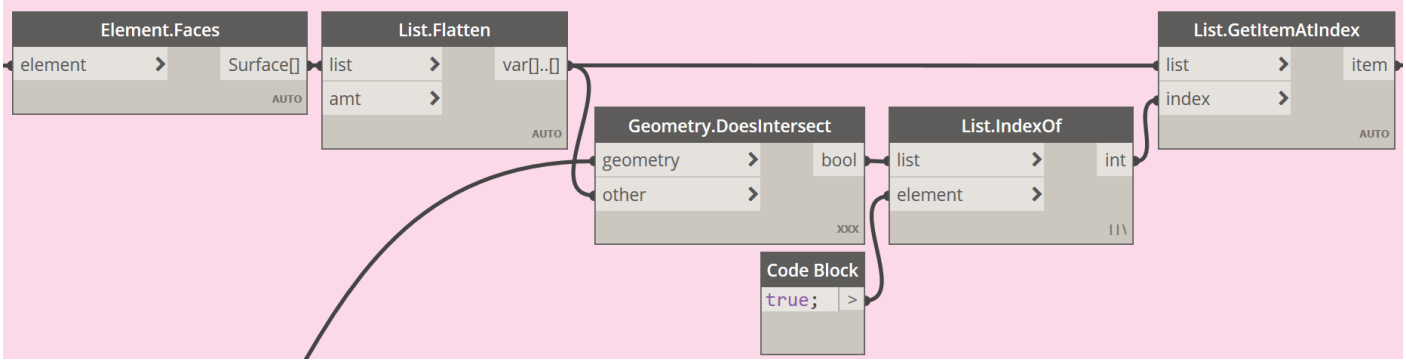


This section finds the number of receptacles required for the space. The gap between each receptacle is set with a slider. The length of the polycurve is then divided by this user-selected gap value. The number is then rounded up to the nearest whole number using the **Math.Ceiling** node.

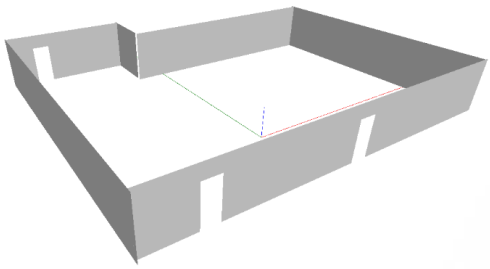
Next, we need to find out how many points are in the potential list of points using the **List.Count** node. We make another series in order to get a list of indices to retrieve points based on our spacing. The series can be seen in the **Code Block** and starts at 0 and goes to the number of potential points, plus 1, at a step value that reflects the number of needed receptacles. The list of indices is then flattened and finally, points at the specified indices for retrieved using the **Get.ItemAtIndex** node.



STEP 6 - GET HOST SURFACE



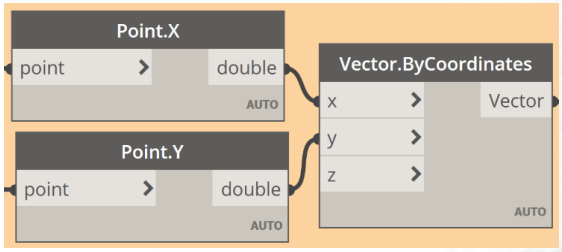
The purpose of this section is to get the host surface for the receptacle families. This needs to be the surface of the wall that the receptacle points are touching so the first step is to get the faces from each wall using the **Element.Faces** node. This list of surfaces can then be flattened and checked for intersection with the points using the **List.Flatten** node and the **Geometry.DoseIntersect** node.



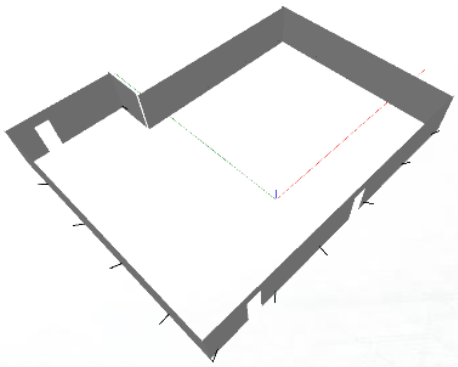
Next, the sub-list of Booleans is searched for true values using the **List.IndexOf** node. Note this node needs to be set to "*longest lacing*" since the input has sub-lists.

Lastly the respective surfaces are gathered using the **List.GetItemAtIndex** Node. The output of this section is a list of surfaces that intersect the list of receptacle points.

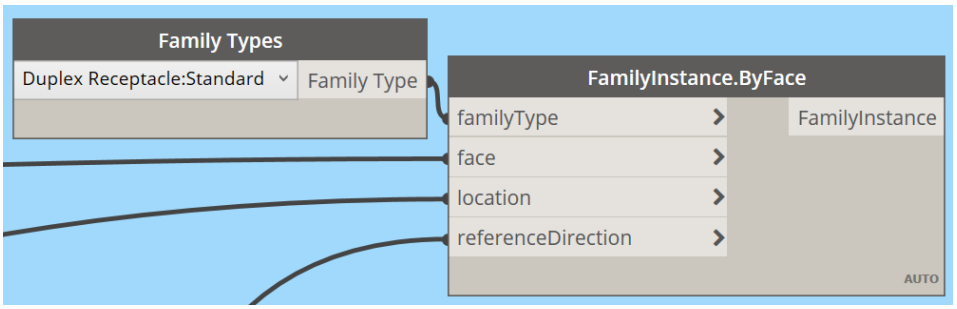
STEP 7 - GET REFERENCE VECTOR



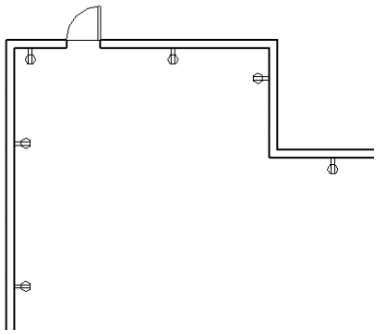
Another input needed in order to place hosted Revit families with Dynamo Is a *Reference Vector*. The reference direction is a vector that defines the direction of placement of the family instance. To get this vector we simply take the *x* and *y* components of the receptacle points using the **Points.X** and **Points.Y** nodes and connect these components to their respective x and y inputs in the **Vector.ByCoordinates** node. The image to the right shows the lines represented by these vectors.



STEP 8 - PLACE RECEPTACLES



The last step is to finally place the receptacles. First, the family type is selected using the **Family Types** node. Note that this family must be a face-hosted family. Then the **FamilyInstance.ByFace** node is used to place the receptacle family throughout the space by connecting the lists of surfaces, receptacle placement points and reference directions.



The output of this algorithm is electrical receptacles that are hosted to the space's wall and governed by a designated gap. The two key takeaways in this lesson are understanding how to get the bounding curves of a Revit space and how to place hosted families using Dynamo.