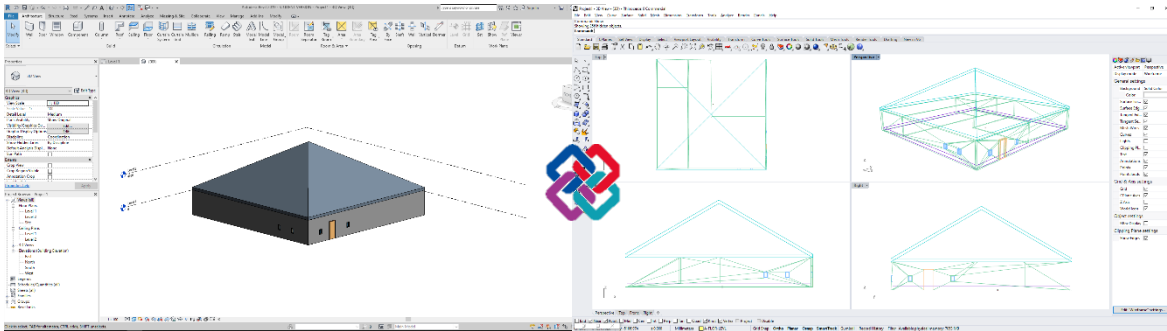


Example - Distance fields & ordered points

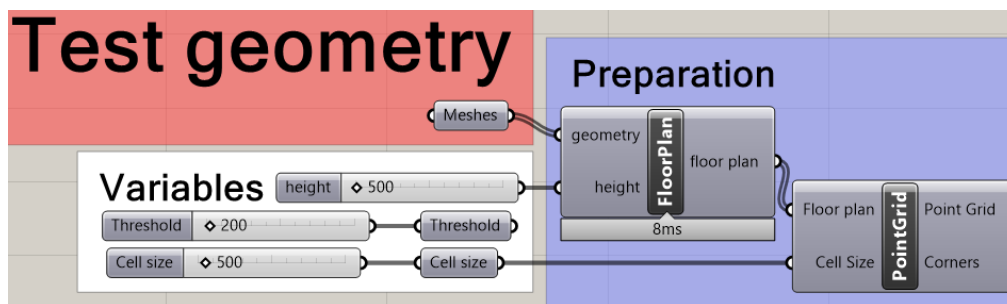
Importing building model

Import the design proposal from your favourite modelling software to Rhino.

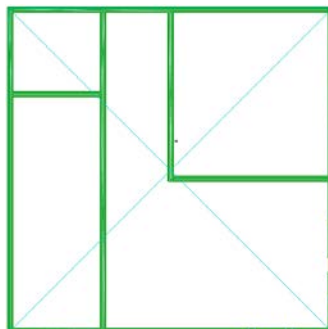


Preparation of geometry

Step 1 – Floor Plan



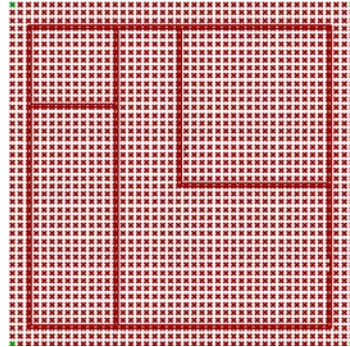
The building model - in this case consisting of 256 meshes - is loaded into Grasshopper through a mesh node. Inputting this in a FloorPlan component creates a plan section of the model in a given height (see variables). 2D curves representing the floor plan are send out of the node.



Step 2 – Point Grid

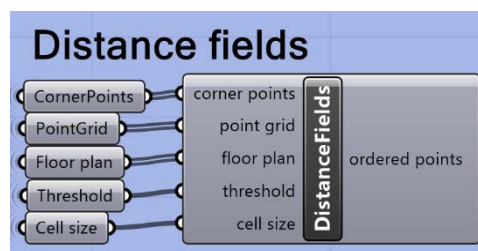
The floor plan curves and the desired cell size are fed into the PointGrid node. Changing the cell size of the point grid will affect both the runtime the whole program and the accuracy of the final room objects.

The outputs of this component are both the entire 2D point grid, and the four grid corners.

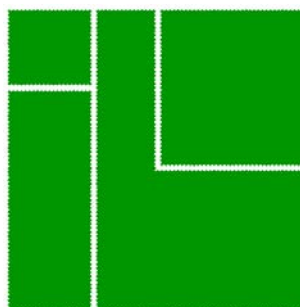


Registration of space boundary data

Step 3 – Distance Fields

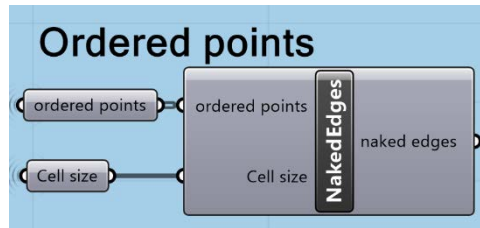


The next step is the hungry DistanceFields. It takes the point grid and removes a lot of points. Both points too close to floor plan curves, and the outliers. Based on the known distances, the rest of the points are grouped into sets of distance fields – each set representing a different room.

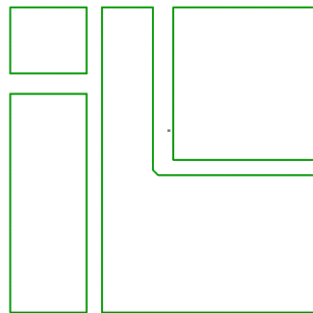


Reconstruction of space entities

Step 4 – Naked edges

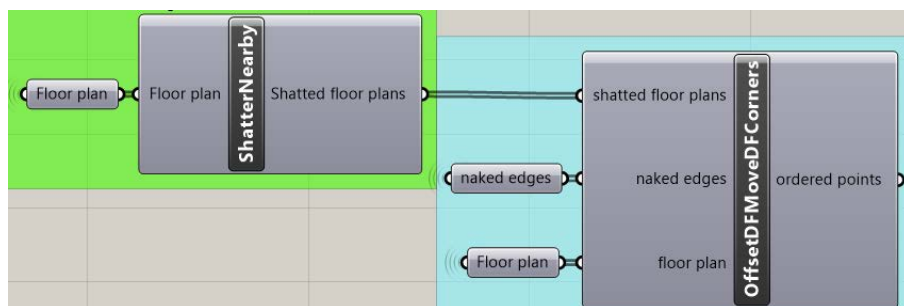


NakedEdges is the component that creates first edition of the room boundaries. It takes all the sets of ordered points/distance fields, make use of Delaunay triangulation, and filter out all edges longer than the diagonal of a cell in the point grid. From each set of Delaunay meshes, the naked edges/boundaries are passed on.



Correction step

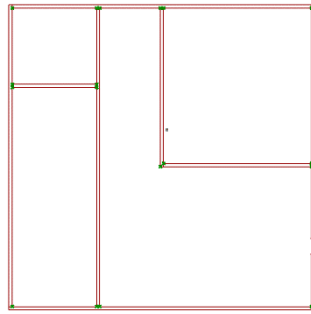
Step 5 – Shatter nearby



FloorPlan curves are shattered using the node ShatterNearby before they are fed into the OffsetDFMoveDFCorners.

Step 6 – Offset by distance fields and move corners

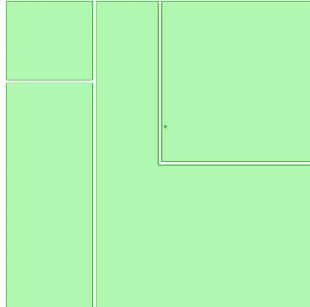
The first edition of the boundaries (naked edges) are corrected to fit the topology of the given floor plan better. The corrected ordered points are passed on to PntSrf.



Step 7 – Surface from Point Trail

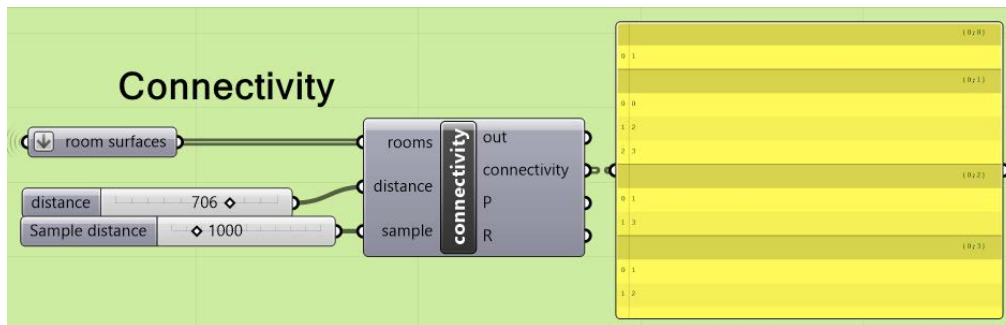


PntSrf creates a surface by creating a closed polyline through the ordered manipulated points. Afterwards the surfaces are extruded by the desired room height, for the analytical rooms.

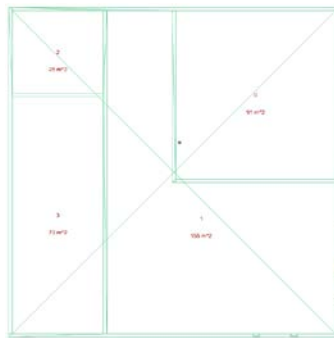


Export for Simulations and Analyses

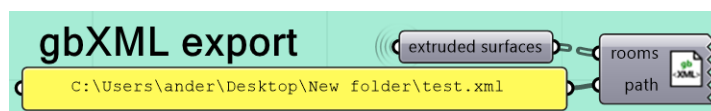
Step 8 – Connectivity



Based on the room surfaces the adjacent rooms for each room is found. The returned adjacency graph contains a list for each room containing the indices of the rooms that are adjacent to it.



Step 9 – gbXML export



The last component takes the extruded building surfaces and a path (with the desired name) for the gbXML export

```
<Campus id="cmps-1">
  <Building id="bldg-1" buildingType="Office">
    <Area>73139994</Area>
    <Space id="sp-0-Open" zoneIdRef="zone-Default">
      <Name>0 Open</Name>
      <Area>73139994</Area>
      <Volume>204791983265</Volume>
      <ShellGeometry id="sg-sp-0-Open" unit="mm">
        <ClosedShell>
          <PolyLoop>
            <CartesianPoint>
              <Coordinate>-9726</Coordinate>
              <Coordinate>2667</Coordinate>
              <Coordinate>0</Coordinate>
            </CartesianPoint>
            <CartesianPoint>
              <Coordinate>-4426</Coordinate>
              <Coordinate>2667</Coordinate>
              <Coordinate>0</Coordinate>
            </CartesianPoint>
          </PolyLoop>
        </ClosedShell>
      </ShellGeometry>
    </Space>
  </Building>
</Campus>
```