Slide 1 **DOTS:**

We would like to present a Grasshopper plugin for architecture and urban planning. This plugin will generate appropriate configs for various types of design problems.

We hope to add analytical components in the near future and hopefully with your inputs, we can develop a comprehensive set of design tools.

We treat the RH-GH environment as a medium to package and distribute our algorithms and try to make it very easy for designers. We have taken care to organize & name the input-output fields so that knowledge of Rhino-GH is not an impediment. It relies mostly on click and drag.

Slide 2 **COMPUTATION & COMPUTABILITY:**

In geometry processing:

1. even simple patterns cannot be generalized to all shapes.
2. Transformation of patterns does not scale well due to inconsistent properties of smooth and discrete segments or drastic variations in angles of segments.
3. From literature, we know that there is additional complexity due to accretion.

It is proposed that typical design configs can be achieved by conducting a sequence of logical and geometric transformations on specific parts of the region under consideration.

1. This step-by-step approach helps in preserving partial successes.
2. Intermediate results help in achieving a larger objective function because in case of an error, the computations do not start from the beginning but an intermediate step.
3. Breaking down the computational process helps in effectively exploring alternatives because we can simply alter some parameters of an intermediate step and it will have a cascading effect.

Slide 3 **COMPONENTS:**

Primarily, there are two main operations - subdividing a large area into smaller pieces or parcels and then placing a spatial object in it.

Right now, we have provided four mechanisms of generating the necessary smaller regions –

1. Recursively subdivide a region
2. Place appropriate parcels along the periphery
3. Place a set of regions in a plane by an exploratory process.
4. By extending segments of an internal polyline

For planning, we have provided a number of components that produce well-known building typologies that can be easily placed in a given region.

These components vary in complexity and even though some of them may be developed using scripting tools, there are few considerations:

1. Can the script be generalized to all projects?
2. How many types of variations can be generated?
3. Can it handle all shapes & sizes of inputs?
4. Learnability & usage.
5. Speed is a primary concern and it is much Faster than grasshopper or python scripting -scale of 500 on average.
6. By classifying massing typologies, we can generate various types of buildings by altering a few parameters.
7. Dots provides a comprehensive approach where the components can be connected to handle many types of problems. It is basically very easy to set up a project pipeline and add custom components for a project.

Our intention is to reduce the time taken to figure out how to approach a problem.

Slide 4 **Software Design:**

The user may think about the dots components as a way of passing inputs into a set of arguments at the end of which, the output is evaluated. The arguments on the inputs run a series of geometric and logical actions. Then, the result of these operations is weighed against the input constraints. If everything is satisfied, output is generated otherwise, some parameters are changed under constraints until satisfactory results are found.

In essence, Dots implements a geometry library with in-built goals. Various permutations of the components will help in design exploration and they are actually meant to be used in conjunction.

As a software design, we treat the output geometric configurations as a language. Each component is a well-known design element- for us it is a symbol or lexeme with internally consistent logic & geometric arguments. The syntax is provided by the user which determines grouping of elements. Finally, once the elements have been laid out based on the syntax, if they form a legible organization, it is accepted as a solution.

Slide 5 FUNDAMENTAL COMPONENTS - **Subdivisions:**

With this information, we can dive into the components. This set of components, will allow the user to quickly break down a large region into smaller regions. There are 4 components that perform the subdivision operations and 1 component to select buildable parcels based on open-space requirements.

The inputs required are simple components for numbers, boolean, and external curve.

Slide 6 FUNDAMENTAL COMPONENTS - **Subdivisions:**

The first 2 components split up the input curve into orthogonal parcels internally and output a set of closed curves and street centerlines, the second component also eliminates an input internal curve from the solution curves. The third component takes in an internal polyline extends the lines and generates parcels for combinations of intersections.

Slide 7 FUNDAMENTAL COMPONENTS - **Subdivisions:**

The first component generates parcels around the periphery which can be combined with massing typologies as shown.

The selection component takes the parcels, FSR & open-space requirements are input and it outputs parcels which should be built and those that can remain open. Either, open parcels are centrally located, or they are randomly dispersed depending on user input. Next-step would be to use one/many attractor points to place the open-space parcels.

Slide 8 FUNDAMENTAL COMPONENTS - **MASSING / TYPOLOGY:**

This set of components are meant to generate massing / typology. We have attempted to minimize the number of components required and still provide a way to generate the major typologies.

The massing typologies are classified according to the topology of the geometry. Their floor-floor heights, setbacks, step-backs, can be easily changed and it will update the geometry very fast.

This is intended to be connected to many parcels output by previous components. And we also intend to add analytical tools to measure what is visible to a person from a point or shading. Then the time taken to generate becomes large. That is why we provide separate components so that the user can judiciously use them and run many iterations in a workflow with multiple components. It is hoped that our components will enable large-scale analysis.

Slide 9 FUNDAMENTAL COMPONENTS - **MASSING:**

The first 2 components illustrate simple extrusions. In the second component, we propose that text panels be used where the required values are separated by a comma.

The third component is a special case where for each parcel, the closest street is found and required setback can be read and used for extrusion.

One thing to notice is that even though the first component can be generated by the second, it is separated because it is very frequently used, and the time taken will be reduced to do just this operation.

Slide 10 FUNDAMENTAL COMPONENTS - **MASSING:**

The courtyard typology is shown. And the second and third components provide a base and towers can be generated from the upper curve. These two are quite sensitive to inputs - generating the towers. There is a subtle difference between them. Interpolated is effective for curves and normal is effective for polylines – sharp corners. Mathematically, it is related to the derivative for finding tangents and normals.

Slide 11  **WORKFLOW:**

Using this component, a number of small parcels along a street network can be extruded based on setback for each street. The street name separated by comma and setback distance separated by comma are input. It references a Rhino3d layer for the street centerline geometry and checks which site is closest to what street. Then the corresponding setback value is used to determine the region which can be extruded. Finally based on the FSR, the region is extruded and massing generated.

Slide 12 **ALTERNATIVES:**

Illustrations of possible variations are shown. The curve can be of any type and the numerical inputs changed at runtime to generate results, very quickly.

Slide 13 **GENERATE LAYOUTS:**

Dots components can help in setting up workflows for urban planning very easily. Here the parcel-street-setback component is used to develop a context. After that the large sites are broken up into smaller pieces and the massing components are used to generate a tentative solution. This is actually quite easy to set this up and apply it or change components for massing typology. Even the partitions generated can be changed in various ways including rotation, number of parcels required or type of partitions.

Slide 14 FUNDAMENTAL COMPONENTS - **CONTROLLED SUBDIVISIONS:**

These components are similar to the previously shown components at urban scale but they are more sensitive to constraints where adjacencies and exact area can be plotted. Typical floor plans may be generated from these but even planning problems can be solved. The input .csv file is simple to understand- the first few columns provide dimensions and the last few cols are meant to provide information about neighbors.

Some components take in a input curve but it is easy to generate the boundary as well. And differences in logic forced us to separate the components.

Slide 15 FUNDAMENTAL COMPONENTS - **CONTROLLED SUBDIVISIONS:**

The first component generates an outer rectangle and places the spaces based on inputs provided. The aspect ratio of the outer rect and rotation can be input to check for various sites.

The second component inputs a site curve and places the spaces based on inputs provided.

Circulation component accepts the spaces and provides circulation – corridor or streets. This component can be used with the simple subdivisions or hand drawn parcels and partition lines as well.

Slide 16 FUNDAMENTAL COMPONENTS - **CONTROLLED SUBDIVISIONS:**

The first component here is exploratory – it places a space on the plane and tries to find the placement for the next best space. This way it proceeds until all spaces are plotted.

The final component places the spaces along the periphery and shuffles them until all neighboring conditions are fulfilled.

Slide 17  **SCOPE & CONSTRAINTS FOR OPTIMIZATION:**

This slide shows the output generated by our components. It takes into account the area required, number of each type of space, and the favorable neighbors. The excel sheet captures the adjacency values. +ve values denote attraction whereas -ve values capture the repulsive force. The number value is the intensity. In this case, a&c should attract each other whereas e should repel other e’s.

Slide 18 **WORKFLOW:**

To a given curve – say floor plate, apply a peripheral space generation then apply a subdivision component successively. The input to .csv file is simply a component which can be set to the required path by double-click.

Once again, the same spreadsheet is used where a&c attract and e repels.

Slide 19 **WORKFLOW:**

This shows three connected components with hierarchies of spatial structures. First, spaces are placed on the plane, then a particular space is occupied along the periphery and finally another space is subdivided internally. For each operation, area and neighborhood is considered.

Slide 20 **APPLICATION:**

A vast range of curves were tested, and it is known that sometimes the output will be garbage But upon changing the parameters slightly, even by decimal values, appropriate results could be retrieved.