

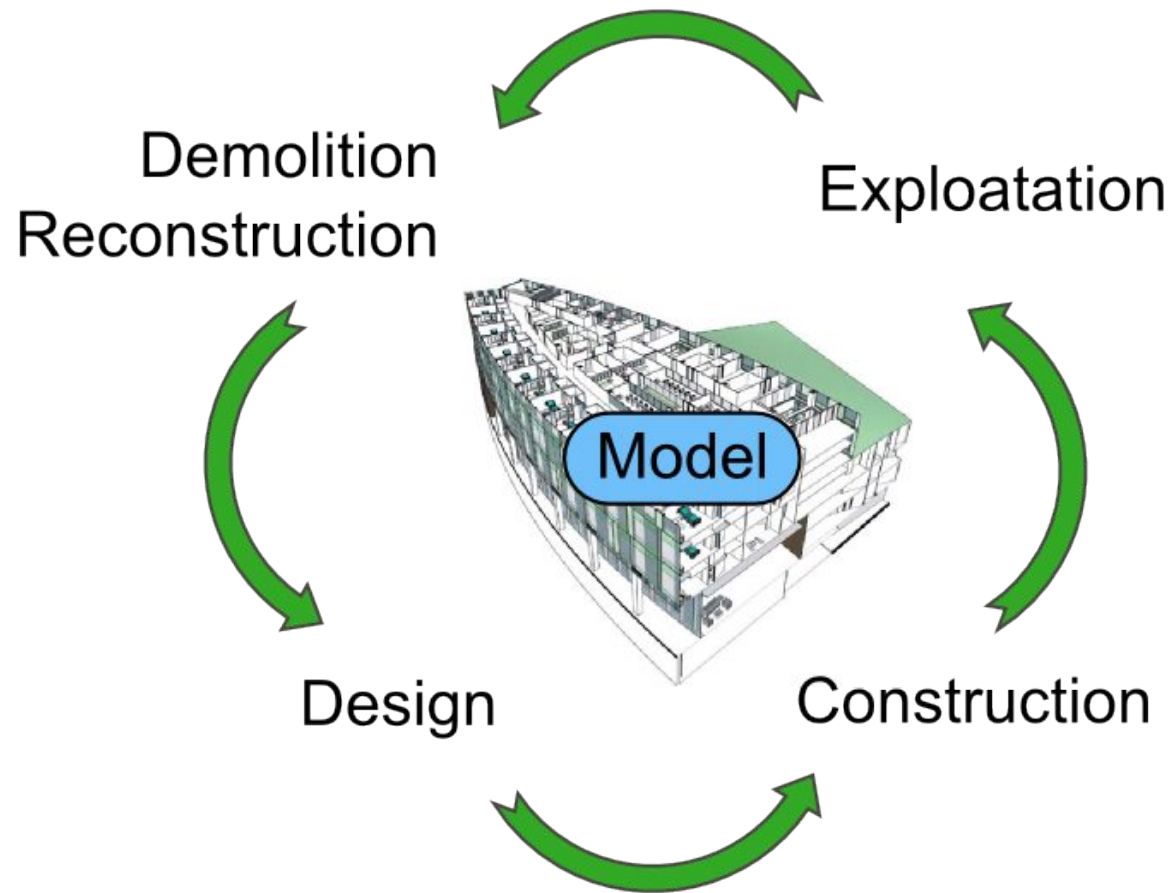
# Blender in architecture

- Everyone knows that blender is cool handling 3dviz/modelling jobs.
  - Some people use it for architecture even now!
  - Still.. there is space for improvement.
- Why it is (could be :) ) better than other tools in market?

# CAD vs BIM

- CAD is about lines, solids and meshes. Out of date.
- BIM is about model in general. You describe the essence of the building. Blueprints should just pop out of it magically.

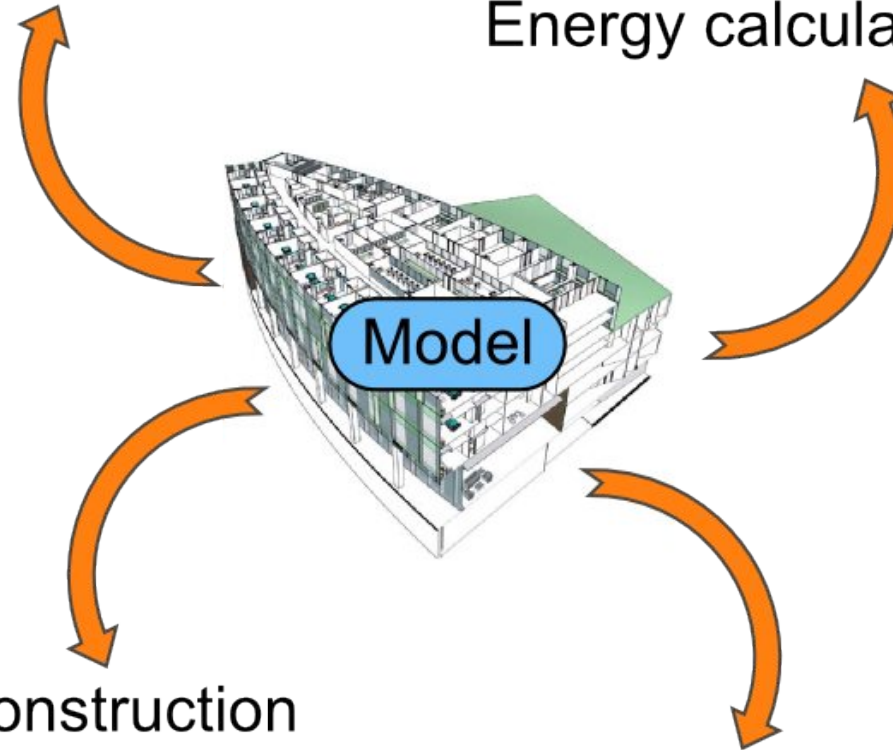
# Building information modeling



# Model

Building management

Energy calculations



Construction  
documentation

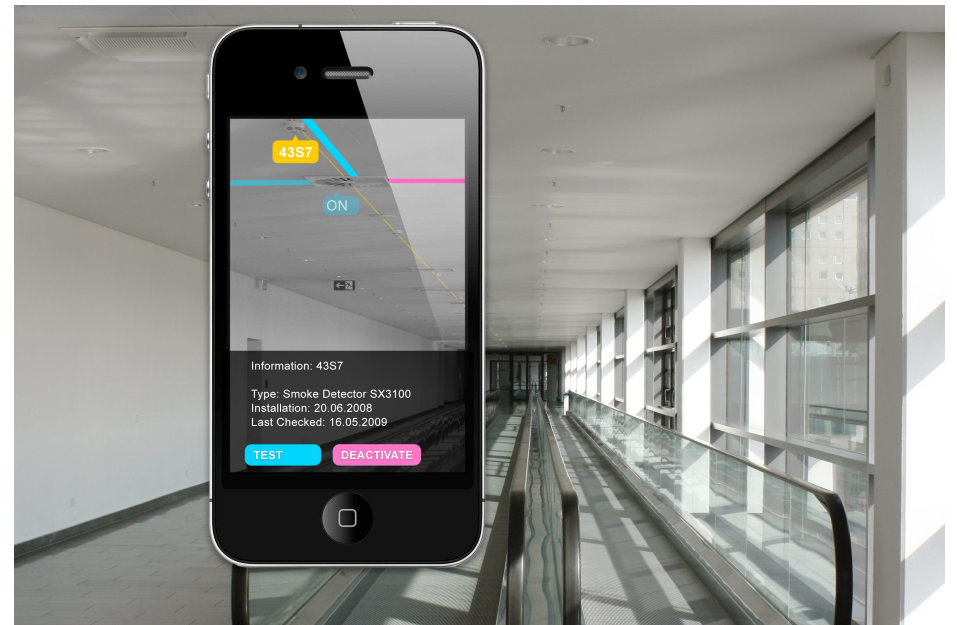
Material count

# Situation in market

- Still, \*\*\* A LOT \*\*\* of architects use old-school CAD here.
- Market demands BIM:
  - Better decisions
  - Greater predictability
  - Less conflicts and collisions
  - Faster project delivery
  - Better project maintainability through all lifecycle

# PLUS

- Augmented reality:
- Ability to see through the wall – pipes, electricity cables etc with iphone
- Easier construction & maintenance



# Big four

- Autodesk Revit
- Autodesk Architectural Desktop
- Graphisoft ArchiCAD
- Bentley Systems

# Competitiveness

- Power of open source!
- Big projects repetitive jobs costs could be cut by customising software – API's available for closed-source projects don't always work out
- Better competition for training, support and development service providers



# Leaps and bounds

- Lack of tools & special functionality
  - not too hard to solve!
  - details in our feasibility study
- It's worth to color the gray area between CAD/Modelling and 3DViz -to have everything in one box

# If talking about BIM...

- Real things are made from objects. Not lines. Not polygons.
- It has to be possible to customize the object without re-designing it completely.
- In example, change height of the table by modifying height parameter.

# Basic parameters

▼ Component	
Info	
Category	Furniture
Name	Table





▼ Component

Table

Hosts

Level	Level 1
Level offset	0

Spatial

Width	200
Depth	300
Height	500

Detailed description: This is a screenshot of a software component inspector. At the top, there is a dropdown menu labeled 'Table' with an orange highlight. Below it, the 'Hosts' section contains two rows: 'Level' with a value of 'Level 1' and 'Level offset' with a value of '0'. The 'Spatial' section contains three rows: 'Width' with a value of '200', 'Depth' with a value of '300', and 'Height' with a value of '500'. Each value is displayed in a light gray box with left and right arrow icons for adjustment.

# How is this possible?

- Reference planes (Refplanes). They are everywhere.

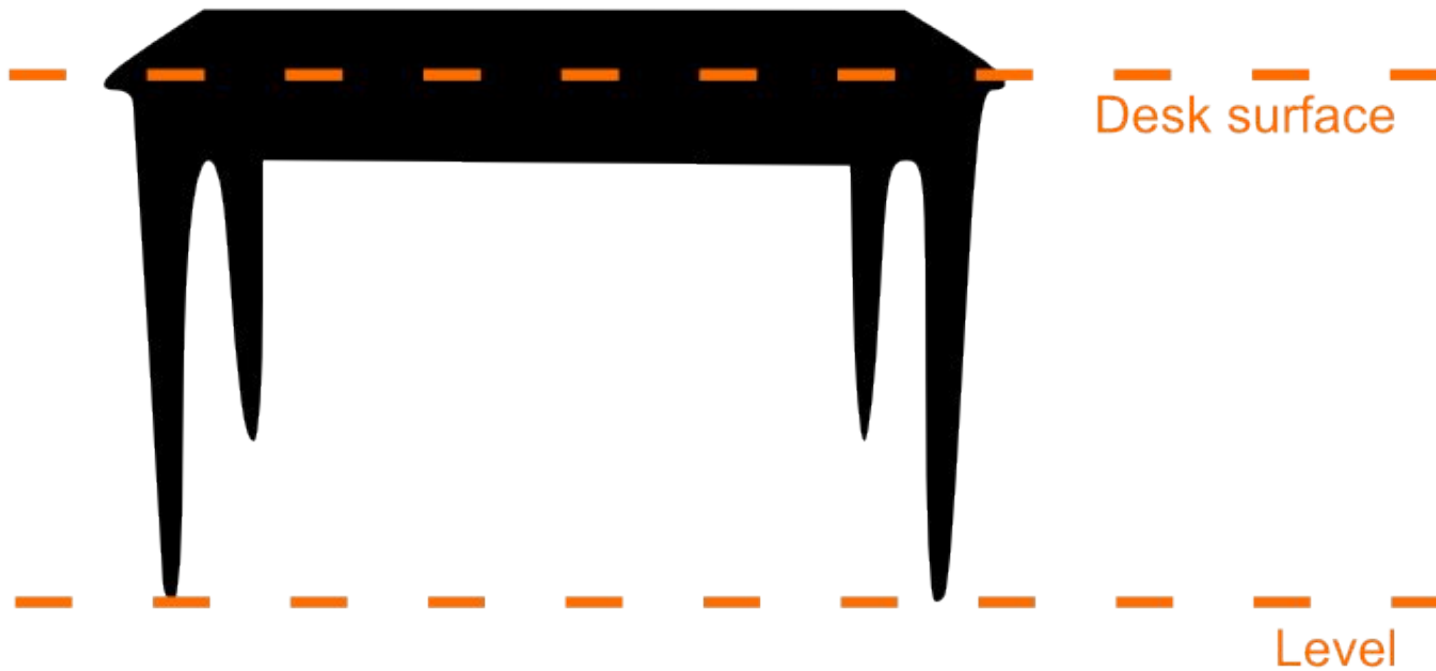
# Refplanes

We have a object. Actually, we have a table.

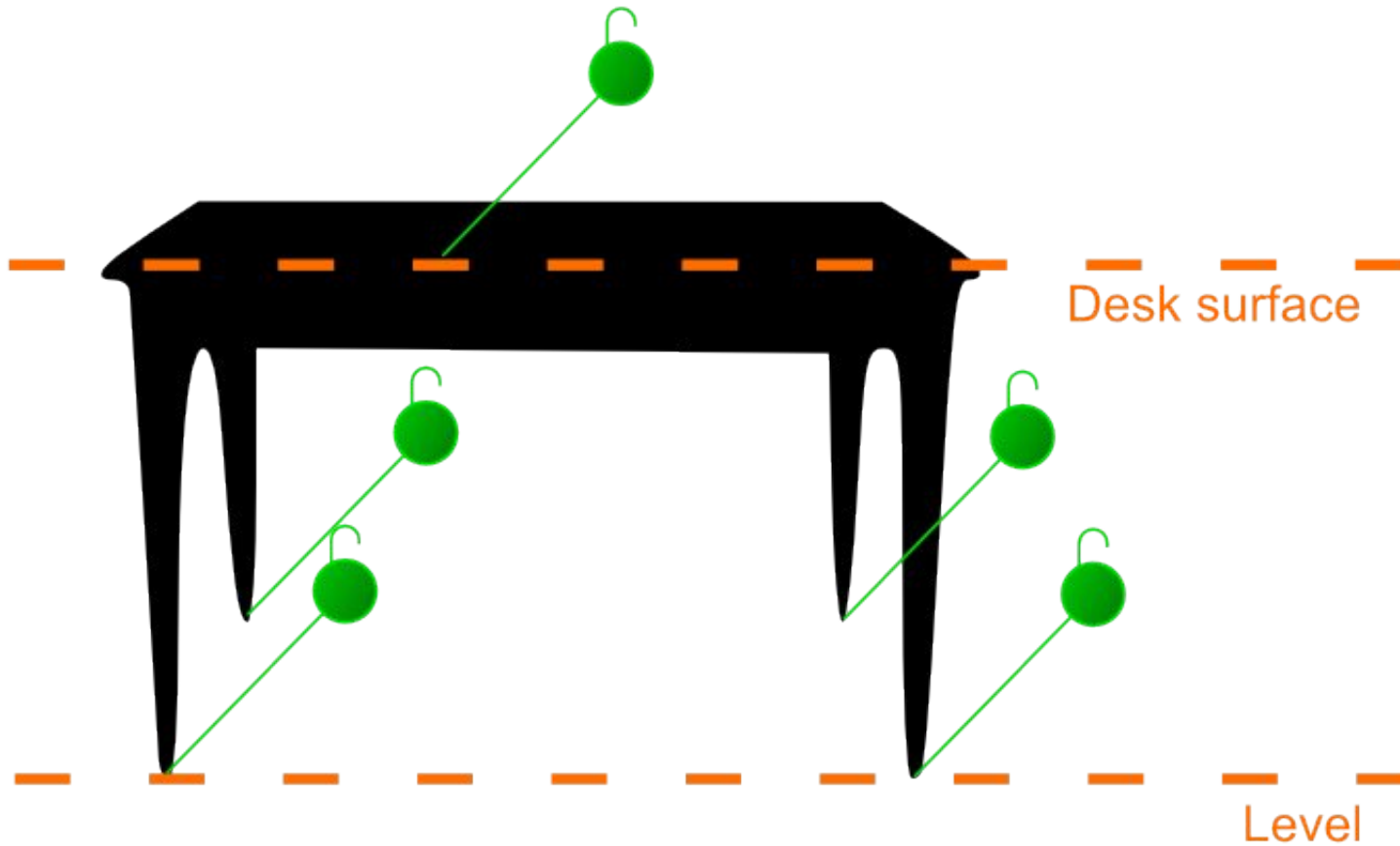


# Refplanes

Let's add it a surface and floor level Refplanes.  
It will be possible to lock it to other objects (in example, room floor).



# Locking geometry to Refplanes

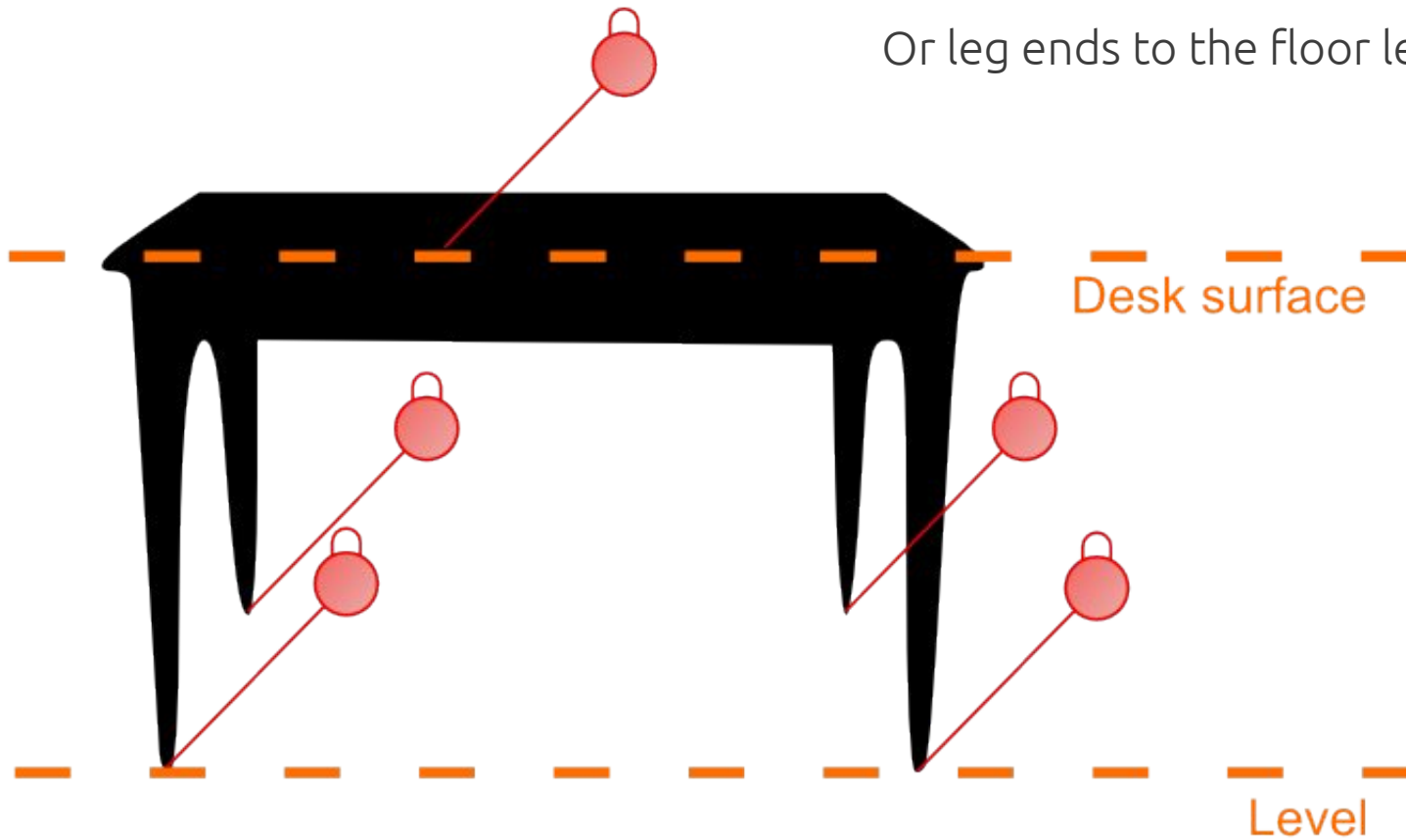




# Locking geometry to Refplanes

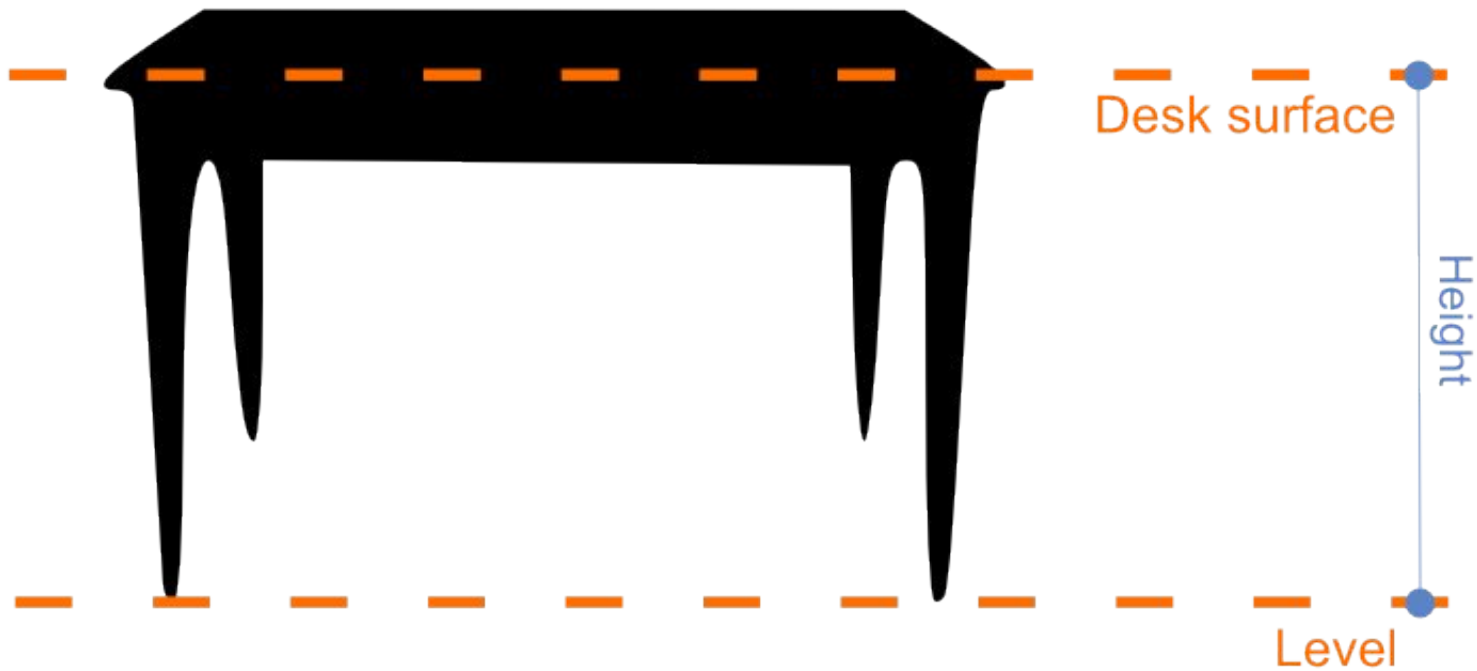
You can lock object parts to refplane.  
For example, desk surface to desk surface refplane.

Or leg ends to the floor level Refplane.



# Adding parameters

Then, it is possible to add parameters between refplanes. Let's specify table legs height for instance.



# Geometry changes according parameter values



Then, when we have instance of the object, our new parameter appears in Object properties box.

# Geometry changes according parameter values

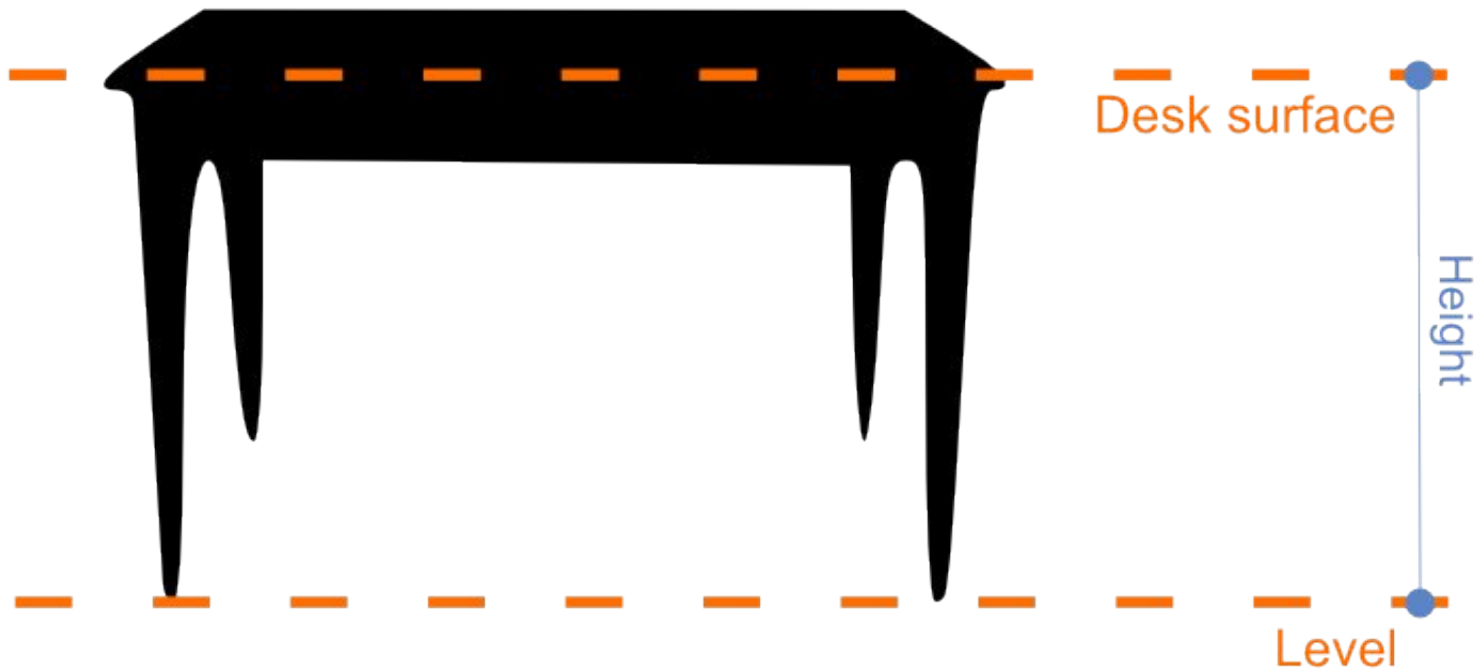


We change it, and table becomes lower.

# Geometry changes according parameter values



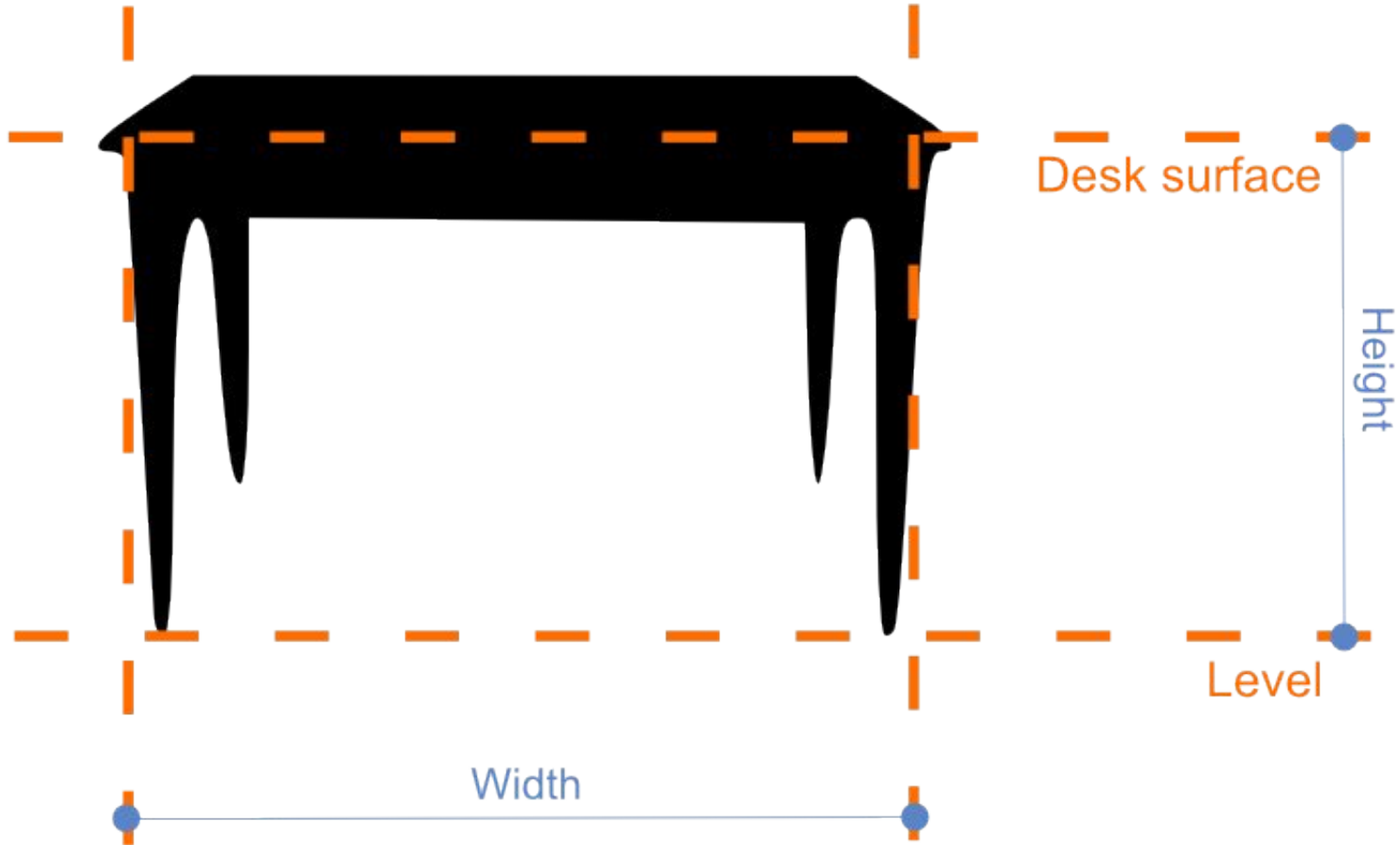
# Controls



Another way to change parameter values

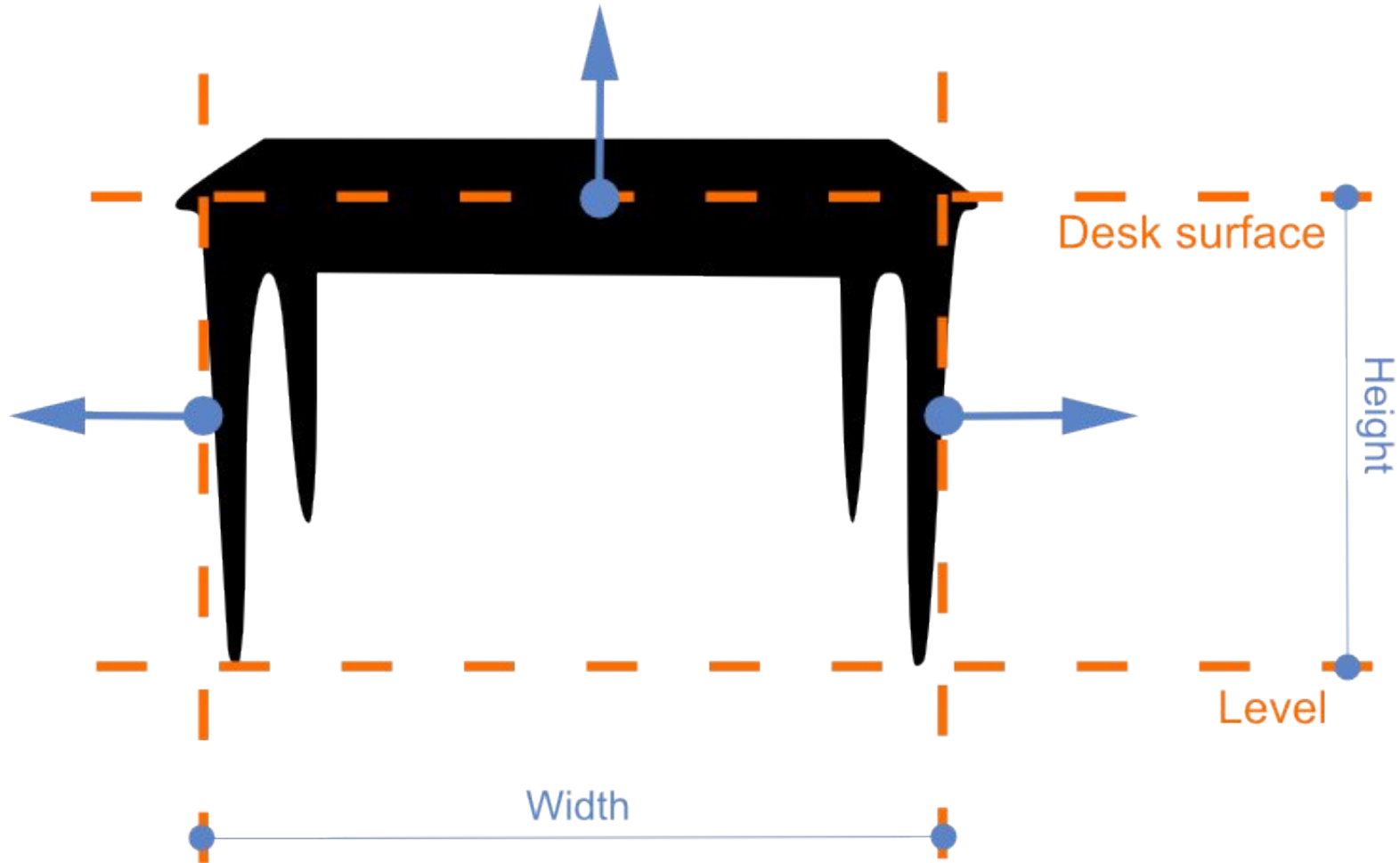
# Controls

Let's add some more replanes



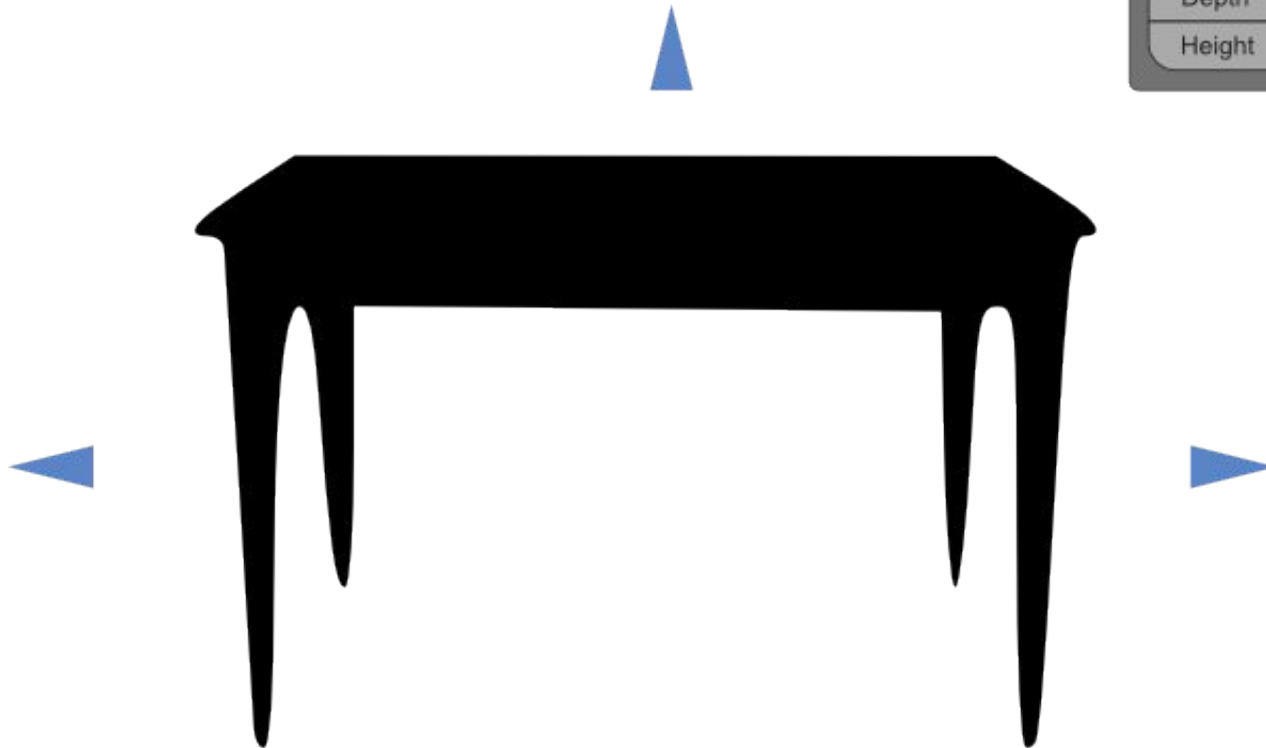
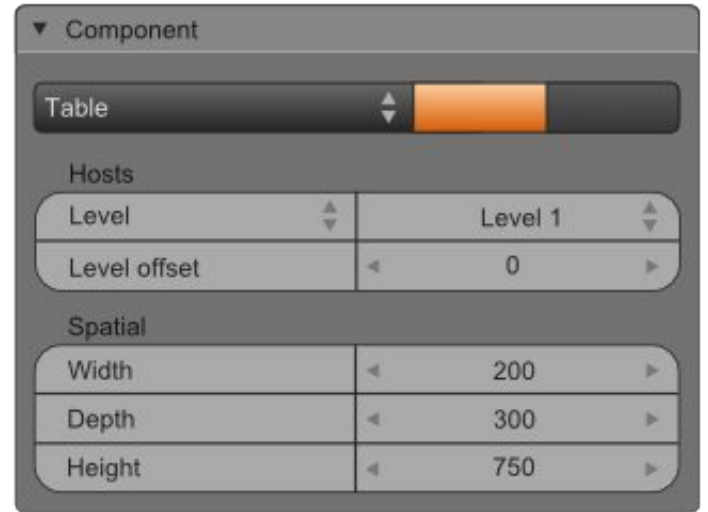
# Controls

And objects defining with refplane user could drag



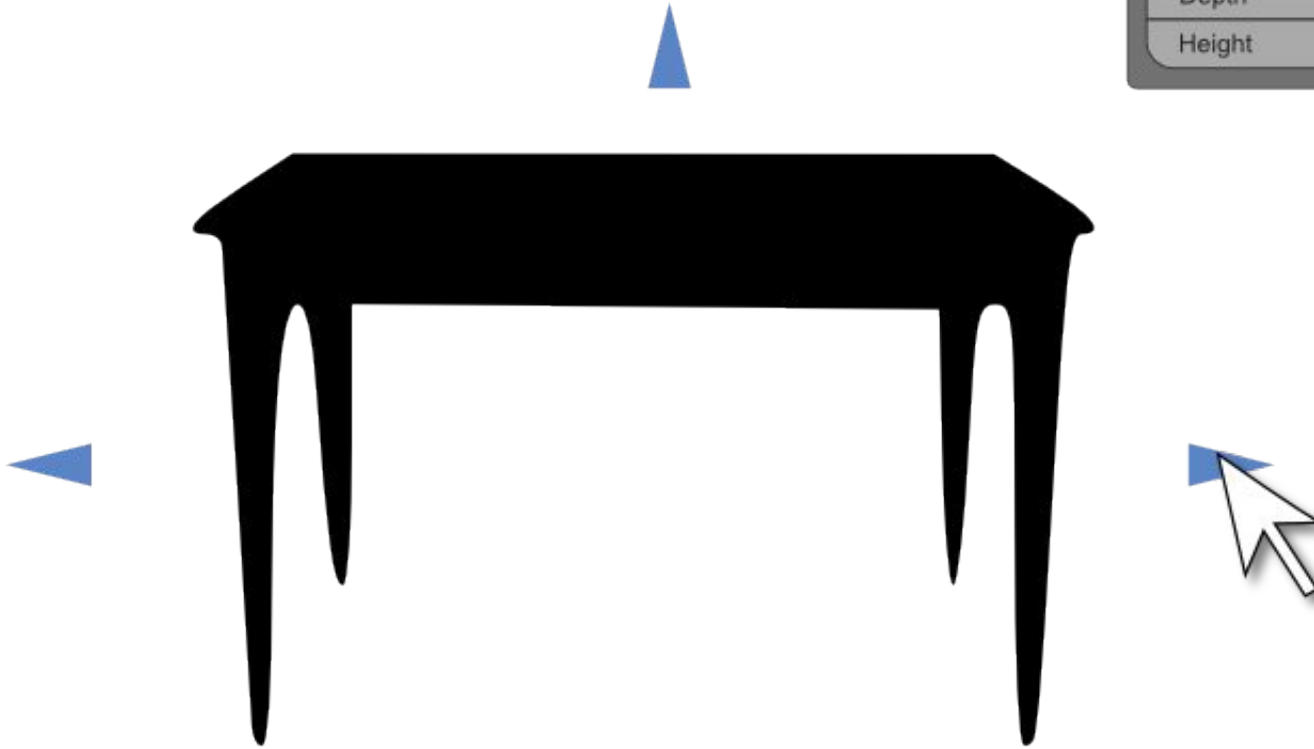


# Controls



So even if we exited edid parametric component mode we can drag now hidden refplanes

# Controls



▼ Component

Table

Hosts

Level	Level 1
Level offset	0

Spatial

Width	1200
Depth	800
Height	750

# Controls

▼ Component

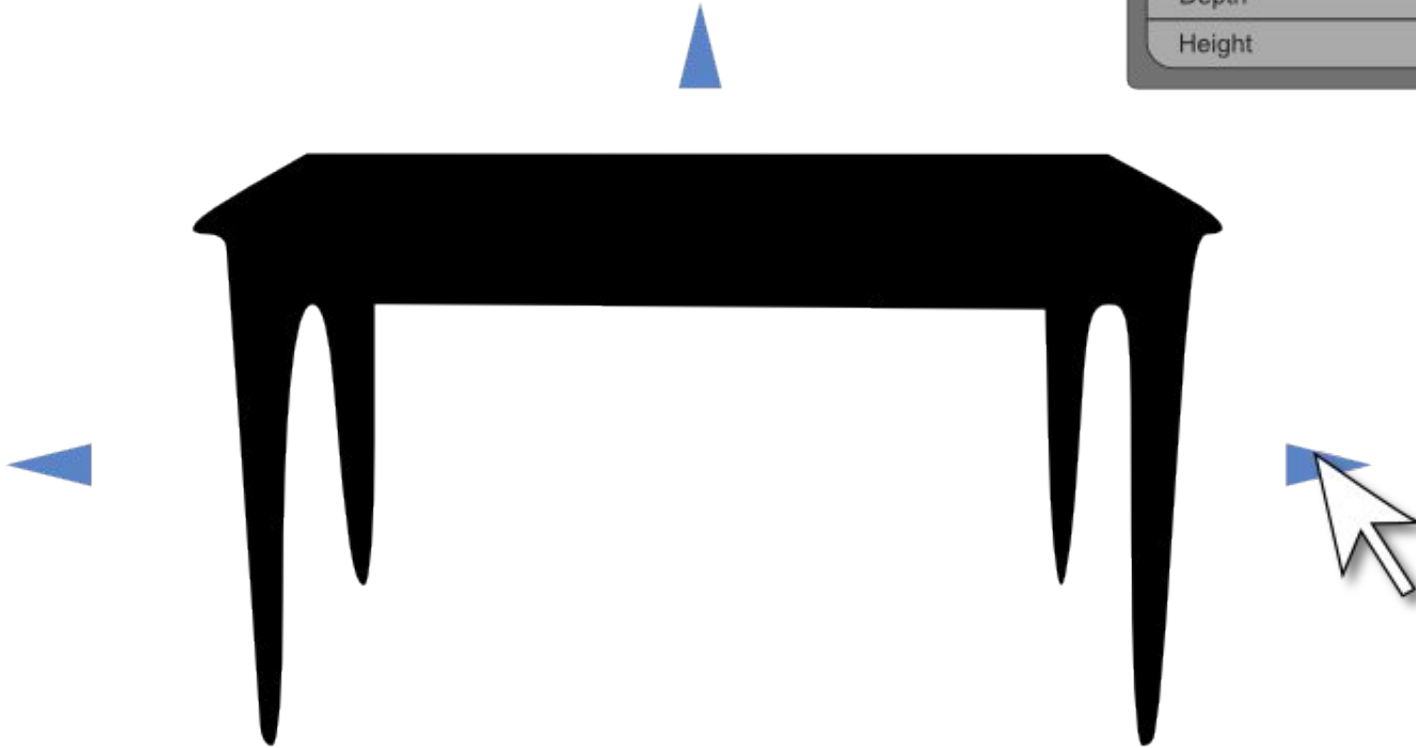
Table

Hosts

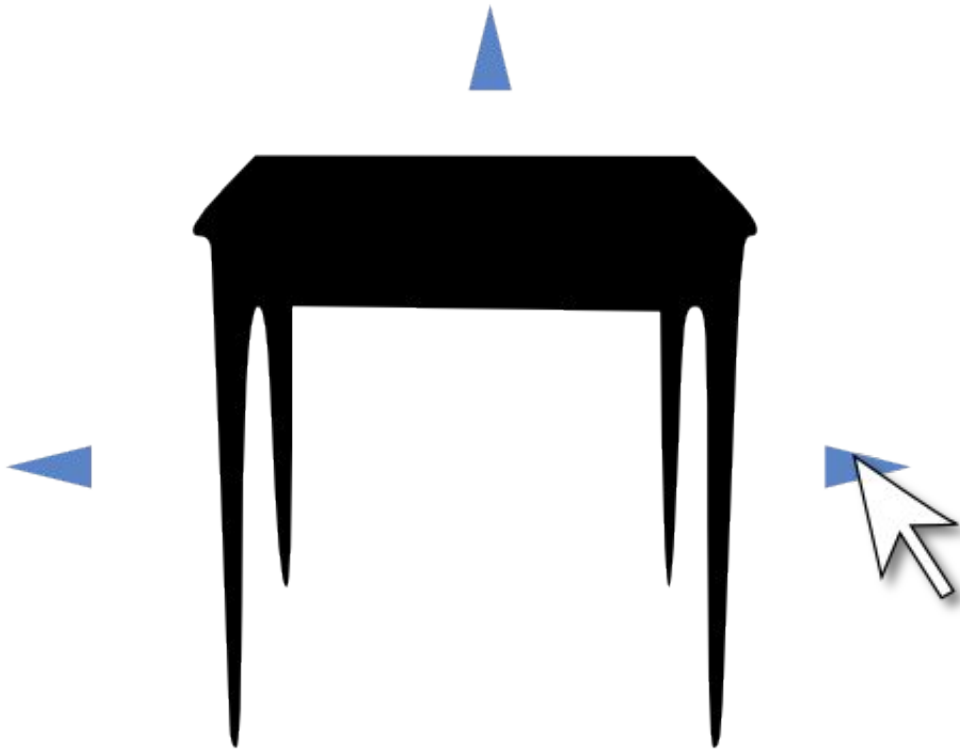
Level	Level 1
Level offset	0

Spatial

Width	1330
Depth	800
Height	750



# Controls



Component

Table

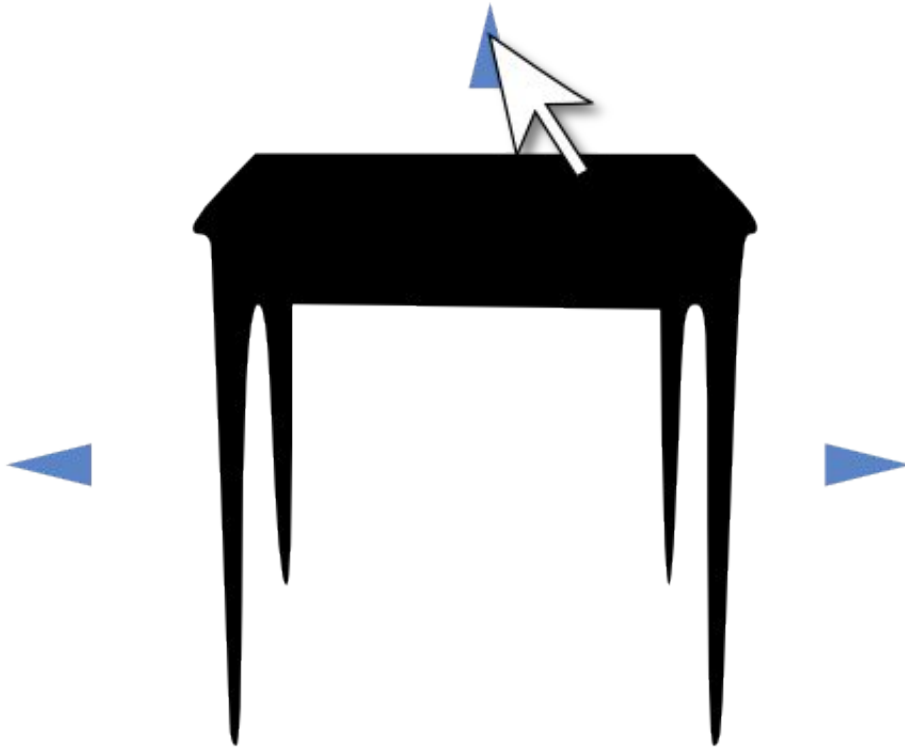
Hosts

Level	Level 1
Level offset	0

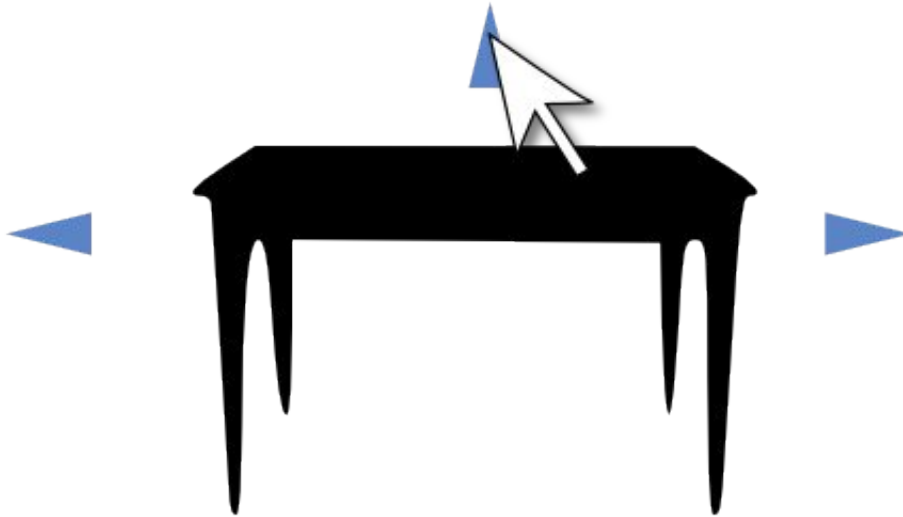
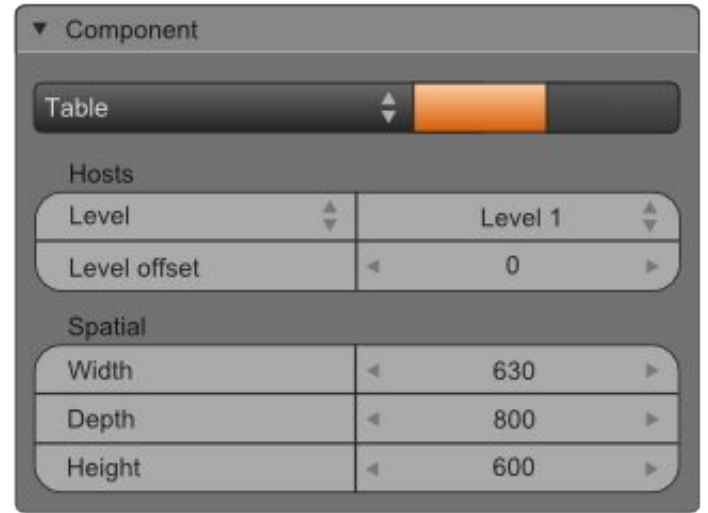
Spatial

Width	630
Depth	800
Height	750

# Controls



# Controls

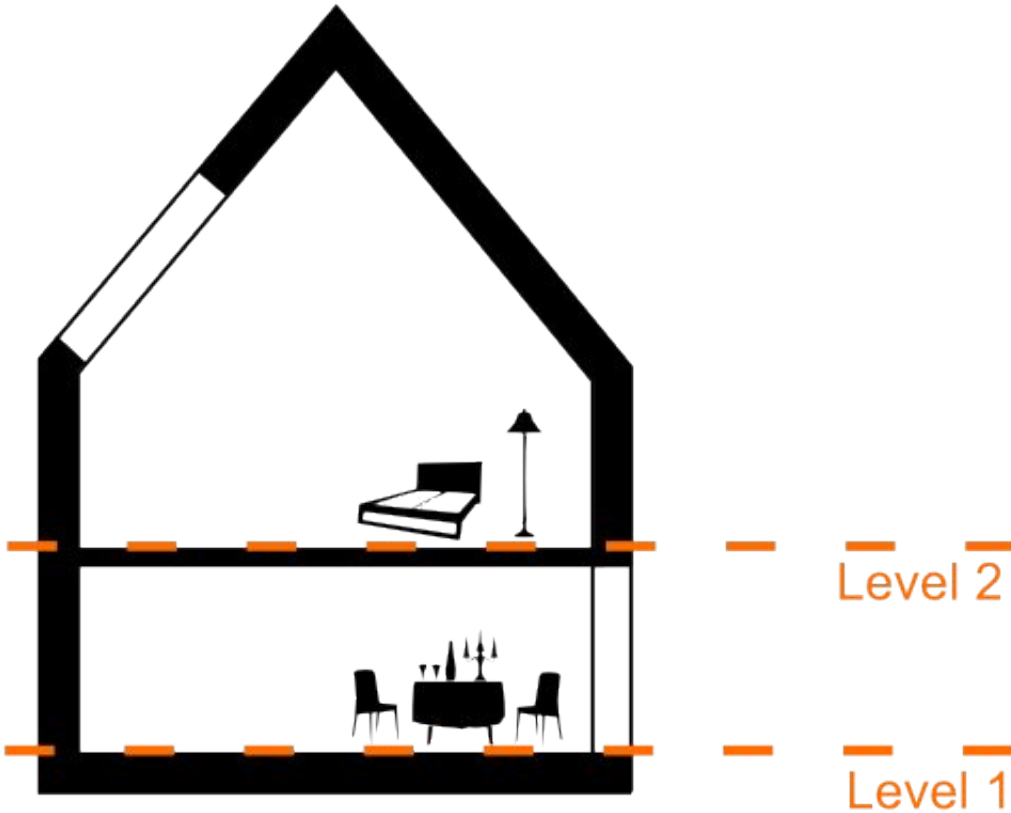


# Very complex situations

- There is nothing that couldn't be programmed by Python on the Parametric Model side
- Powerful API of POs to be developed

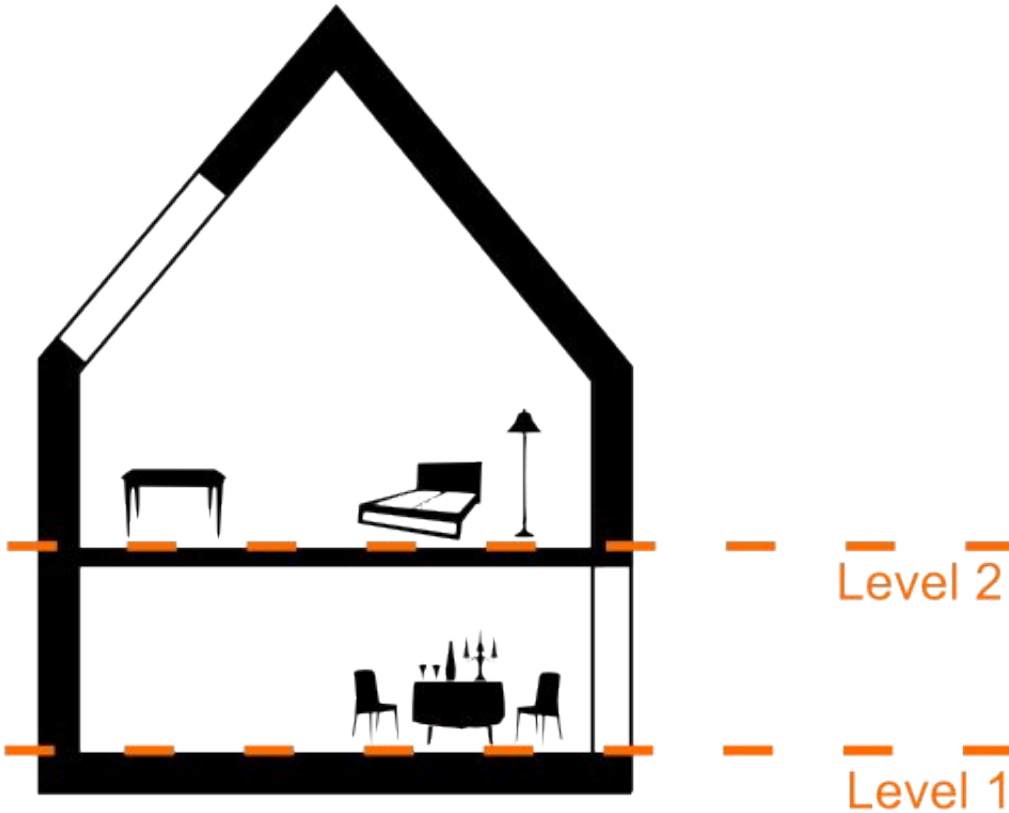


# Refplanes and hosted components



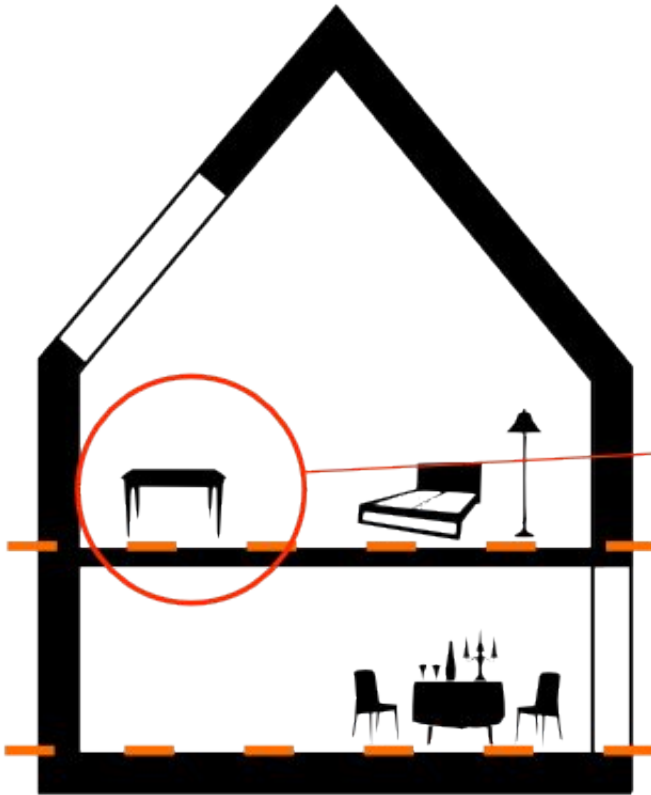


# Refplanes and hosted components



Also refplanes can be used to snap object to parent level refplane.

# Refplanes and hosted components



Level 2

Level 1

▼ Component

Table

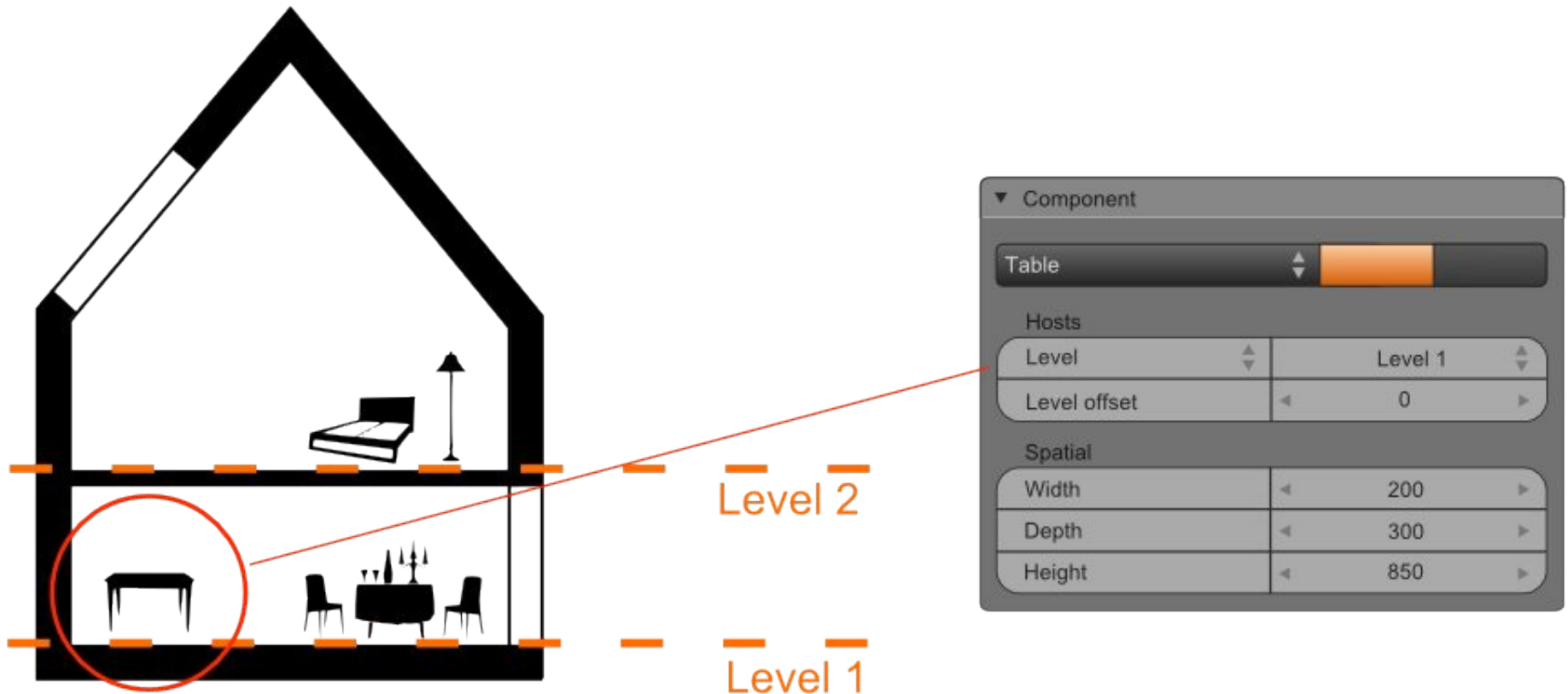
Hosts

Level	Level 2
Level offset	0

Spatial

Width	200
Depth	300
Height	850

# Refplanes and hosted components



You just change hosting parameter value, and object appears in another floor  
Offset allows table to levitate :-)

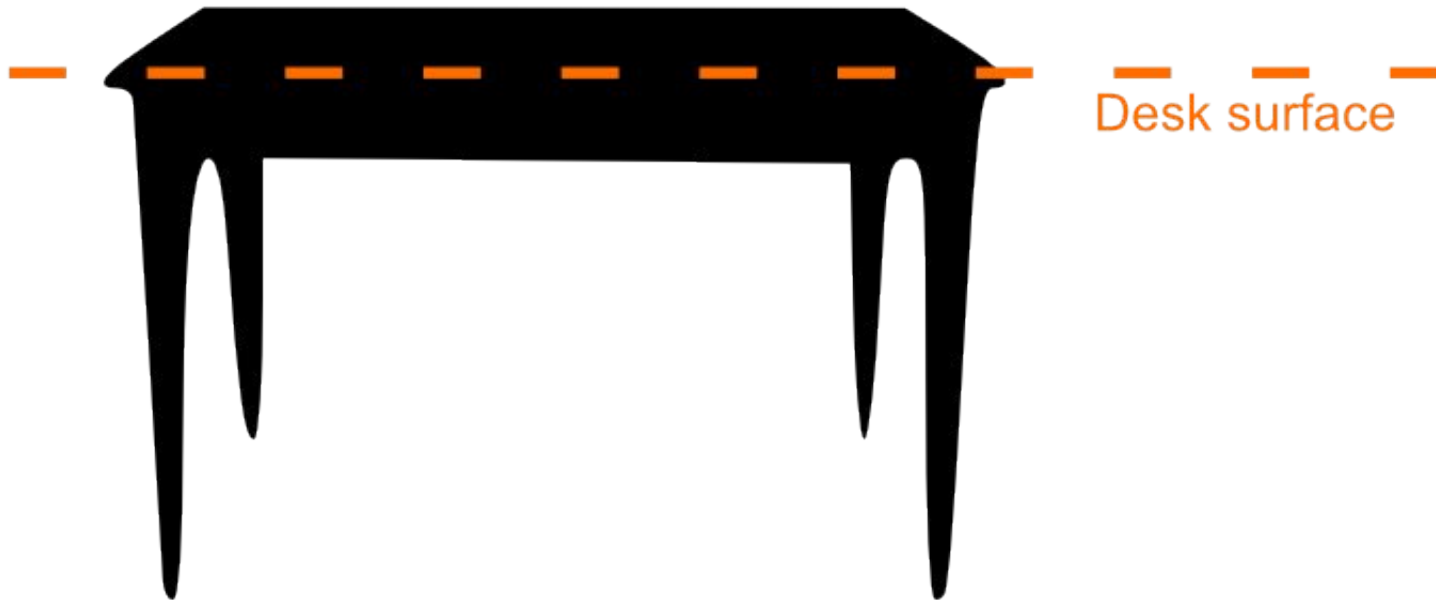
# Refplanes and hosted components



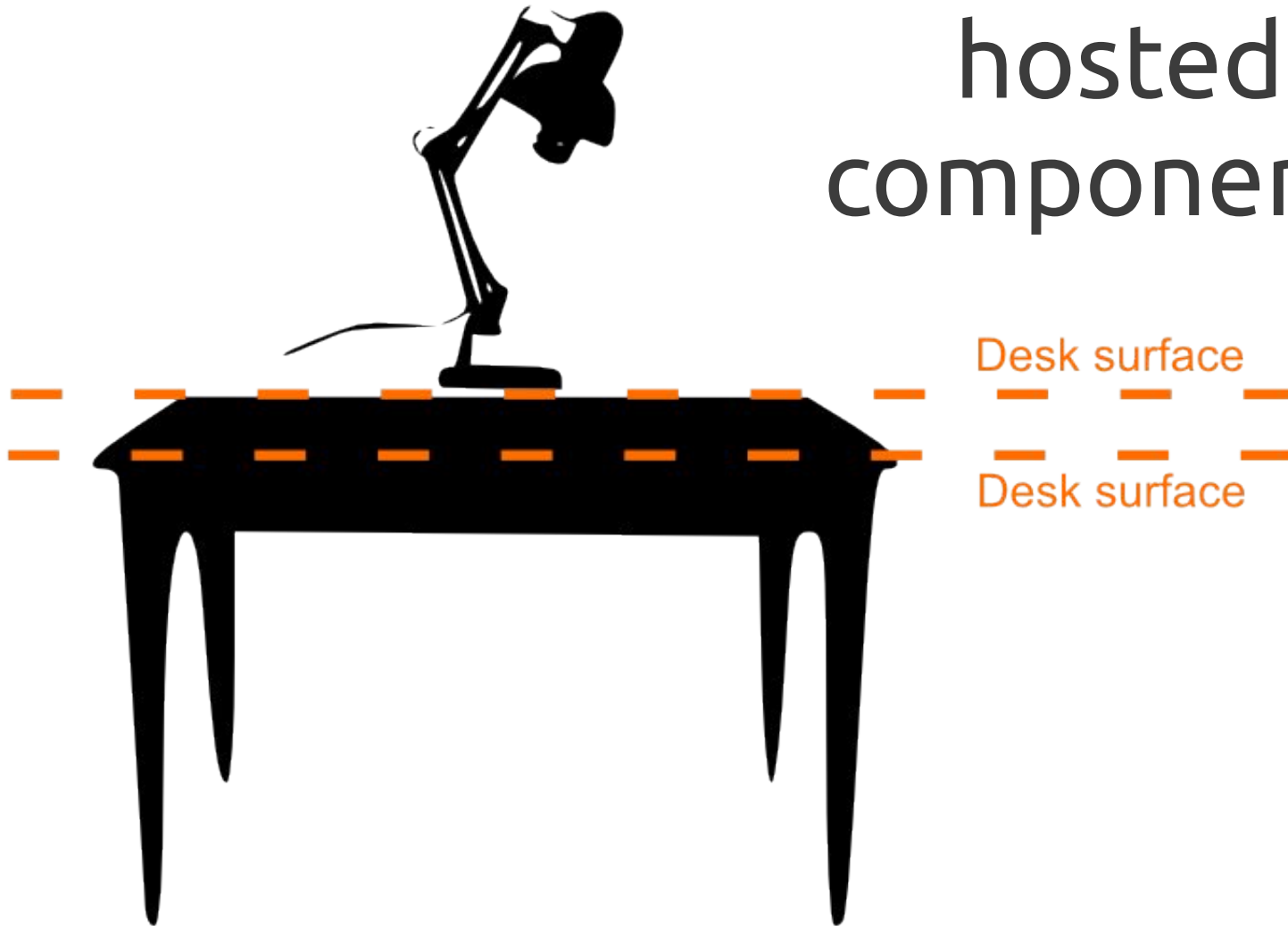
Desk surface



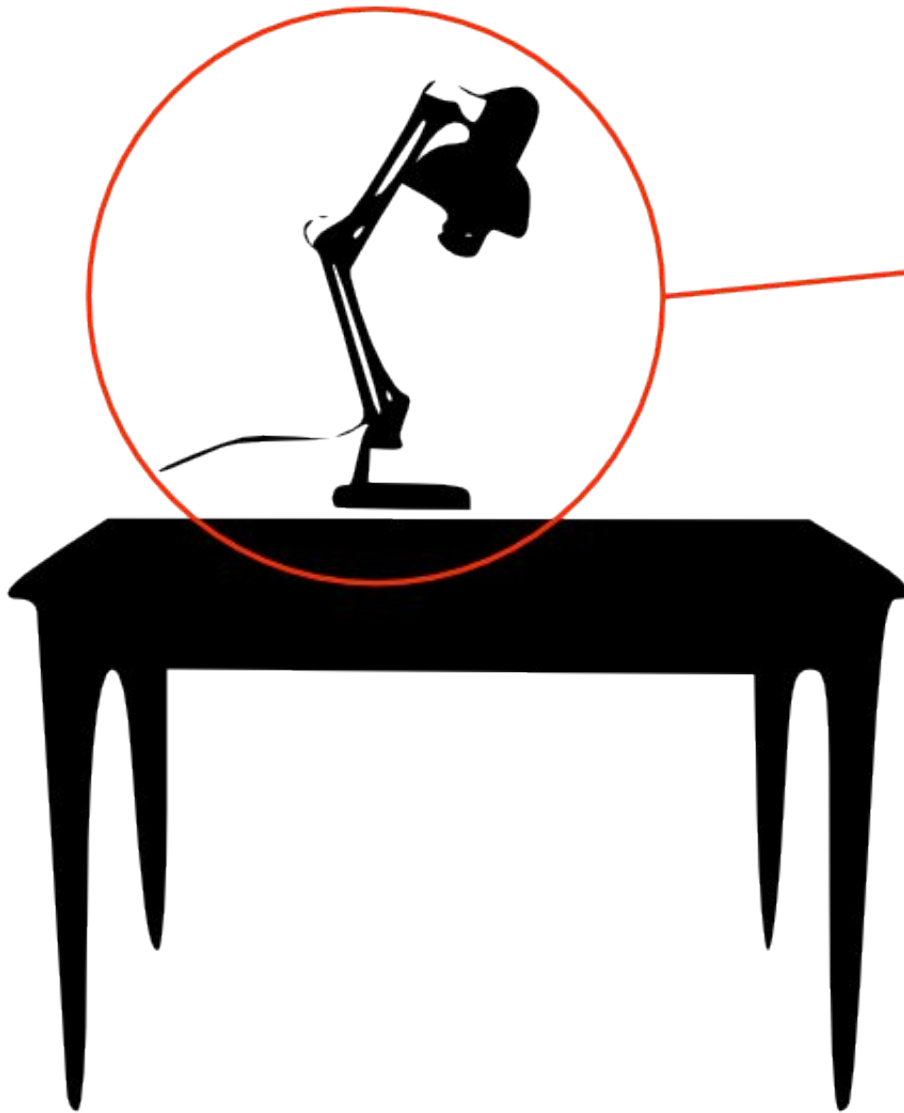
# Refplanes and hosted components



# Refplanes and hosted components



# Offset from host refplane



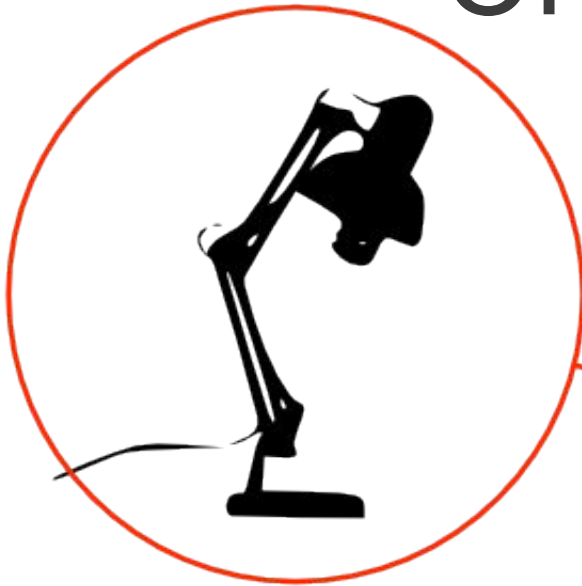
▼ Component

Lamp

Hosts

Desk surface	Desk surface
Offset	0

# Offset from host refplane



Component

Lamp

Hosts

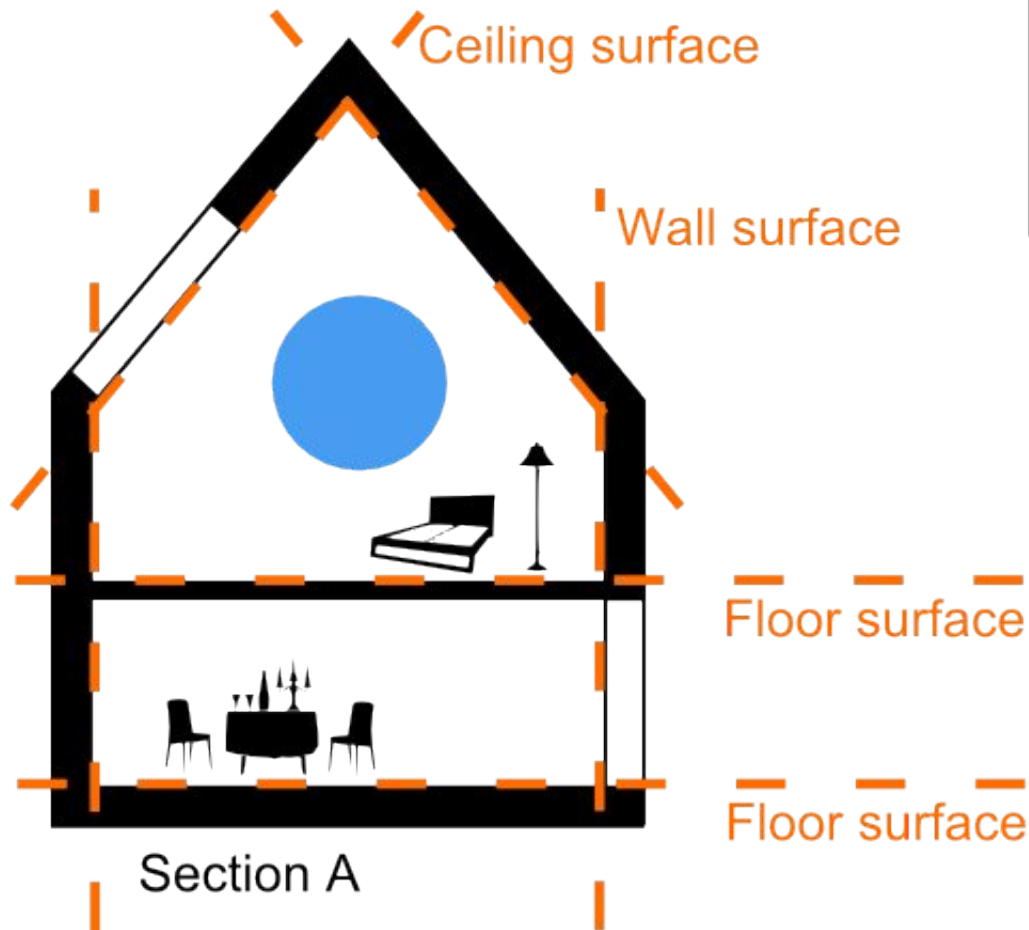
Desk surface	Desk surface
Offset	600

The control panel is a grey rectangular window. At the top, it has a dropdown menu labeled 'Component' with a downward arrow. Below this is a slider control for 'Lamp', with an orange bar indicating its position. Underneath is a section labeled 'Hosts' containing two 'Desk surface' entries, each with a vertical double-headed arrow. At the bottom, there is an 'Offset' entry with a numerical value of '600' and horizontal arrows on either side.



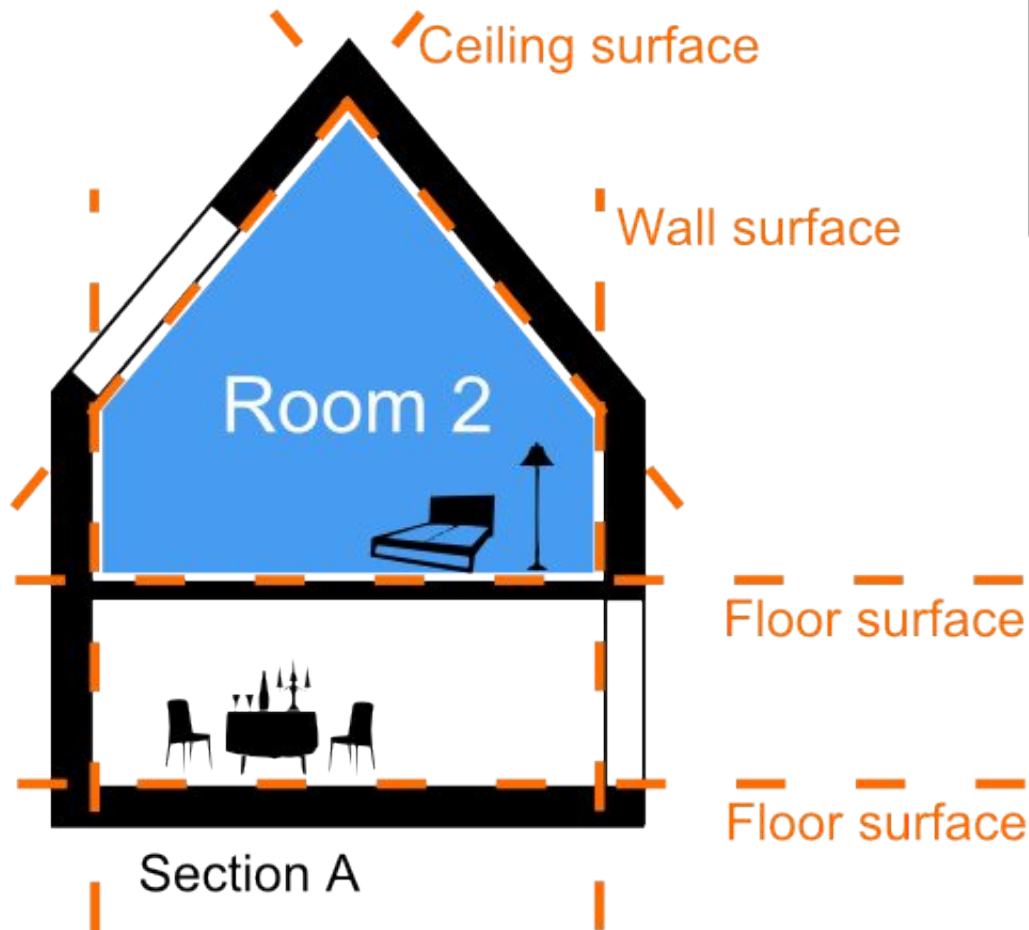


# Adaptive components



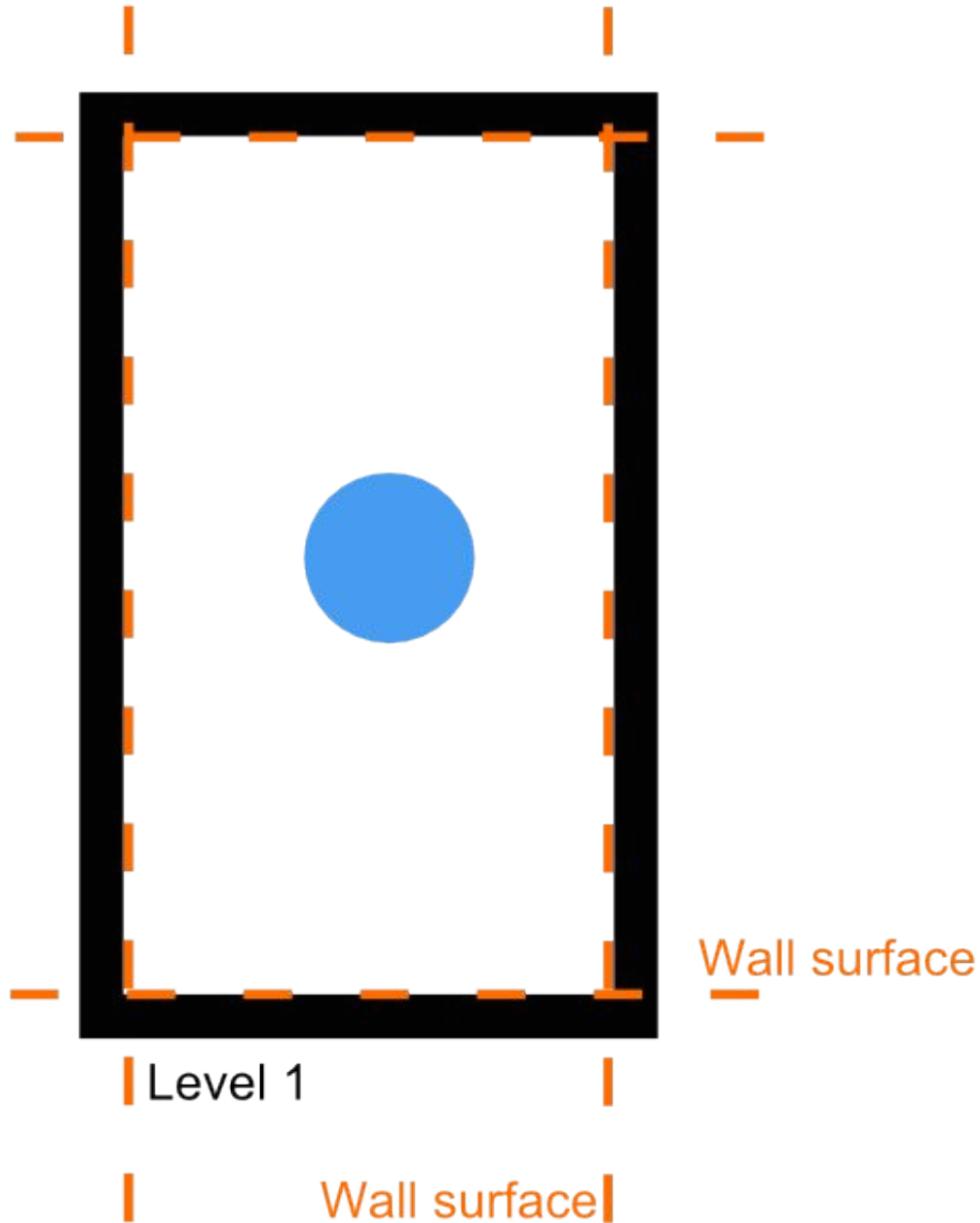
Sometimes it is required to have components that automatically expands  
In desired space, delimited by desired replanes.

# Adaptive components

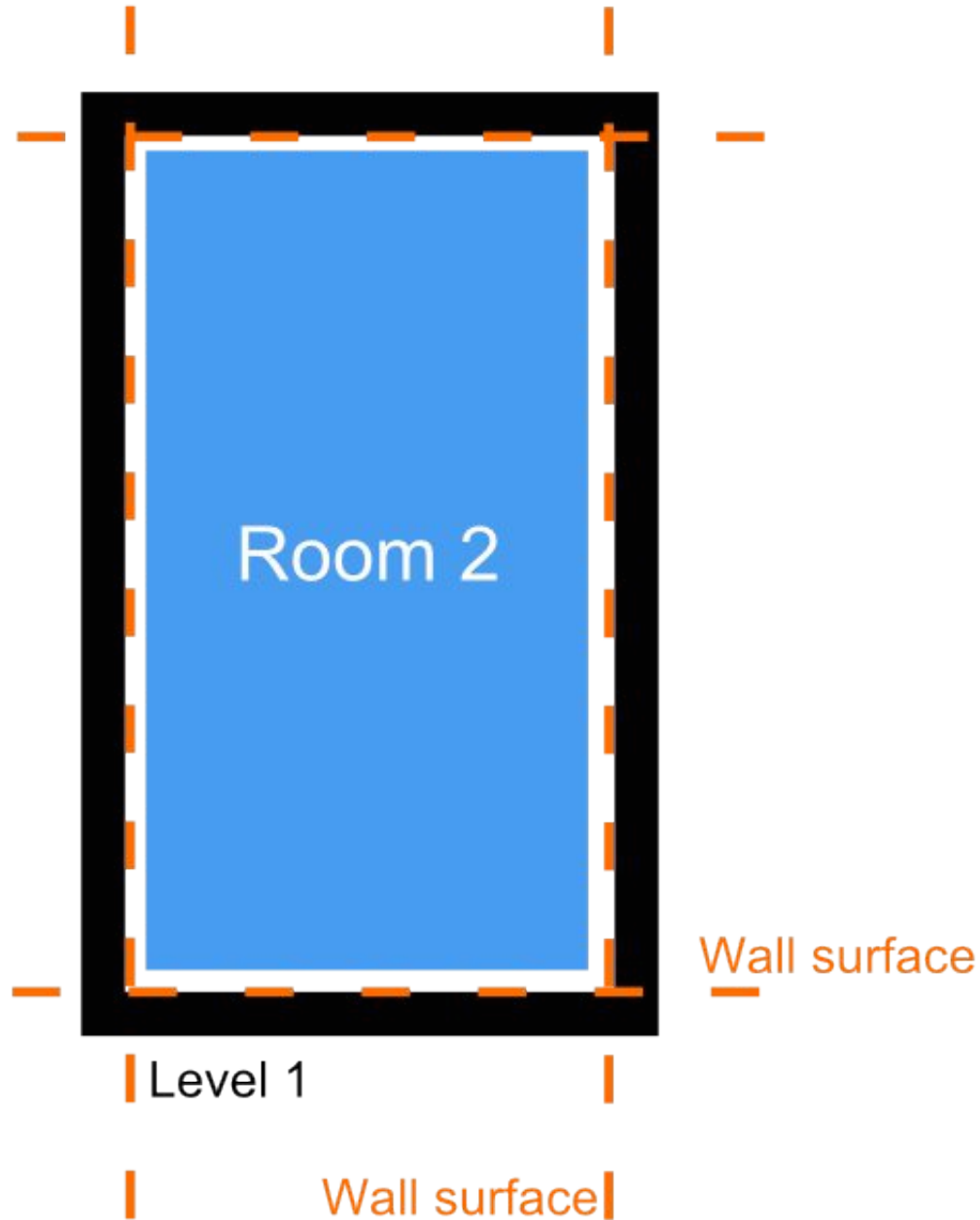


This example allows to calculate room volume. Or to specify a purpose for the area.  
Living room, kitchen, lounge, etc...

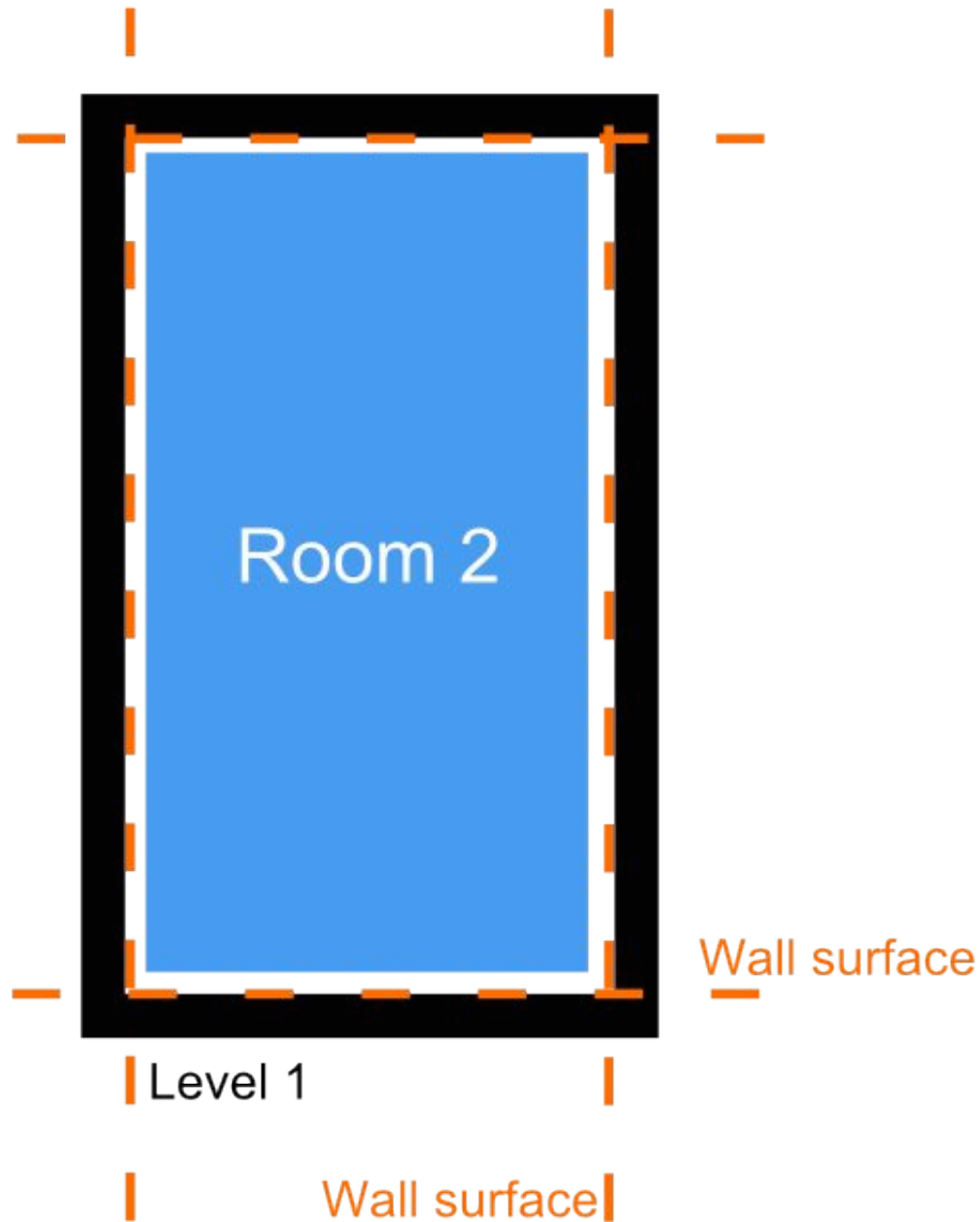
# Adaptive components



# Adaptive components



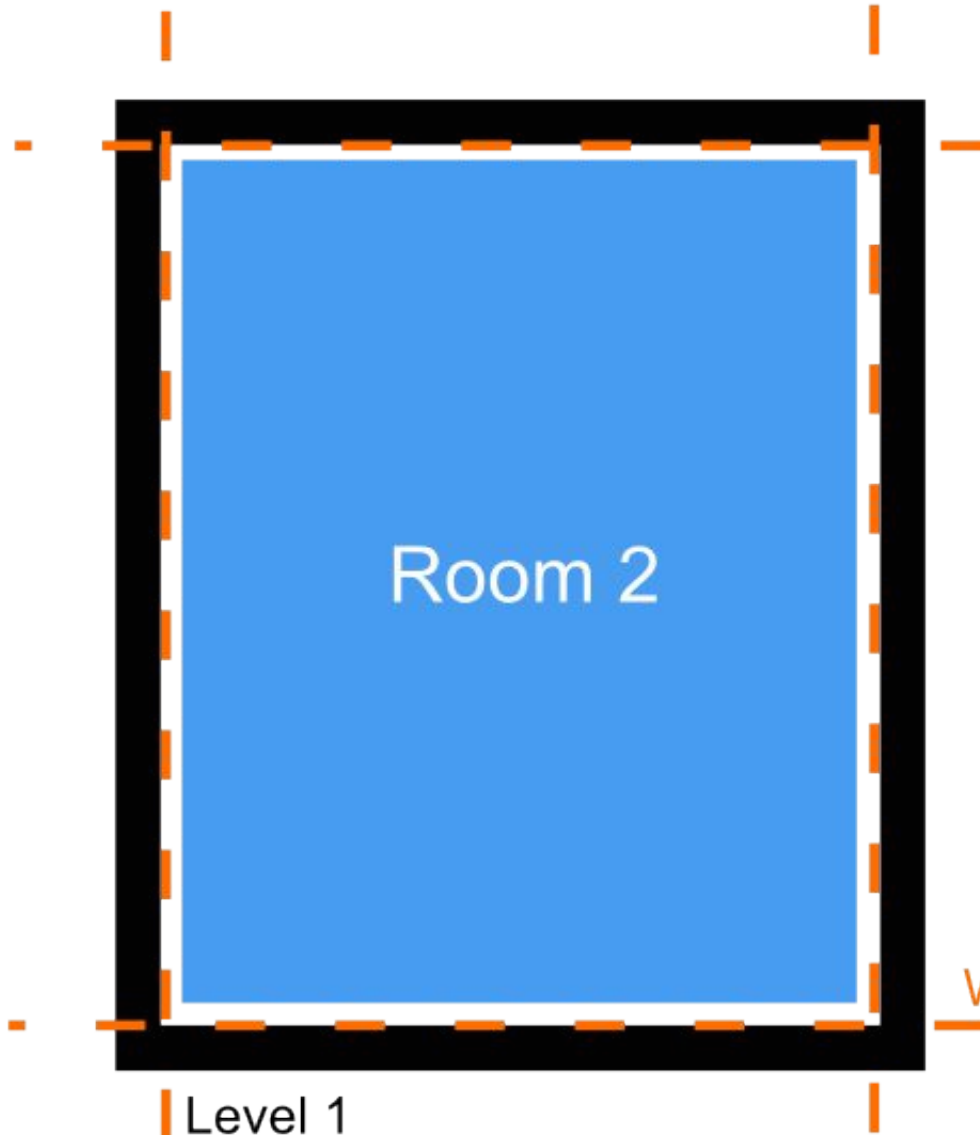
# Adaptive components



Name	Area
Room 1	50 m <sup>2</sup>
Room 2	60 m <sup>2</sup>

In this example (plan view)  
We calculate area of floors by  
counting room components  
and showing them in shedule.  
Shedule column shows area  
parameter value of room  
components.

# Adaptive components



▼ Component

Space

Hosts

Level | Level 1

Constrains

Top	Ceiling surface
Bottom	Floor surface
Side	Wall surface

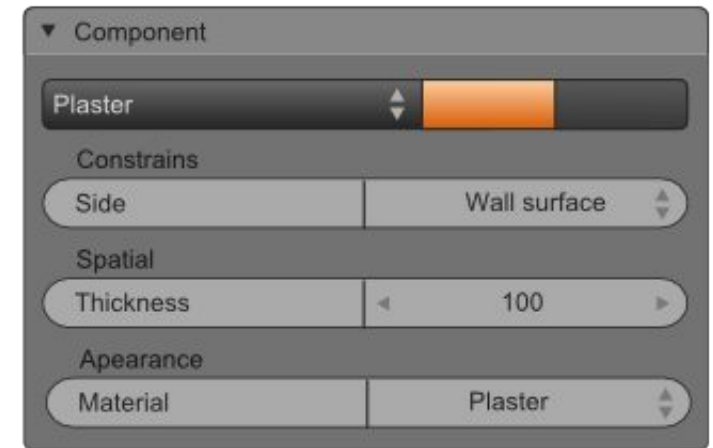
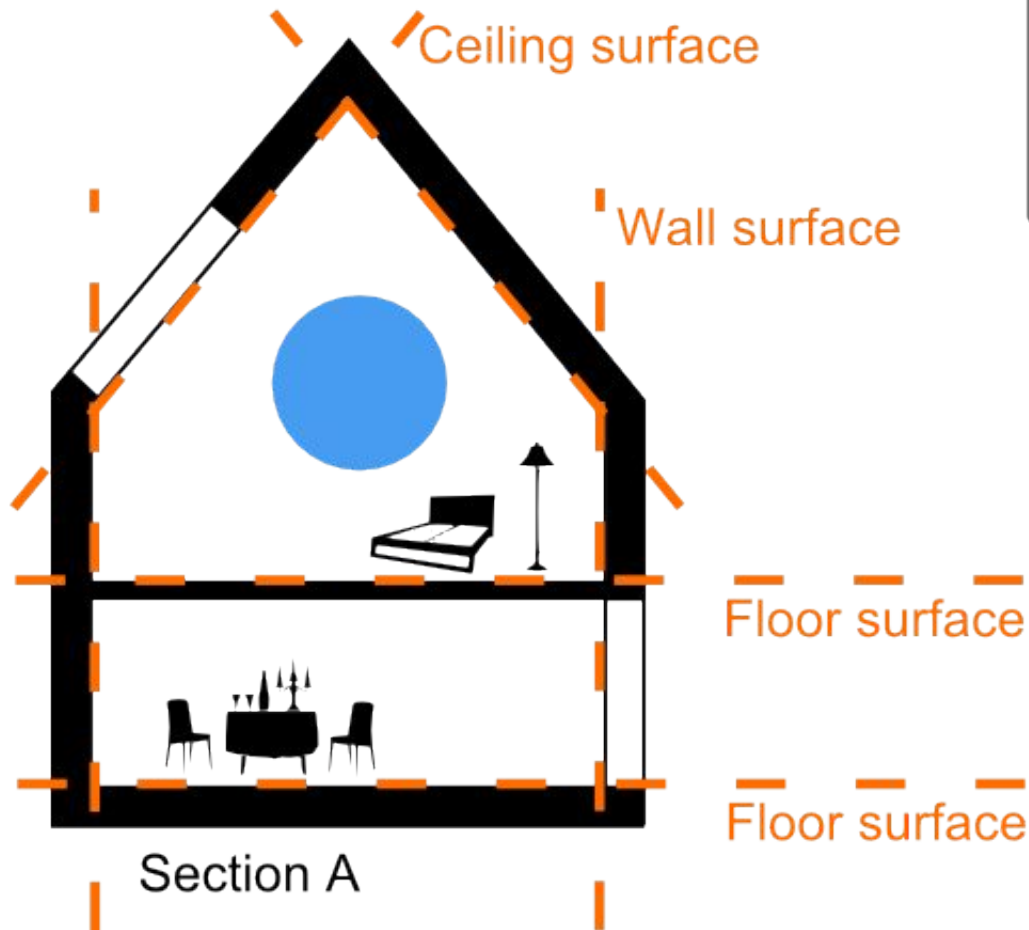
Name	Area
Room 1	50 m <sup>2</sup>
Room 2	80 m <sup>2</sup>

If we move wall, room component automatically extends to fill new space.  
Area parameter updates

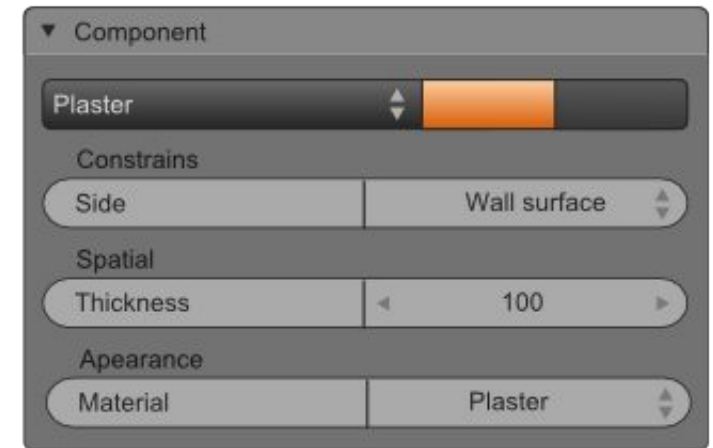
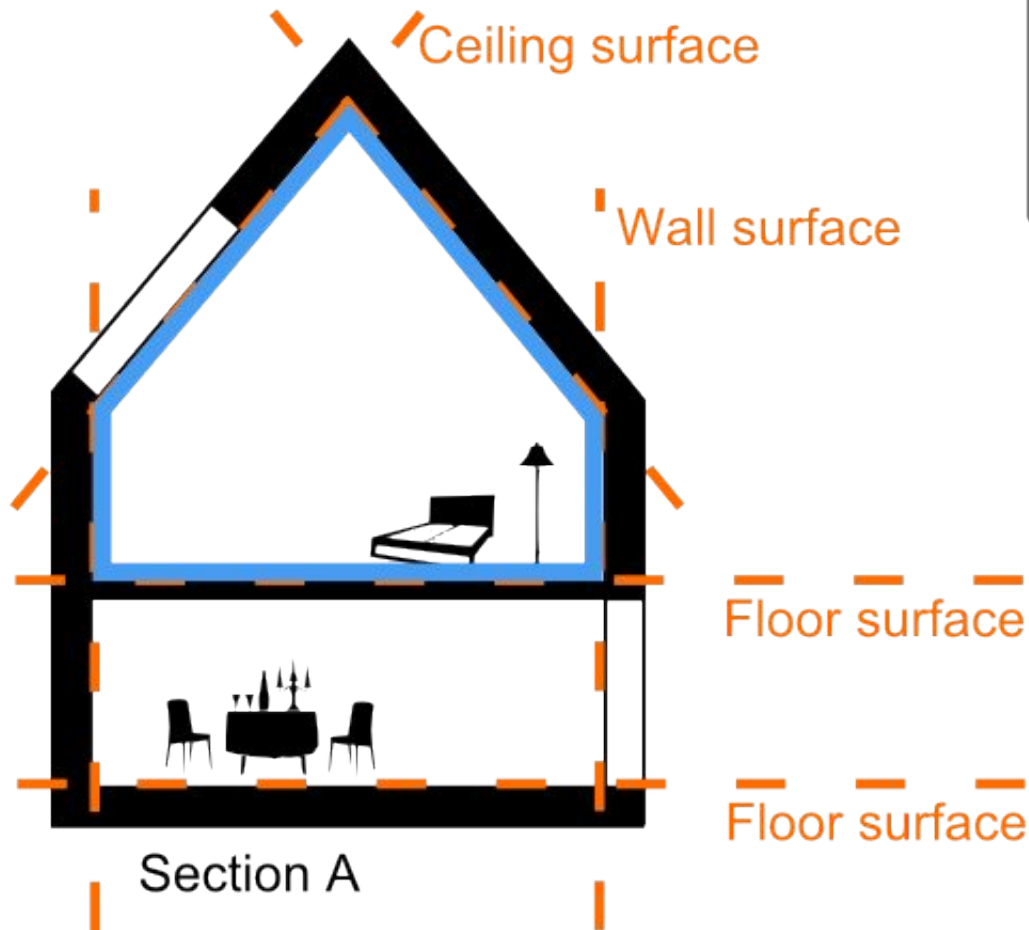
Wall surface

Wall surface

# Adaptive components



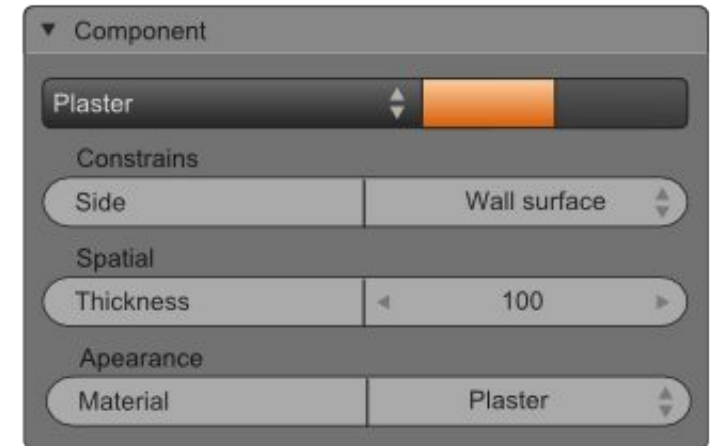
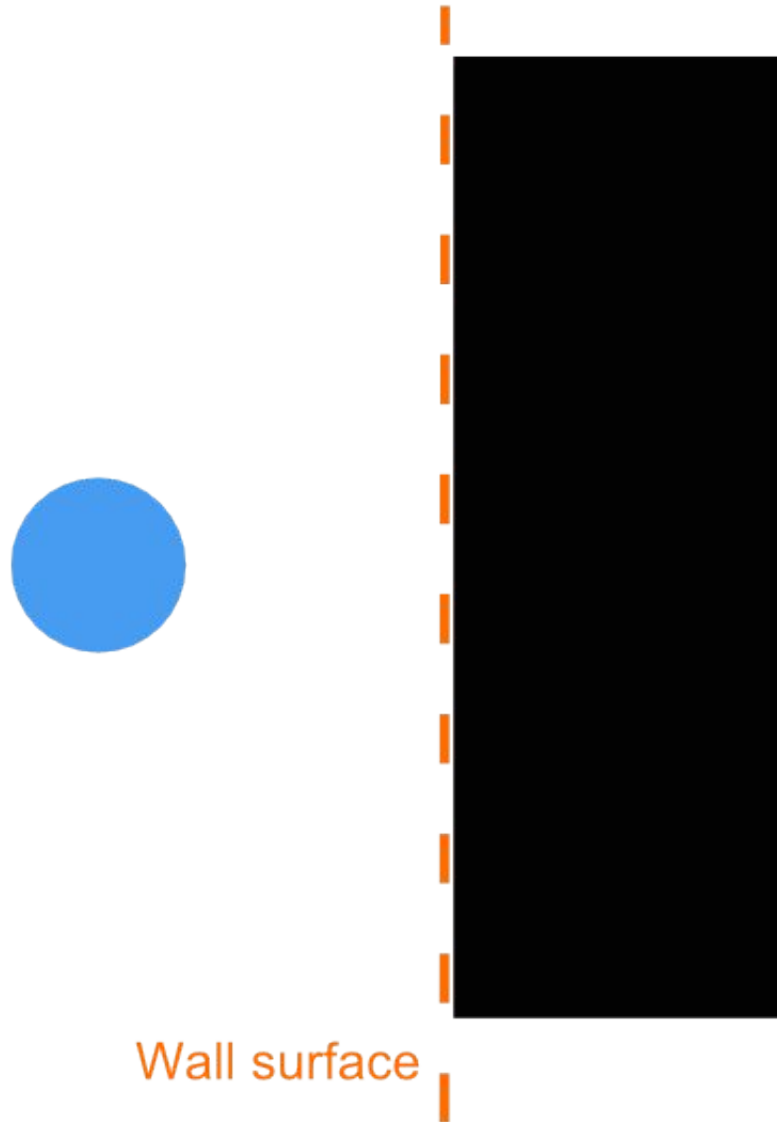
# Adaptive components



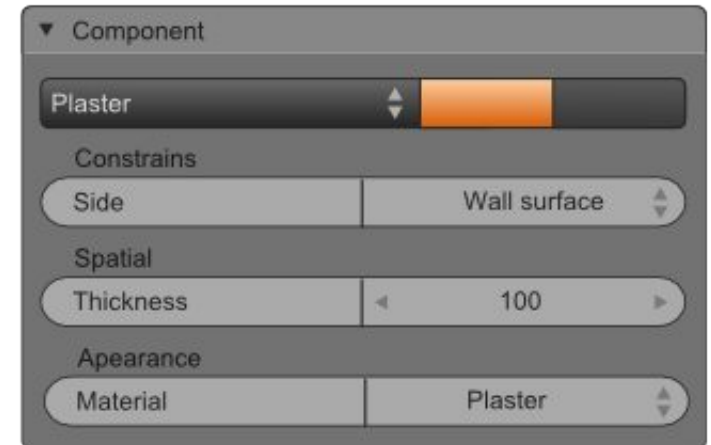
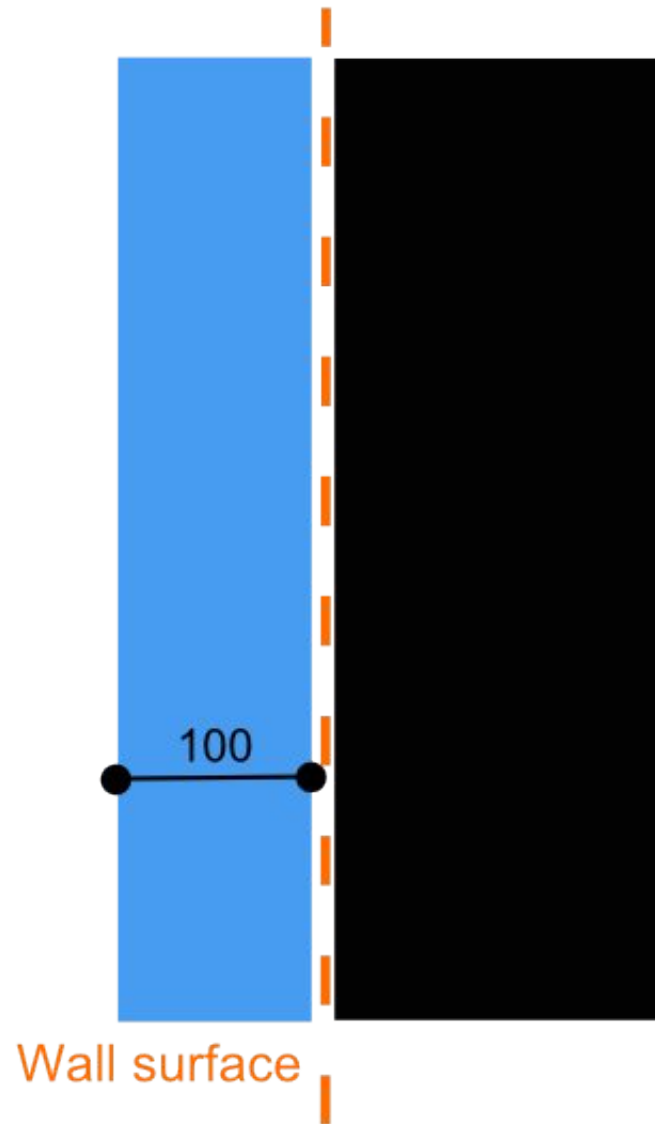
This tool can be also used to add some materials onto all walls, for instance, plaster or paint.



# Adaptive components



# Adaptive components

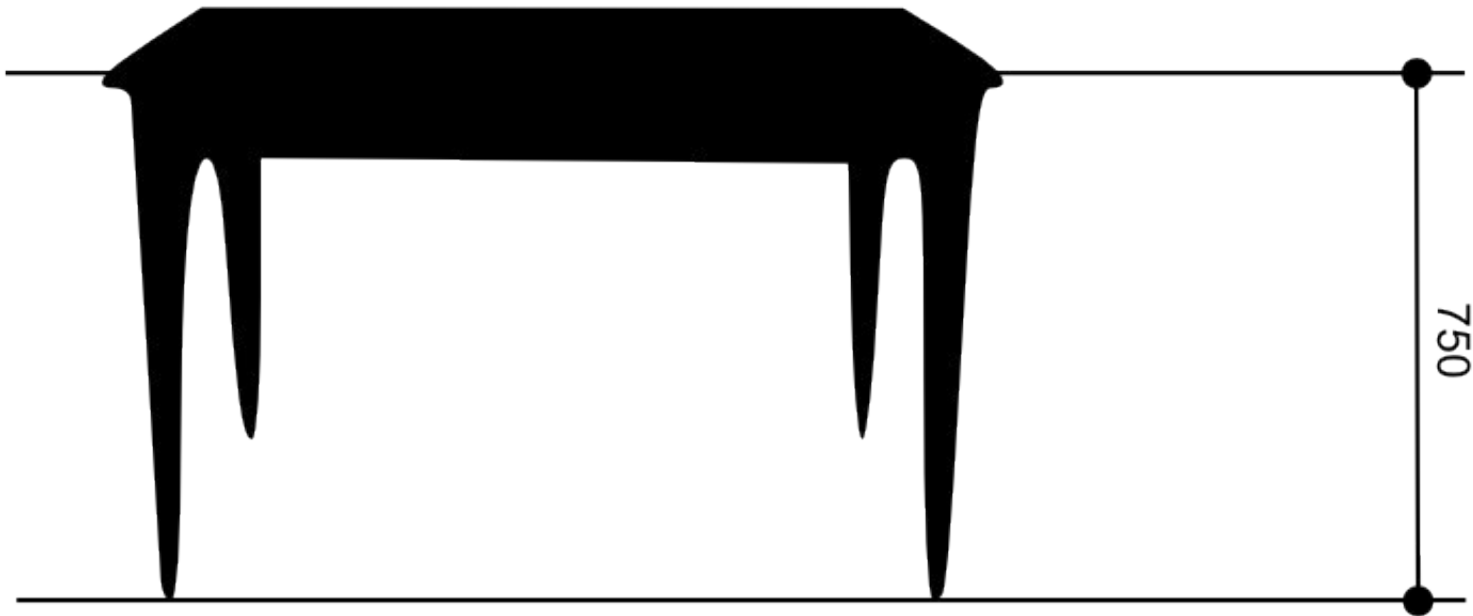


Example of applying some plaster to the wall.

# Refplane and adaptive components benefits to mainstream version of Blender

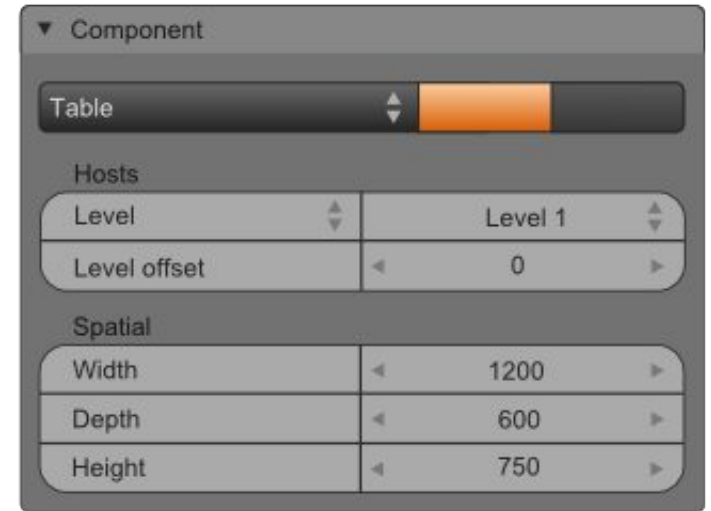
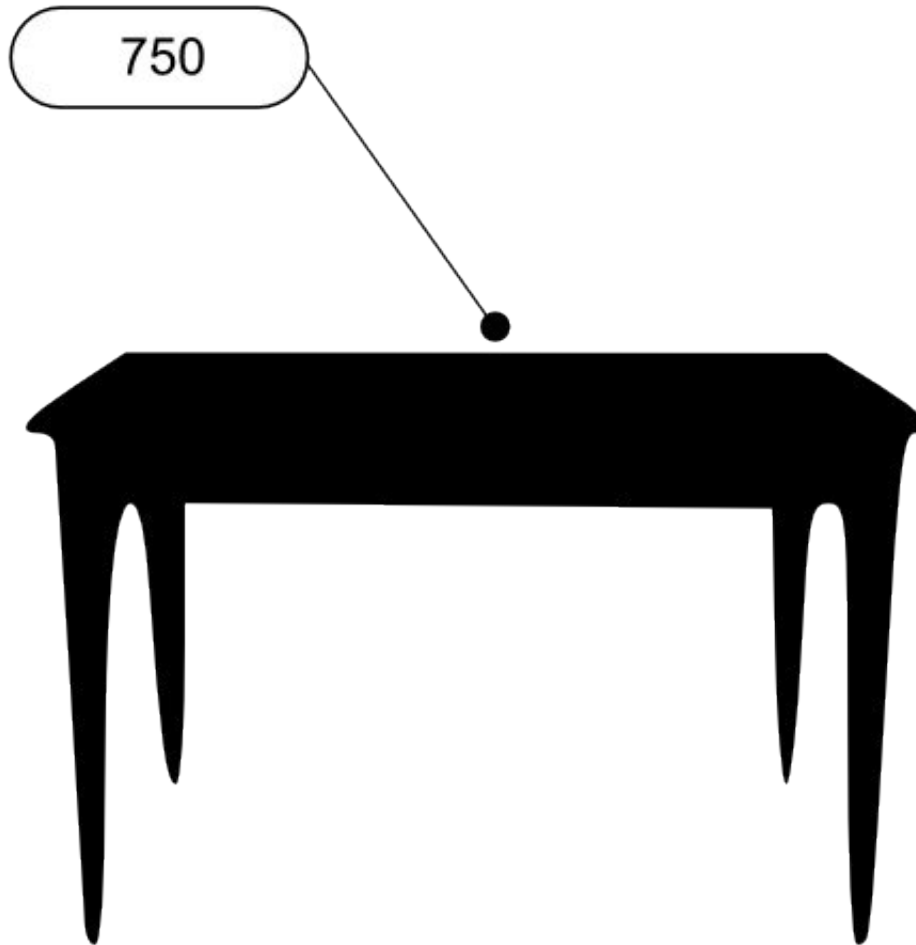
- Optimized workflow
- Increased flexibility
- Automatic object resizes, placements, etc.
- More generalized, less repeating objects
- Adaptive components saves time and hassle

# Dimension tool



To make drawings display actual information, it is vital to have the ability to draw dimensions for some important distances, angles or altitudes

# Tag



This 2D component displays particular parameter value of another component

# Tag

Table



▼ Component

Table

Hosts

Level	Level 1
Level offset	0

Spatial

Width	1200
Depth	600
Height	750

# Tag

1200 x 600



▼ Component

Table

Hosts

Level	Level 1
Level offset	0

Spatial

Width	1200
Depth	600
Height	750

750



Changing value in Tag also change tagged component parameter value



750



Component

Table

Hosts

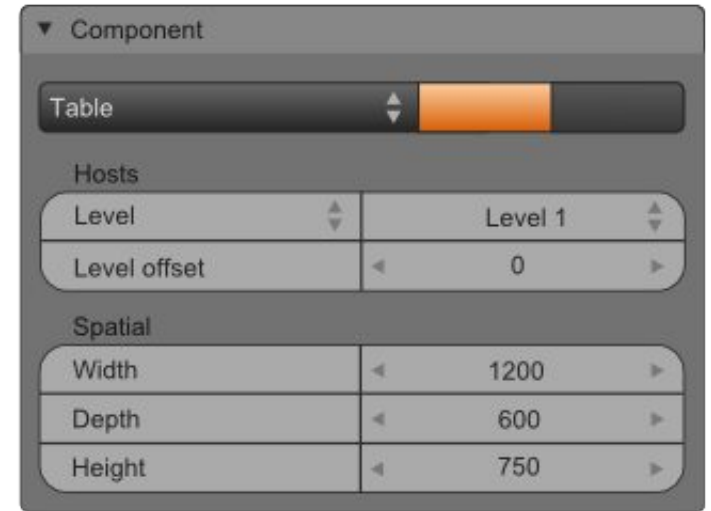
Level	Level 1
Level offset	0

Spatial

Width	1200
Depth	600
Height	750

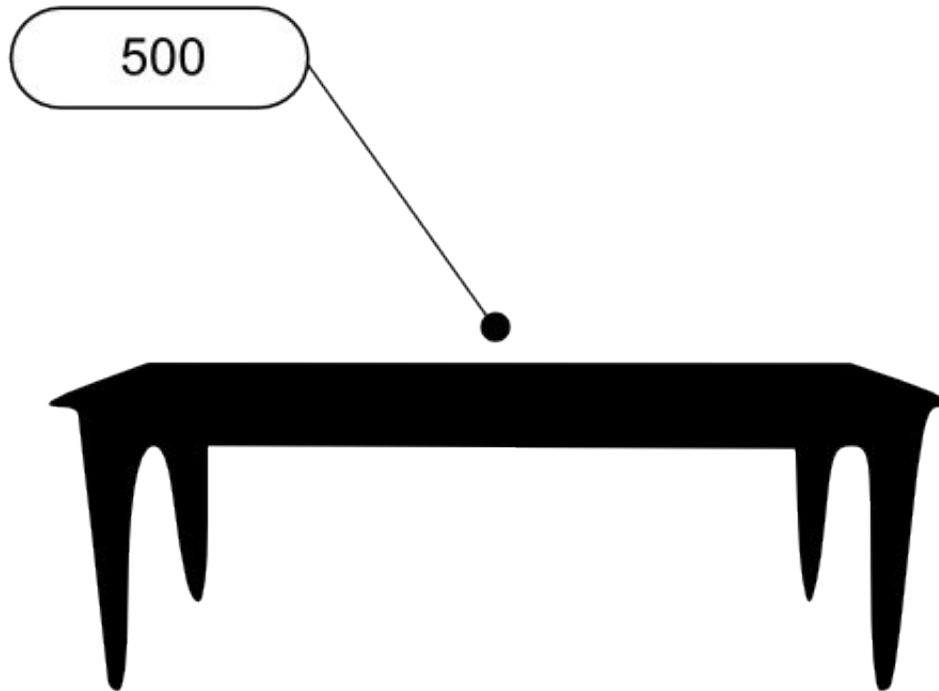
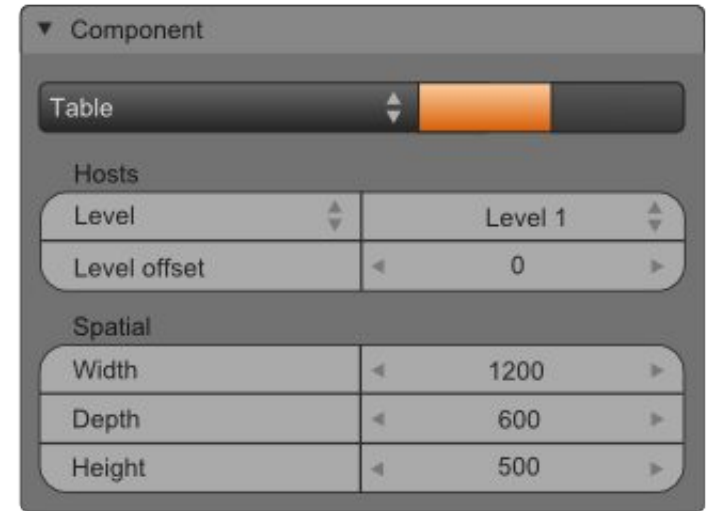
Changing value in Tag also change tagged component parameter value

500



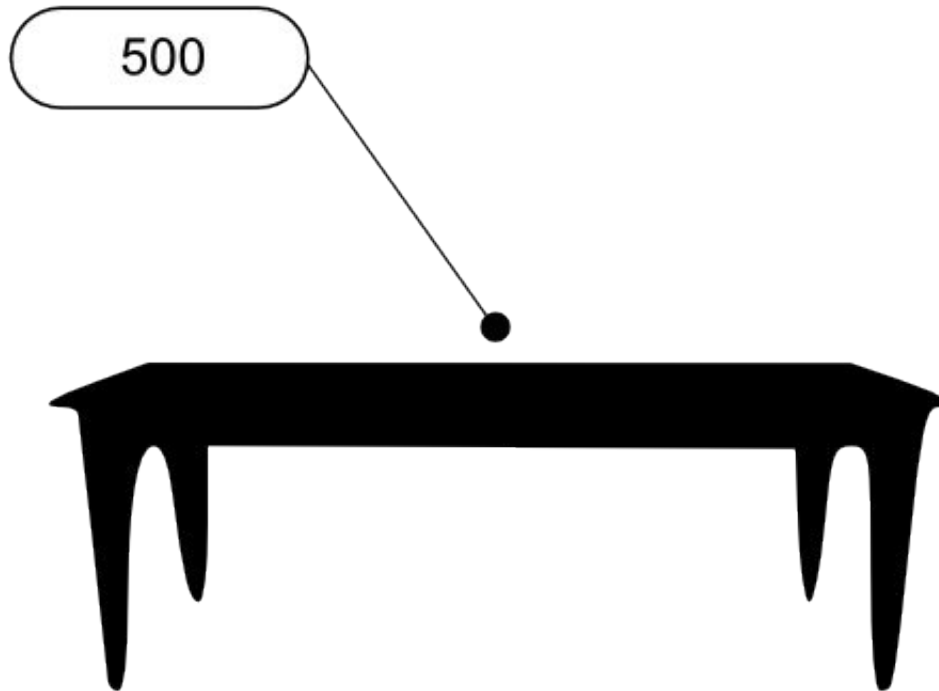
Changing value in Tag also change tagged component parameter value

Name	Dimensions	Count
Chair	450x450x800	2
Table	1200x600x500	3



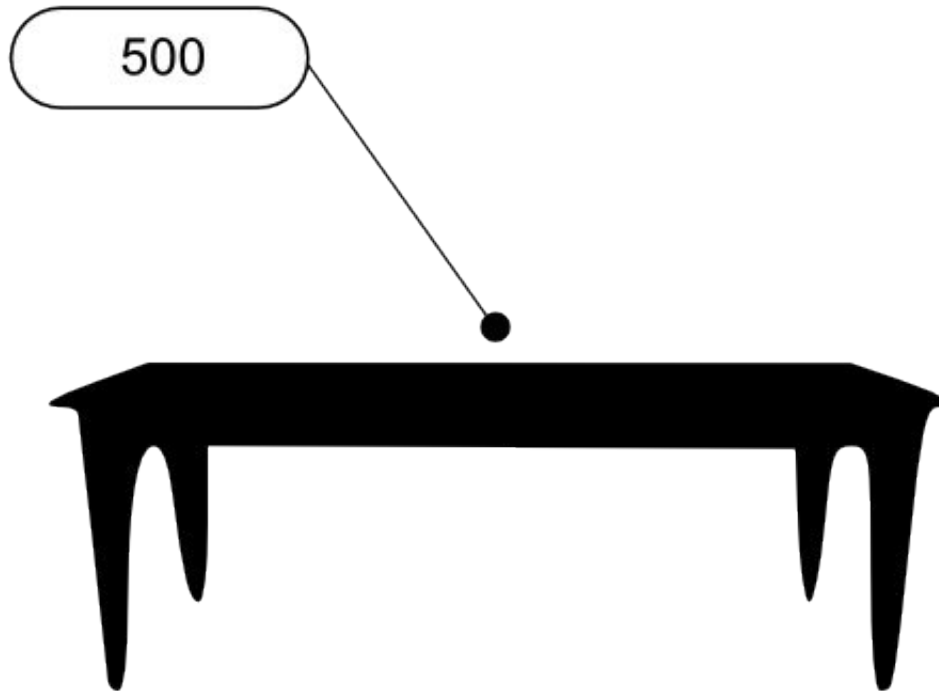
Changing value in Tag also change tagged component parameter value

Name	Dimensions	Count
Chair	450x450x800	2
Table	1200x600x500	3



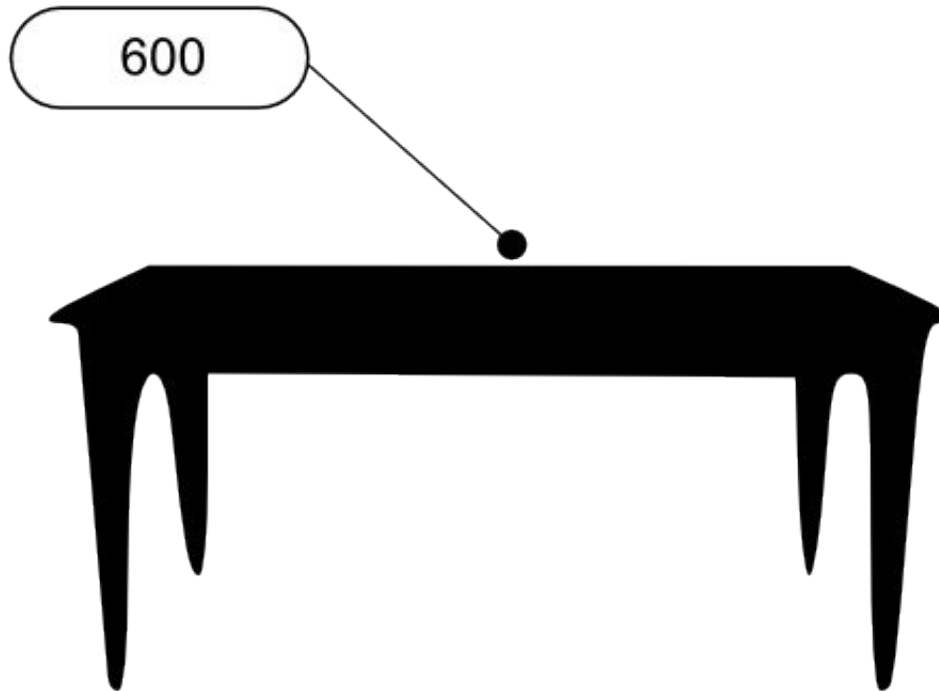
Changing value in Tag also change tagged component parameter value

Name	Dimensions	Count
Chair	450x450x800	2
Table	1200x600x600	3



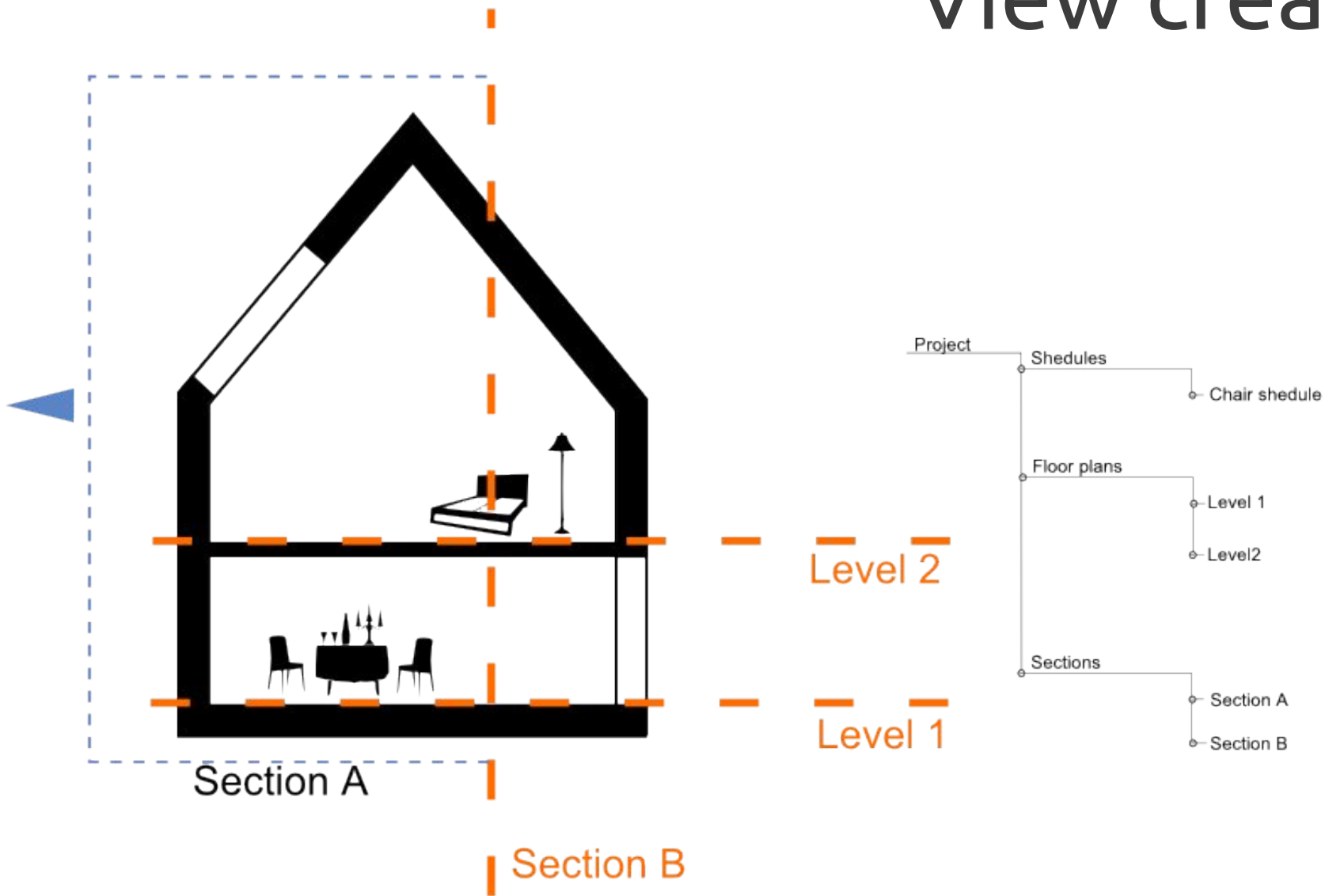
Changing value in Tag also change tagged component parameter value

Name	Dimensions	Count
Chair	450x450x800	2
Table	1200x600x600	3



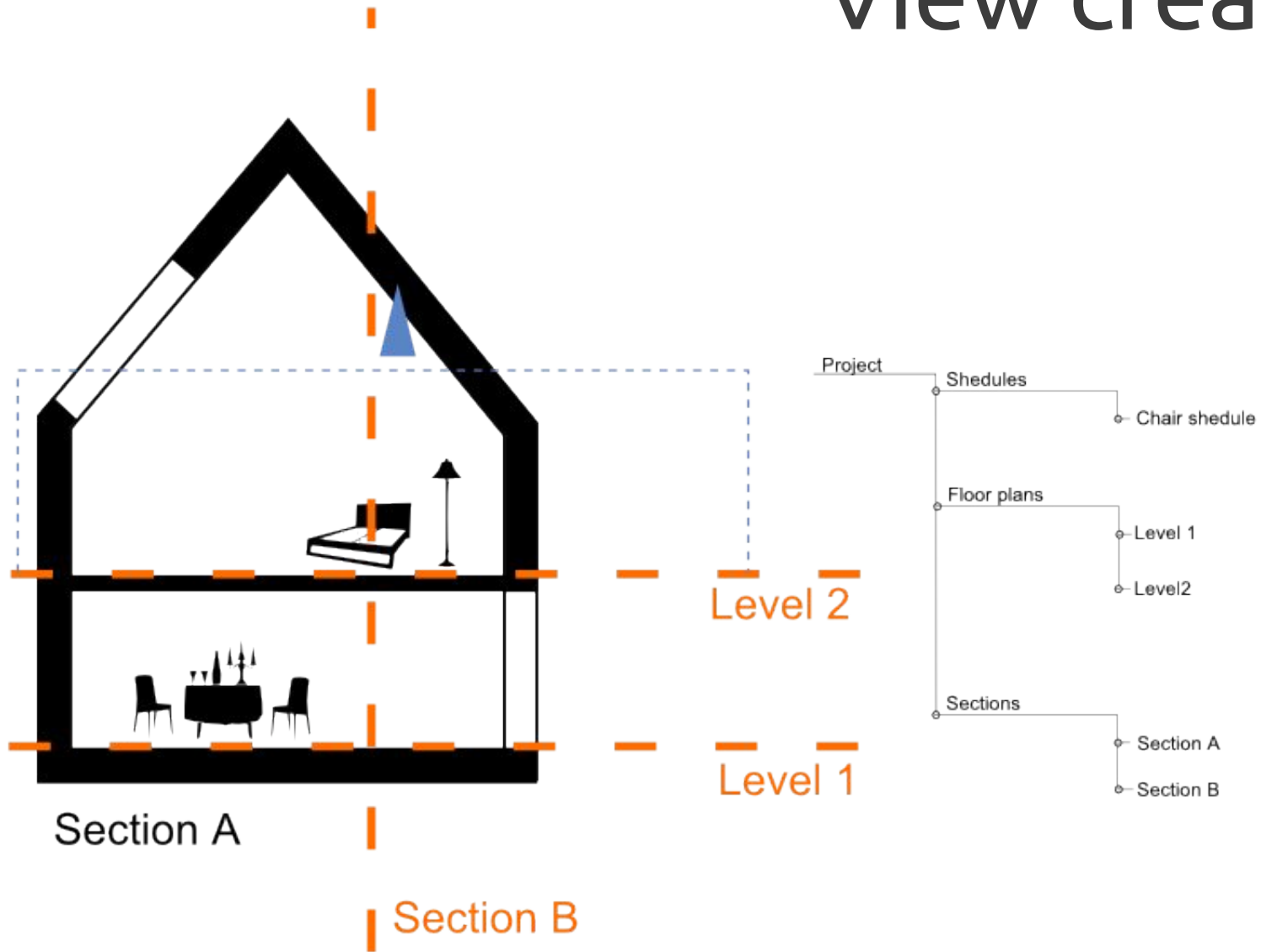
Changing value in Tag also change tagged component parameter value

# View creation



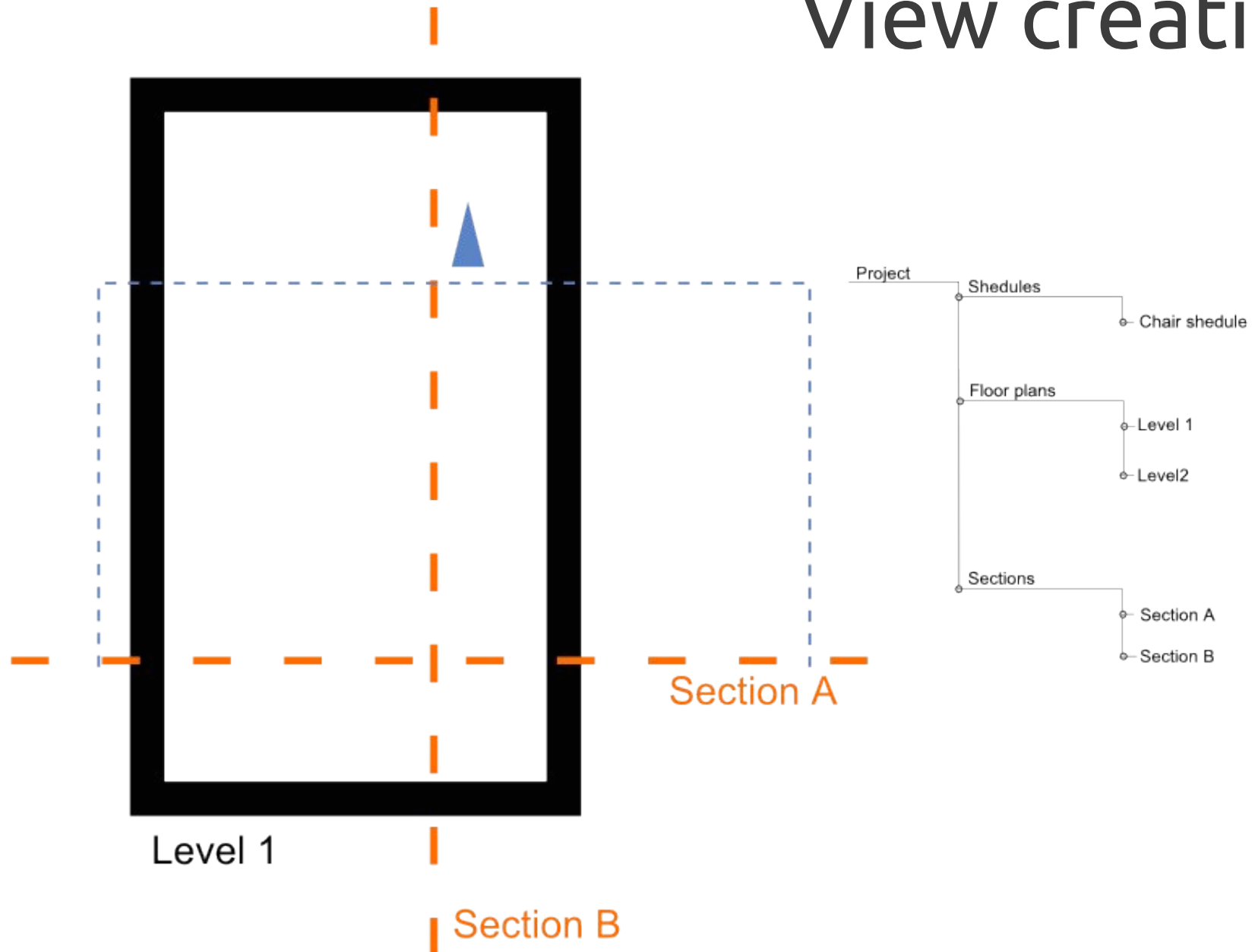
Even if we start nanolathing directly from the project model one day, still, drawings are at least interesting to print out.

# View creation

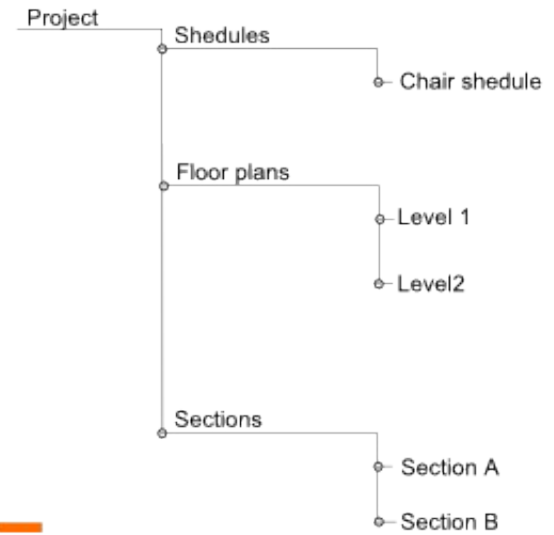
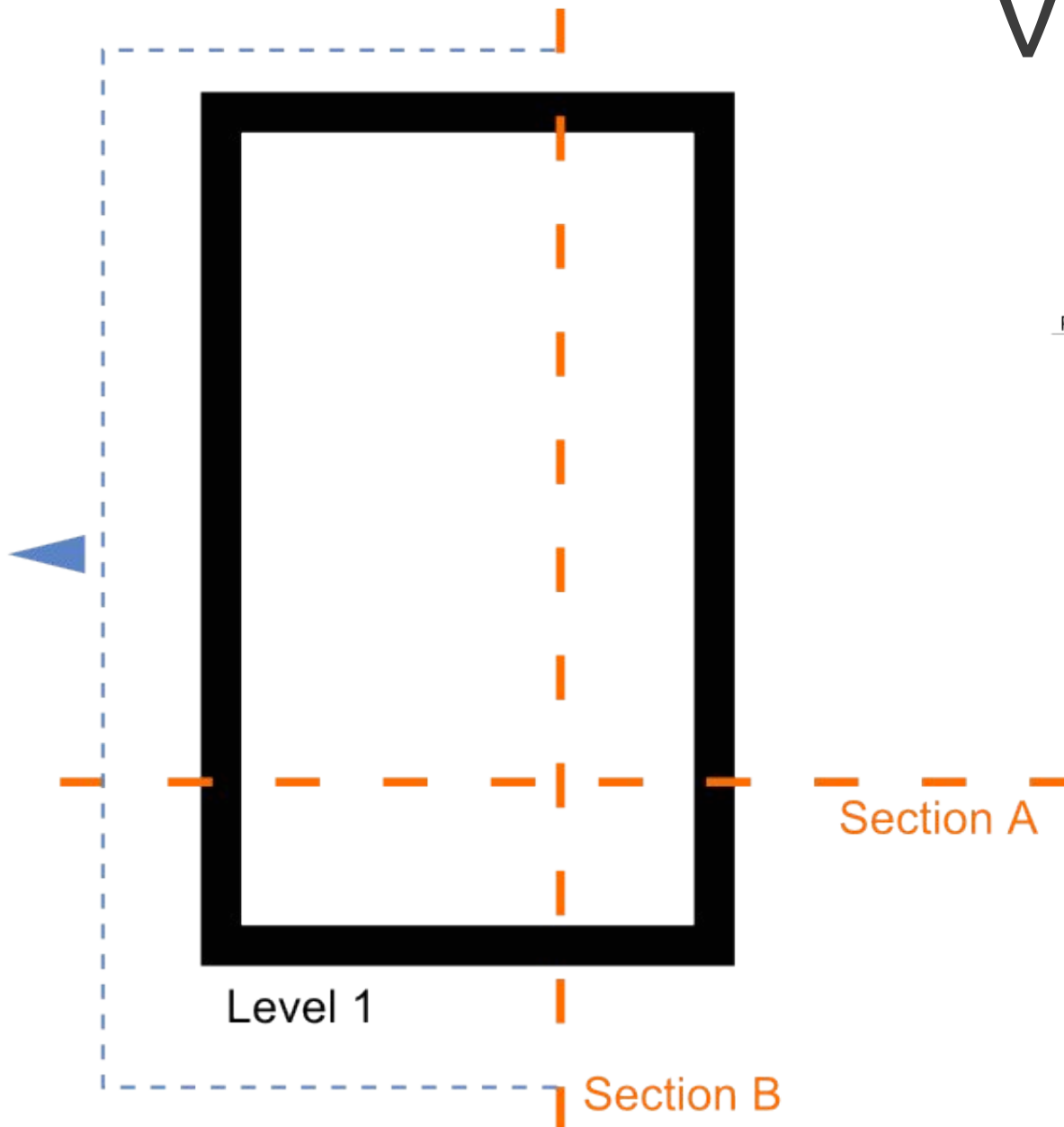




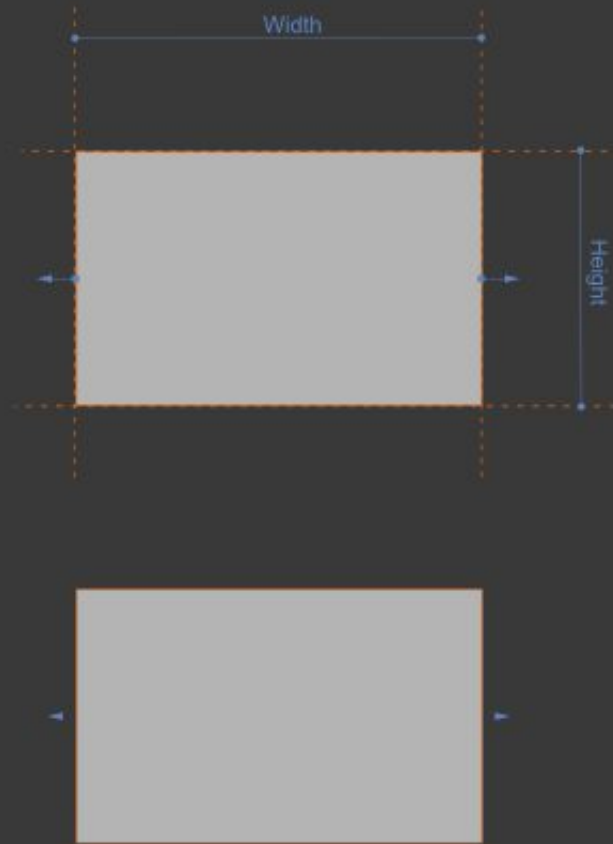
# View creation



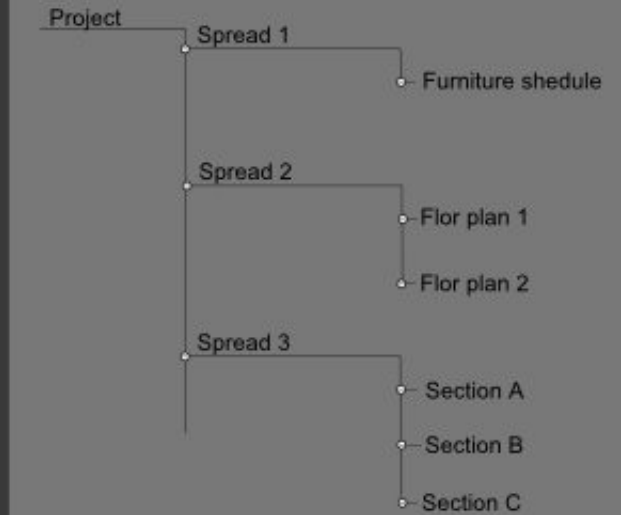
# View creation



# Model space



...When You click on a page or spread in project tree, you get special interface to organize the 2D drawings



## ▼ Component



Chair Wood

### Hosts

Level

Level 1

Level offset

0

### Spatial

Width

200

Depth

300

Height

200

### Apearance

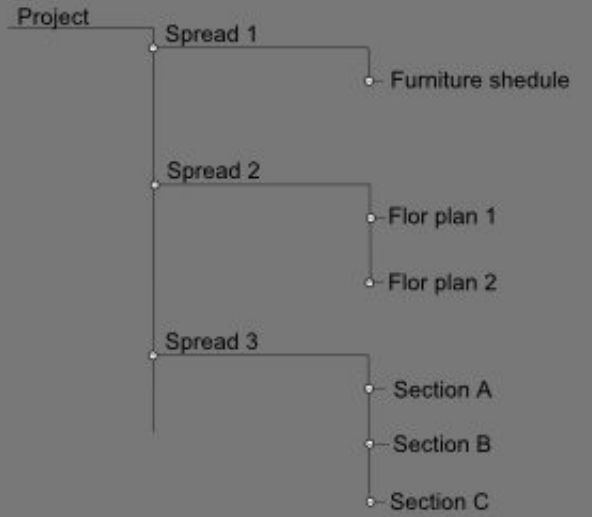
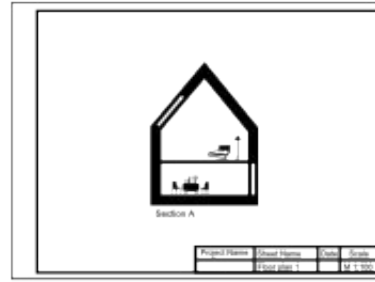
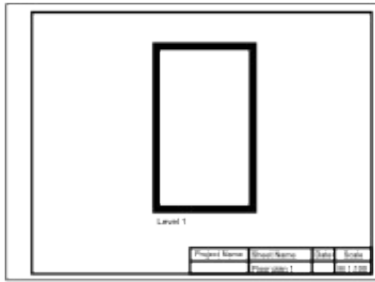
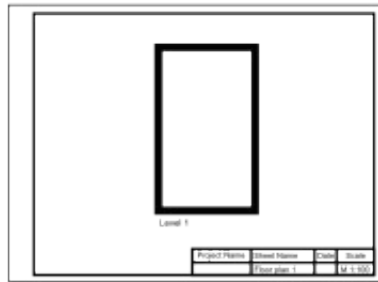
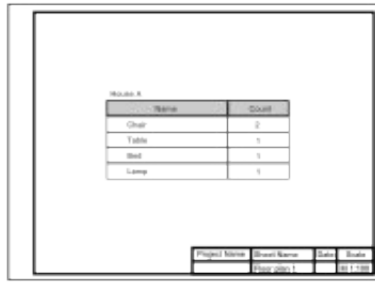
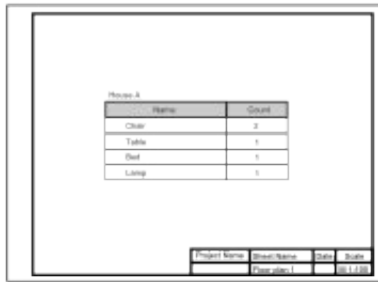
Seat material

Wood

Leg material

Chrome

# Paper space



## Component

Titleblock

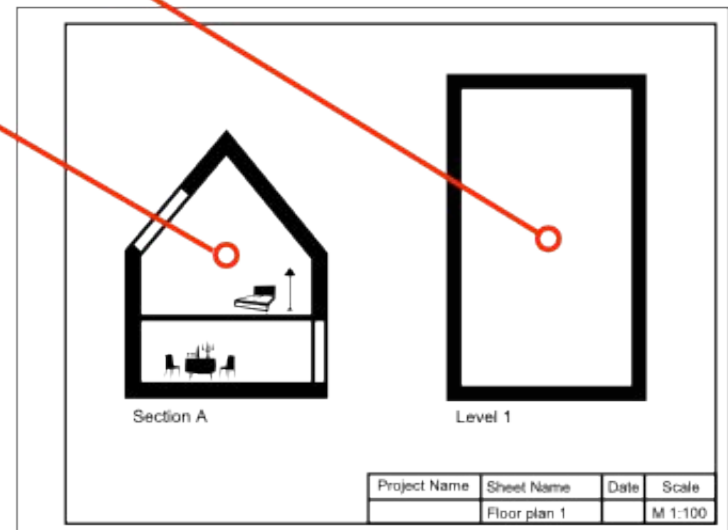
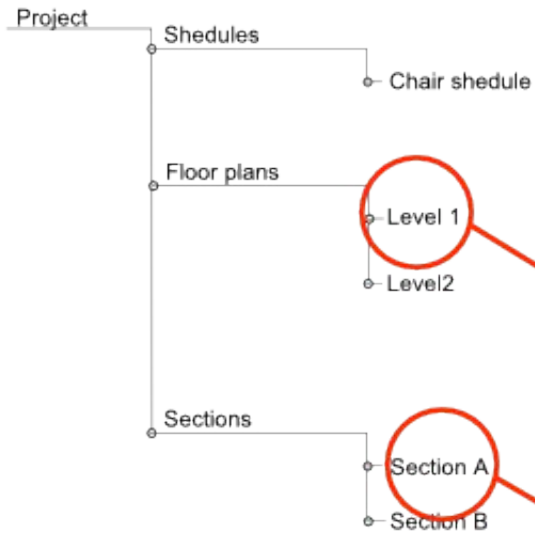
Spatial

Width 200

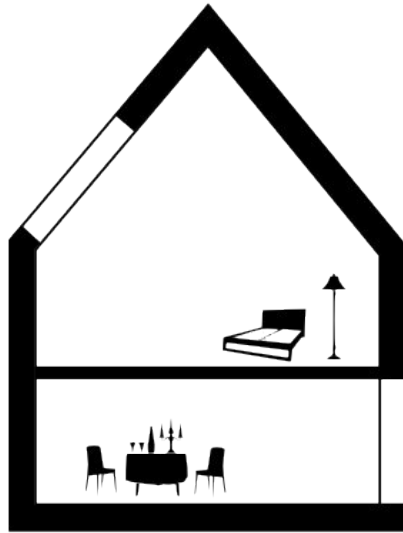
Depth 300

Height 200

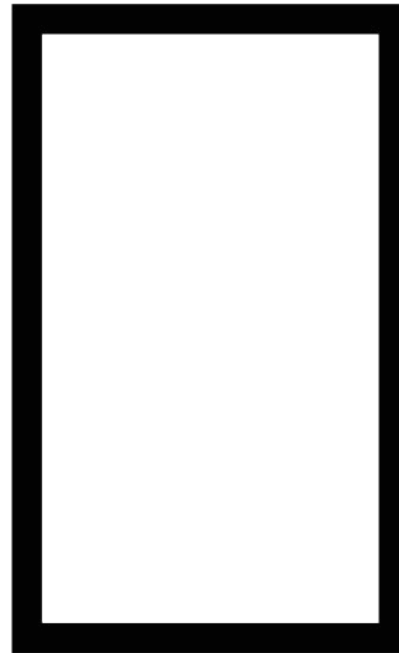
# Views added to sheets



# Views added to sheets



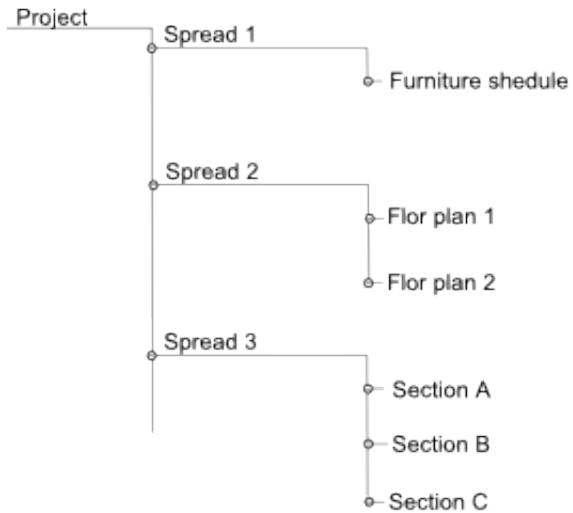
Section A



Level 1

Project Name	Sheet Name	Date	Scale
	Floor plan 1		M 1:100

# Exporting spreads to PDF



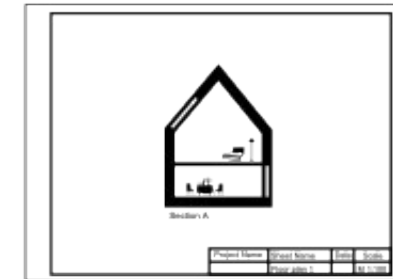
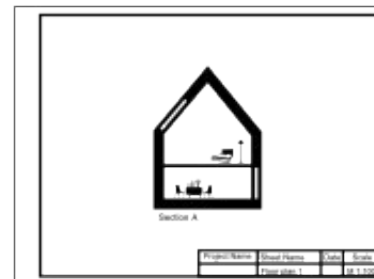
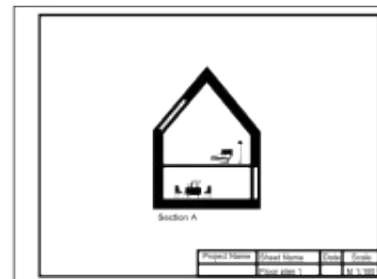
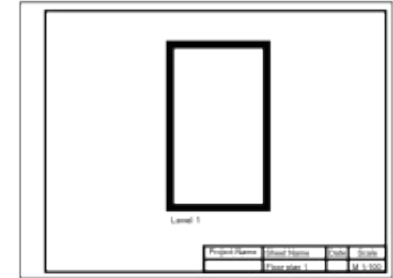
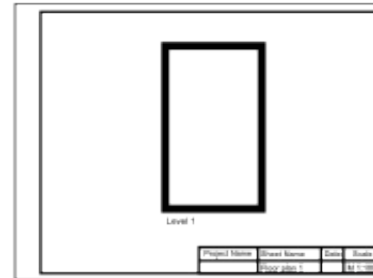
Thumbnail of the Furniture shedule spreadsheet. It features a table with the following data:

Name	Count
Chair	2
Table	1
Bed	1
Lamp	1

At the bottom right, there is a metadata table:

Project Name	Sheet Name	Date	Scale
Project 1	Furniture 1		M 1:100

Thumbnail of the Furniture shedule spreadsheet, identical to the one on the left.



# Exporting spreads to PDF



House A			
Name	Count		
Chair	2		
Table	1		
Bed	1		
Lamp	1		

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500

House A			
Name	Count		
Chair	2		
Table	1		
Bed	1		
Lamp	1		

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500



Level 1			

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500

Level 1			

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500



Section A			

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500

Section A			

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500

Section A			

Project Name	Sheet Name	Units	Scale
House plan 1	Floor plan 1		M 1:500



# Layout engine benefits to mainstream version of Blender

- Suitable as a documentation tool.

# Online parametric component library

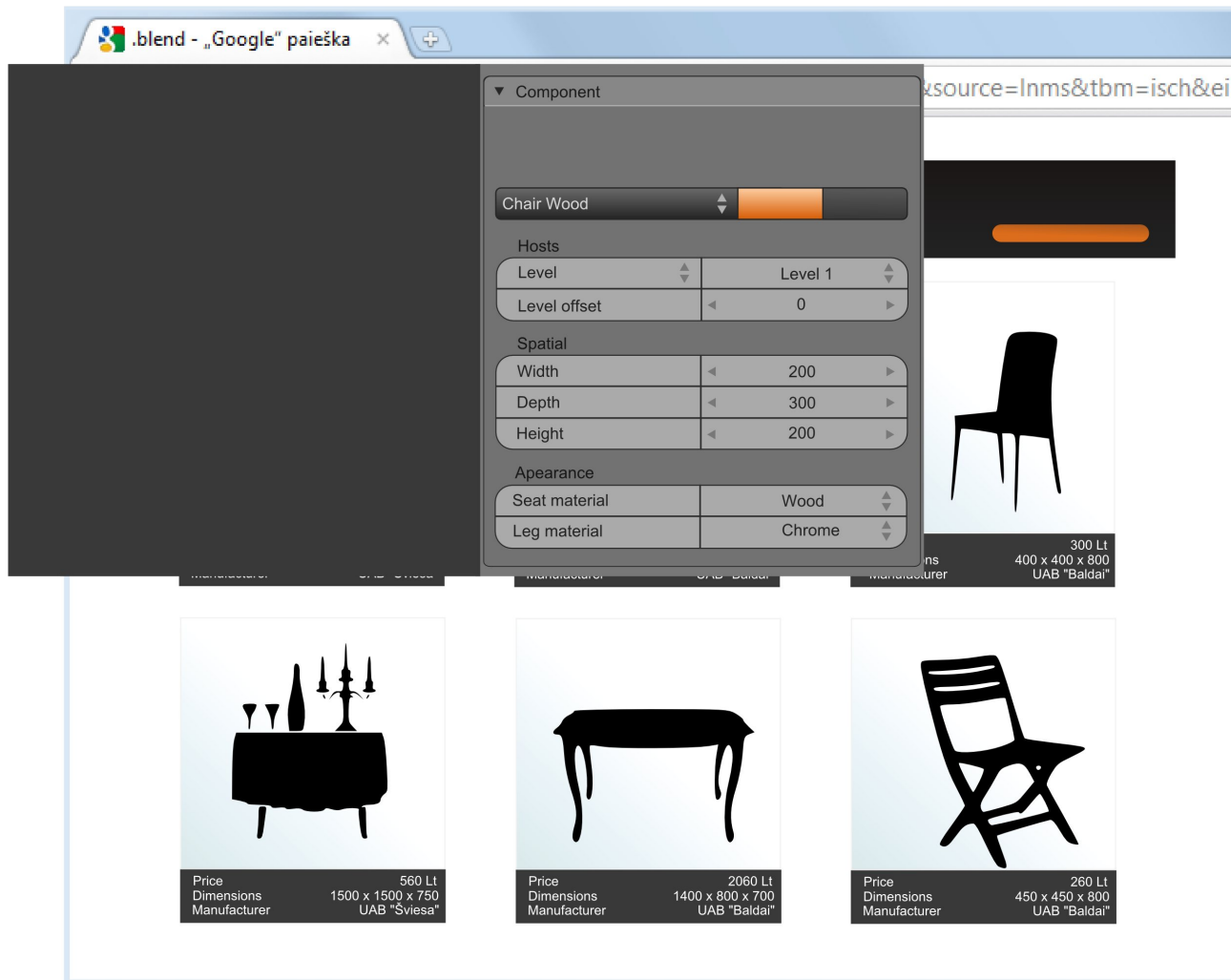
The screenshot shows a web browser window with the URL [www.google.lt/search?q=.blend&hl=lt&qscrl=1&prmd=ivns&source=lnms&tbm=isch&ei=](http://www.google.lt/search?q=.blend&hl=lt&qscrl=1&prmd=ivns&source=lnms&tbm=isch&ei=). The page features a navigation bar with the ".blend" logo, a search bar, and filters for "category", "manufacturer", and "submit blend". Below the navigation bar is a grid of six furniture items, each with a silhouette image and a data box containing price, dimensions, and manufacturer information.

Item	Price (Lt)	Dimensions	Manufacturer
Desk Lamp	160	200 x 300 x 600	UAB "Sviesa"
Table	500	1200 x 700 x 720	UAB "Balda"
Chair	300	400 x 400 x 800	UAB "Balda"
Dining Table with Candles	560	1500 x 1500 x 750	UAB "Sviesa"
Table	2060	1400 x 800 x 700	UAB "Balda"
Chair	260	450 x 450 x 800	UAB "Balda"

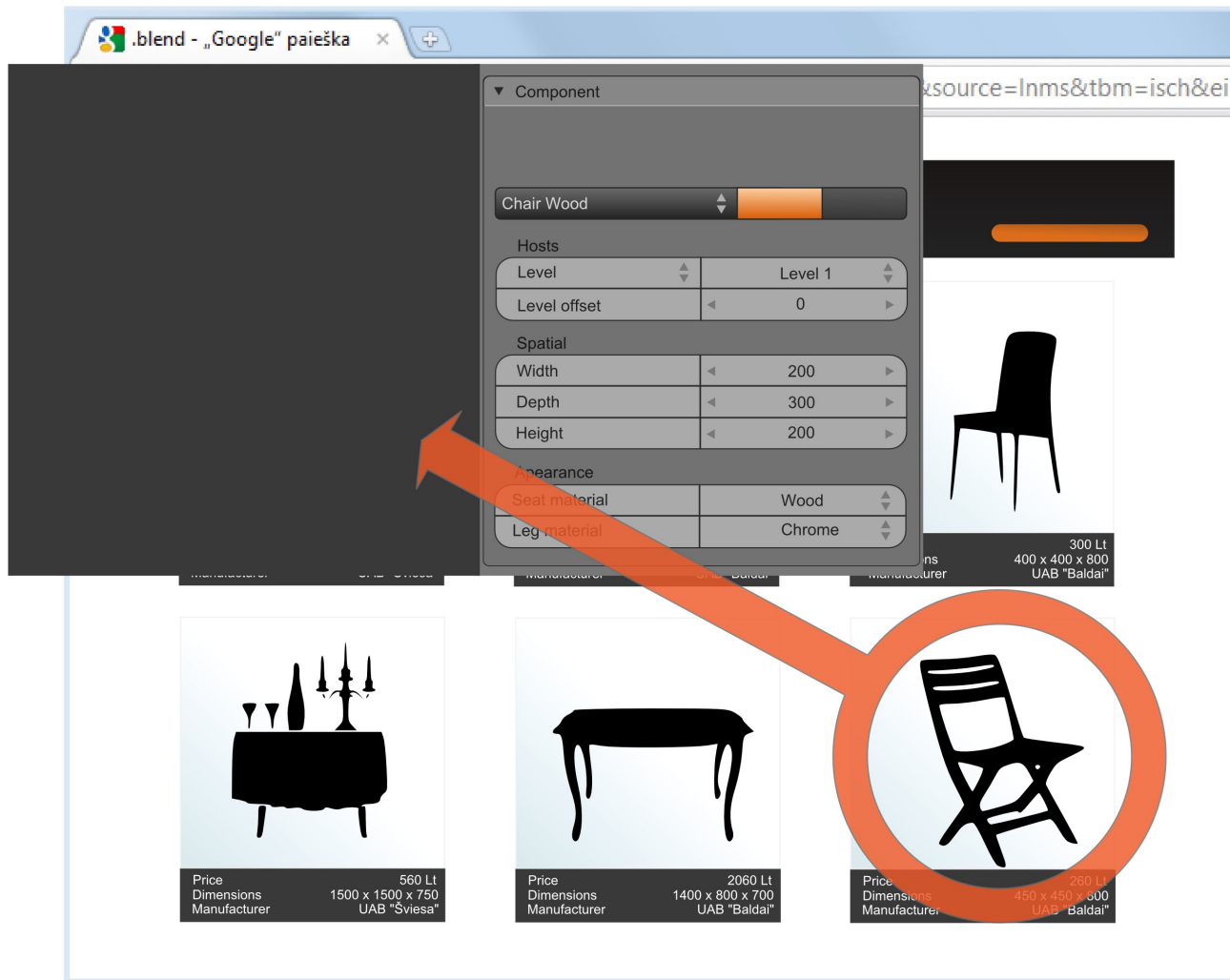
# Important

- Connection between online library and local project
- Version & updates control
- Teamwork & project workflow management
- Use cases:
  - Real-life 3D model gallery/database
  - Intranet for developing complex projects

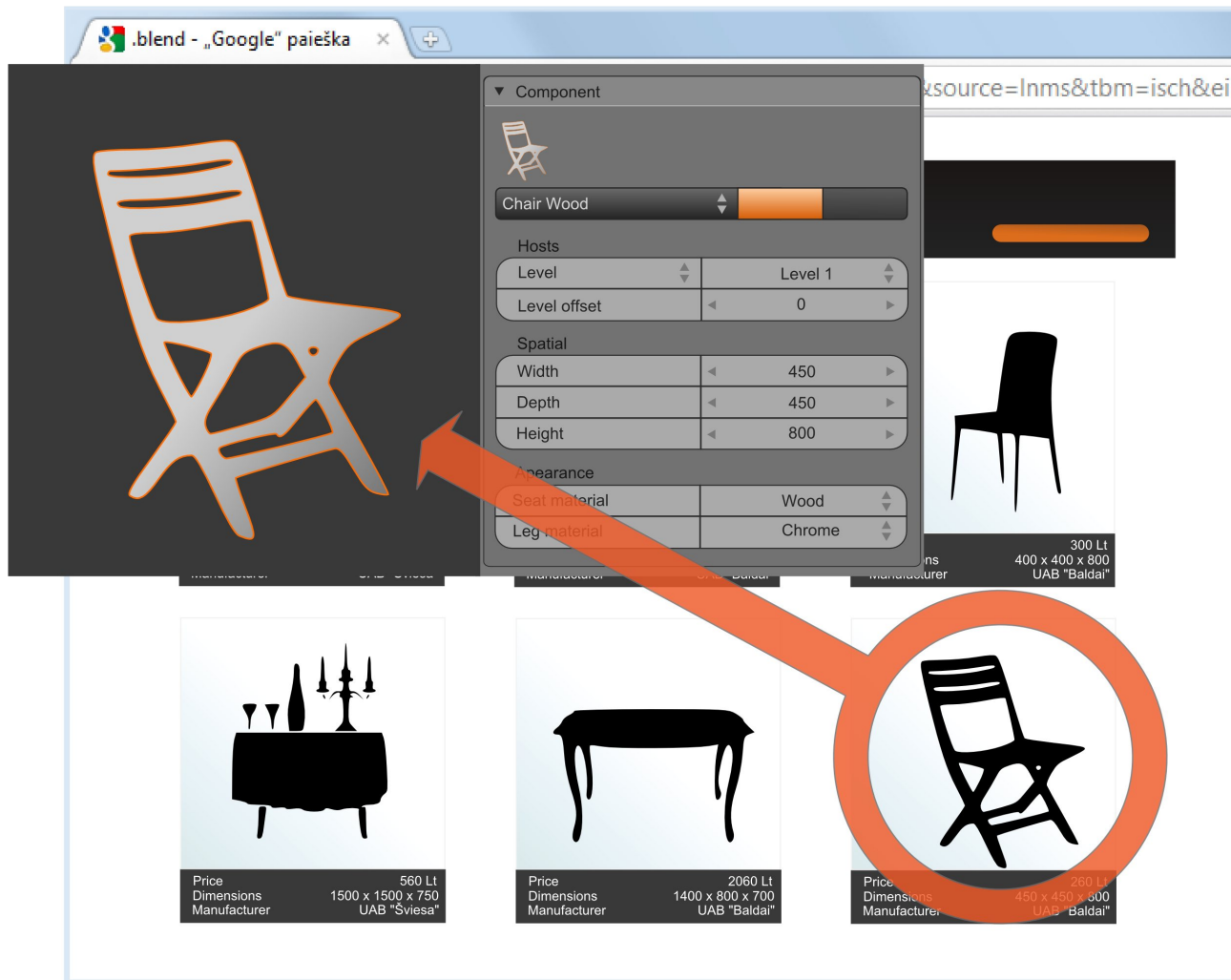
# Online parametric component library



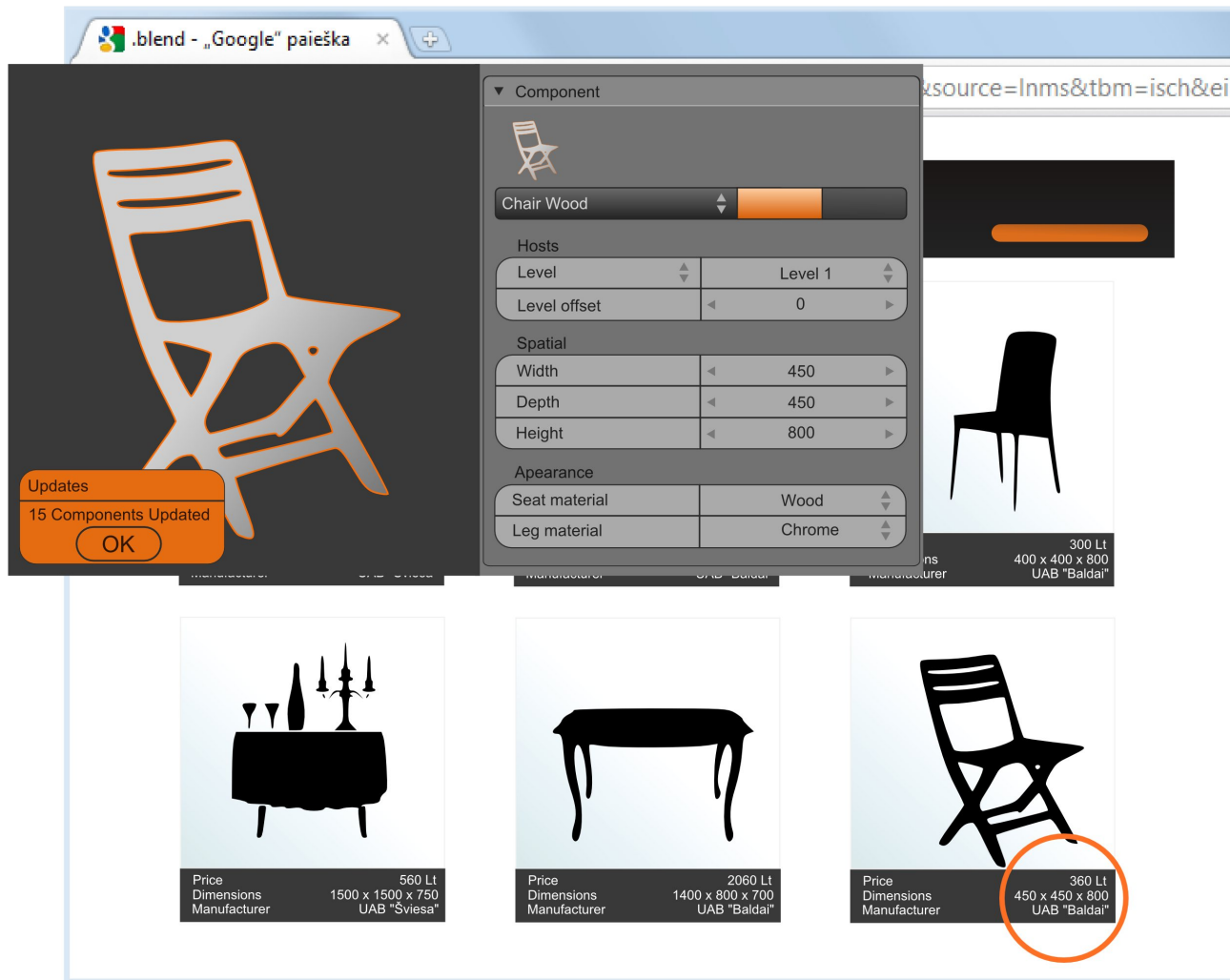
# Online parametric component library



# Online parametric component library



# Online parametric component library



When we open old model and Blender sees updates online, it is possible to update to new version with one click.

# Online parametric component library

- Library acts like Linux package repository
- Parametric components are like packages
- One module can have a loads of requirements, version dependencies etc...



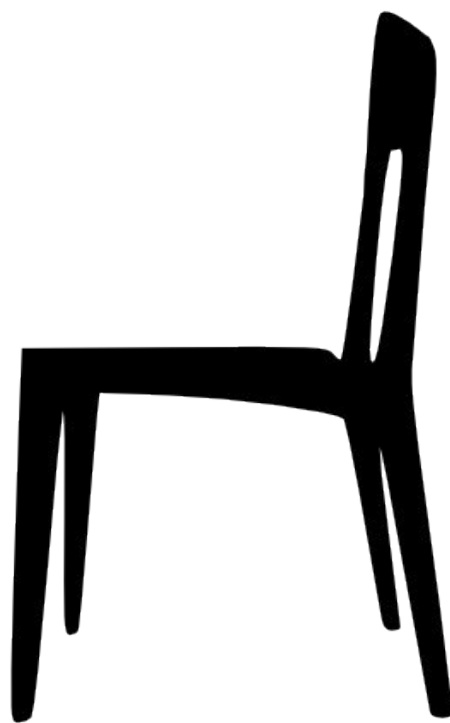
# Problem of consistency

- Like in coding, it is very important, that everyone would create models and name refplanes using determined names.
- Imagine, if someone starts naming floor as „Bottom“, and other - „Level“, tables made to snap level „Floor“ will not work on any of these.
- Strict control is compulsory, as like in accepting code to mainstream!

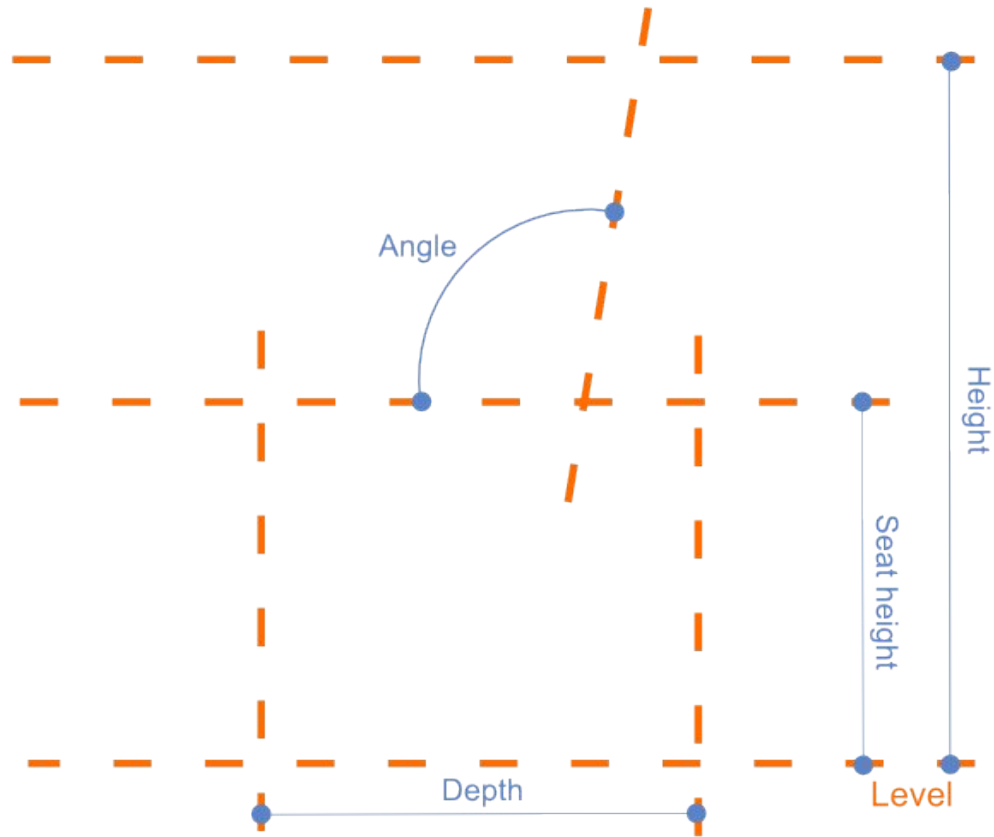
# Tools to manage consistency of components

- Standard templates
- Validators

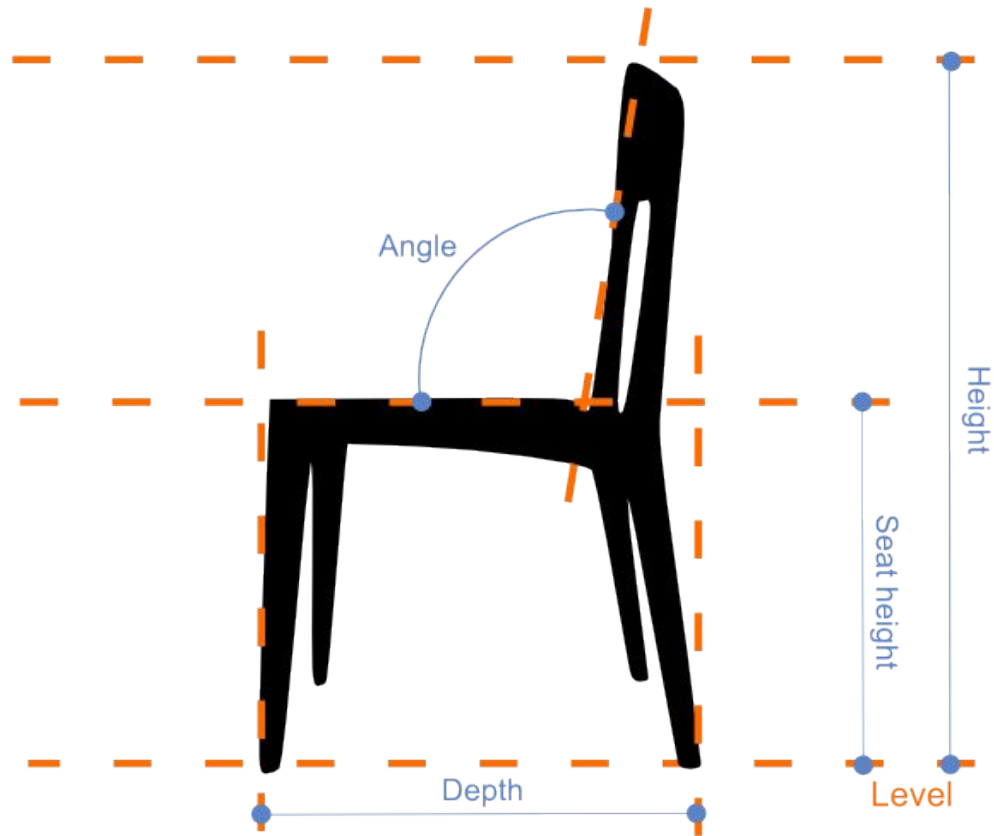
# Chair geometry



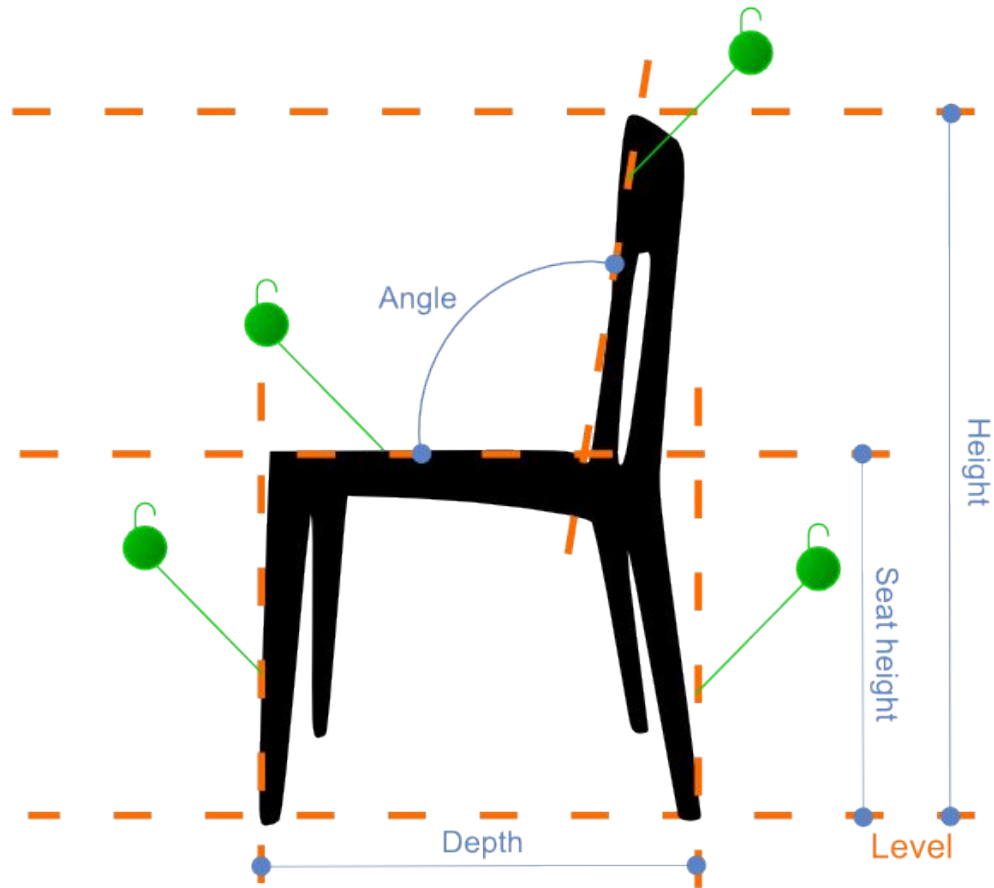
# Chair template



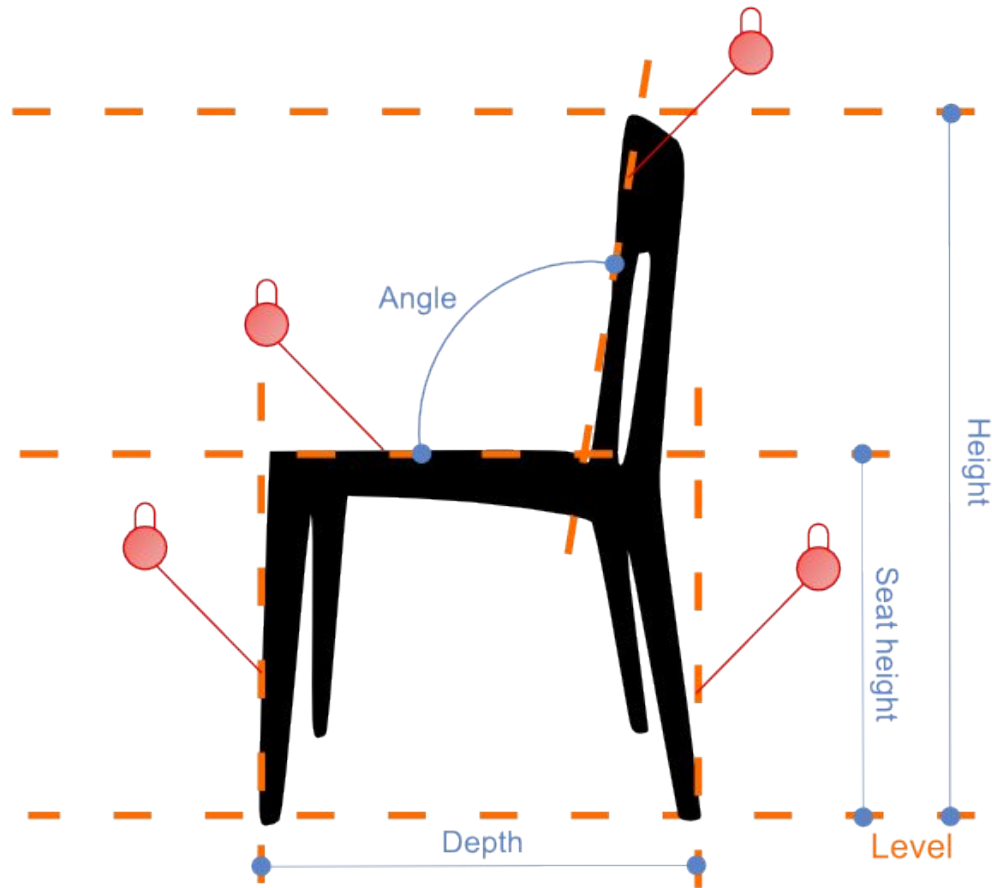
# Template added to geometry



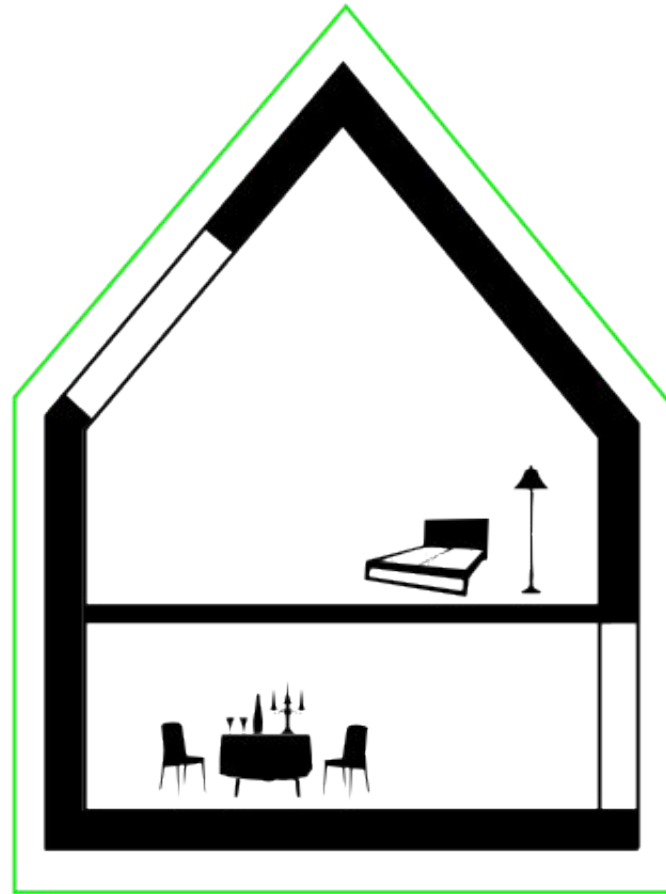
# Geometry locked to refplanes



# Geometry locked to refplanes



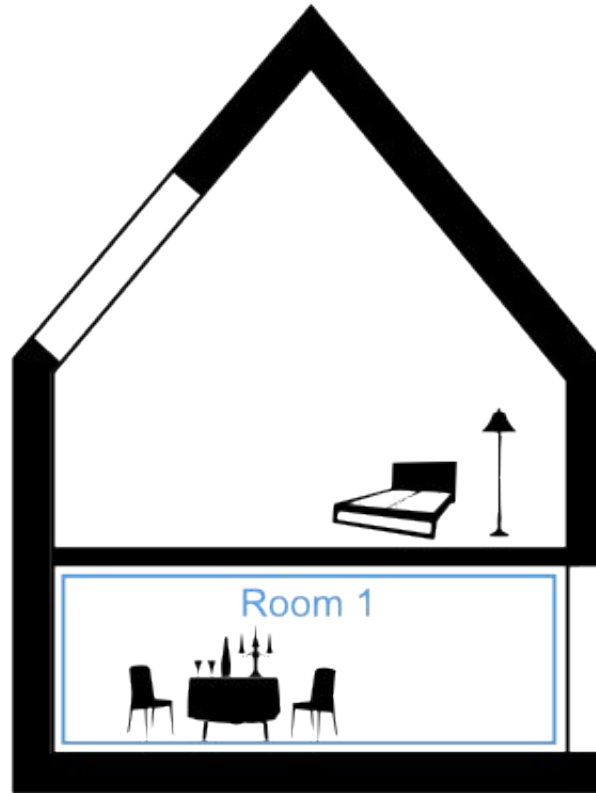
# Scope boxes



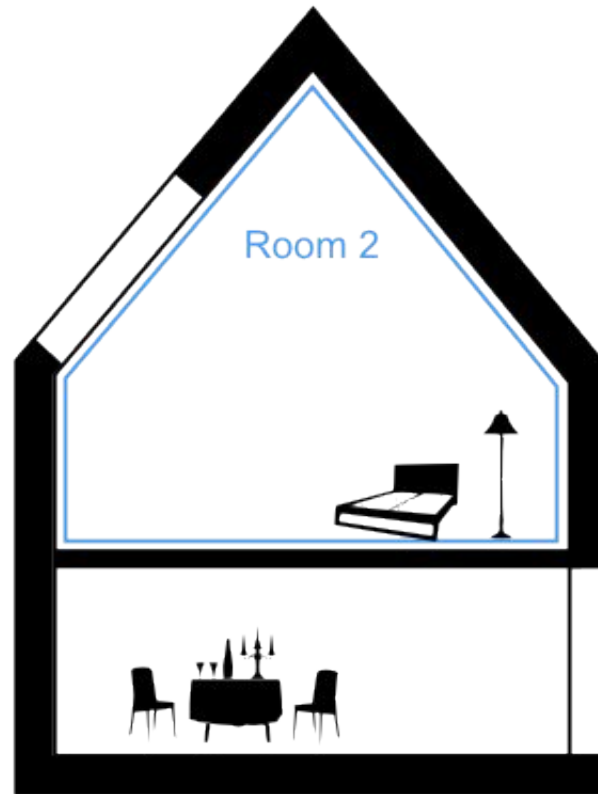
House A



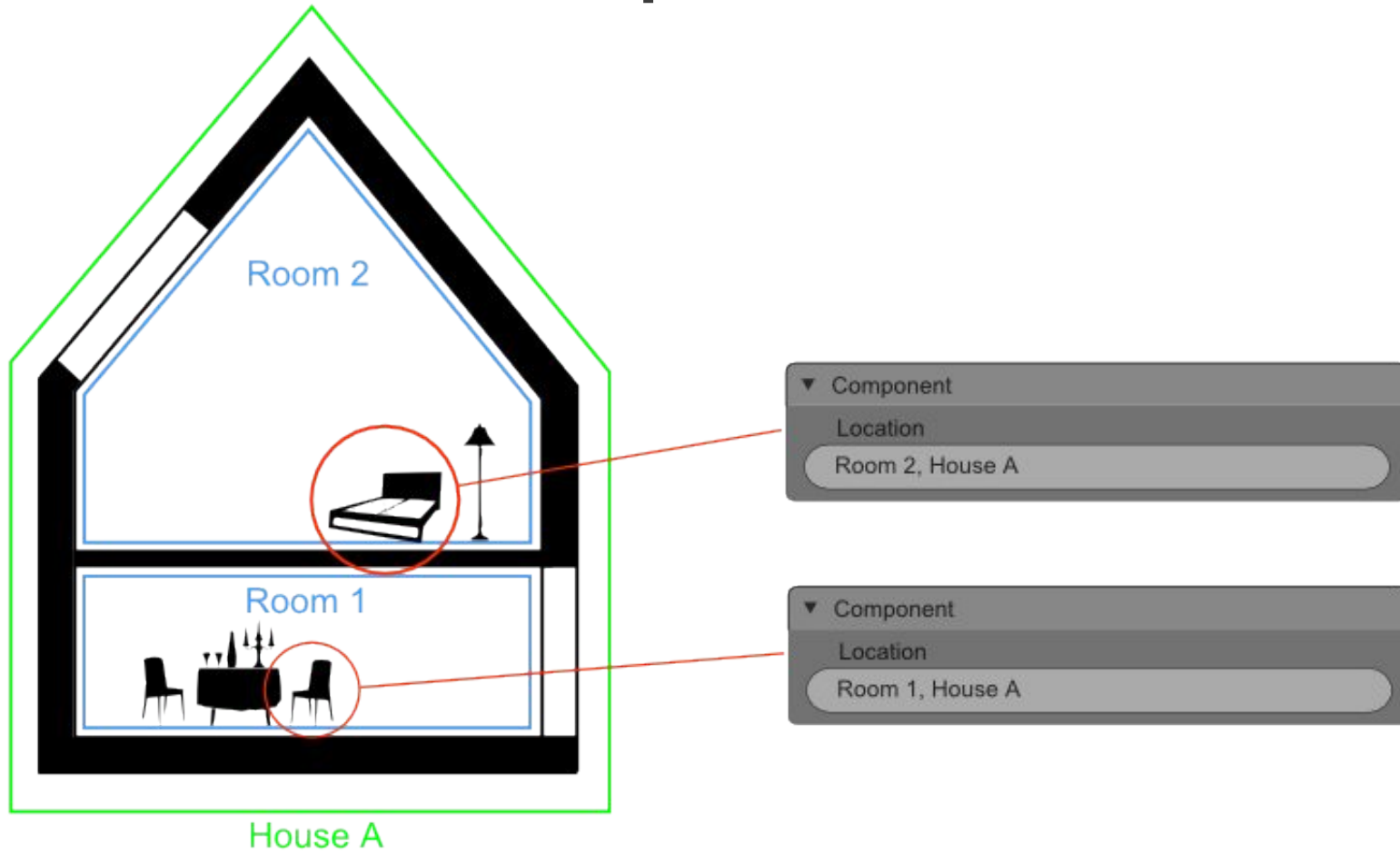
# Scope boxes



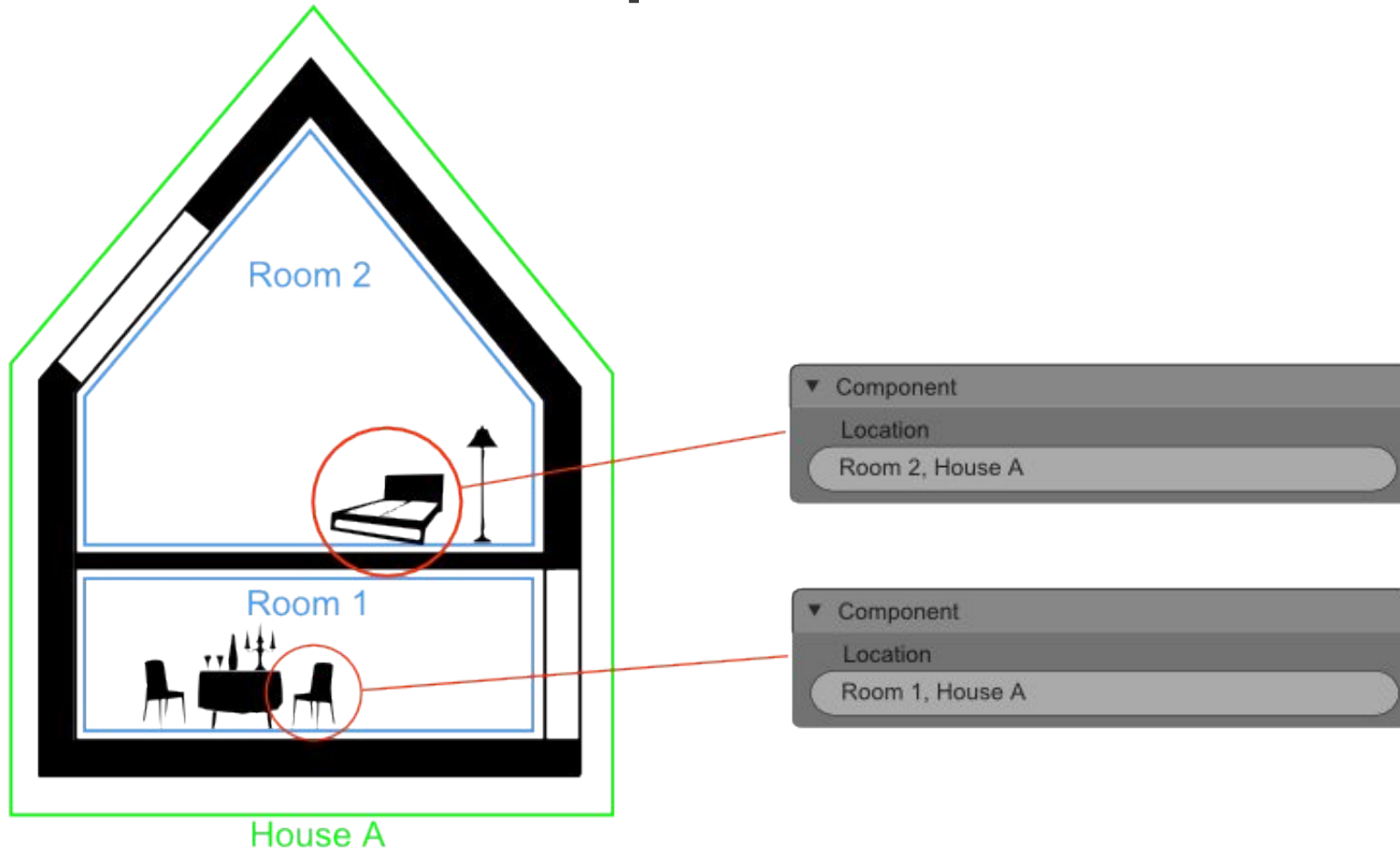
# Scope boxes



# Scope boxes



# Scope boxes



Room 1

Name	Count
Chair	2
Table	1

House A

Name	Count
Chair	2
Table	1
Bed	1
Lamp	1

# Objects sorted by scope box parameter

Room 1

Name	Count
Chair	2
Table	1

House A

Name	Count
Chair	2
Table	1
Bed	1
Lamp	1

# Beating performance bottlenecks

Complex model = slow user experience

Possible solutions:

- Use appropriate hardware :-)
  - x It's expensive
- Distribute expensive calculations over the cloud
  - x Won't scale in some cases; Works only for final rendering
- Strict Level of Detail
  - We don't see screws of the windows anyway :-)

# Automatic Level of Detail

- Has to be automatically managed when working with models;
- Flexibility to customize component priority / rules of LoD is a must when creating PO
- Profit for the mainstream version is obvious! (faster work with complex objects)

# Complex commercial formats

- No documentation
- Closed-source projects poorly support other closed-source formats
- Errors and data fidelity losses are common





# .ifc (Industry Foundation Classes)

- Intended to describe building and construction industry data
- neutral and open specification
- not controlled by a single vendor or group of vendors
- well-known standard in architectural design field
- Still.. Commercial software doesn't export/import this format very well

# Thank You for attention!

Justas Ingelevičius

+370 613 53 947

[Justas@zvejone.lt](mailto:Justas@zvejone.lt)

Justinas Jaronis

+370 601 36 912

[justinas.jaronis@aksprendimai.lt](mailto:justinas.jaronis@aksprendimai.lt)

<http://www.aksprendimai.lt/english/>