

COMPONENT-BASED MODELING OF COMPLETE BUILDINGS

Luc Leblanc Jocelyn Houle Pierre Poulin

Dept. I.R.O. 
Université
de Montréal

Graphics Interface
St. John's, May25-27, 2011

GOALS

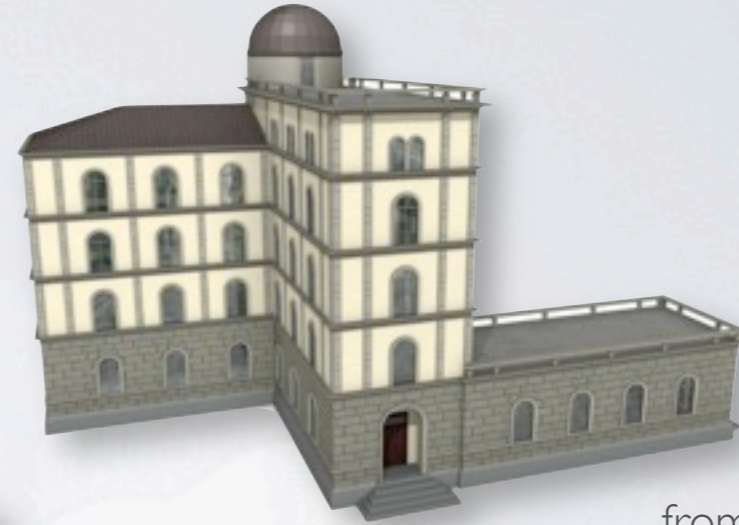
- Procedural generation of buildings
- Exteriors and interiors
- Complete control over the results
- High quality geometry

WHAT IT IS NOT

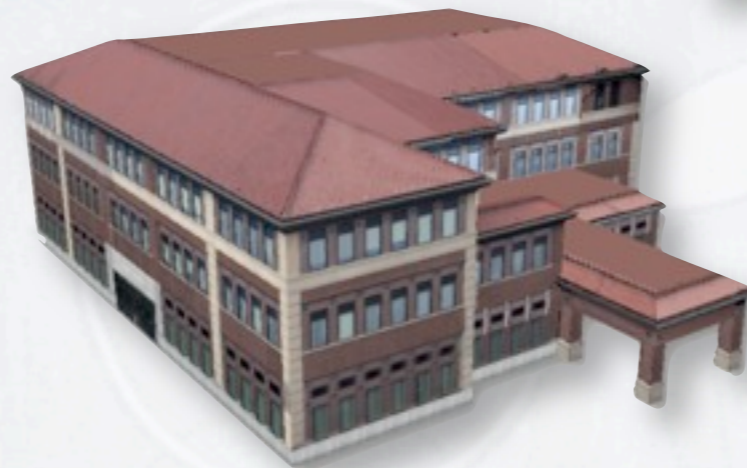
- Not a system for architects
 - although if they can program...
- Not a system to explore designs
 - although if they like looking at some random aspects...
- Not a library of functionalities
 - more like a special language with regular language concepts (variables, loops, randomness, etc.)

PREVIOUS WORK

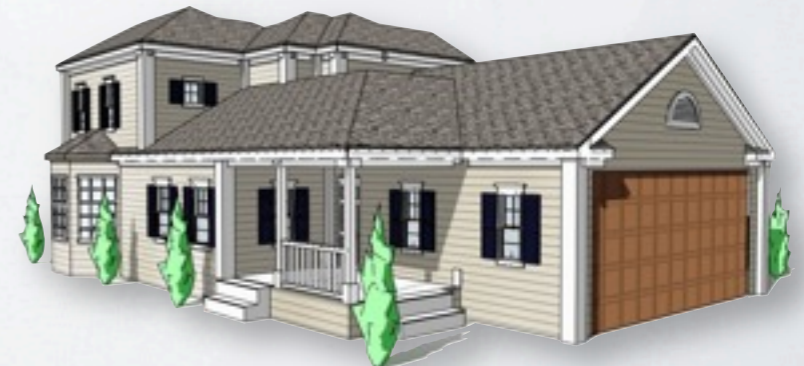
- Mostly façades and building shapes
 - Shape/split grammars
- 3D Reconstruction
 - photos
- Floor plans
 - optimization



from Müller et al.



from Aliaga et al.



from Merrell et al.

CHANGES

- Linear order of execution of rules
- Rules become statements with queries
- A symbol becomes a component
 - persistency
 - inheritance
- Operation execution
 - applied on any subset of components
 - applied individually or on all components

CONCEPTS

- Components
 - spatially and semantically define elements
 - organized in tree(s)
- Program
 - statements formed by queries and operations
- Geometry
 - attached to some components
 - created from some components

COMPONENT

- 2D or 3D boundary with a bounding box
 - Labels
 - User attributes
 - Child components
 - Regions
 - Connector
- 

PROGRAM

- List of statements

- queries
- conditions
- operations

- for c in query (« label1 ») do
 operation1 (c, ...)
 operation2 (c, ...)

...

end

- operation (query (« label1 »), ...)

- for c in query (« label1 ») do
 operation (c, query (« label2»), ...)
end

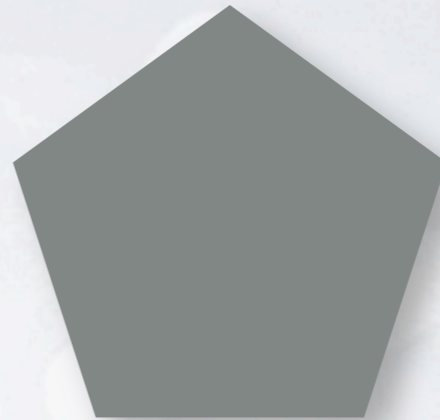
QUERY

- Subset of elements
 - components
 - faces
 - regions
- Conditions
 - labels
 - attributes
 - visibility



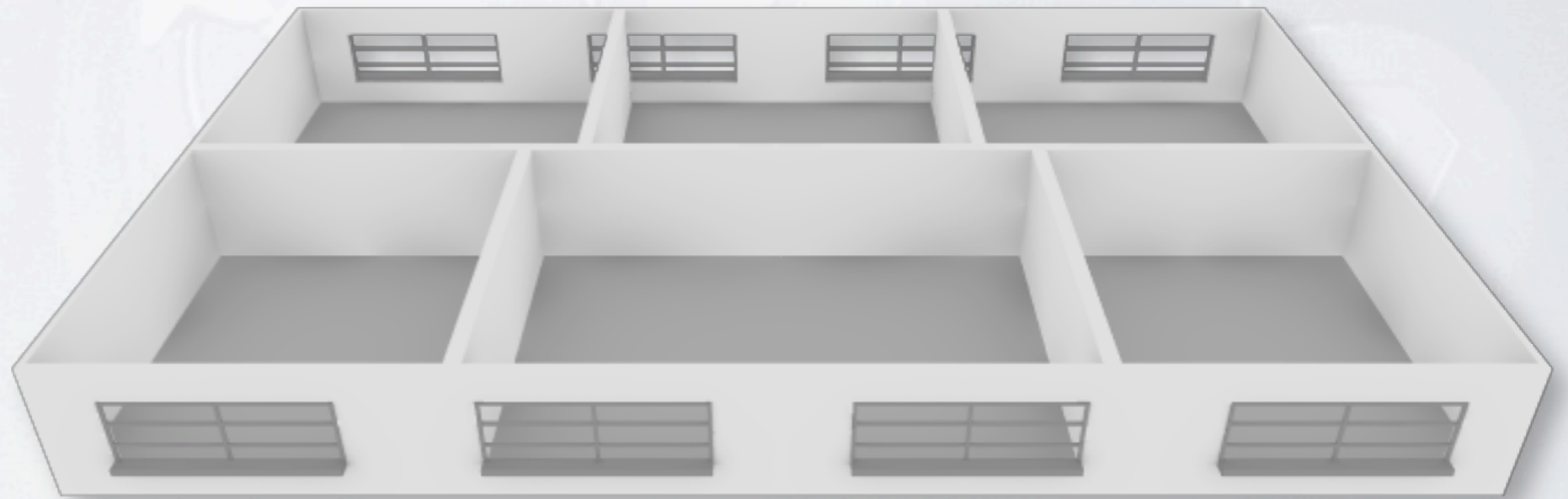
OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



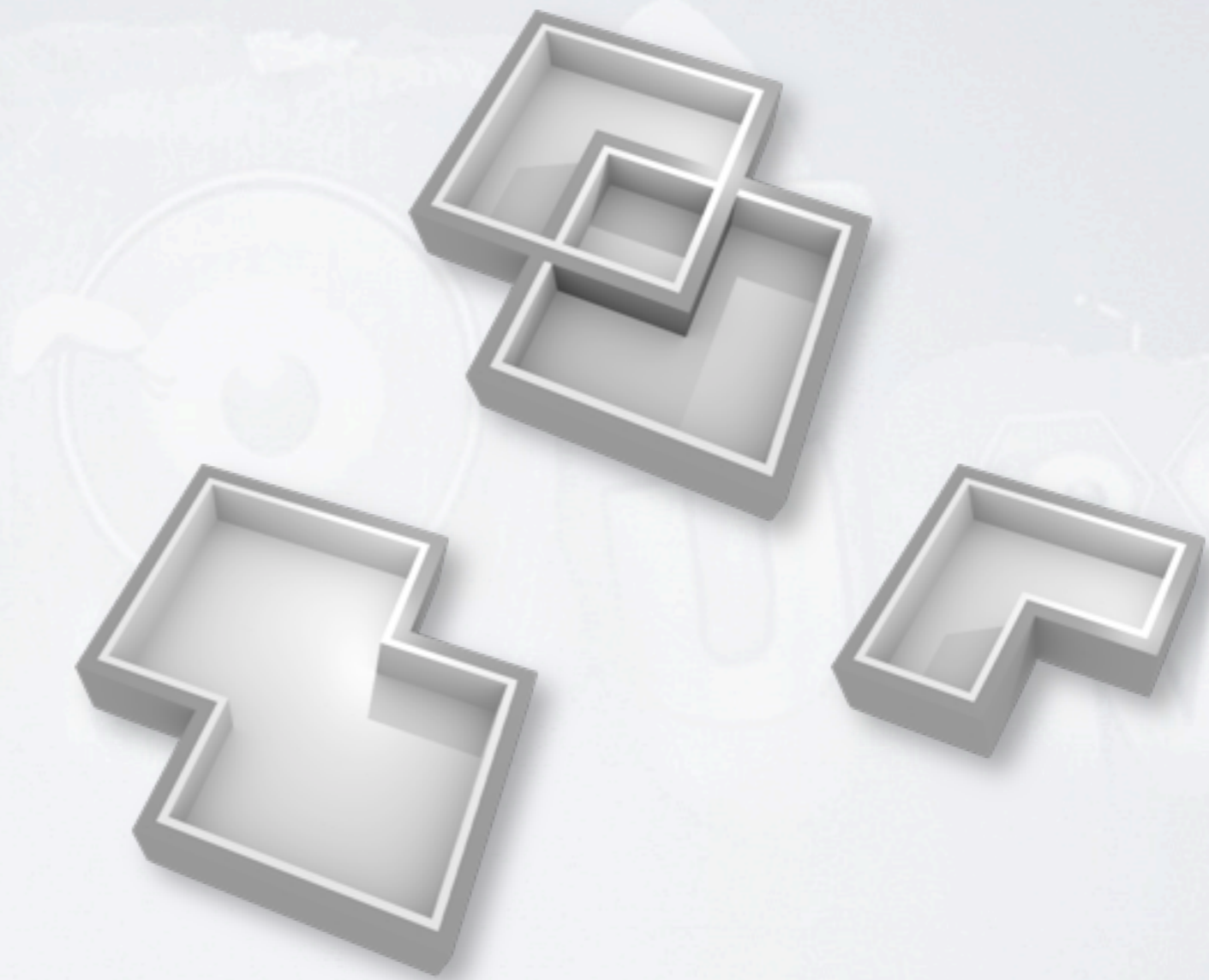
OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



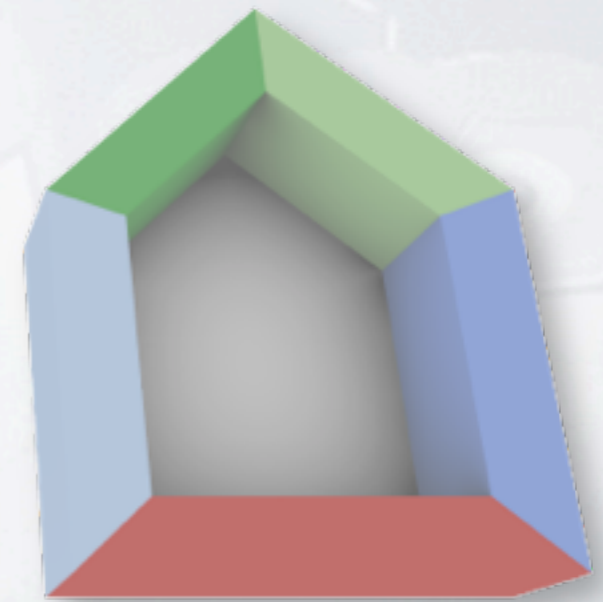
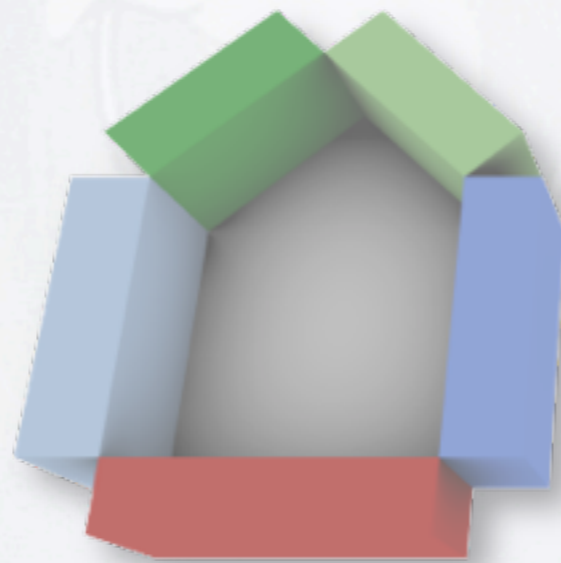
OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



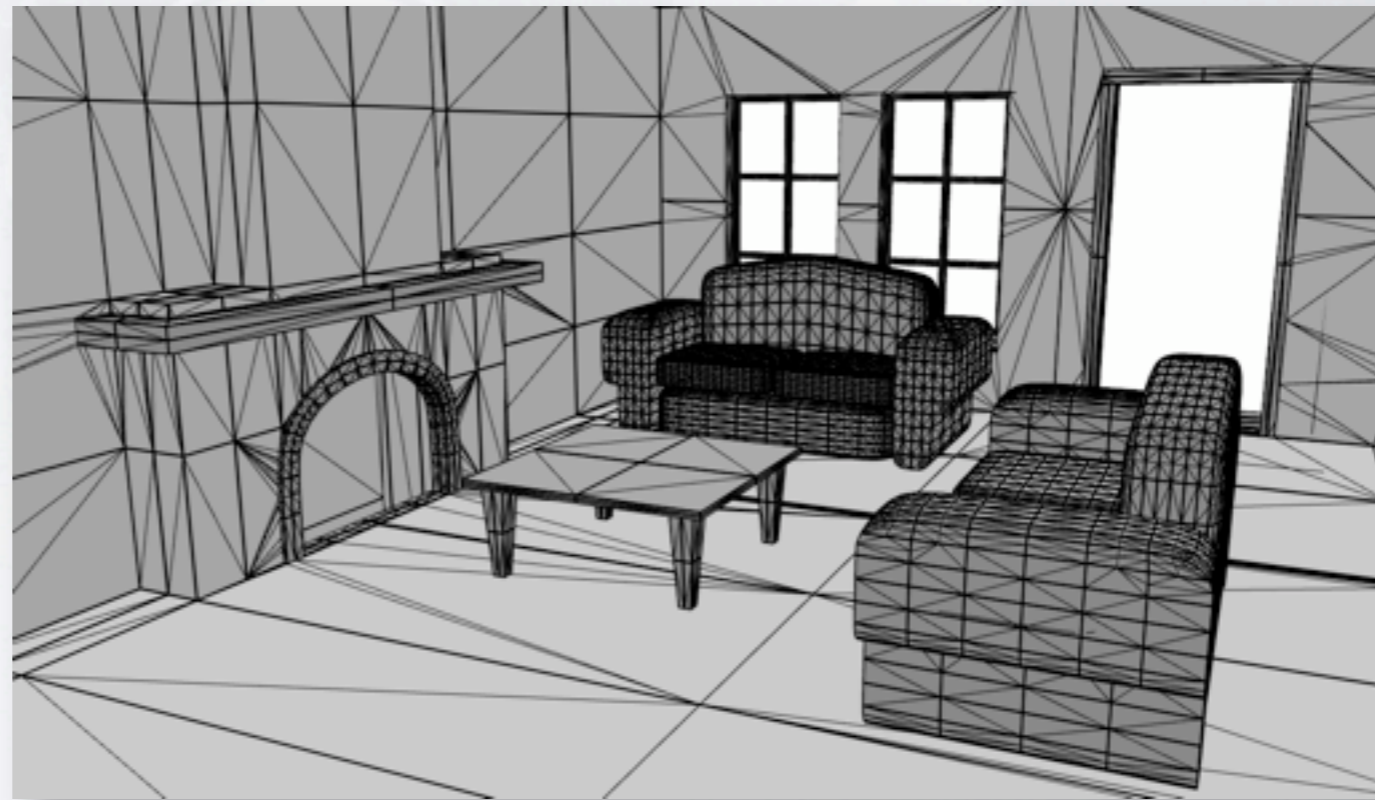
OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



OPERATIONS

- Creation
- Split and slice
- Boolean
- Extrusion
- Connection
- Geometry



BUILDINGS

- Space partitioning
 - building shape
 - subdivision into rooms
- Base geometry
 - walls, floors and ceilings
- Architectural elements
 - doors, windows, staircases, balconies, ...
- Furniture, decorations


```
-- Main subdivision
```

```
component{ label="main", size={10,5,10}, color=1 }
```

```
for c in query( "main" ) do  
  slice( c, "Y", { label="story", 2.5, level=count() } )  
end
```

```
for c in query( "story" ) do  
  split( c, "Z", { label="living space", rel=2 },  
        { label="corridor", color=2, abs=2 },  
        { label="living space", rel=2 } )  
end
```

```
for c in query( "living space" ) do  
  split( c, "X", { label="apartment", rel=1 },  
        { label="apartment", rel=1 } )  
end
```

```
component{  
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},  
  color=3  
}
```

```
-- Priorities.
```

```
for c in query( "apartment" or "corridor" ) do  
  subtract( c, query( "elevator" ), { label="room" } )  
end
```

```
-- Frames
```

```
for c in query( "room" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="wall", boundary=f }  
  end  
  for f in fquery( c, "BOTTOM" ) do  
    component{ c, label="floor", boundary=f }  
  end  
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )  
end
```

```
for c in query( "main" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="facade", boundary=f }  
  end  
end  
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )
```

```
-- Regions
```

```
for c in query( "wall" and parent("apartment")  
              and occlusion("corridor") > 0 ) do  
  region{ c, label="door" }  
end
```

```
for c in query( "wall" and parent("elevator") ) do  
  slice( c, "Y", { id="elwall", 2.5 } )  
end
```

```
for c in query( "elwall" and occlusion("corridor") > 0 ) do  
  region{ c, id="door" }  
end
```

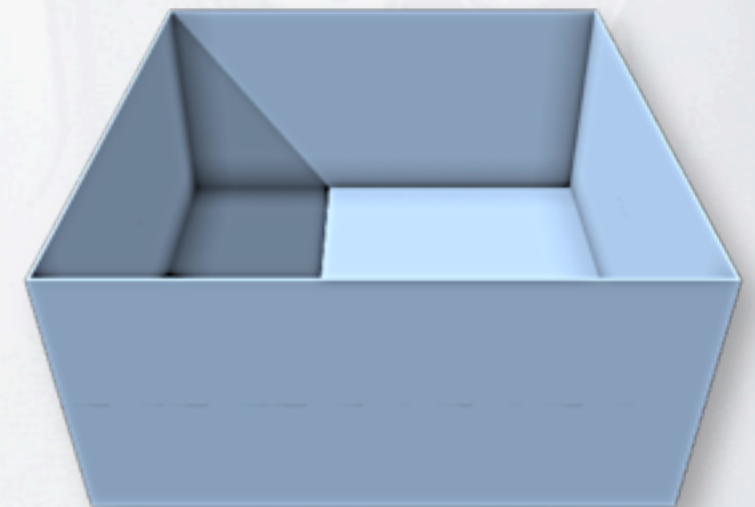
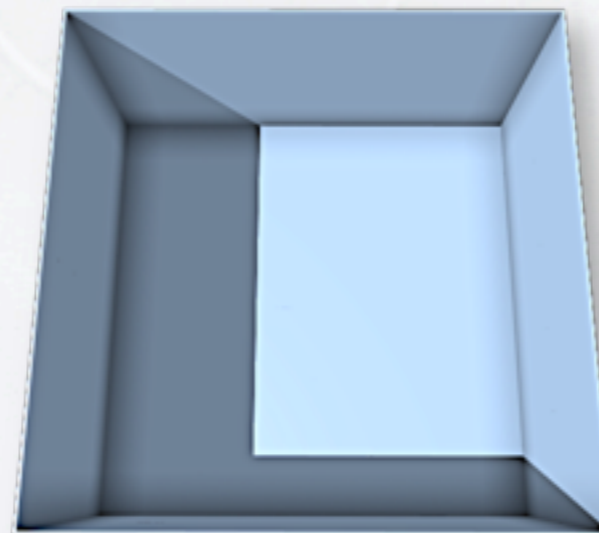
```
---- Doors
```

```
for r in rquery( "door" ) do  
  connect( componentFromFile("door"), r )  
end
```

```
-- Create actual geometry.
```

```
for c in query( "iwall", "ewall" ) do  
  solidGeometry( c, c.color )  
end
```

main



```
-- Main subdivision.  
component{ label="main", size={10,5,10}, color=1 }
```

```
for c in query( "main" ) do  
  slice( c, "Y", { label="story", 2.5, level=count() } )  
end
```

```
for c in query( "story" ) do  
  split( c, "Z", { label="living space", rel=2 },  
         { label="corridor", color=2, abs=2 },  
         { label="living space", rel=2 } )  
end
```

```
for c in query( "living space" ) do  
  split( c, "X", { label="apartment", rel=1 },  
         { label="apartment", rel=1 } )  
end
```

```
component{  
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},  
  color=3  
}
```

```
-- Priorities.  
for c in query( "apartment" or "corridor" ) do  
  subtract( c, query( "elevator" ), { label="room" } )  
end
```

```
-- Frames  
for c in query( "room" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="wall", boundary=f }  
  end  
  for f in fquery( c, "BOTTOM" ) do  
    component{ c, label="floor", boundary=f }  
  end  
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )  
end
```

```
for c in query( "main" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="facade", boundary=f }  
  end  
end  
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )
```

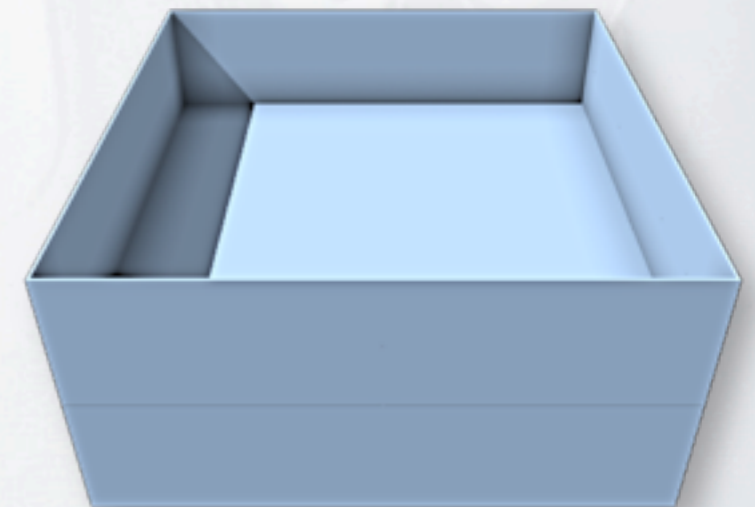
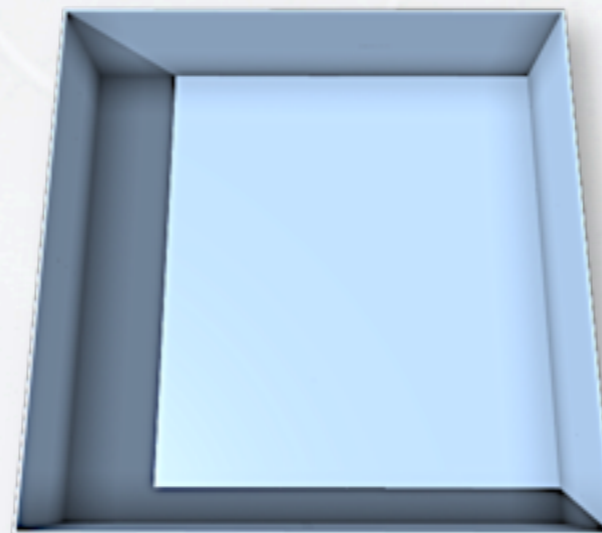
```
-- Regions  
for c in query( "wall" and parent("apartment")  
               and occlusion("corridor") > 0 ) do  
  region{ c, label="door" }  
end
```

```
for c in query( "wall" and parent("elevator") ) do  
  slice( c, "Y", { id="elwall", 2.5 } )  
end
```

```
for c in query( "elwall" and occlusion("corridor") > 0 ) do  
  region{ c, id="door" }  
end
```

```
---- Doors  
for r in rquery( "door" ) do  
  connect( componentFromFile("door"), r )  
end
```

```
-- Create actual geometry.  
for c in query( "iwall", "ewall" ) do  
  solidGeometry( c, c.color )  
end
```




```
-- Main subdivision.  
component{ label="main", size={10,5,10}, color=1 }
```

```
for c in query( "main" ) do
```

```
for c in query( "story" ) do  
  split( c, "Z", { label="living space", rel=2 },  
        { label="corridor", color=2, abs=2 },  
        { label="living space", rel=2 } )  
end
```

```
split( c, "X", { label="apartment", rel=1 },  
      { label="apartment", rel=1 } )  
end
```

```
component{  
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},  
  color=3  
}
```

```
-- Priorities.
```

```
for c in query( "apartment" or "corridor" ) do  
  subtract( c, query( "elevator" ), { label="room" } )  
end
```

```
-- Frames
```

```
for c in query( "room" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="wall", boundary=f }  
  end  
  for f in fquery( c, "BOTTOM" ) do  
    component{ c, label="floor", boundary=f }  
  end  
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )  
end
```

```
for c in query( "main" ) do  
  for f in fquery( c, "SIDE" ) do  
    component{ c, label="facade", boundary=f }  
  end  
end  
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )
```

```
-- Regions
```

```
for c in query( "wall" and parent("apartment")  
              and occlusion("corridor") > 0 ) do  
  region{ c, label="door" }  
end
```

```
for c in query( "wall" and parent("elevator") ) do  
  slice( c, "Y", { id="elwall", 2.5 } )  
end
```

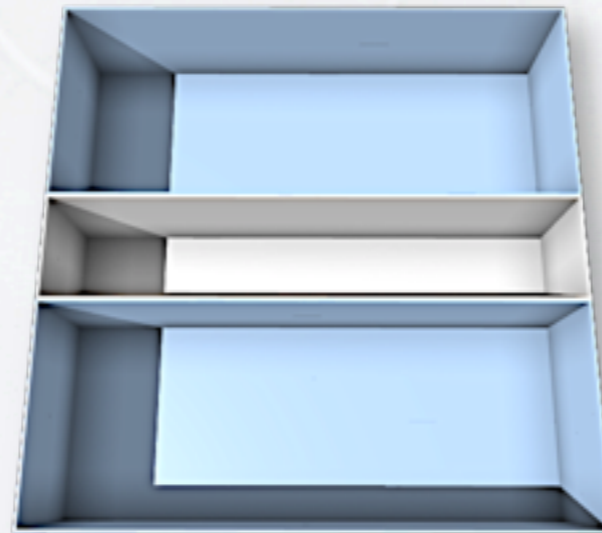
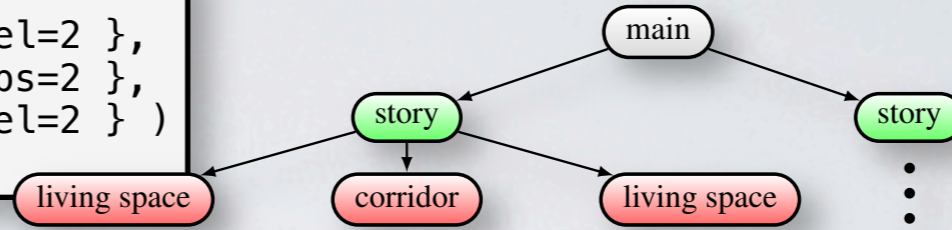
```
for c in query( "elwall" and occlusion("corridor") > 0 ) do  
  region{ c, id="door" }  
end
```

```
---- Doors
```

```
for r in rquery( "door" ) do  
  connect( componentFromFile("door"), r )  
end
```

```
-- Create actual geometry.
```

```
for c in query( "iwall", "ewall" ) do  
  solidGeometry( c, c.color )  
end
```



```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

```

```

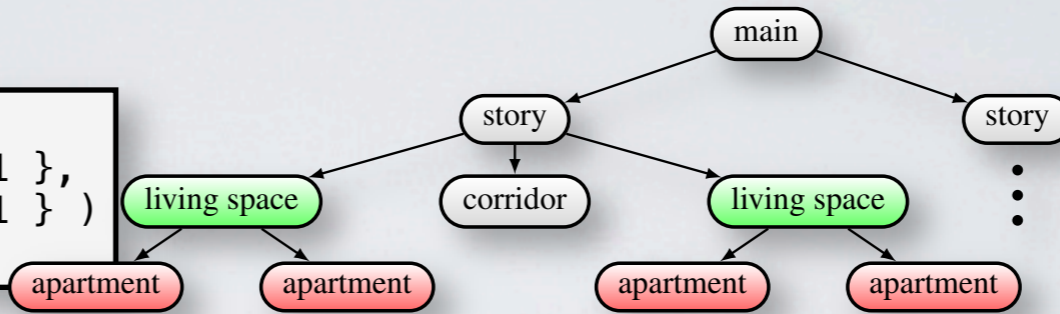
for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 } )
end

```

```

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

```



```

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```

```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

```

```

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

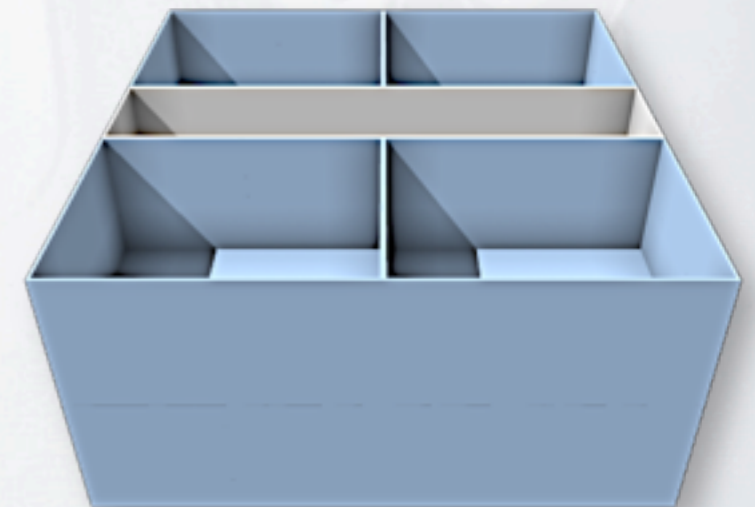
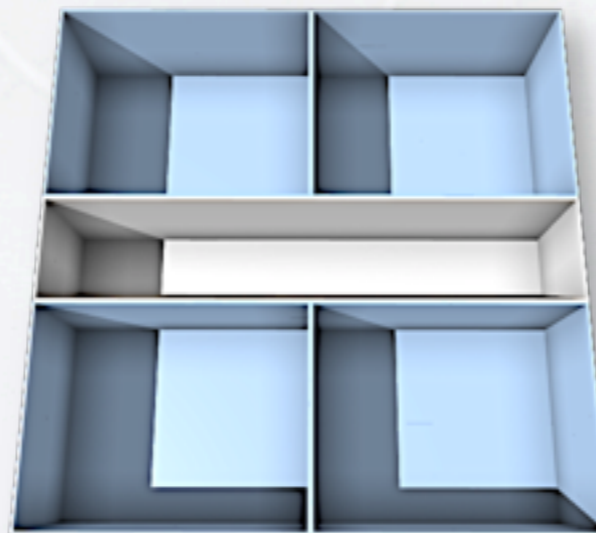
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

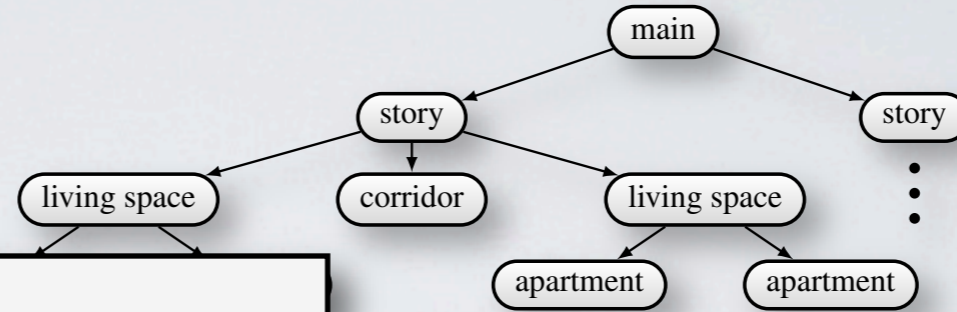
-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

```



```

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```

elevator, room

```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

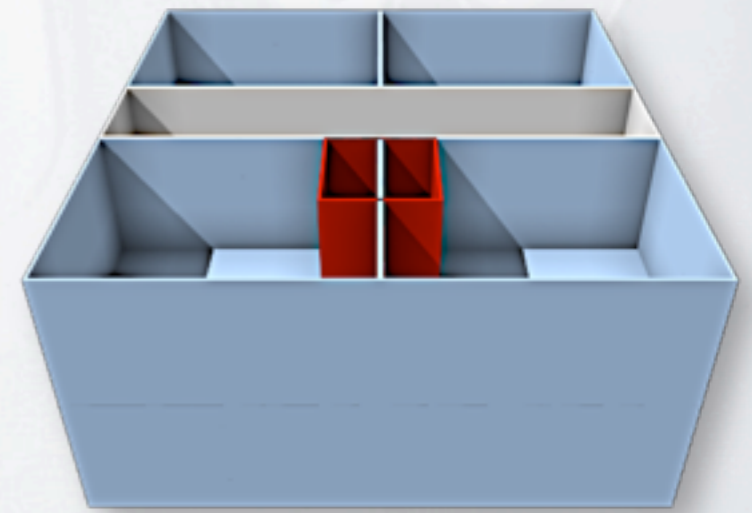
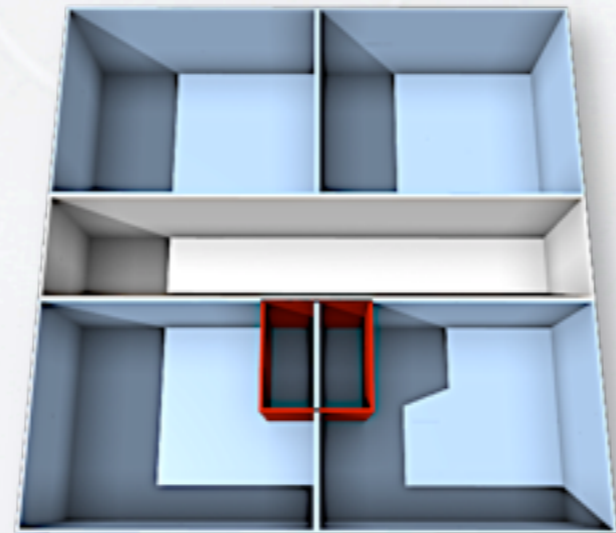
for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```



```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

```

```

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

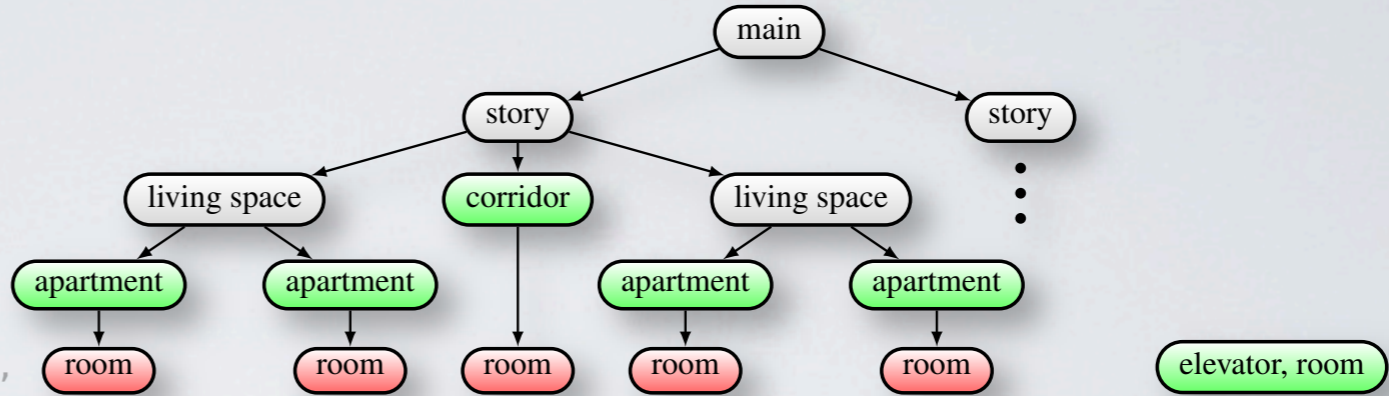
for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

```

```

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

```

```

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

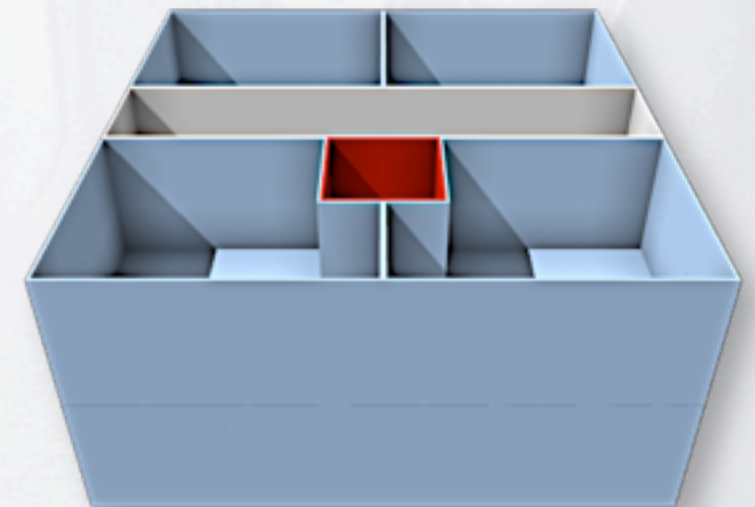
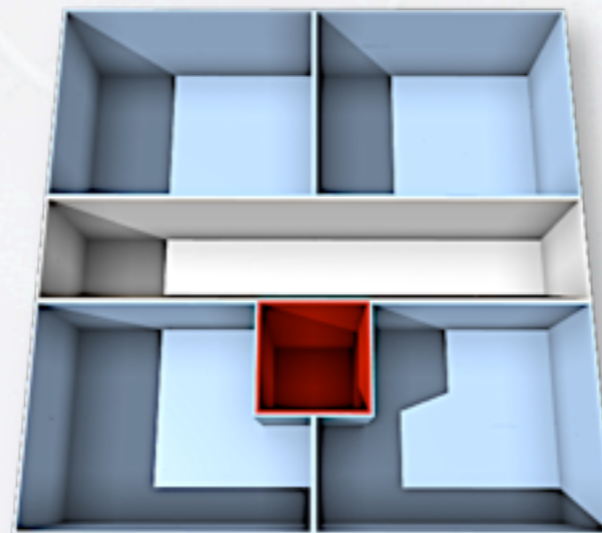
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

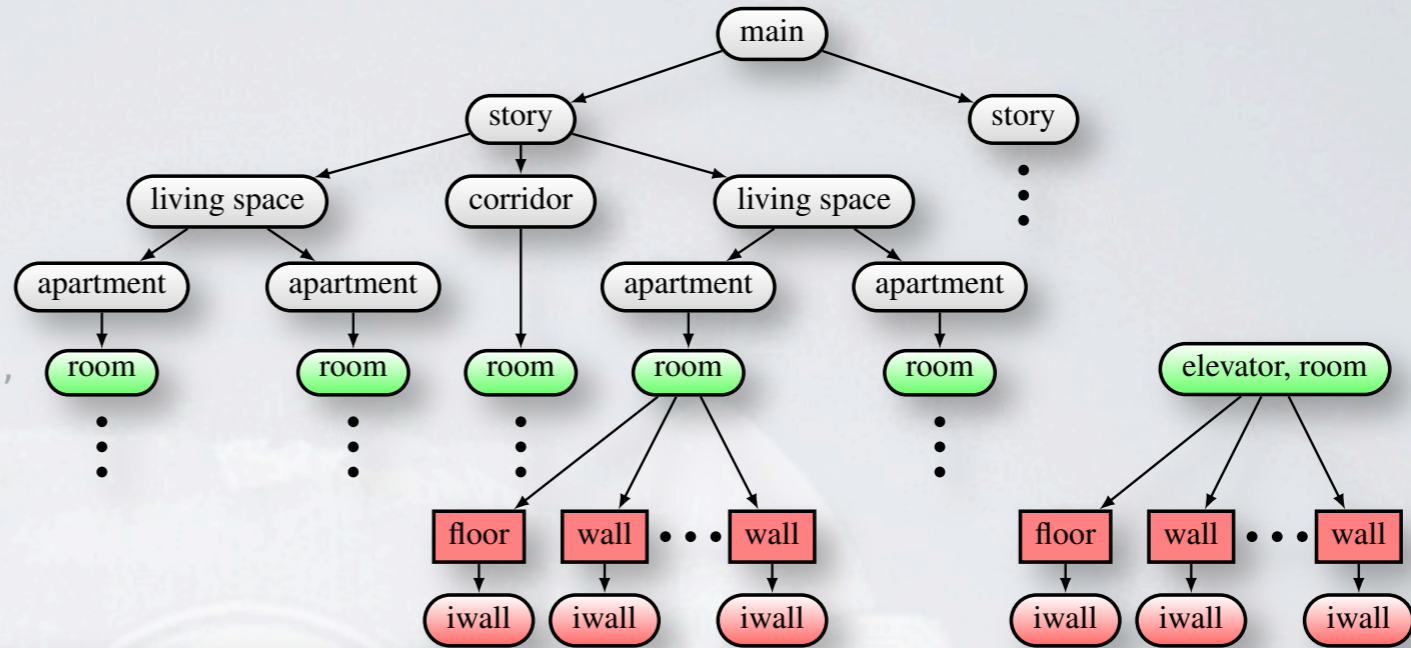
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

```

-- Frames

```

for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

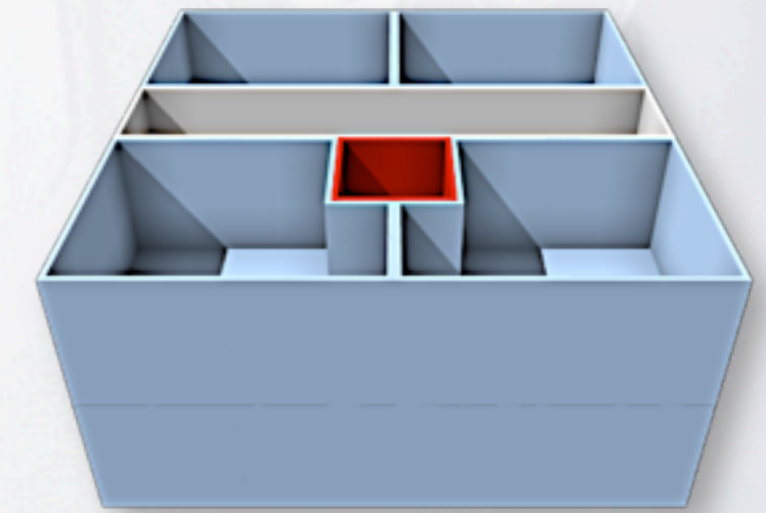
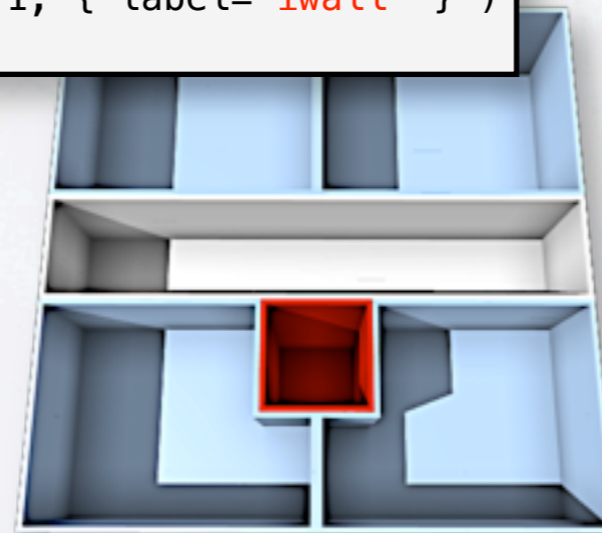
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```



```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

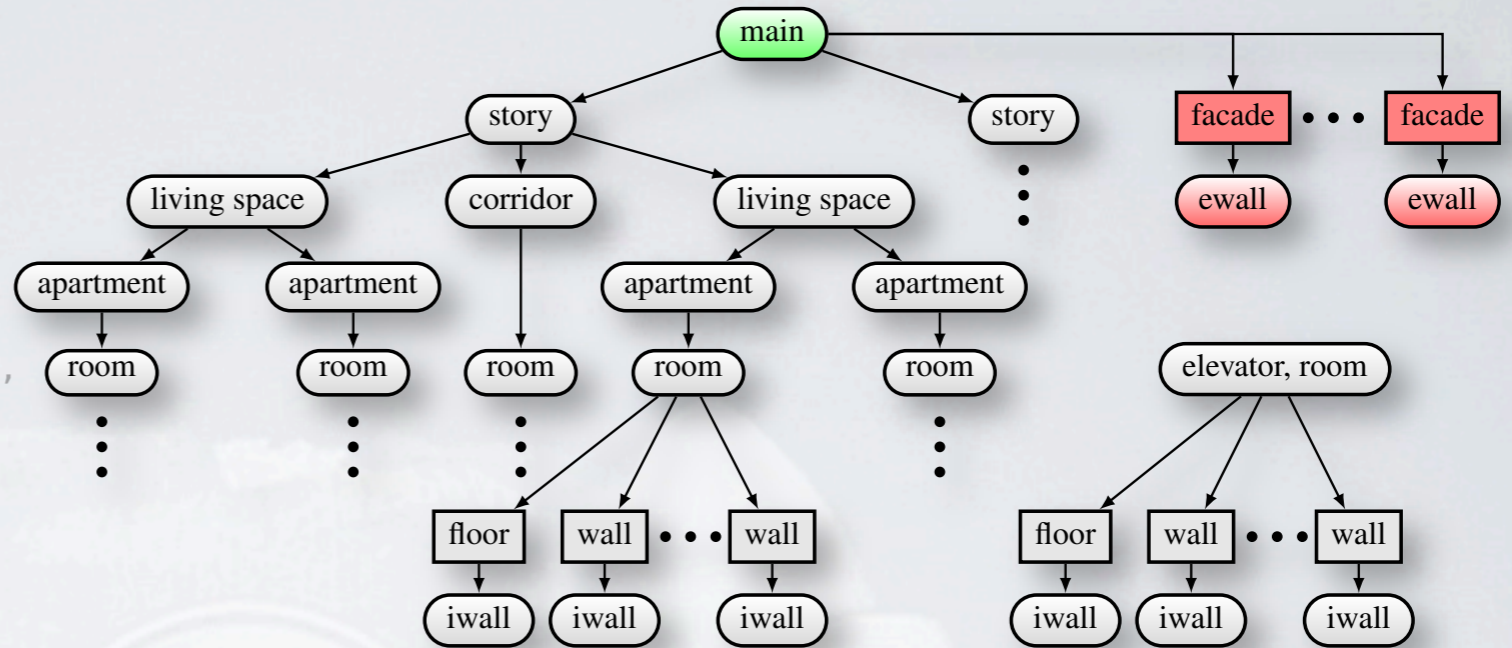
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

```

```

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

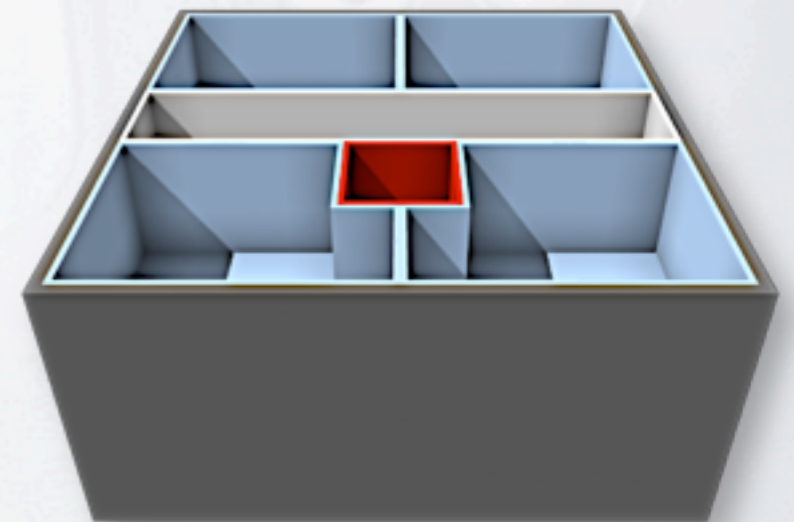
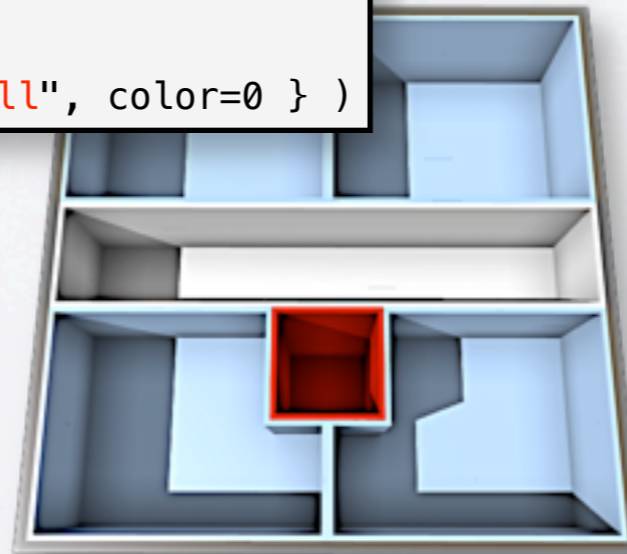
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

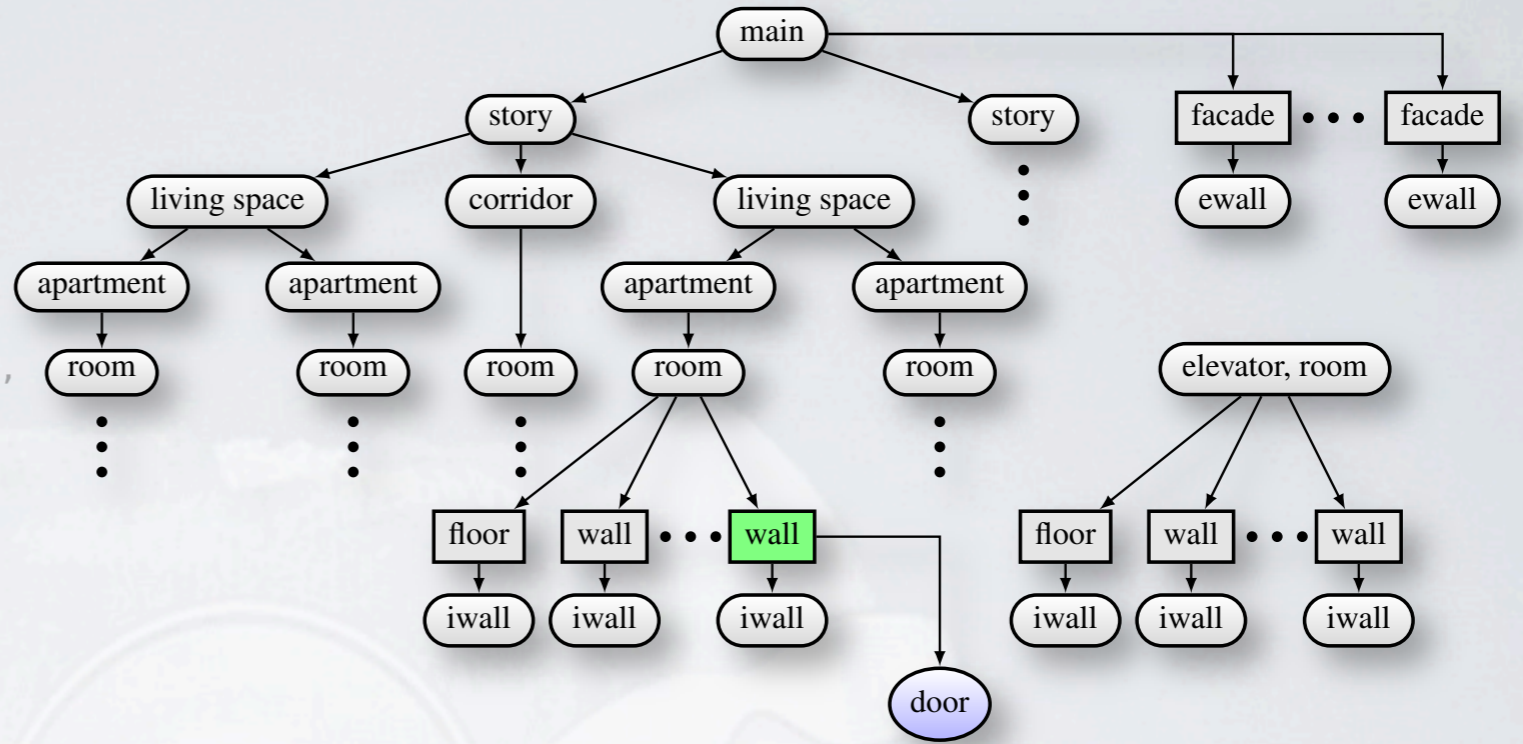
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end

```

```

for c in query( "wall" and parent("apartment")
                and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

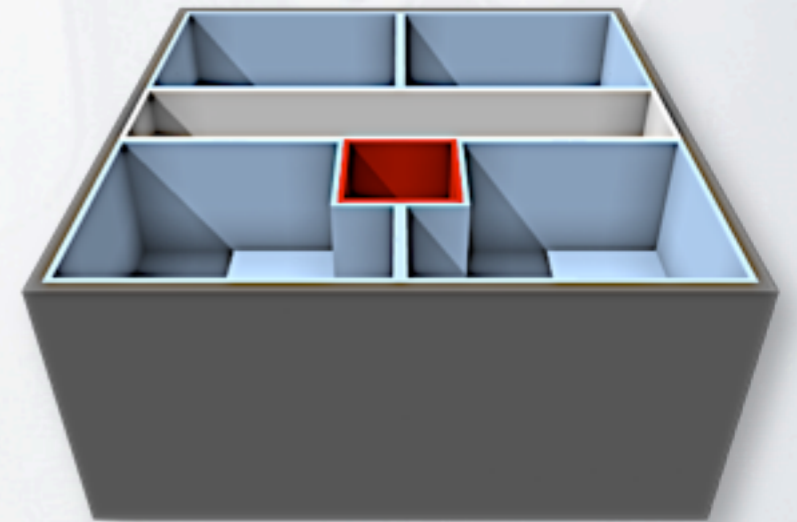
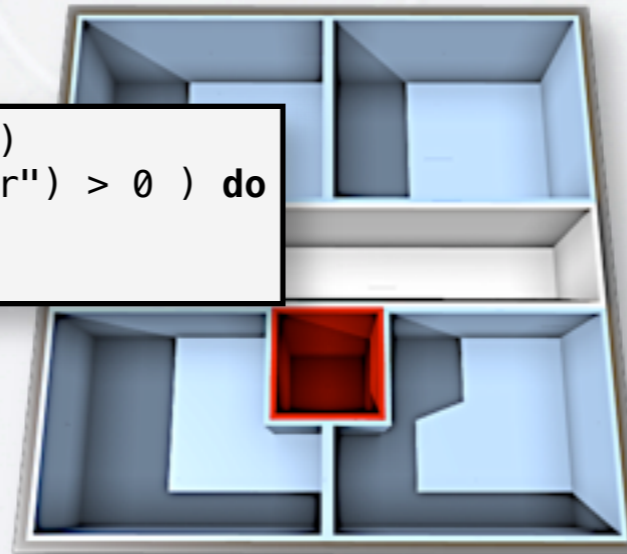
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```



```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

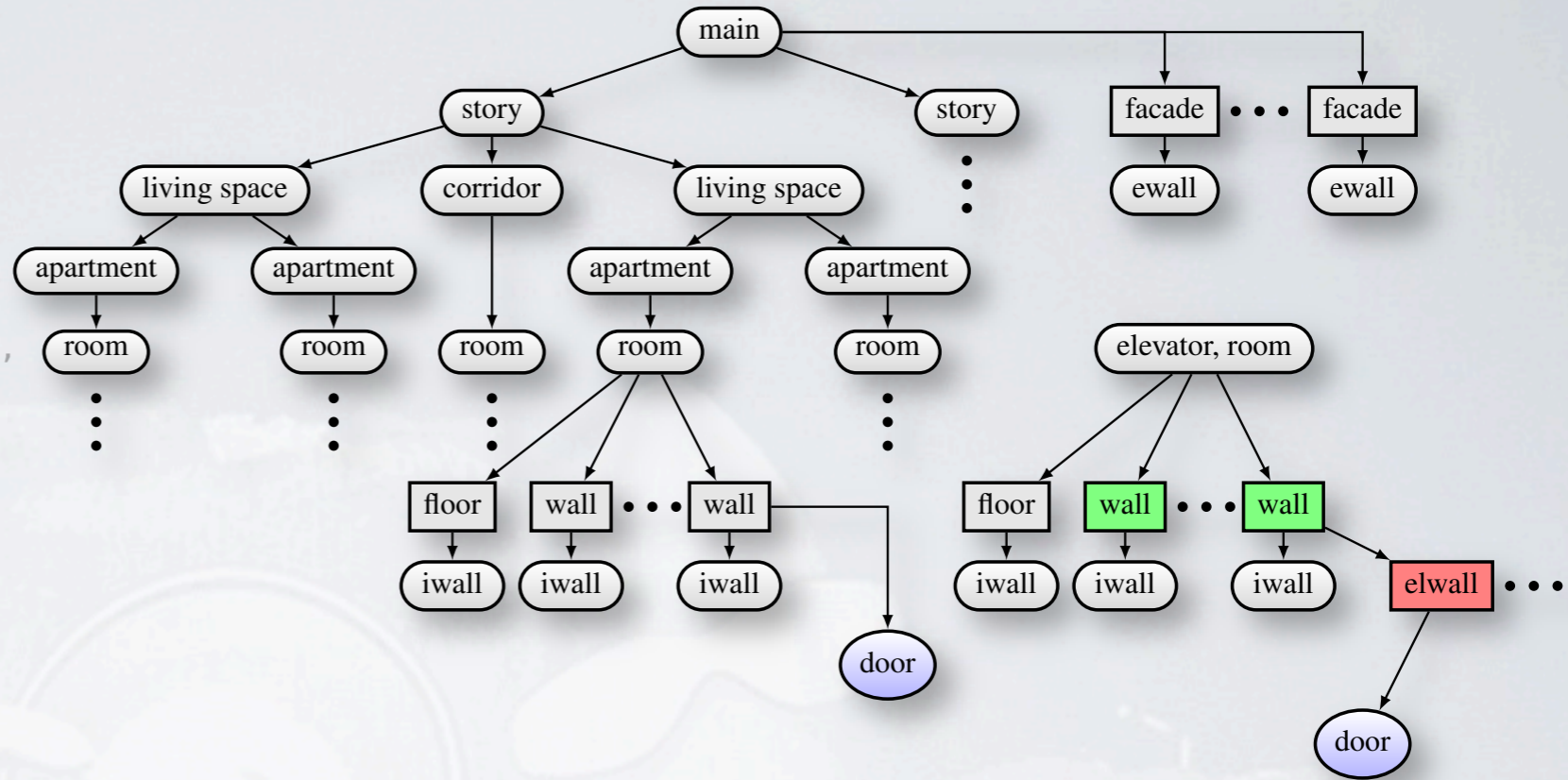
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

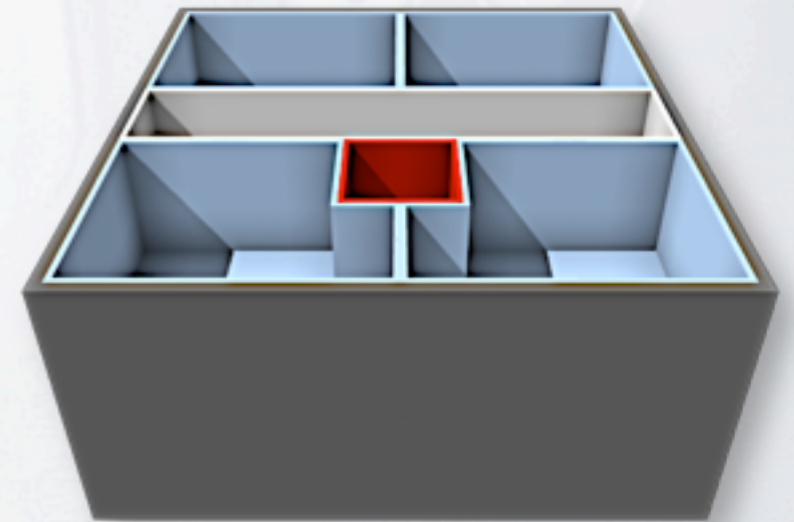
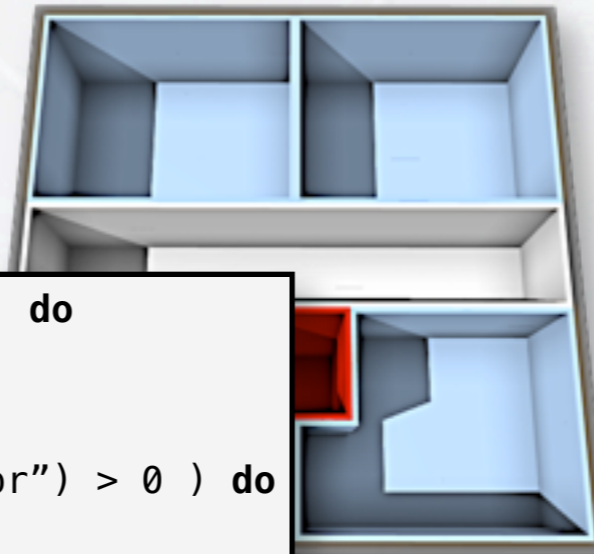
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

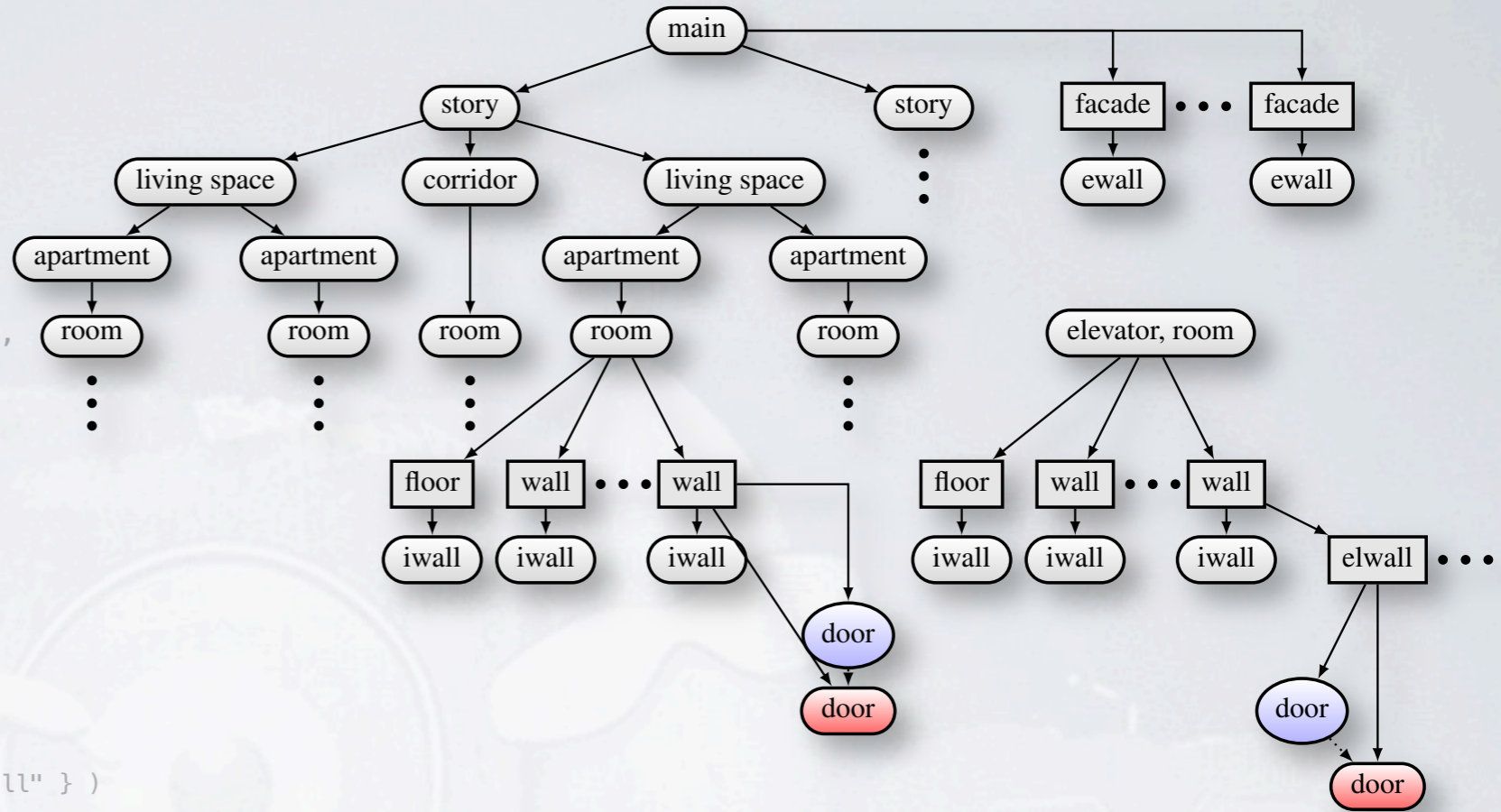
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

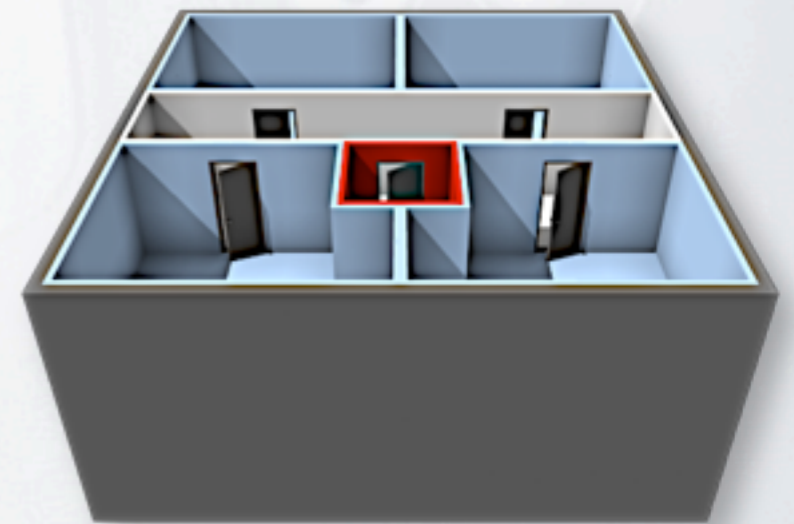
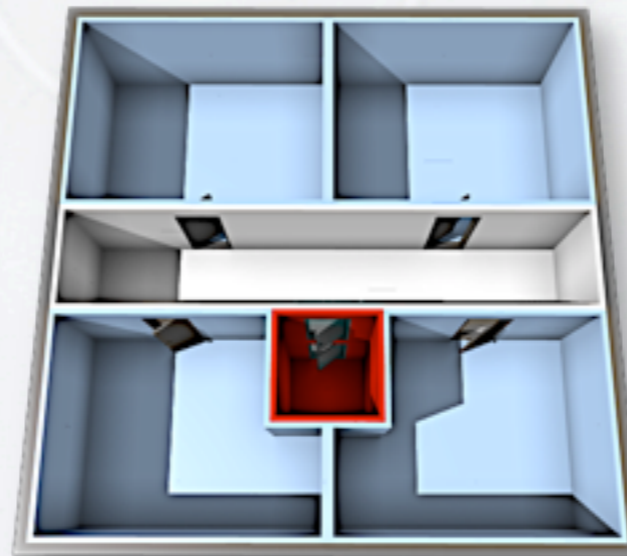
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

-- Create actual geometry.
for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

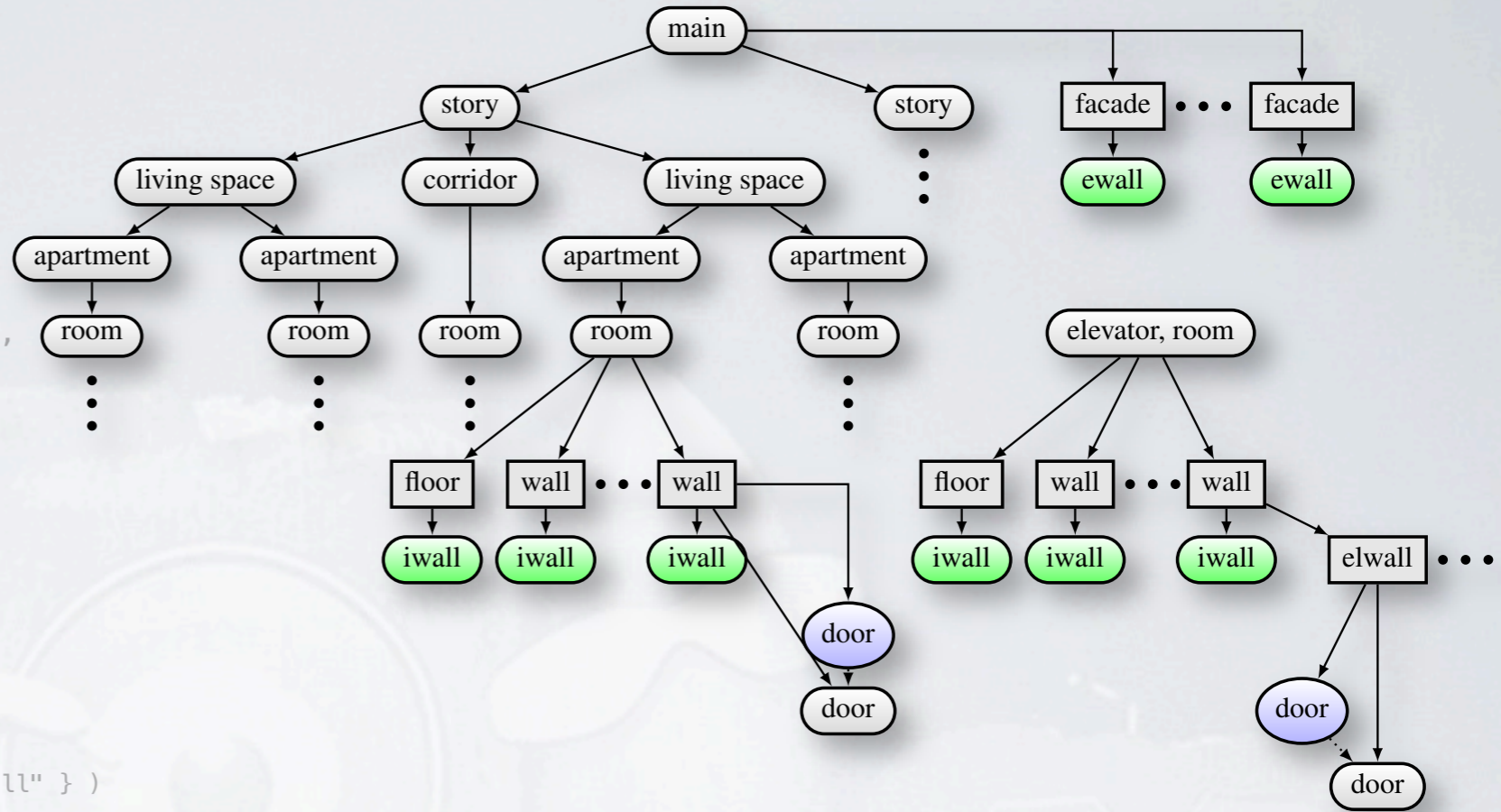
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

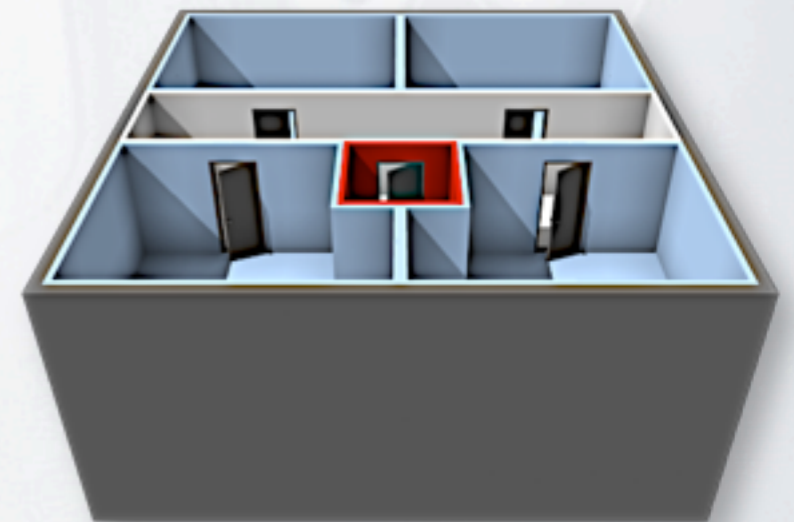
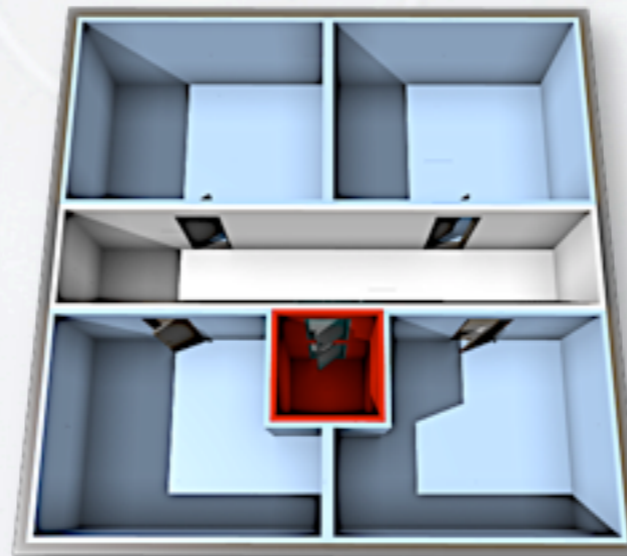
---- Doors
for r in rquery( "door" ) do
  connect( componentFromFile("door"), r )
end

```

```

for c in query( "iwall", "ewall" ) do
  solidGeometry( c, c.color )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

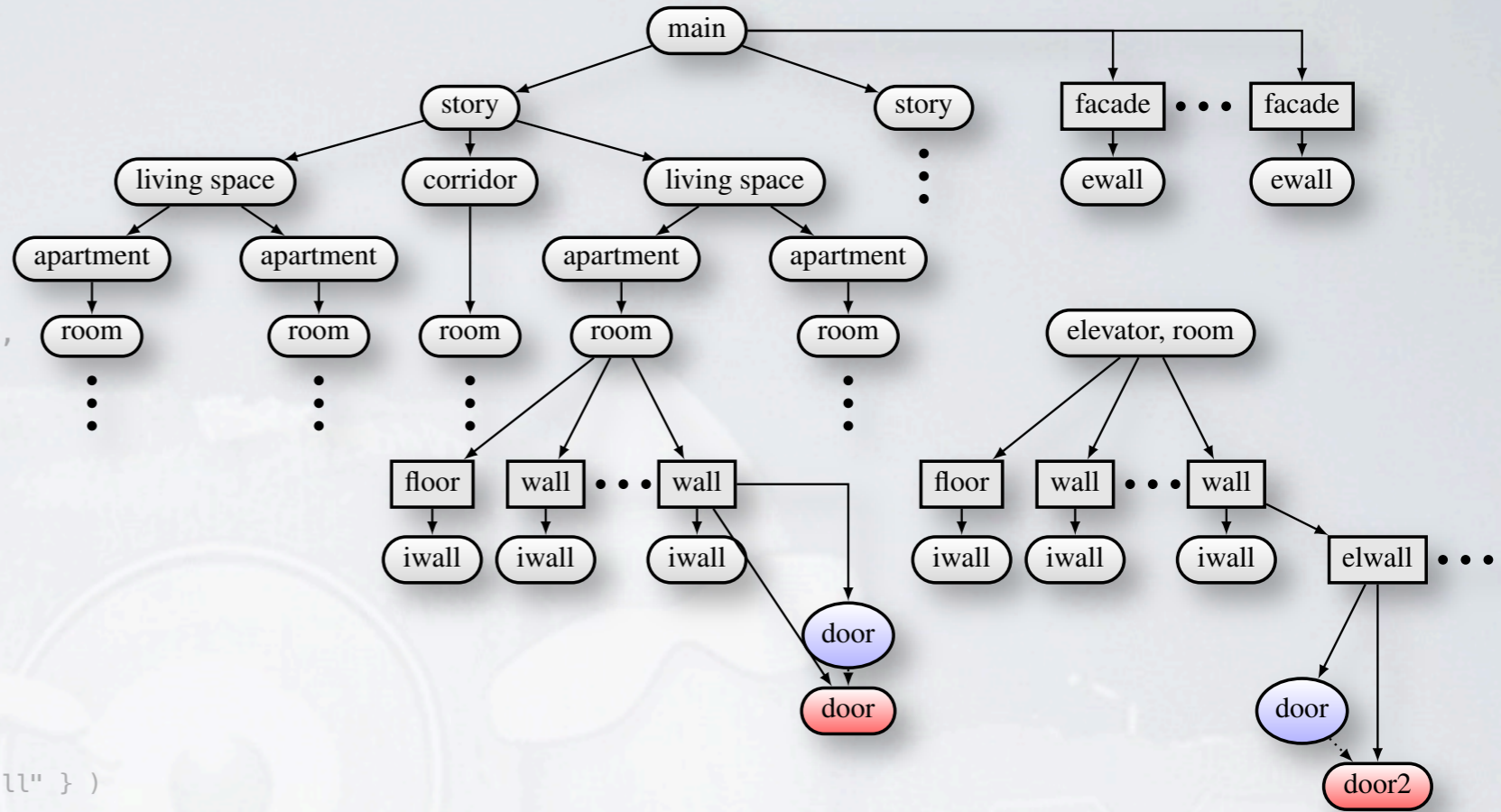
for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```



```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "main" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
extrude( query( "facade" ), 0.2, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door2" }
end

```

```

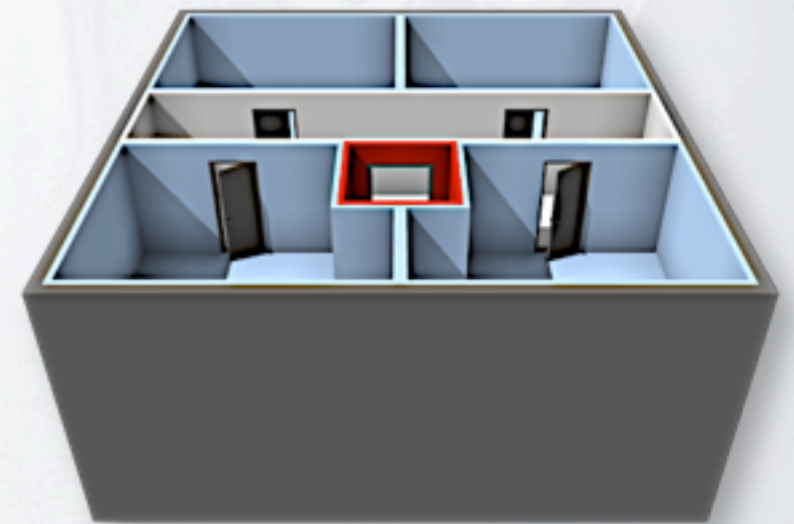
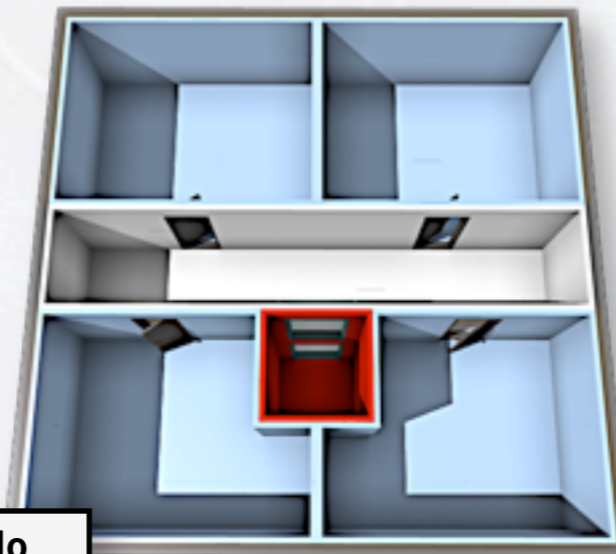
for r in rquery( "door" and parent("wall") ) do
  connect( componentFromFile("door"), r )
end
for r in rquery( "door" and parent("elwall") ) do
  connect( componentFromFile("door2"), r )
end

```

```

-- Create actual geometry
for c in query( "iwall", "ewall" ) do

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```

```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

```

```

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "story" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end
for c in query( "facade" ) do
  split( c, "Y", { label="f1", rel=1 }, { label="ledge", 0.2 } )
end
extrude( query( "f1" ), 0.2, { label="ewall", color=0 } )
extrude( query( "ledge" ), 0.3, { label="ewall", color=0 } )

```

```

for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

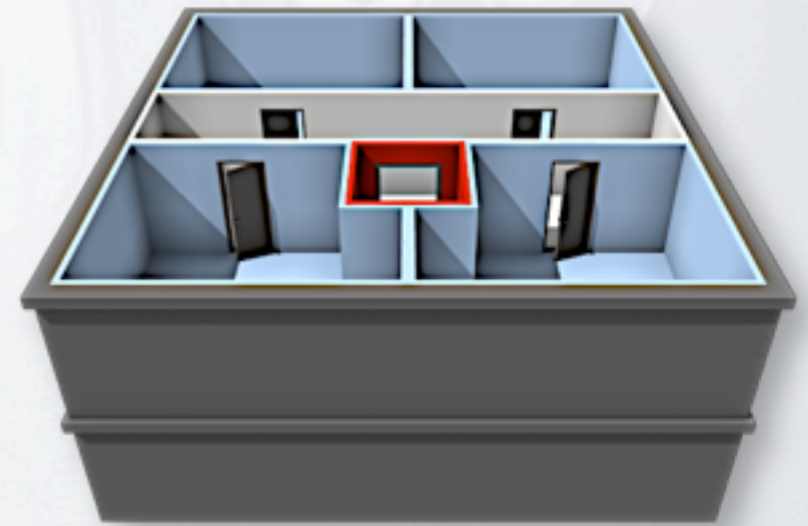
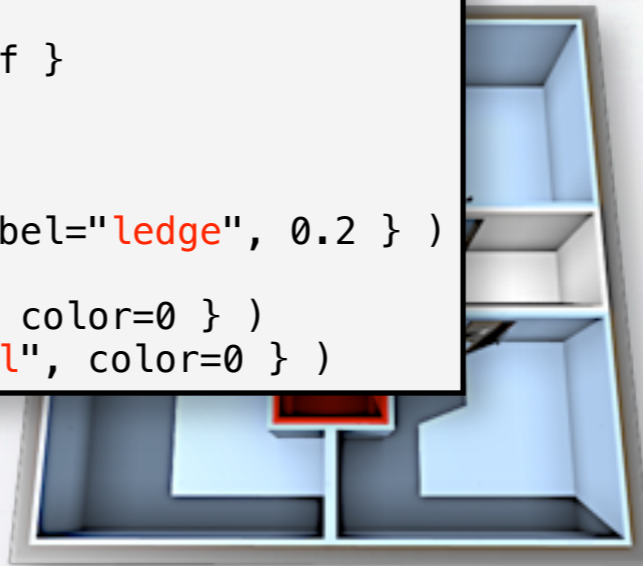
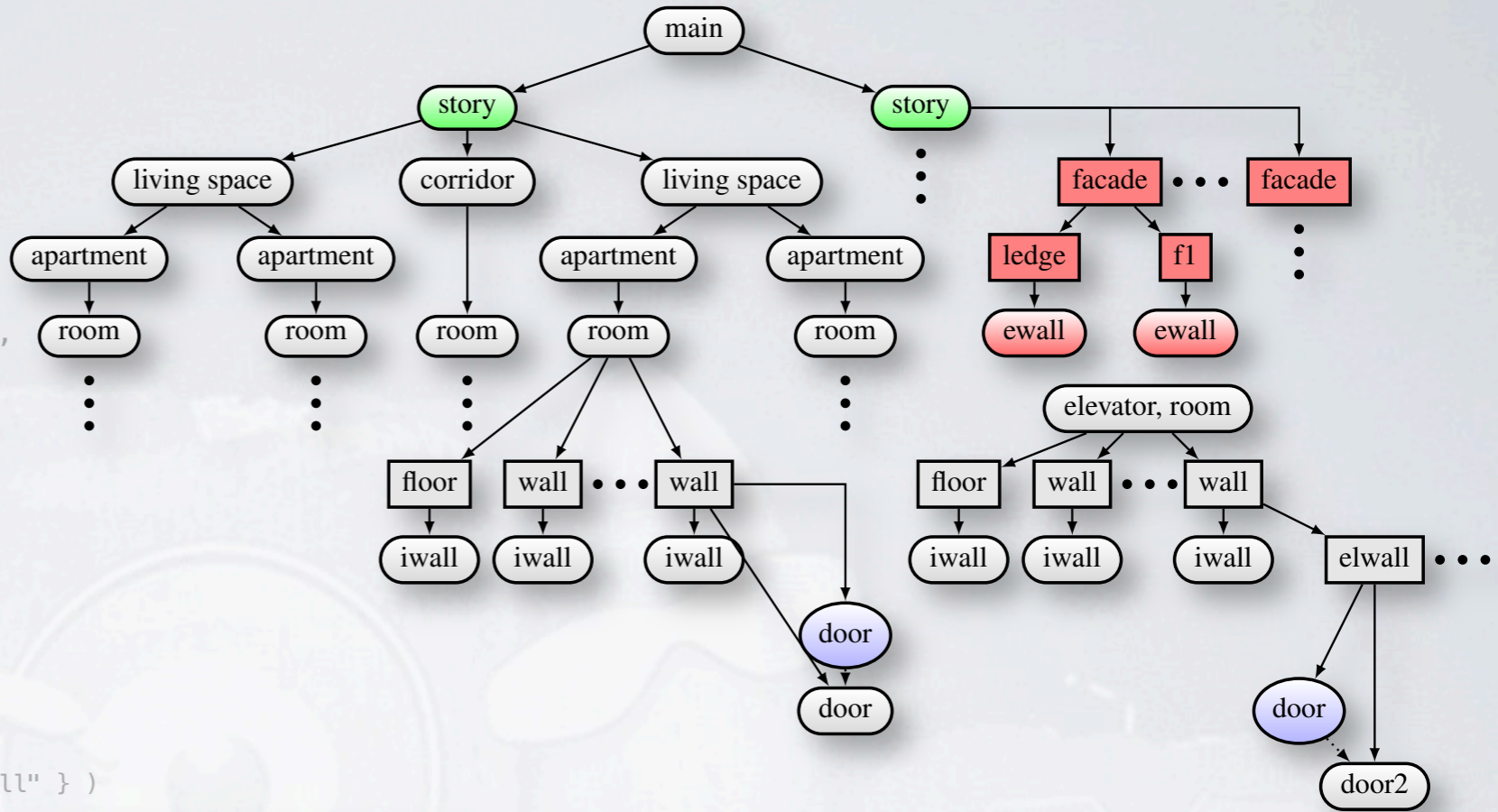
for c in query( "elwall" and occlusion("corridor") > 0 ) do
  region{ c, id="door" }
end

```

```

---- Doors
for r in rquery( "door" and parent("wall") ) do
  connect( componentFromFile("door"), r )
end
for r in rquery( "door" and parent("elwall") ) do
  connect( componentFromFile("door2"), r )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```

```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "story" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end

for c in query( "facade" ) do
  split( c, "Y", { label="f1", rel=1 }, { label="ledge", 0.2 } )
end
extrude( query( "f1" ), 0.2, { label="ewall", color=0 } )
extrude( query( "ledge" ), 0.3, { label="ewall", color=0 } )

```

```

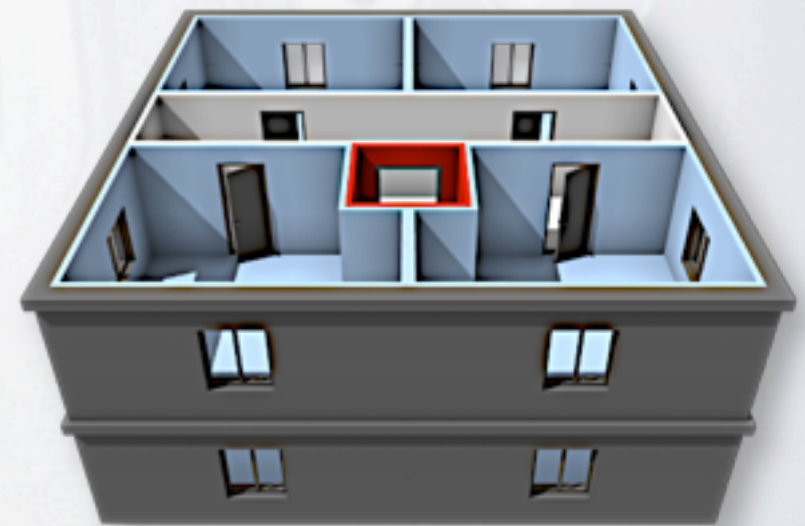
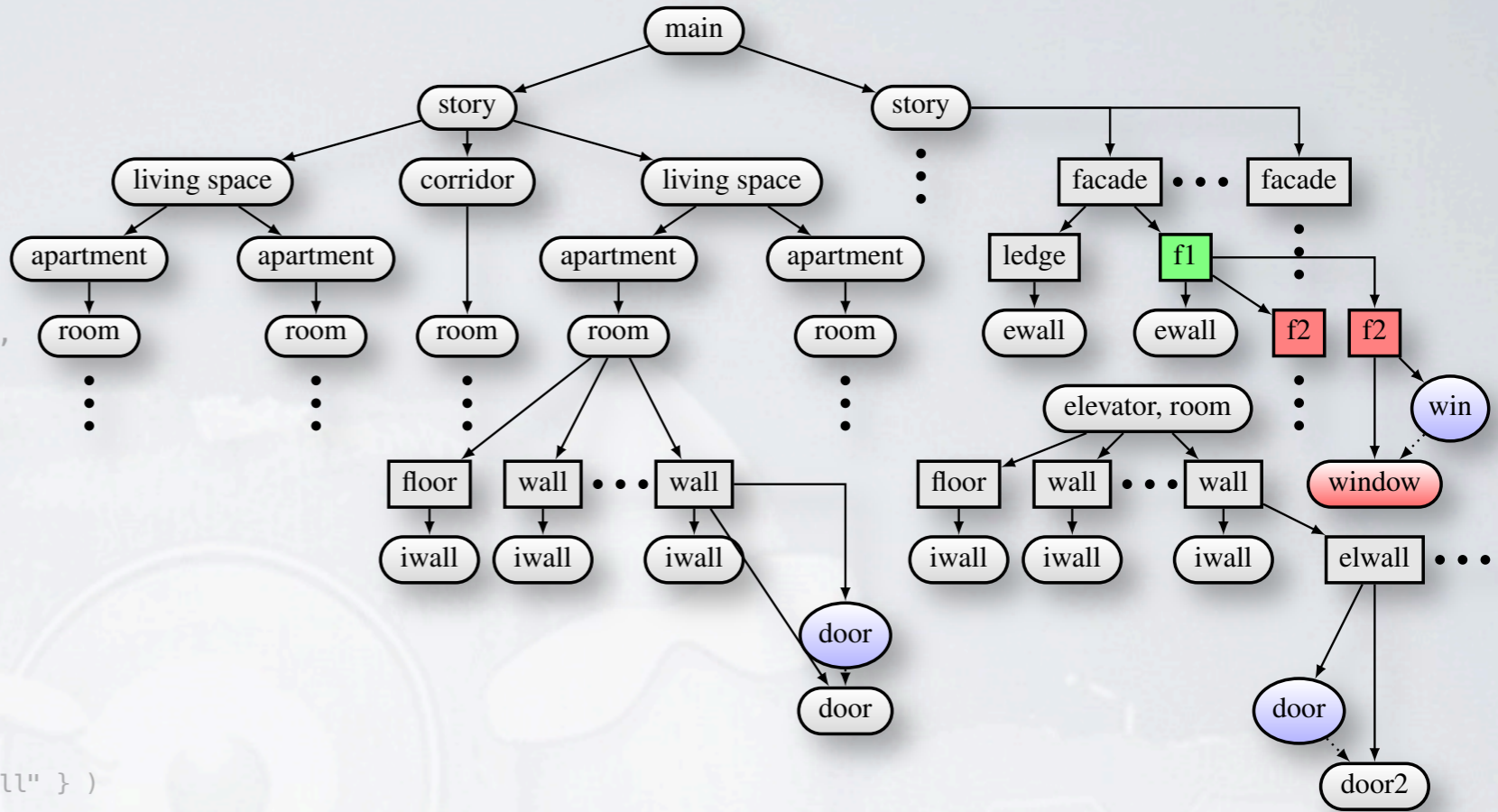
-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

for c in query( "f1" ) do
  slice( c, "X", { label="f2", 4 } )
end
for c in query( "f2" ) do
  region{ c, label="window" }
end
for r in rquery( "window" ) do
  connect( componentFromFile("window"), r )
end

```




```

-- Main subdivision.
component{ label="main", size={10,5,10}, color=1 }

for c in query( "main" ) do
  slice( c, "Y", { label="story", 2.5, level=count() } )
end

for c in query( "story" ) do
  split( c, "Z", { label="living space", rel=2 },
        { label="corridor", color=2, abs=2 },
        { label="living space", rel=2 } )
end

for c in query( "living space" ) do
  split( c, "X", { label="apartment", rel=1 },
        { label="apartment", rel=1 } )
end

component{
  label={"elevator", "room"}, size={2,5,2}, position={4,0,2},
  color=3
}

```

```

-- Priorities.
for c in query( "apartment" or "corridor" ) do
  subtract( c, query( "elevator" ), { label="room" } )
end

-- Frames
for c in query( "room" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="wall", boundary=f }
  end
  for f in fquery( c, "BOTTOM" ) do
    component{ c, label="floor", boundary=f }
  end
  extrude( query( c, "wall" or "floor" ), -0.1, { label="iwall" } )
end

```

```

for c in query( "story" ) do
  for f in fquery( c, "SIDE" ) do
    component{ c, label="facade", boundary=f }
  end
end

for c in query( "facade" ) do
  split( c, "Y", { label="f1", rel=1 }, { label="ledge", 0.2 } )
end
extrude( query( "f1" ), 0.2, { label="ewall", color=0 } )
extrude( query( "ledge" ), 0.3, { label="ewall", color=0 } )

```

```

-- Regions
for c in query( "wall" and parent("apartment")
               and occlusion("corridor") > 0 ) do
  region{ c, label="door" }
end

```

```

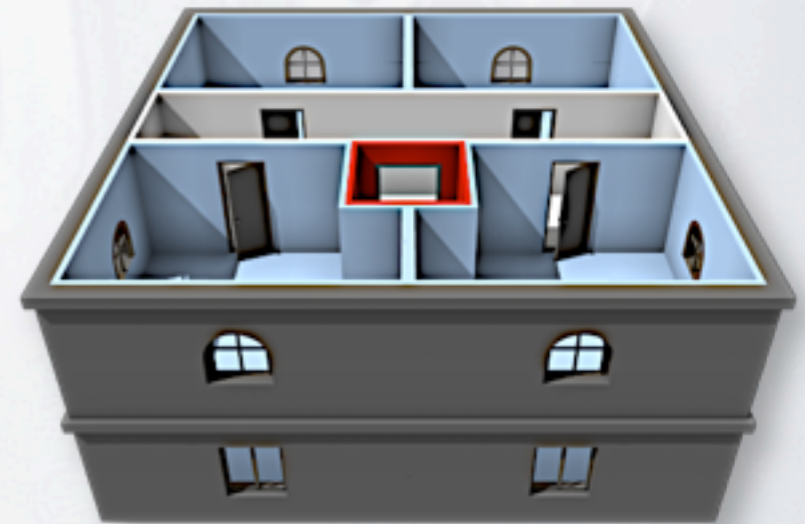
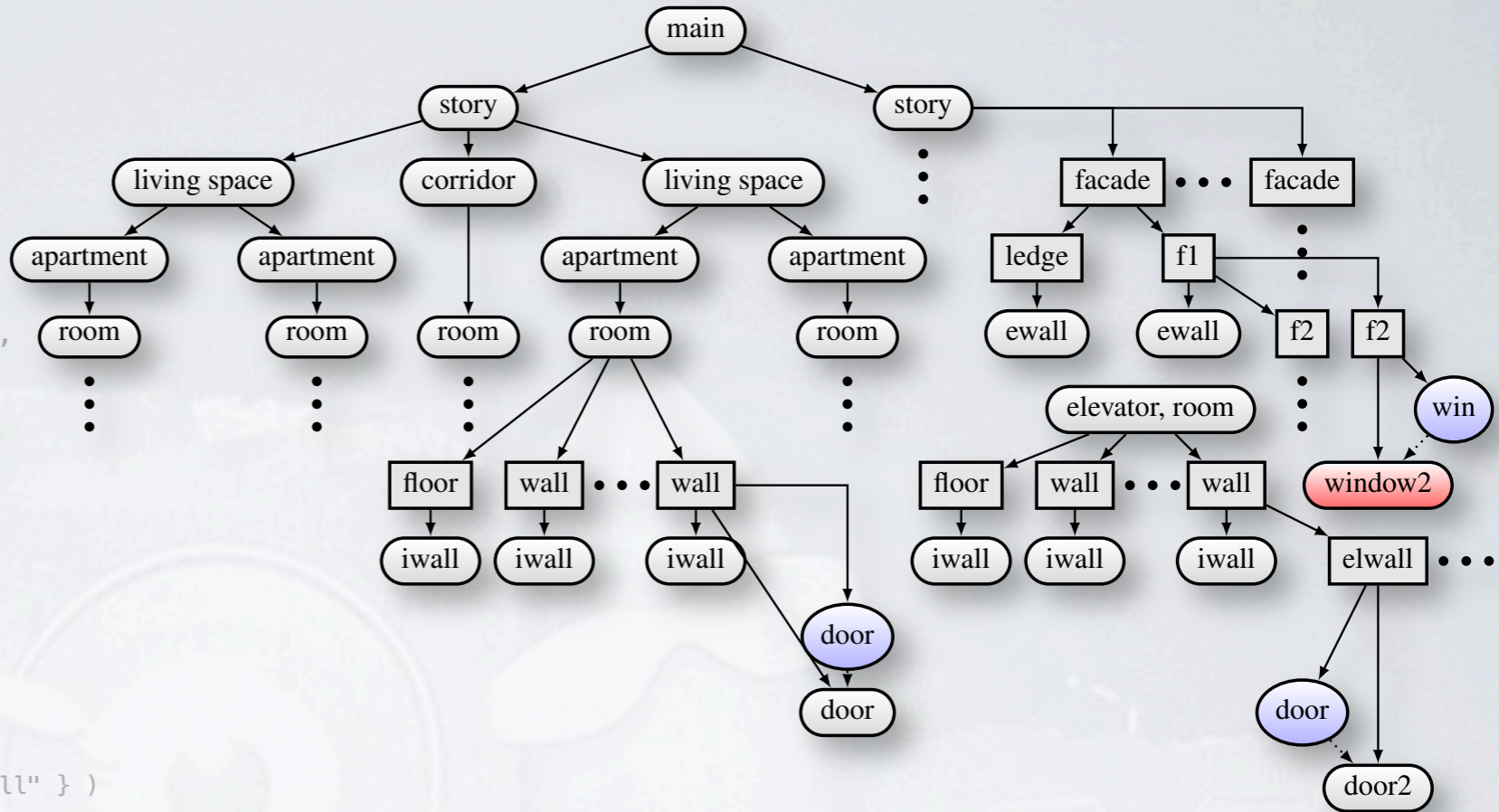
for c in query( "wall" and parent("elevator") ) do
  slice( c, "Y", { id="elwall", 2.5 } )
end

```

```

for r in rquery( "window" and level==1 ) do
  connect( componentFromFile("window"), r )
end
for r in rquery( "window" and level>1 ) do
  connect( componentFromFile("window2"), r )
end

```



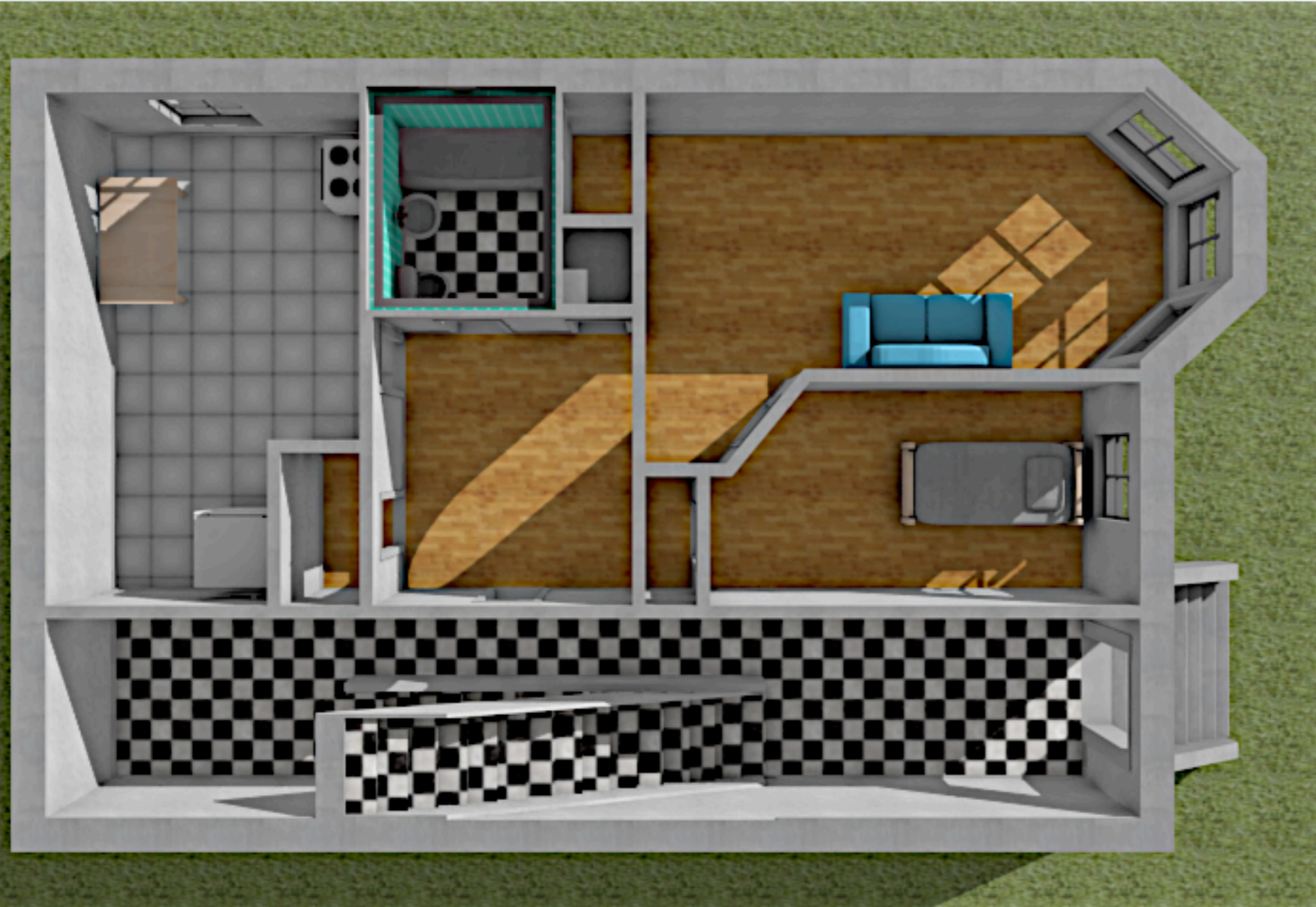
RESULTS



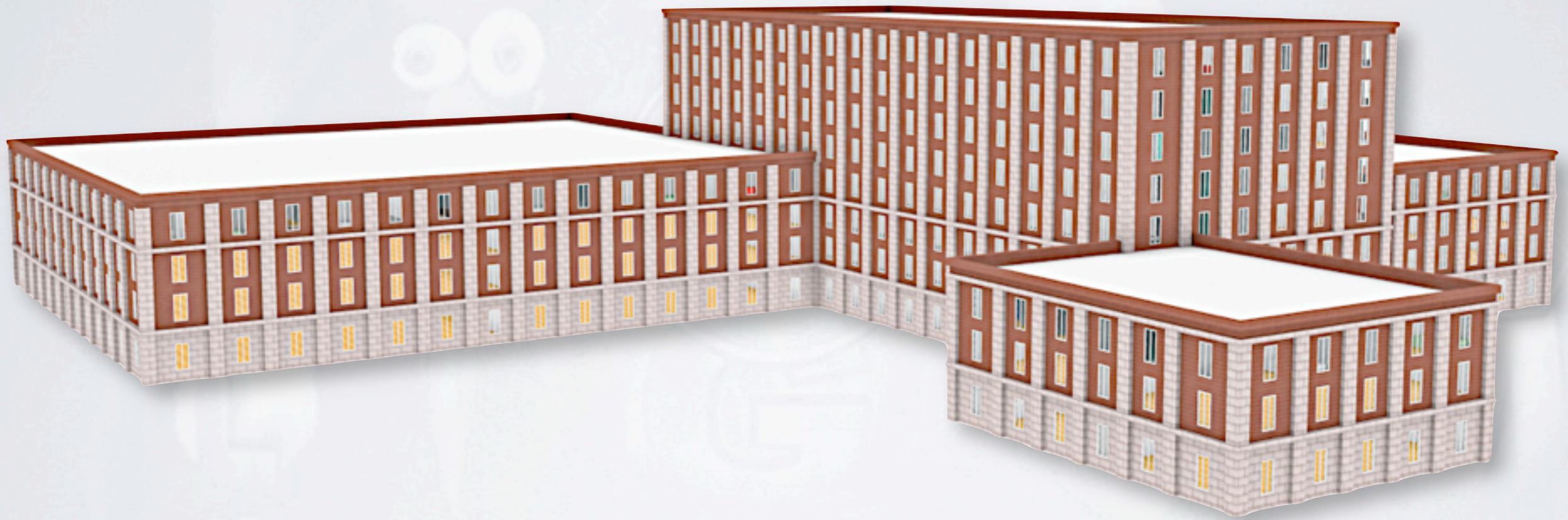




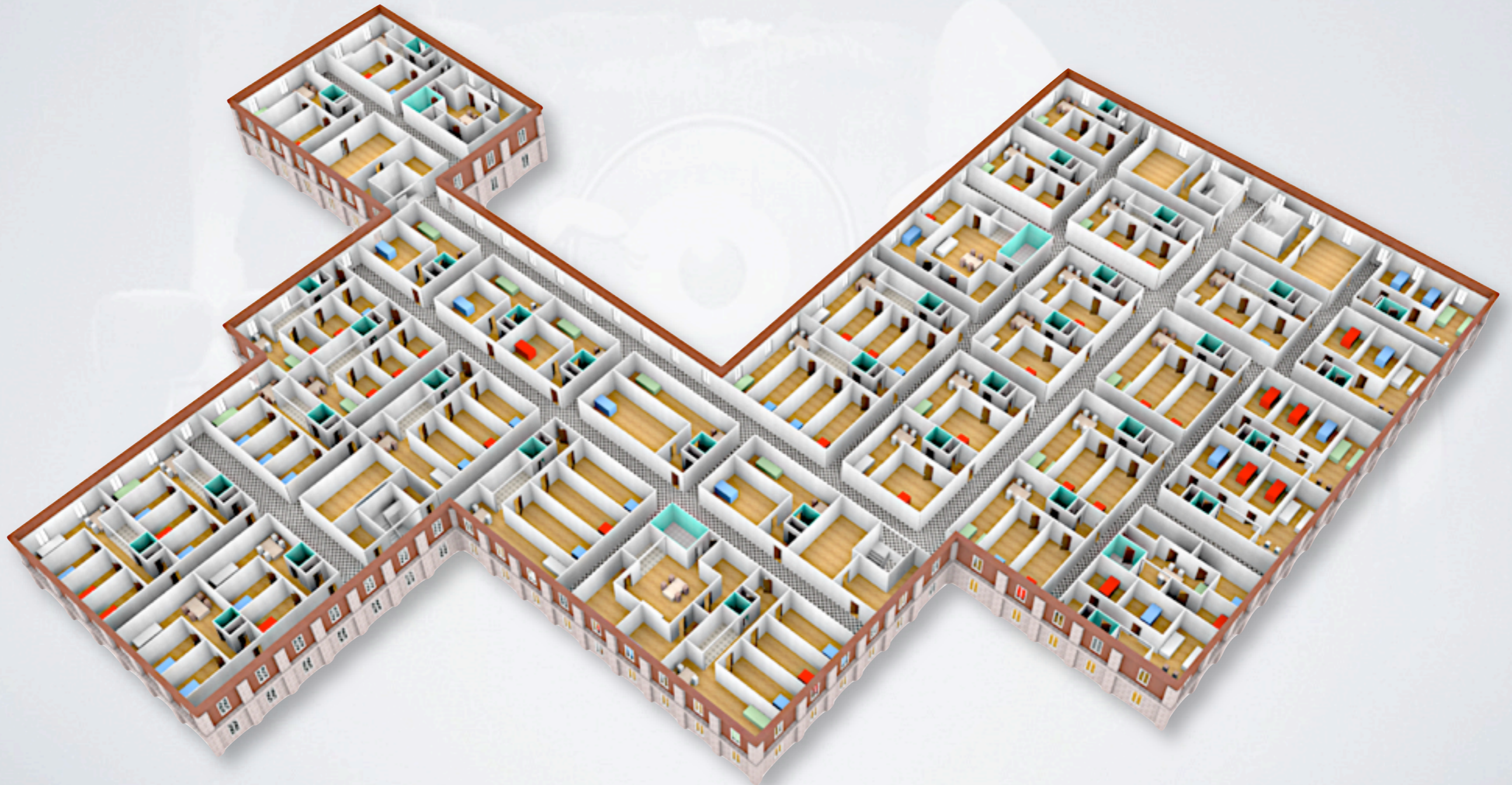








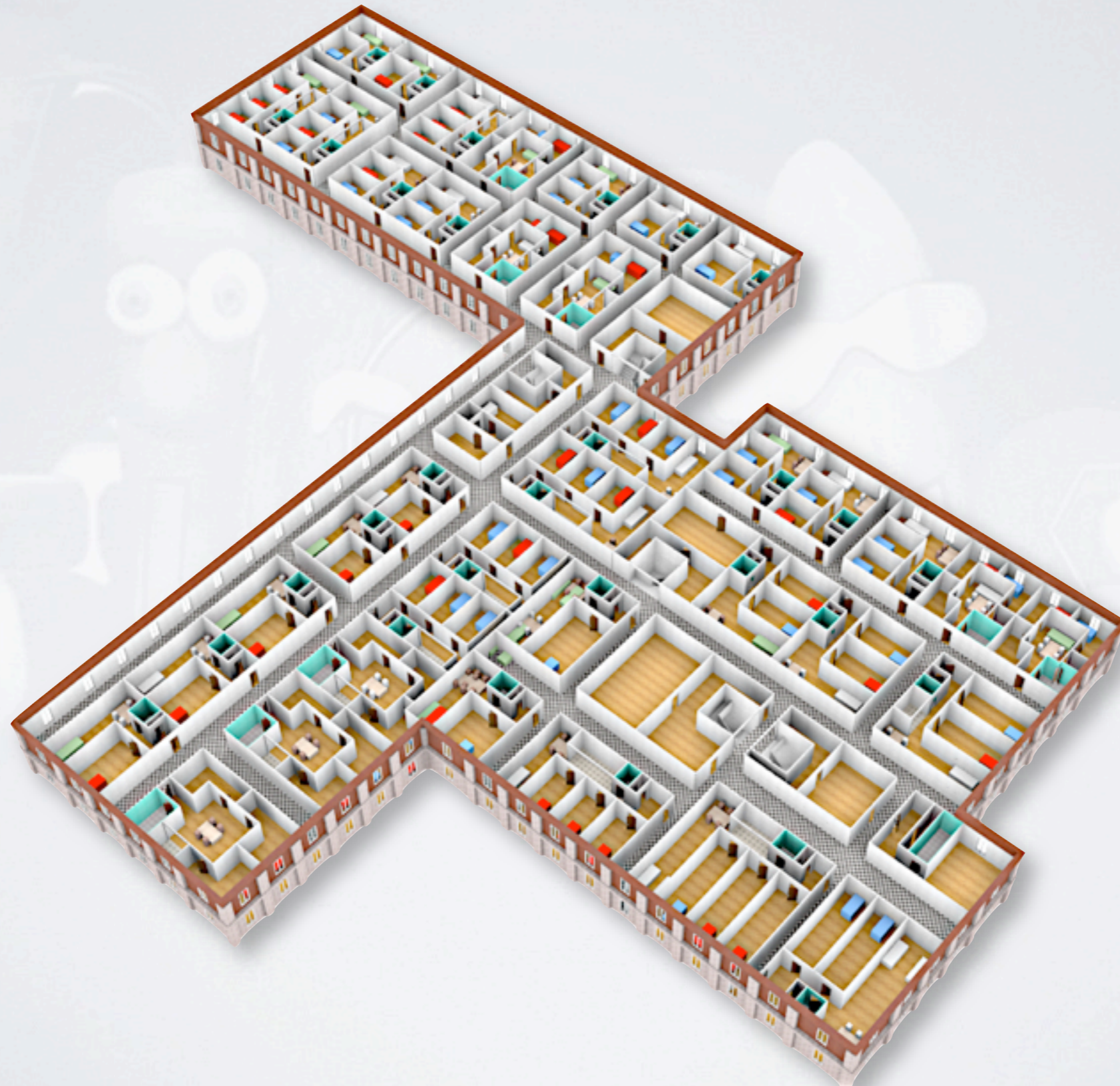
INTERIOR VARIATION



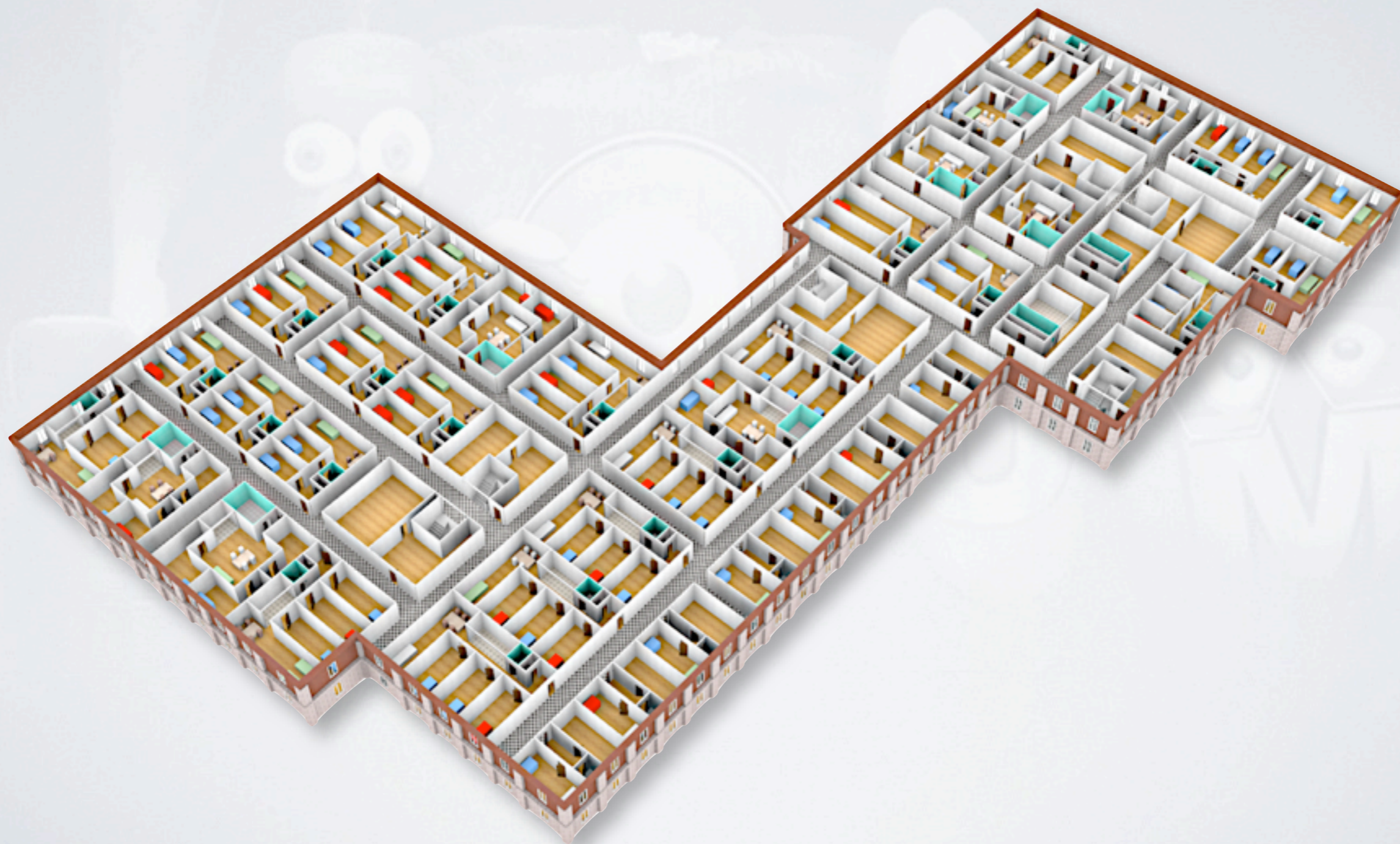
INTERIOR VARIATION

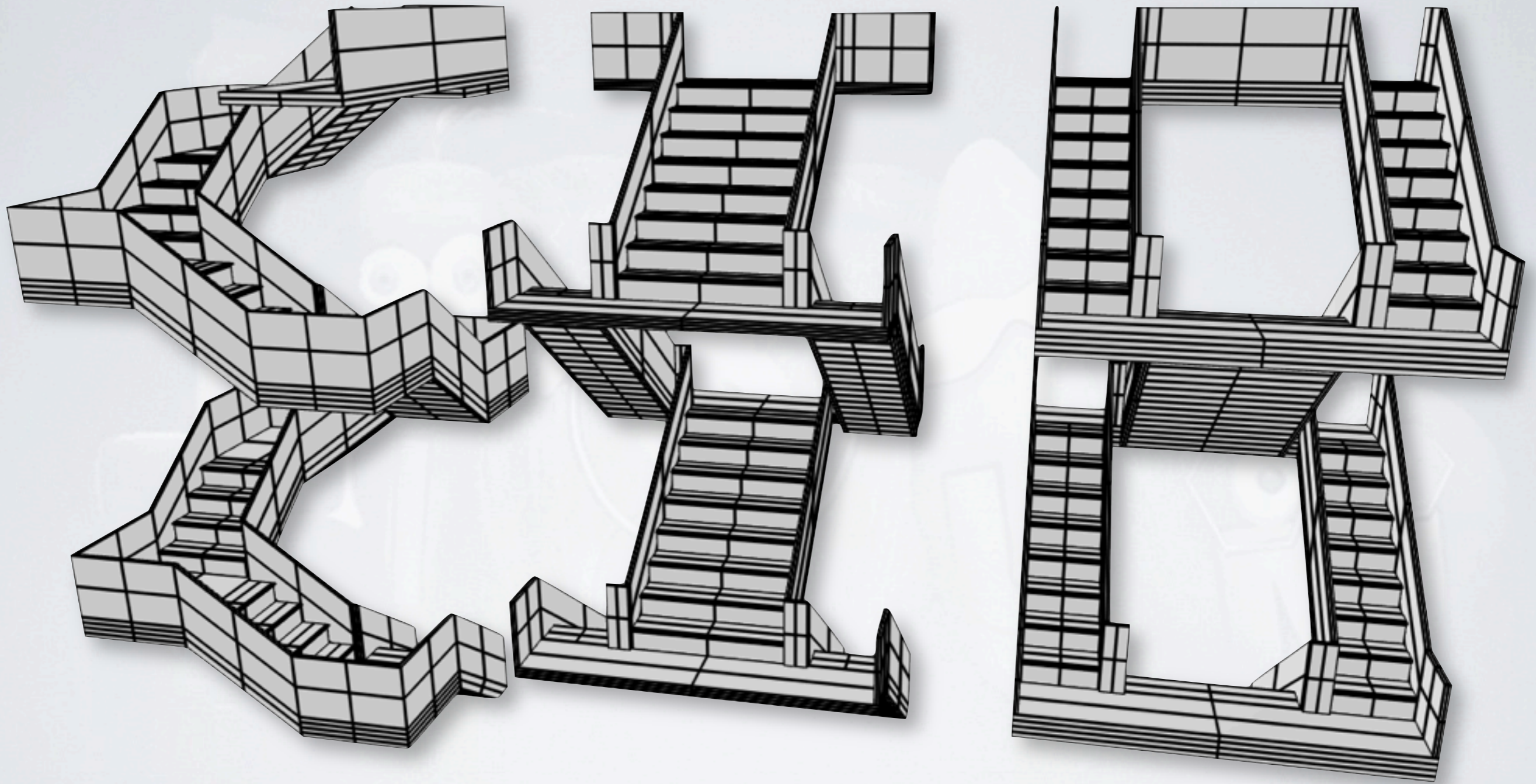


EXTERIOR VARIATION



EXTERIOR VARIATION





CONCLUSION

- Interiors and exteriors
- Programming environment
 - code reuse (libraries)
- Flexible and powerful
- Complex task

FUTURE WORK

- Optimization operation for high level space partitioning
- Higher level coding
 - interface
 - photos
- Lazy generation

QUESTIONS

- www.iro.umontreal.ca/labs/infographie/papers/Leblanc-2011-CMCB/
 - paper
 - videos
 - slides