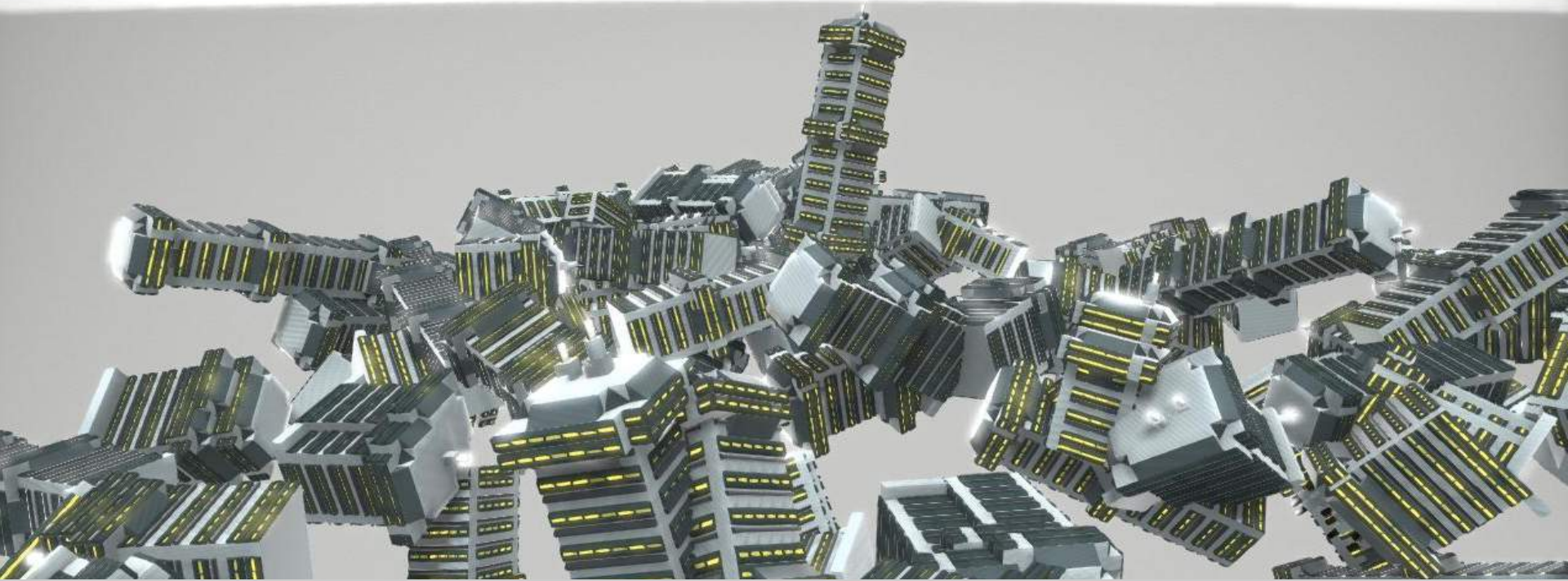
Procedural Cities



Kyle Cheng, Devin Gardella, Eli Goldstein, Matt LaRose, Tony Liu,
Diwas Timilsina, Kai Wang, Kelly Wang, David Yan

→ **Background**

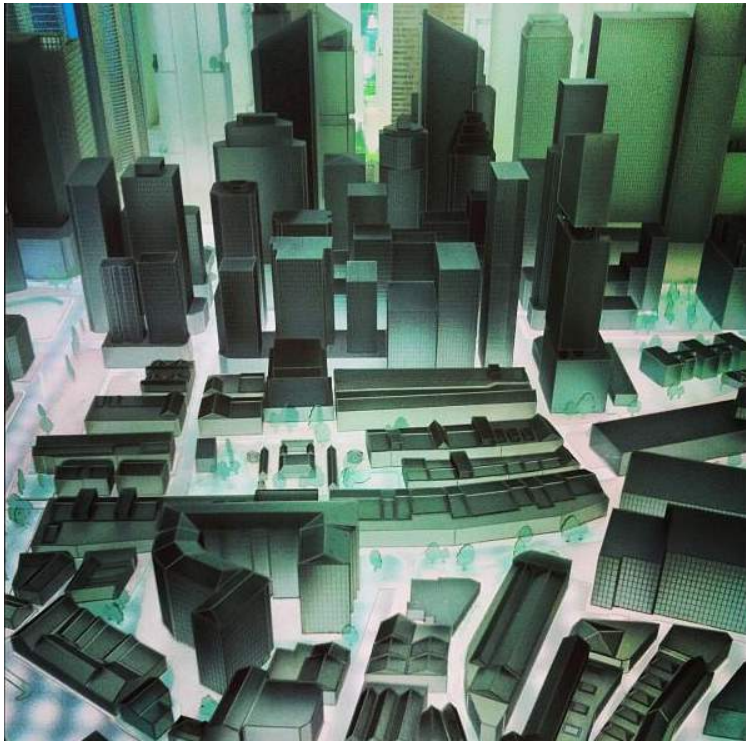
Building Grammars
- stack-based
- rectangle-based

City Layout

Film Development

Inspiration

Cyberpunk - dark, sharp, futuristic, sprawling



Background

Building Grammars

→ stack-based

rectangle-based

City Layout

Film Development

Stack-Based Building Grammar

used by Kowloon, EliBuilding, TonyBuilding

Grids of stacks of OBJs, where each OBJ stores its own information about rotation, translation, and scale

Floorplan

↓

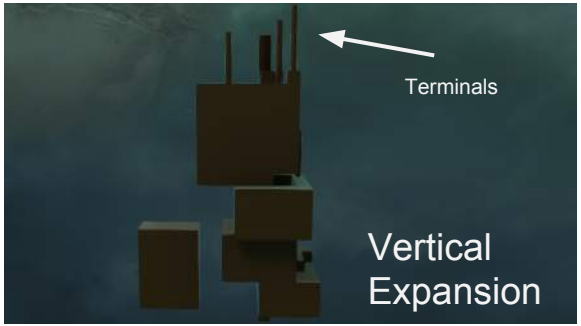
Axioms

↓

Expansion rules

↓

Termination



Basic Stack-Grammar Building Examples

Nonterminals

Rules

Terminals

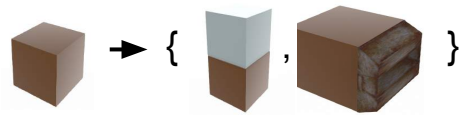
Horizontal (decorators)

Vertical (roofs, roof ornaments)

Basic Stack-Grammar Building Examples

Nonterminals

Rules



Terminals

With the following simple context:

```
if height > 1:
        stack.terminateVertical()
else:
        stack.terminateHoriz() and or
        stack.repeatUp()
```

We can get a building similar to the one on the right

Background

Building Grammars

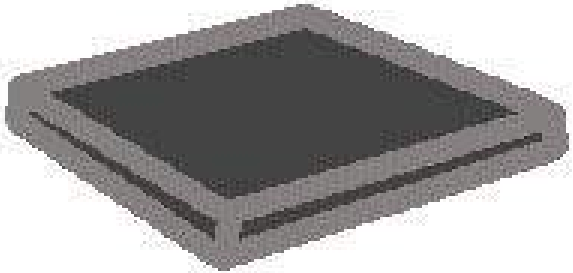
- stack-based
- rectangle-based

City Layout

Film Development

Rectangle-based Building Grammar

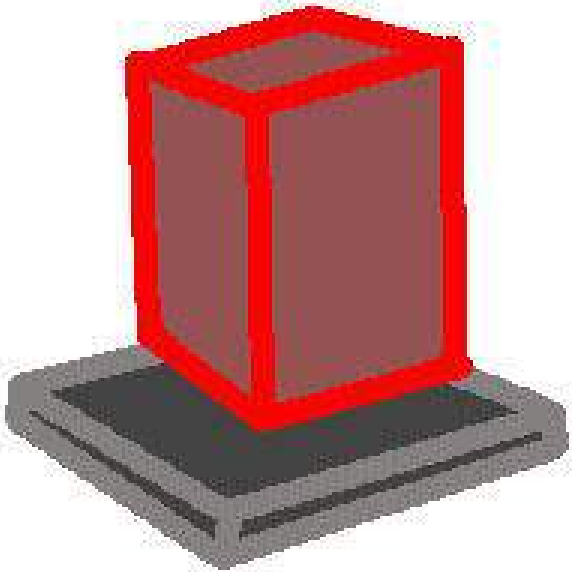
used by KyleBuilding



Inputs:

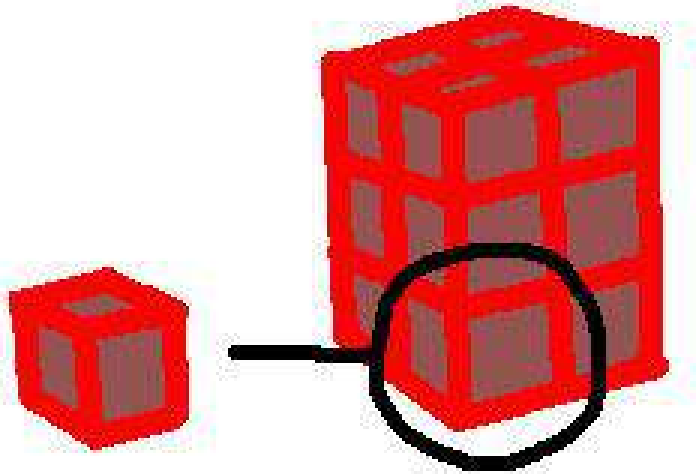
- width
- length
- height
- seed

Rectangle-based Building Grammar



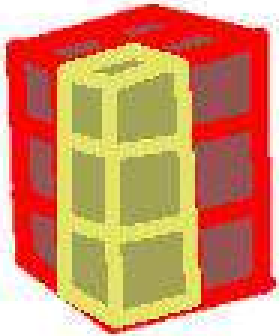
Build Rectangular Object

Rectangle-based Building Grammar



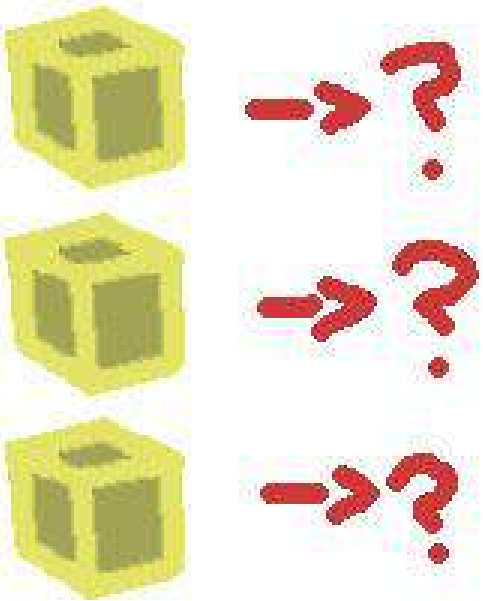
Rectangular object composed of 3d grid of cubes

Rectangle-based Building Grammar



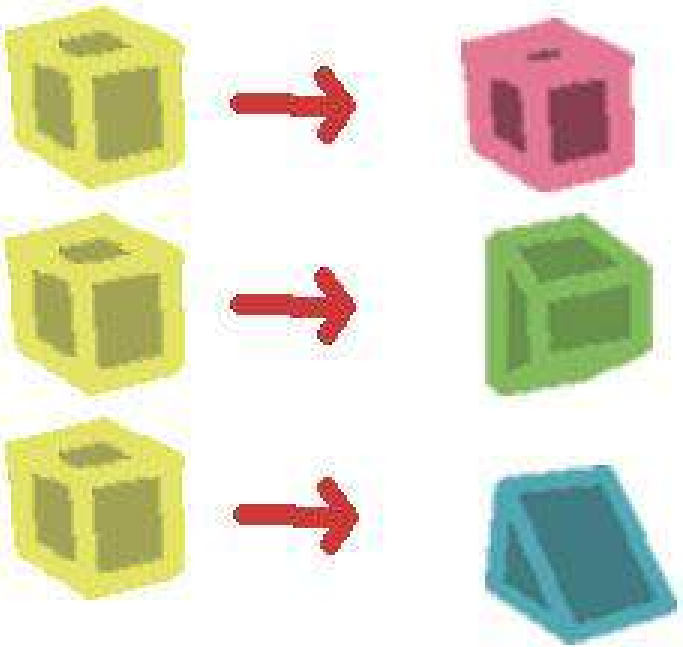
Define behavior for xz cube stacks

Rectangle-based Building Grammar



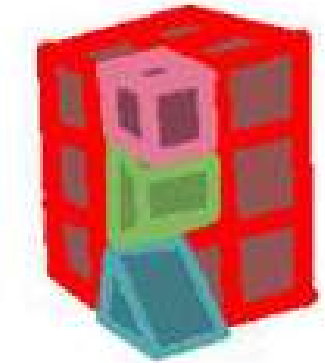
Define behavior for xz cube stacks

Rectangle-based Building Grammar



Example of cube stack expansion

Rectangle-based Building Grammar

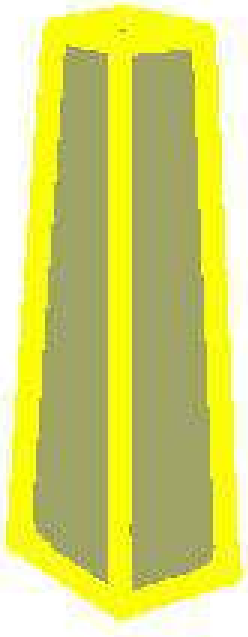


Expansion in context of rectangular object

Rectangle-based Building Grammar

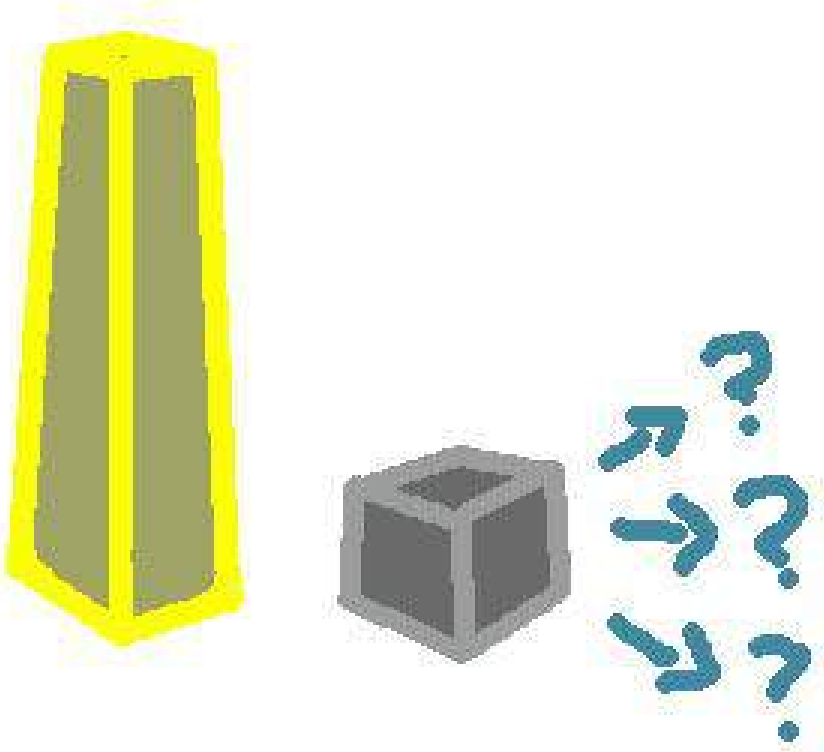
Buildings composed of multiple rectangular objects

Rectangle-based Building Grammar



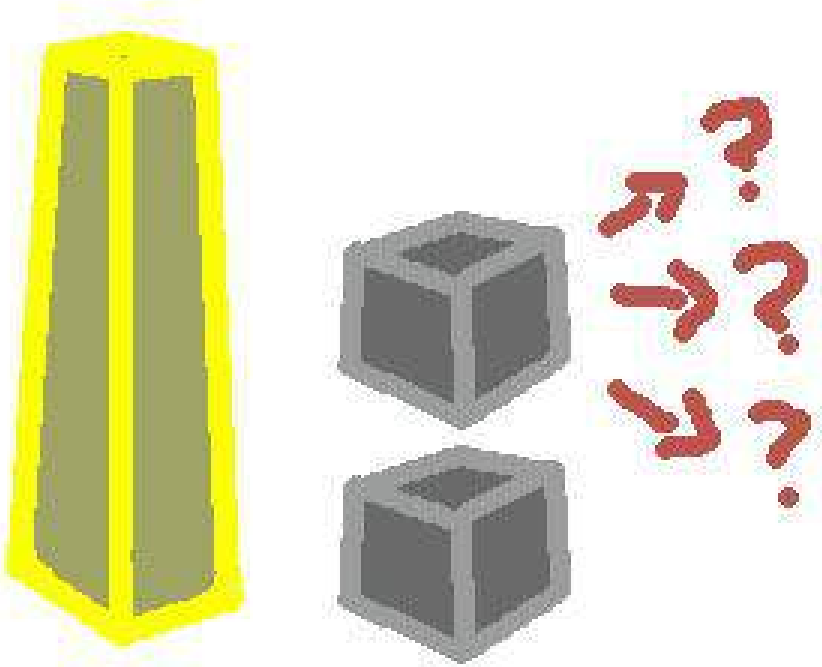
More Detail : Each colored object is several unique stacked rectangular objects

Rectangle-based Building Grammar

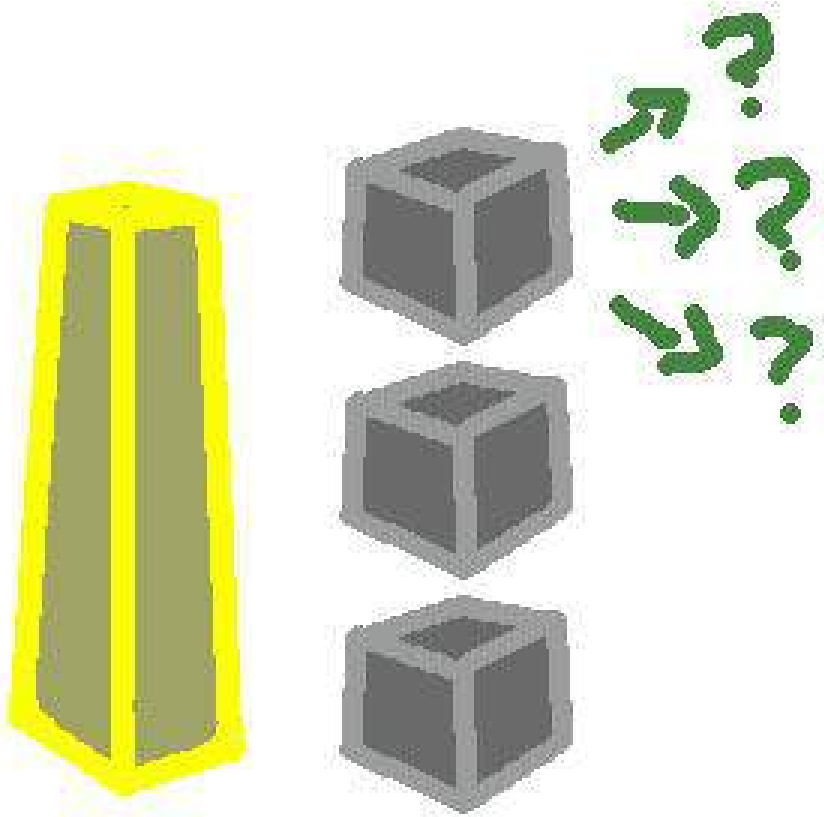


Choose from set of rectangular objects

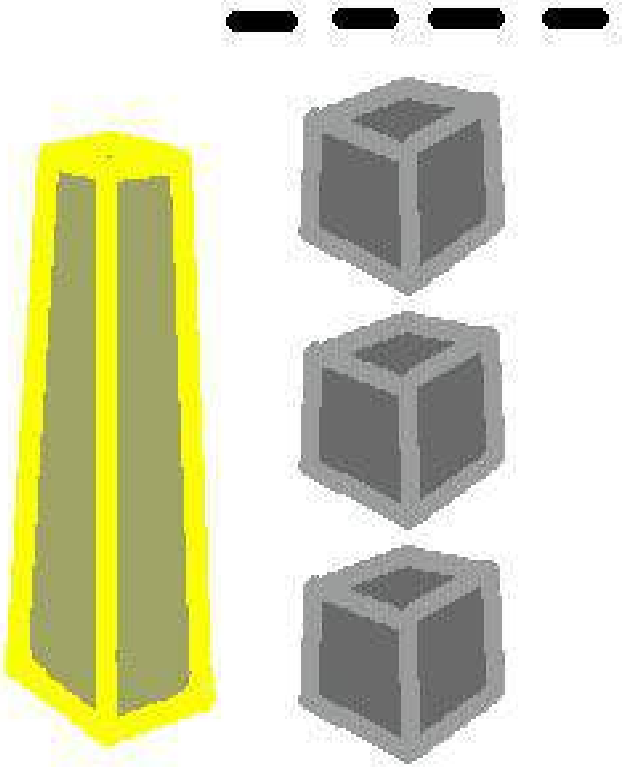
Rectangle-based Building Grammar



Choose next Expander, repeat

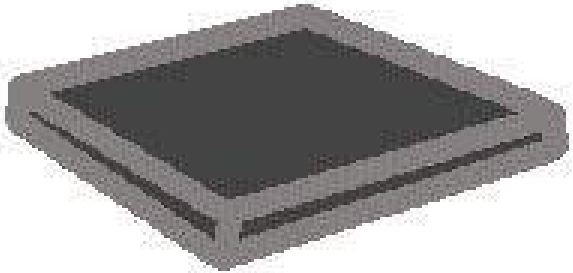


Choose next Expander, repeat



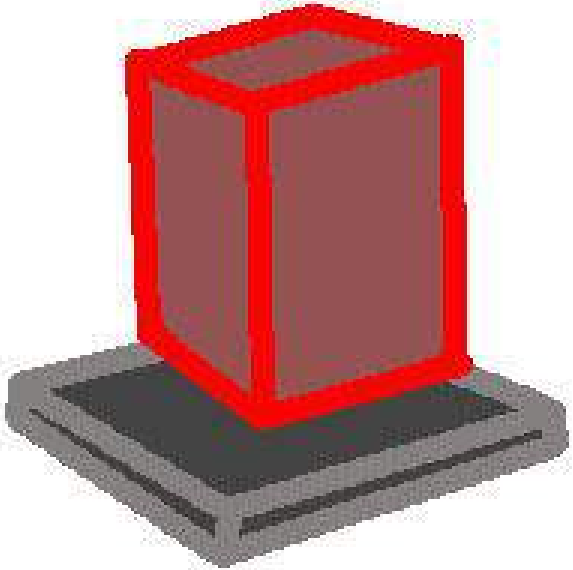
Stop iteration at *max_height*

Rectangle-based Building Grammar

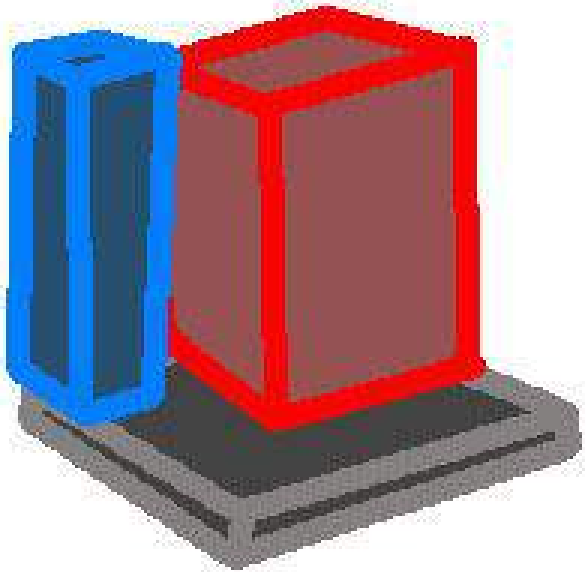


Expand out several colored objects

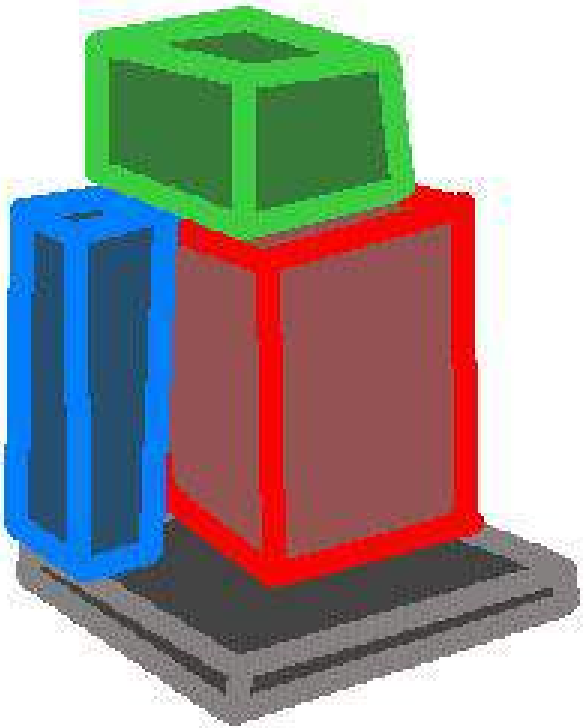
Rectangle-based Building Grammar



Rectangle-based Building Grammar

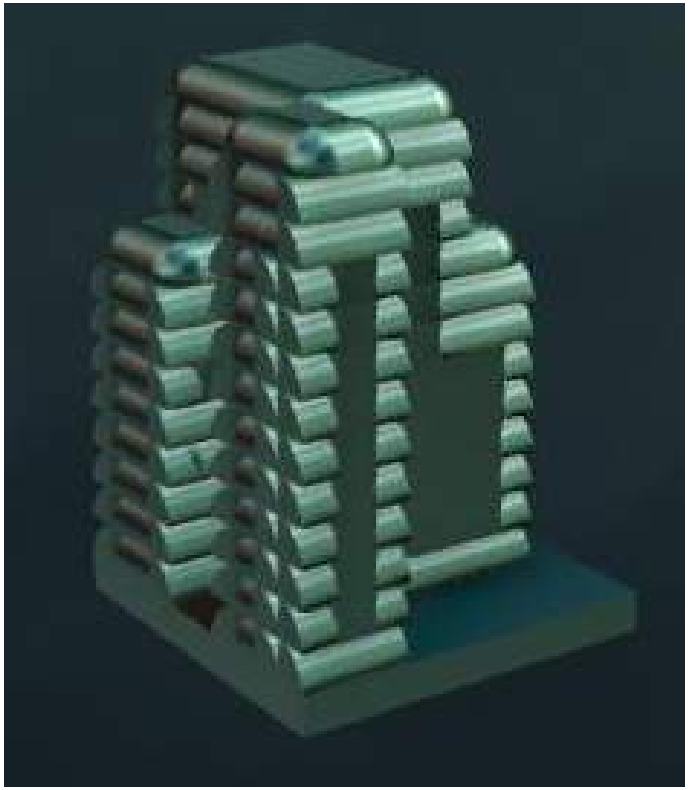


Rectangle-based Building Grammar

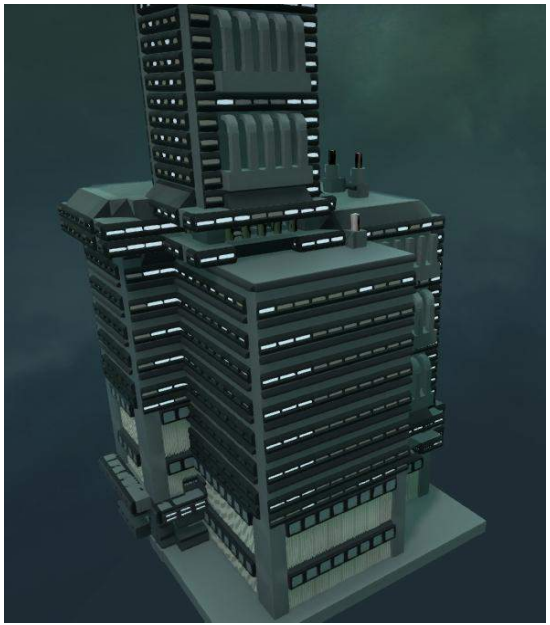


Rectangle-based Building Grammar

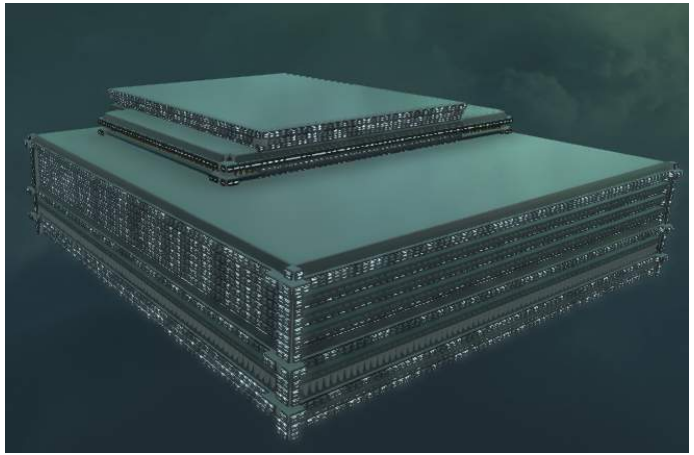
Rendered Result

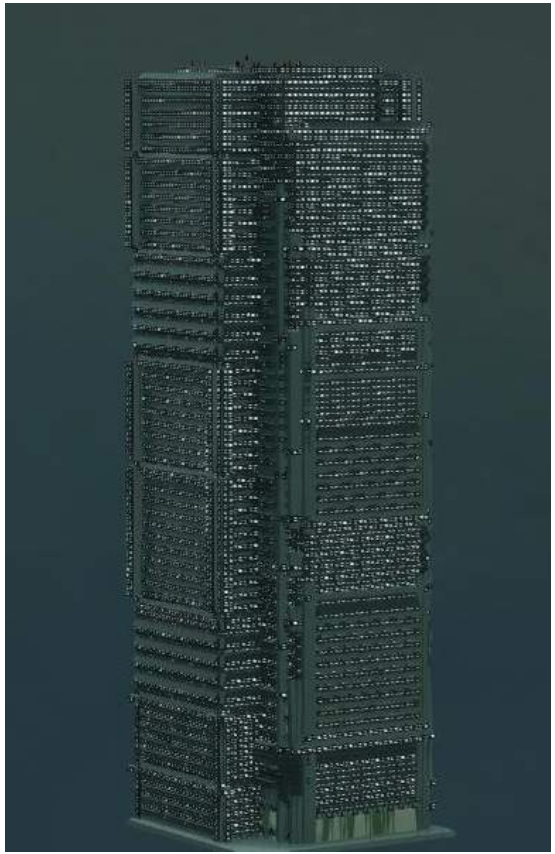


Grammar Elements and Well-Defined Rec Behavior



Scalability





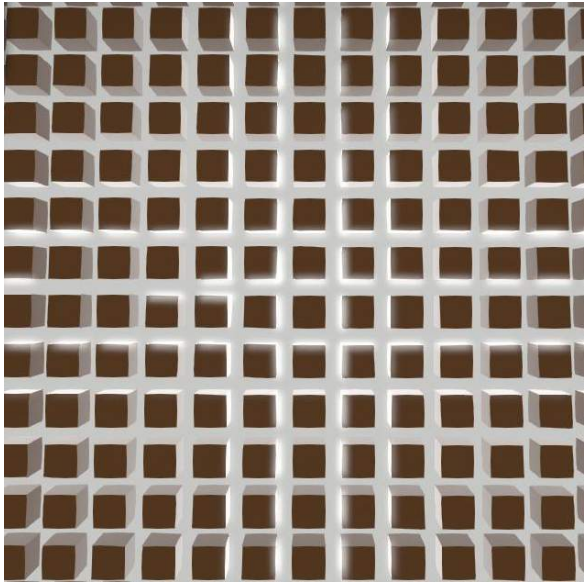
Background

Building Grammars
- stack-based
- rectangle-based

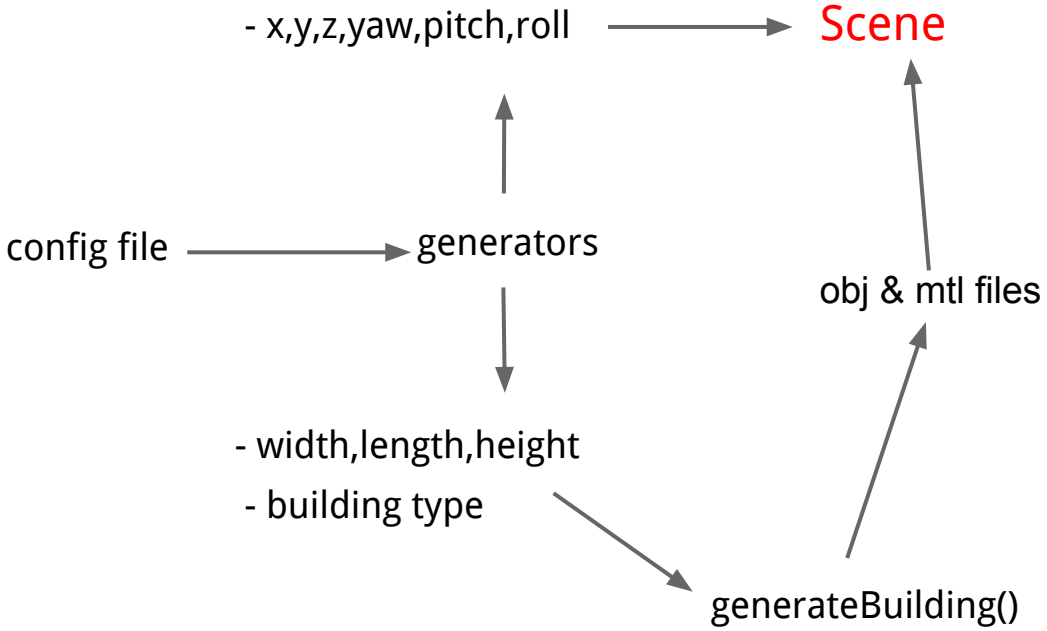
→ **City Layout**

Film Development

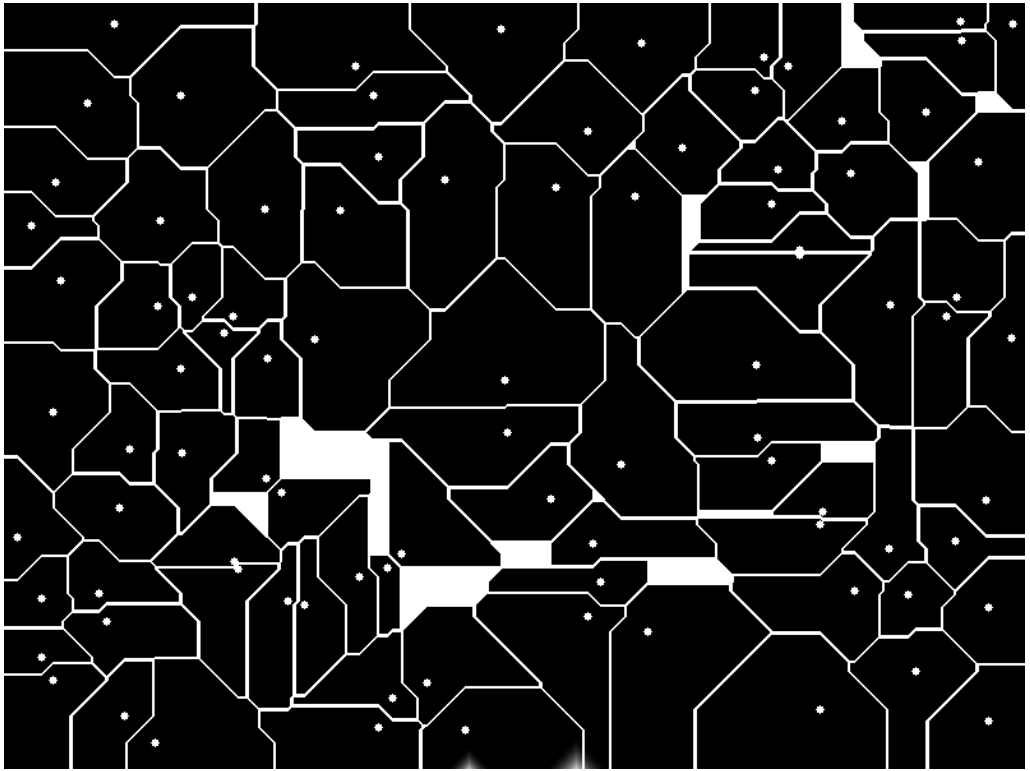
City Layout



Simple Grid Generator



Voronoi Diagram



Randomly Generated Diagram

- Region defined by the closest point

- In terms of Manhattan distance:
$(x - x_0) + (y - y_0)$

Tuning the Diagram

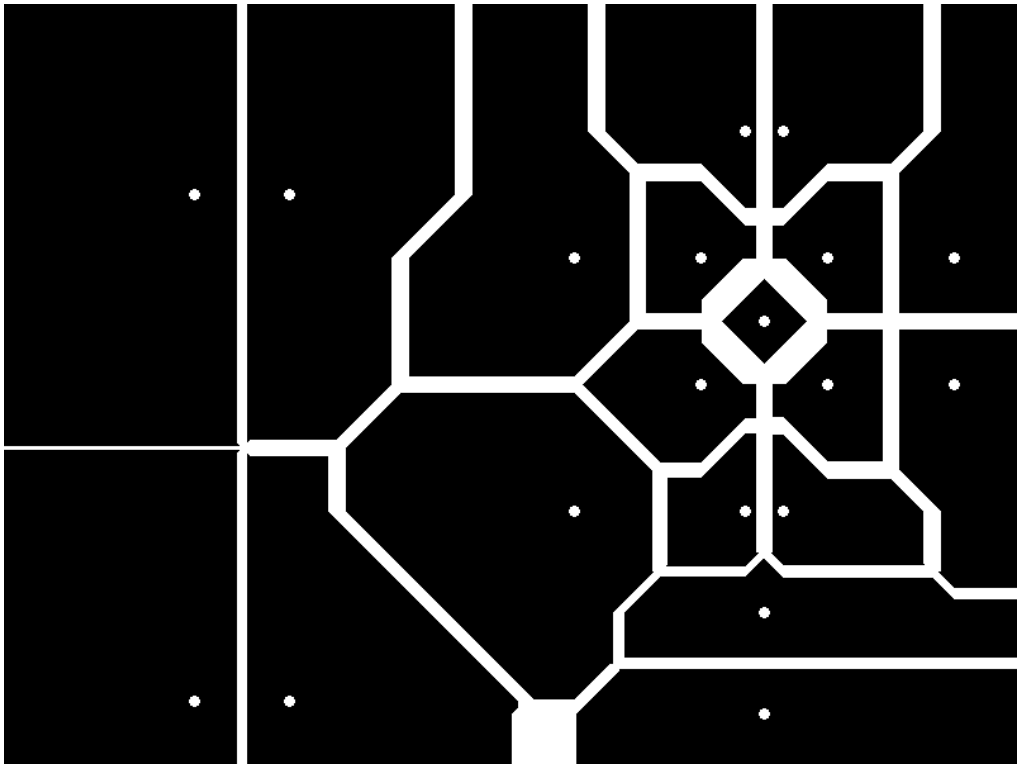


Diagram Used in the Scene

- Manually place points

Tuning the Diagram

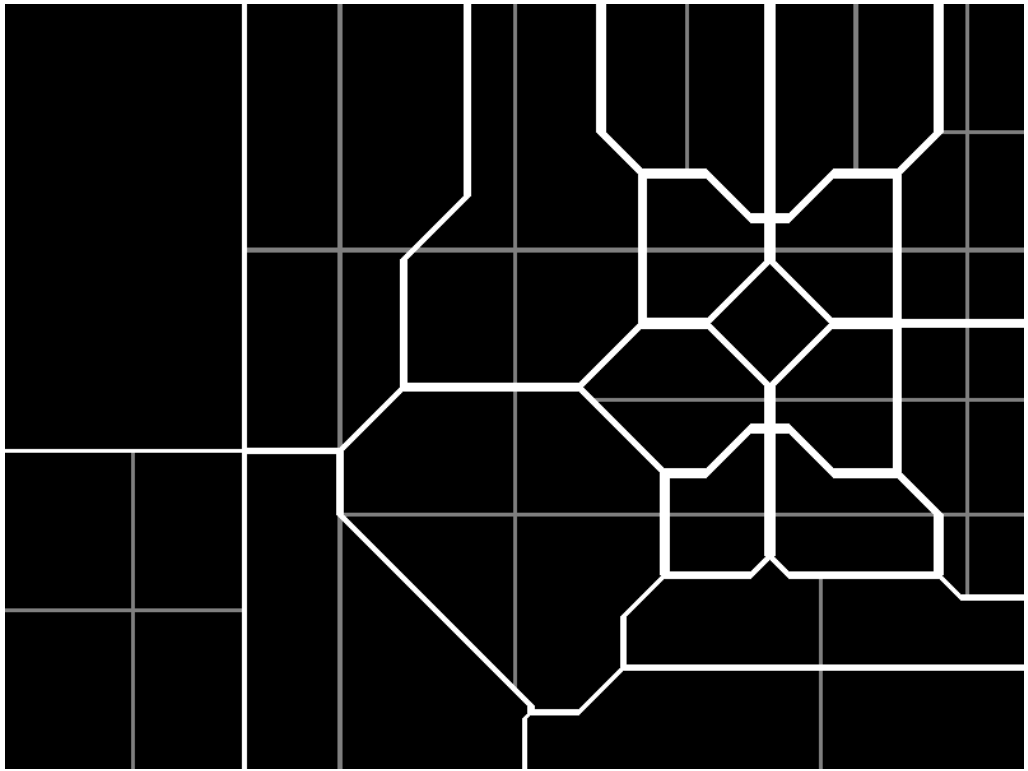


Diagram Used in the Scene

- Adding Sub-Districts

- Second-order Voronoi

Building Placement

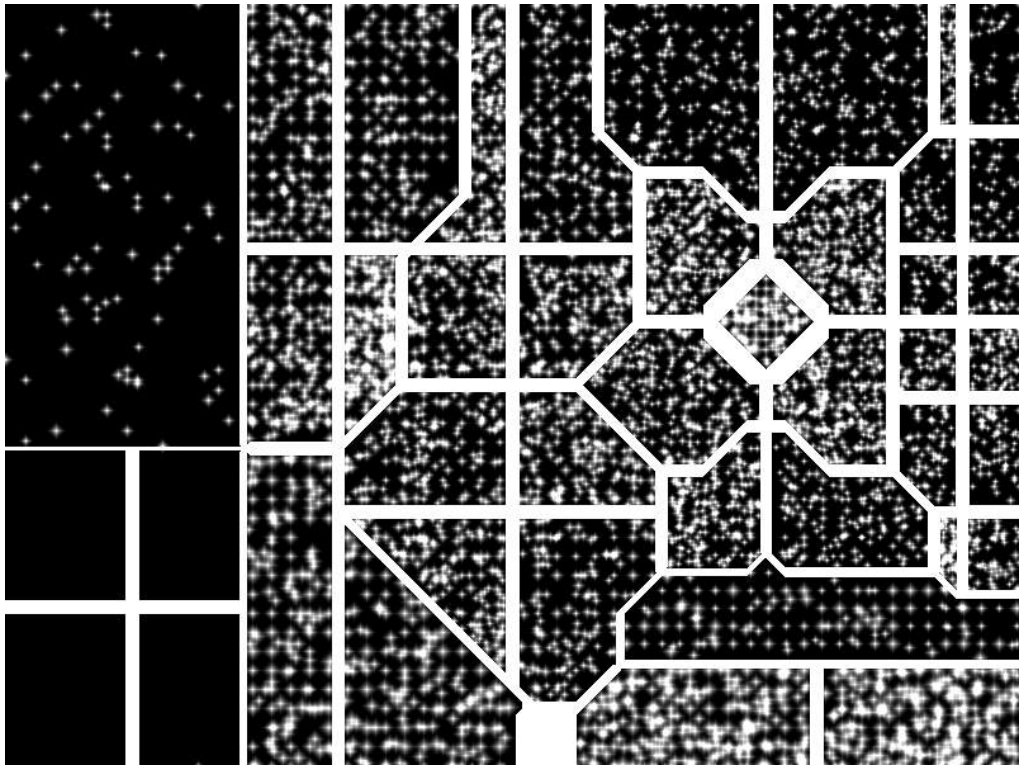


Diagram with Buildings

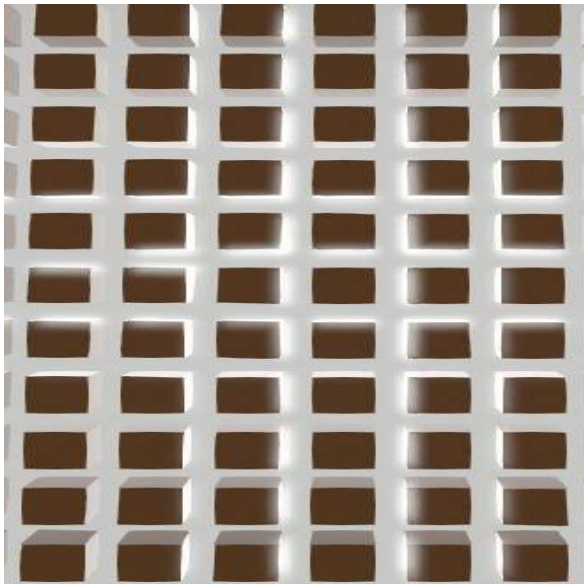
- Customize each district (.vor)

- Road size
- Building density
- Probability for each type of building
- Tidiness
- Grid size
- Height, width and length function

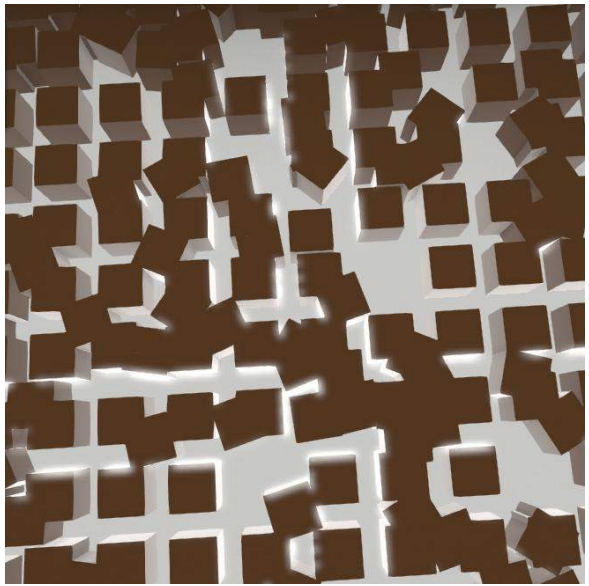
- Global parameters (.cfg)

- Scale
- Overall density
- Overall road size
- Cube or actual building

Tidiness



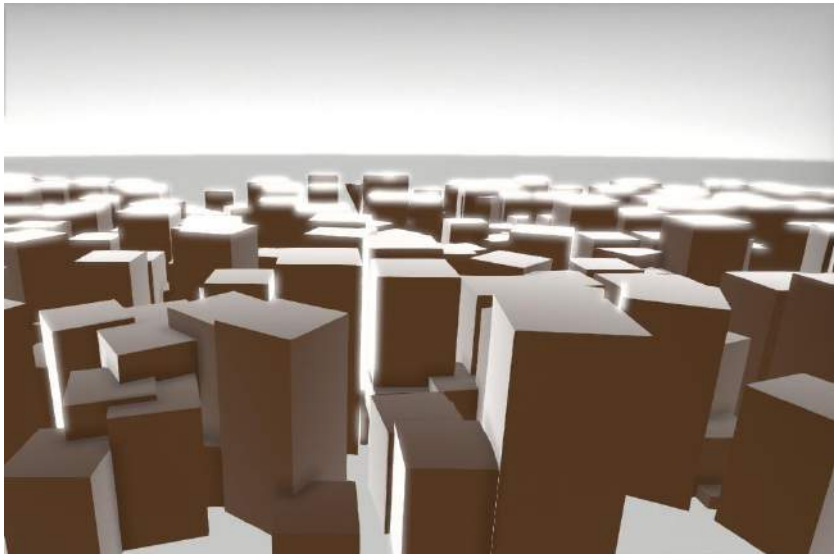
Tidiness = 1.0
Grid Generator



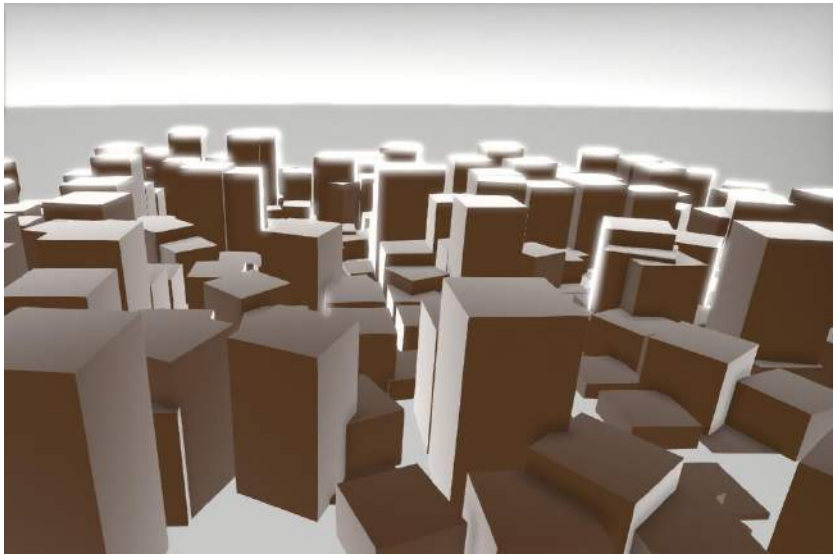
Tidiness = 0.7
Default Setting

Tidiness = 0.0
Random Placement

Height function

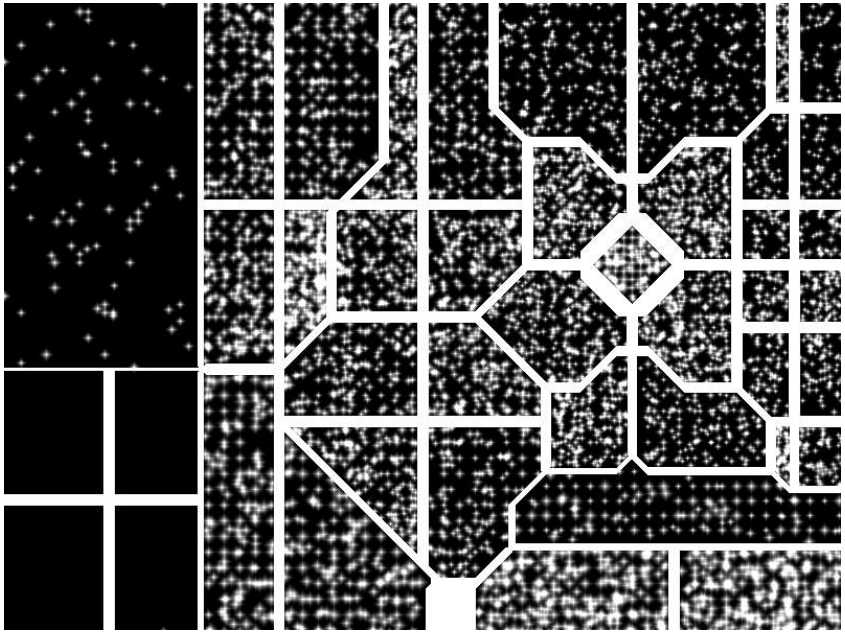


Linear height function
h = a + bx
Uniform height
distribution

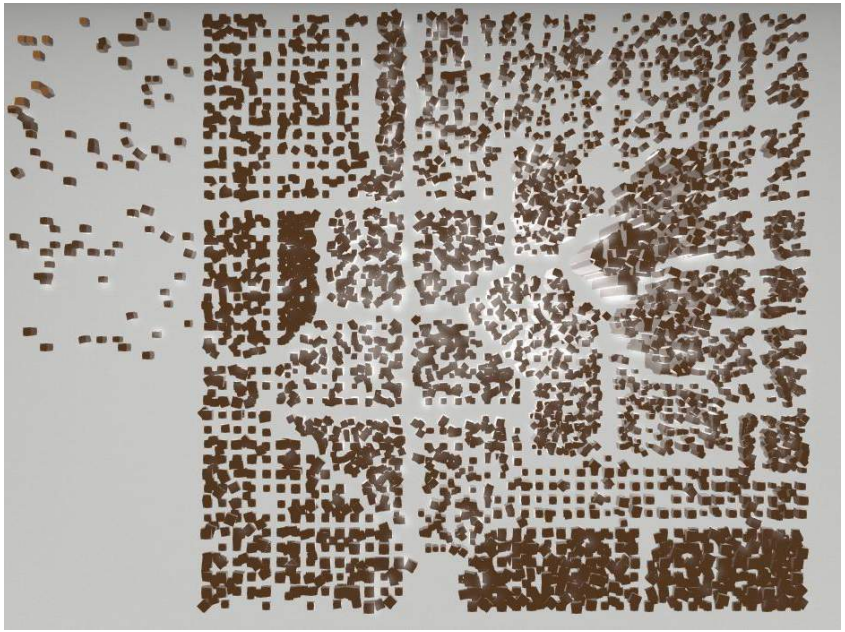


Cubic height function
h = a + bx^3
More shorter buildings

Finished Layout (~4500 buildings)

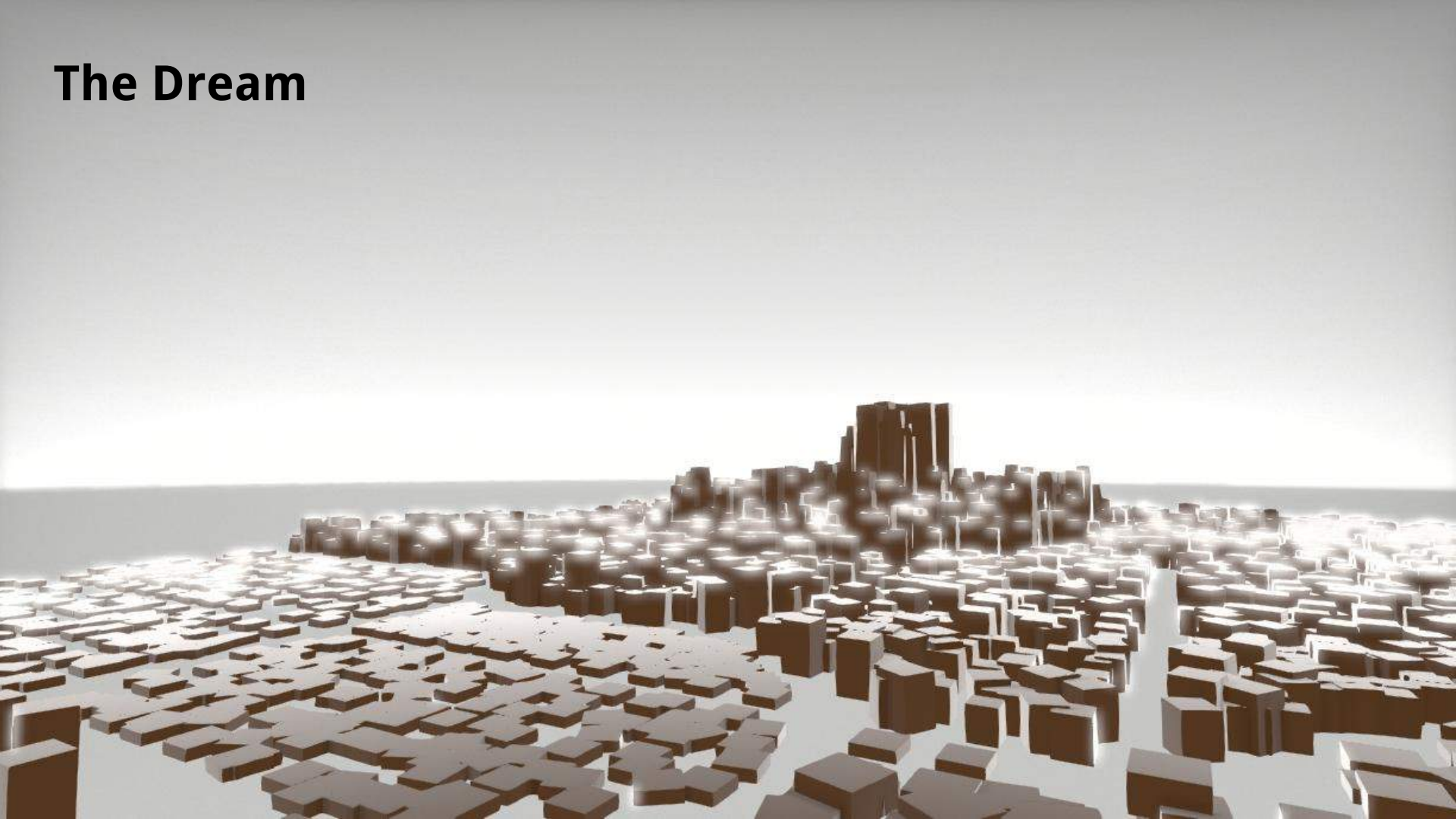


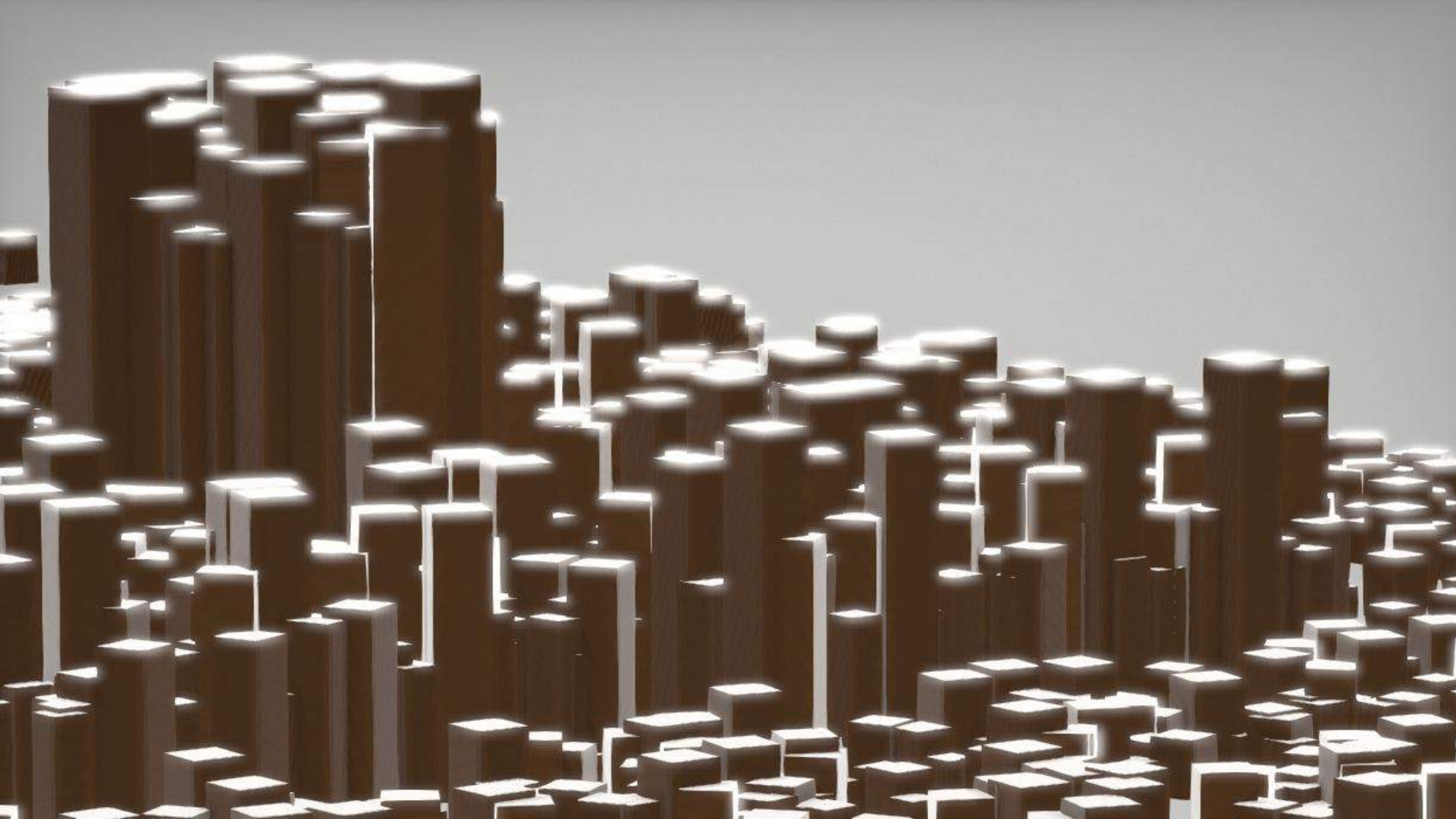
2D Layout



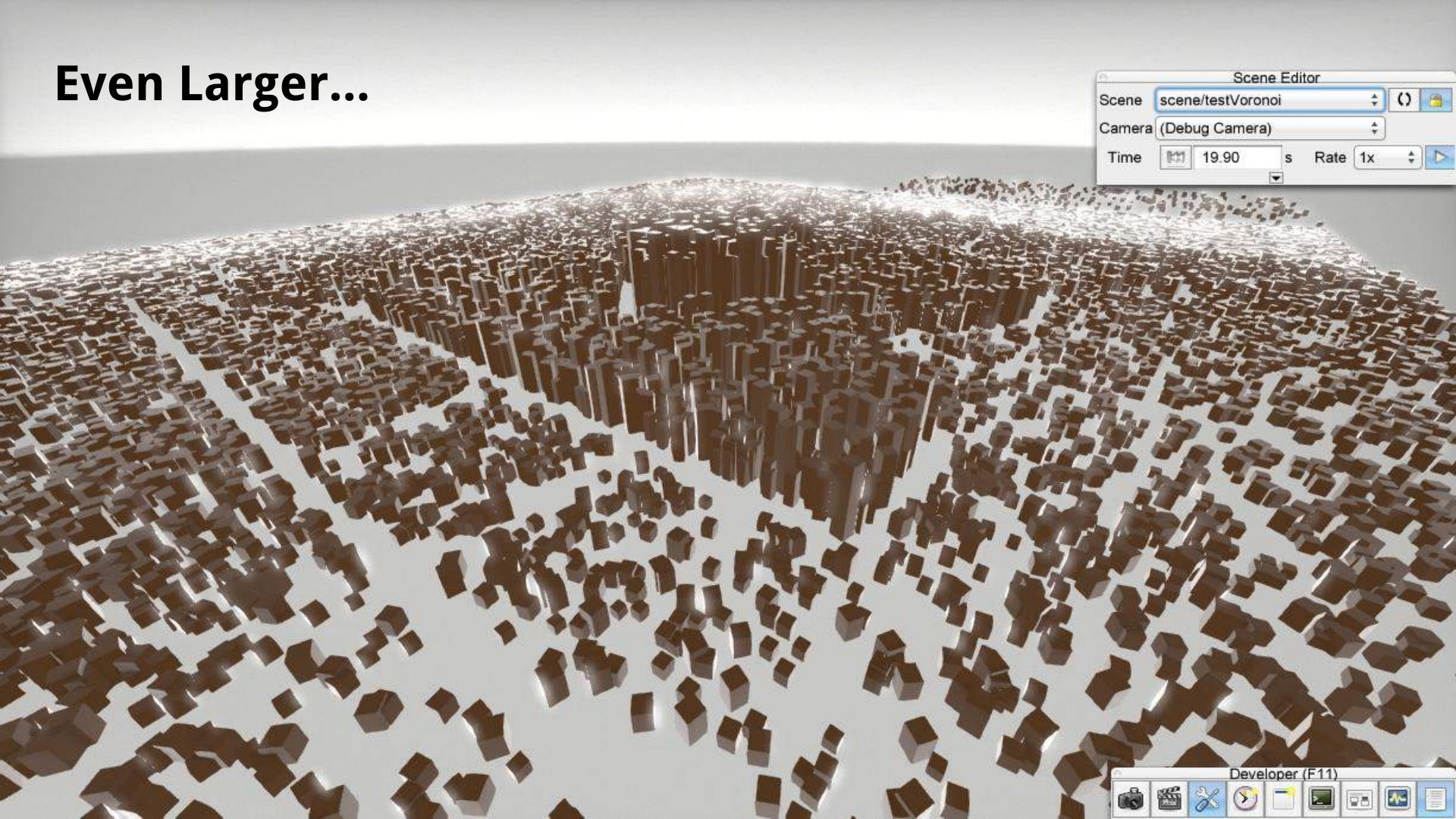
3D View from top

The Dream

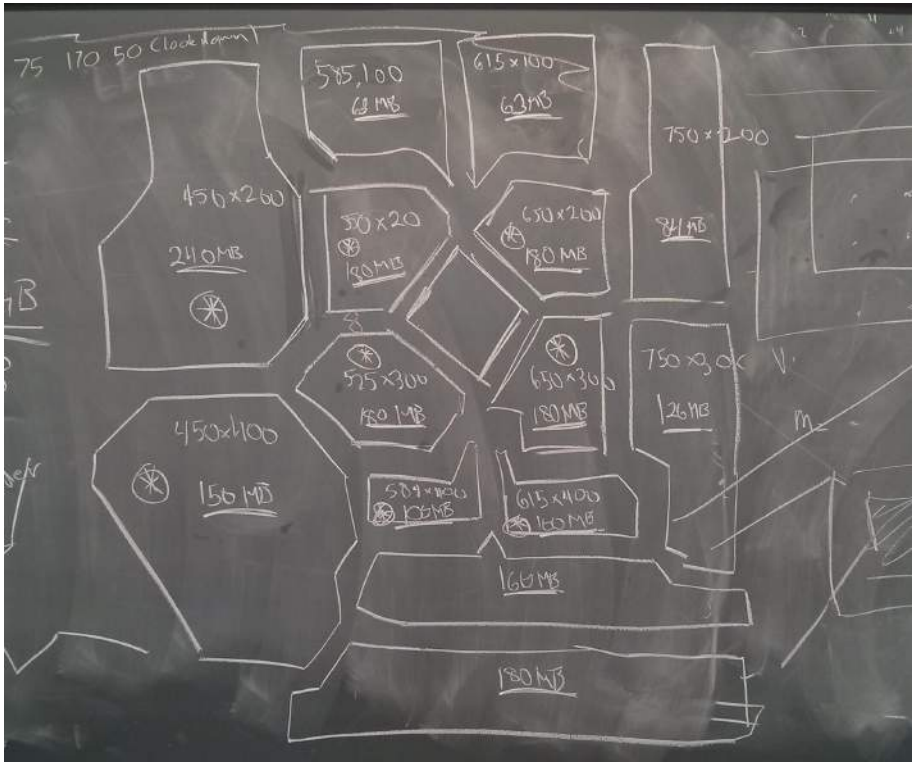




Even Larger...


Scene Editor
Scene
scene/testVoronoi
Camera
(Debug Camera)
Time
19.90
s
Rate
1x
Developer (F11)

Memory Issues



Planning Memory Usage

High memory consumption

- 200 Buildings = ~2 GB

Optimization

- Reduce number of triangles
- Reduce precision
- Better/more computers
- Less buildings

Background

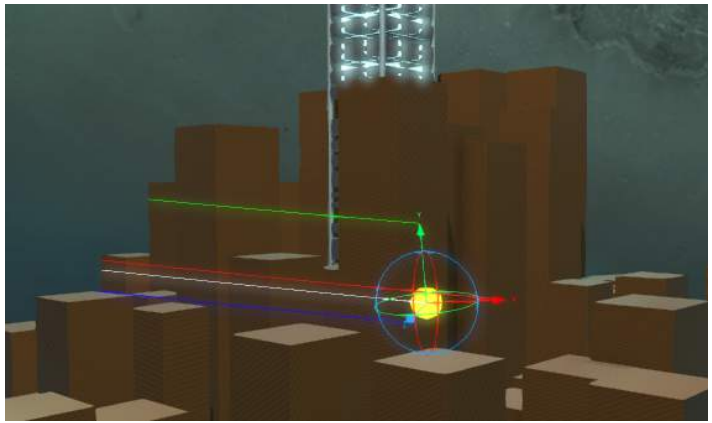
Building Grammars
- stack-based
- rectangle-based

City Layout

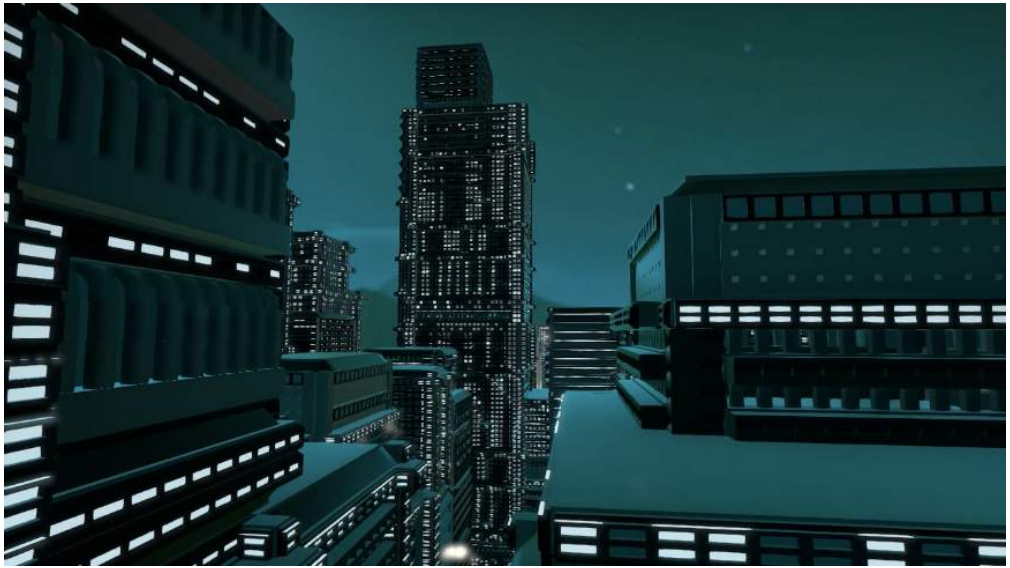
→ **Film Development**

Creating a Compelling Film

- Inspiration from “the Timeless” and city tourist videos
- Hand placed splines in a dummy city
- Replaced cubes with actual buildings



First set camera splines in a demo city



A screenshot from the final film

Background

Building Grammars
- stack-based
- rectangle-based

City Layout

Film Development

→ **Thank you**