Procedural City Generation

Karl Volkenandt, Jakob Krause

Freie Universität Berlin

September 7, 2018

Motivation

Problem:

Creating high quality virtual worlds by hand has the following drawbacks:

- expensive
- cumbersome
- time consuming

Solution:

Smart Algorithms that do the work for us...

Requirements for procedural worlds

- believable
  - no obvious patterns
  - Close to the original

- attributes to parameterize
  - size
  - terrain features, snow, population …
  - architecture style

- variational

- Acceptable computation time
  - real-time in games

- editable output format

Prominent examples

- ## The Elder Scrolls 2: Daggerfall (1996) :487.000 km^2

    - very generic

- ## No Man's Sky (2018)

    - procedural solar system and creatures

    - 18 quintillion planets

    - Algorithm is kept a secret
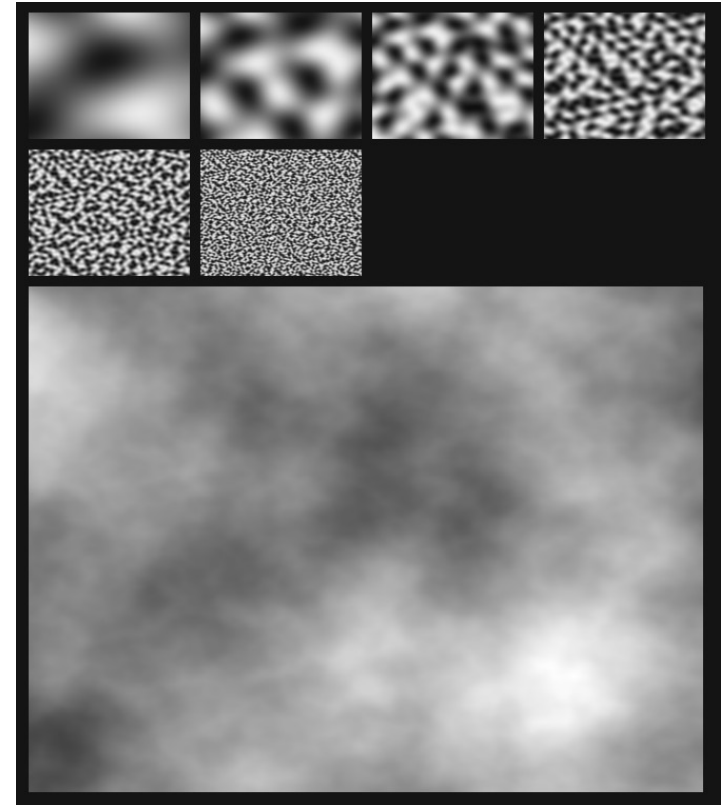
City Generation

Challenges:

- realistic road systems

- public transportation

- generation of buildings

- building type distribution
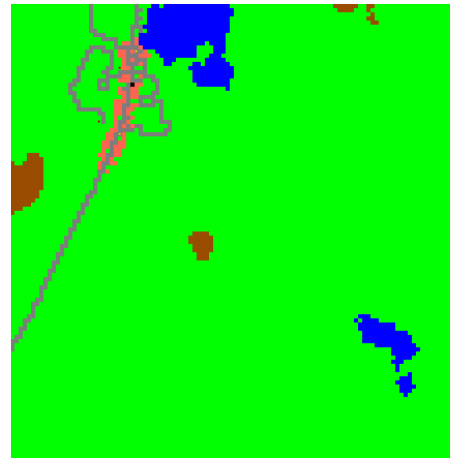
Generation

The key is to find the right abstraction:

- L-Systems
  - alphabet, rules, constants, start state
  - works well for plants, roads

- Perlin Noise
  - layered by different frequencies
  - typically used for generating terrain

- Voronoi cells

- mapping basic street template to terrain maps

Agent-based generation

- different agents follow different goals

  - street building

  - house building

  - planning authorities

- starts in an empty terrain

- heat maps for evaluate space

- evolution can be observed

- complex
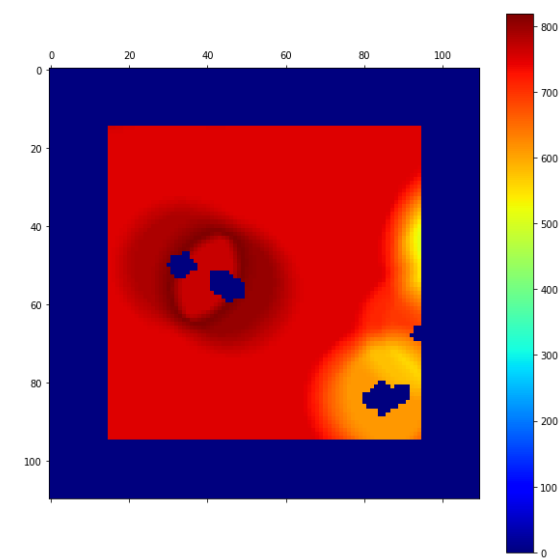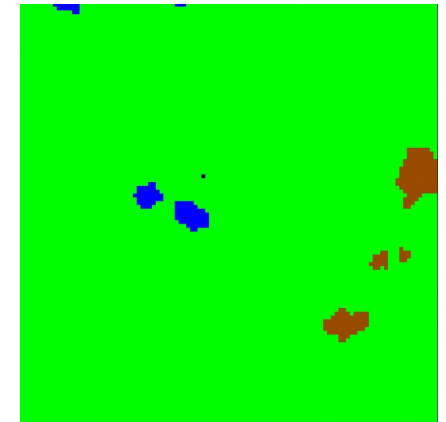
- tile based graphic

- renderable format

  - "SSS
    HSH
    TSS"

- 1. street system based on voronoi cells

  - Every street is connected to each other

- 2. street system based on agents

- Generation pipeline: (terrain) → streets → house

- 4 Tile types (nature, street, house, tower) with randomized graphics

- higher population towards the center of the city or pole points

Demonstration ...

Improvements
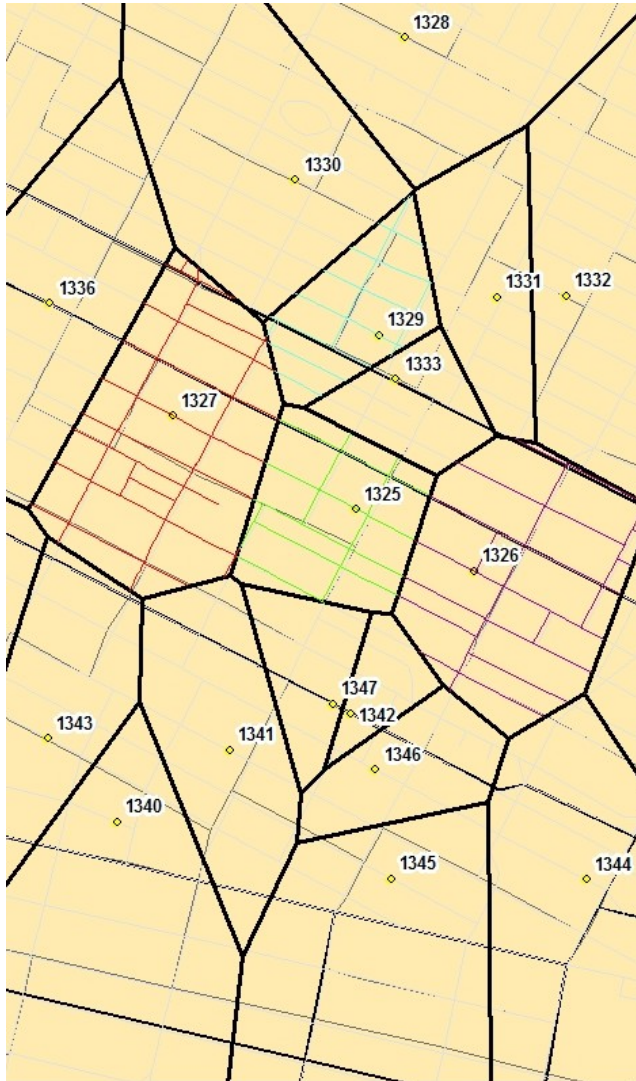
- L-Systems for district streets

- Connected street tiles

- Industrial tiles

- Special buildings

- ….

Questions ?