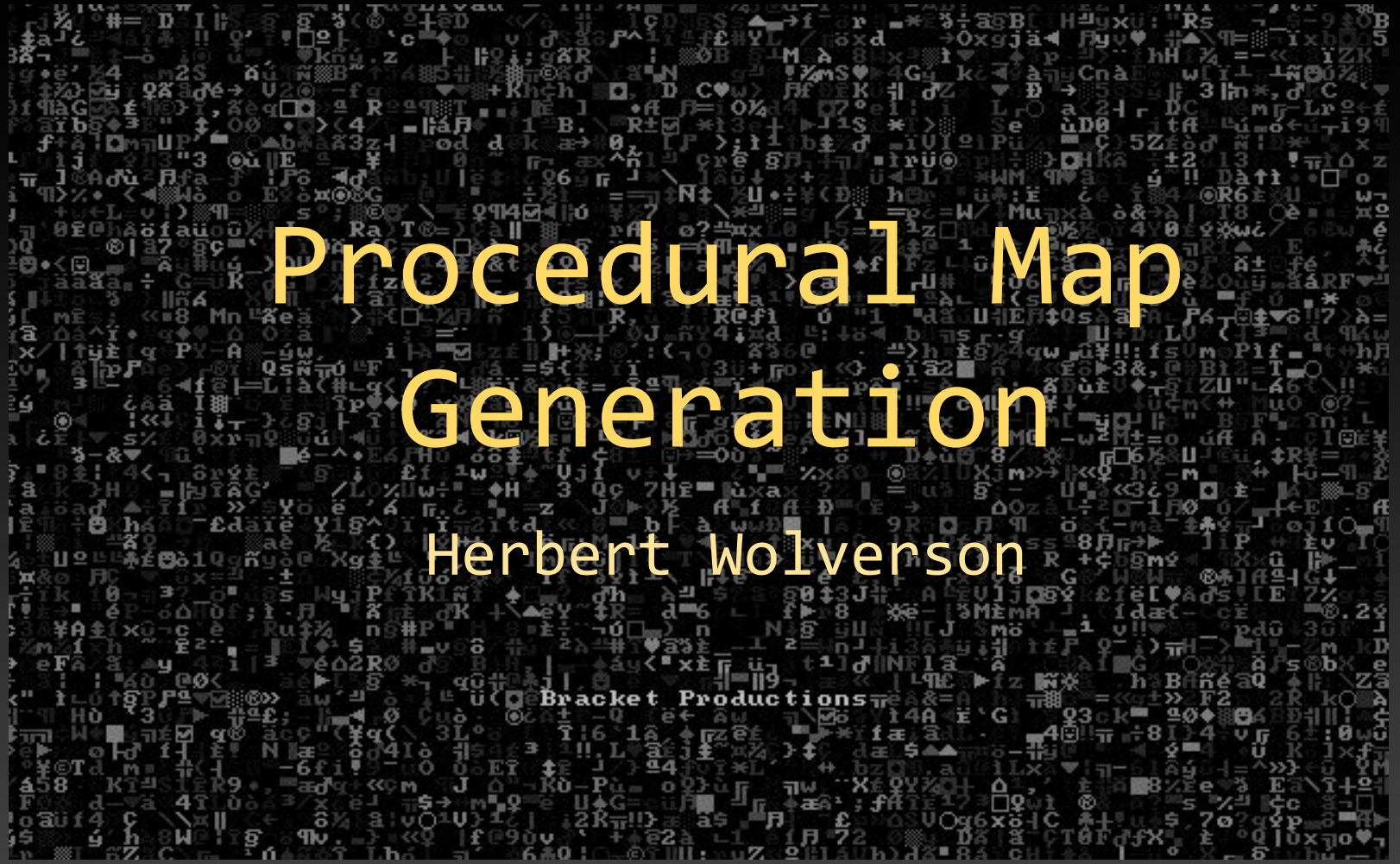


Roguelike Celebration 2020

Who is Herbert Wolverson?

Author of the *Rust Roguelike Tutorial*
http://bfnightly.bracketproductions.com/rustbook/

Talks too much on Twitter: @herberticus
Writes far too much on Reddit: /u/thebracket

Upcoming book: *Hands-on Rust: Effective Learning through 2D Game Development and Play*
Look for it soon on https://PragProg.com/

Source code for this talk:
https://github.com/thebracket/roguelike-celebration-2020

Rogue (1980)

We wouldn’t be here without *Rogue*.

Generates up to 9 rooms, connects them randomly.

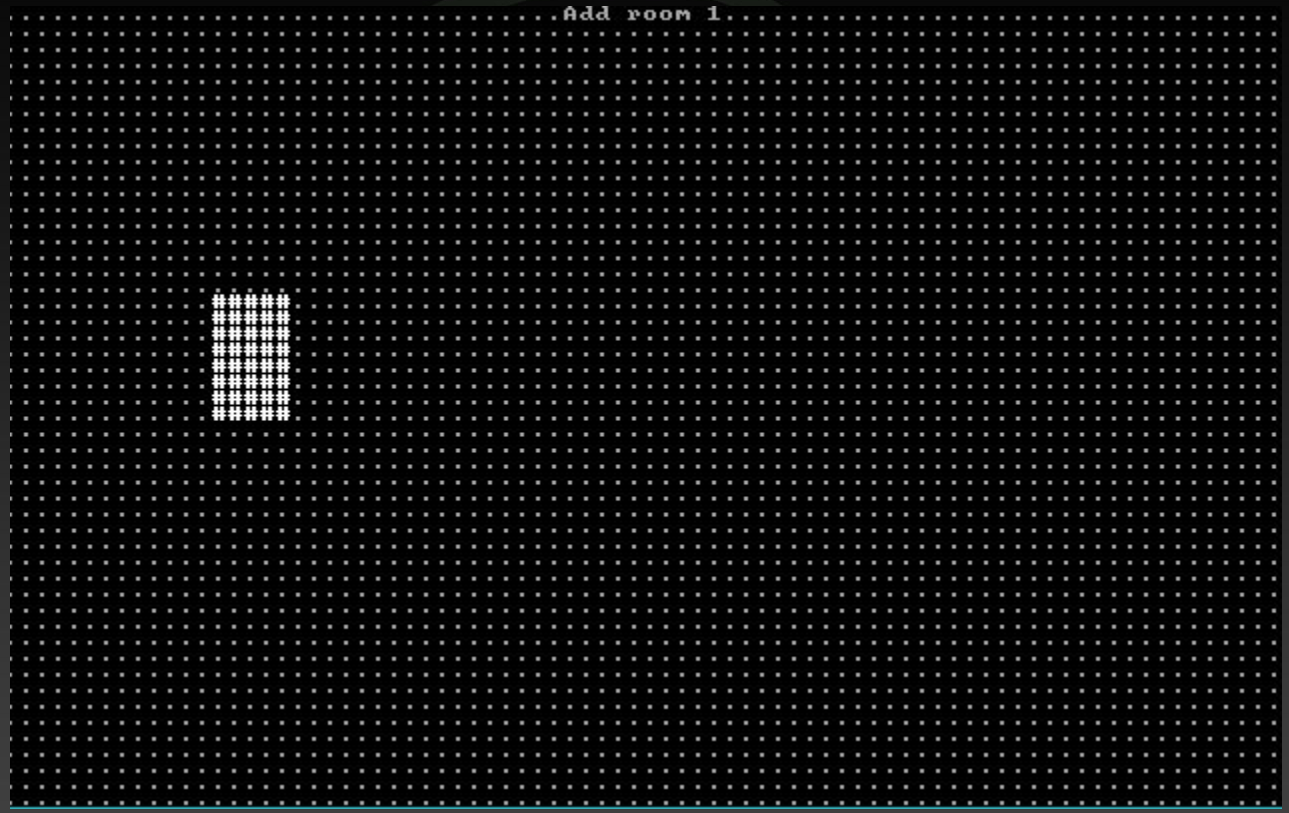
Dwarf Fortress (2006 - Present)

Simple Room-Placement

If you've done any Roguelike Development tutorials - you know this one.

1. Fill the map with walls.
2. Randomly pick a room location.
    - If it's not already occupied, add the room.
3. Keep picking rooms.
4. Join the rooms you kept with corridors.

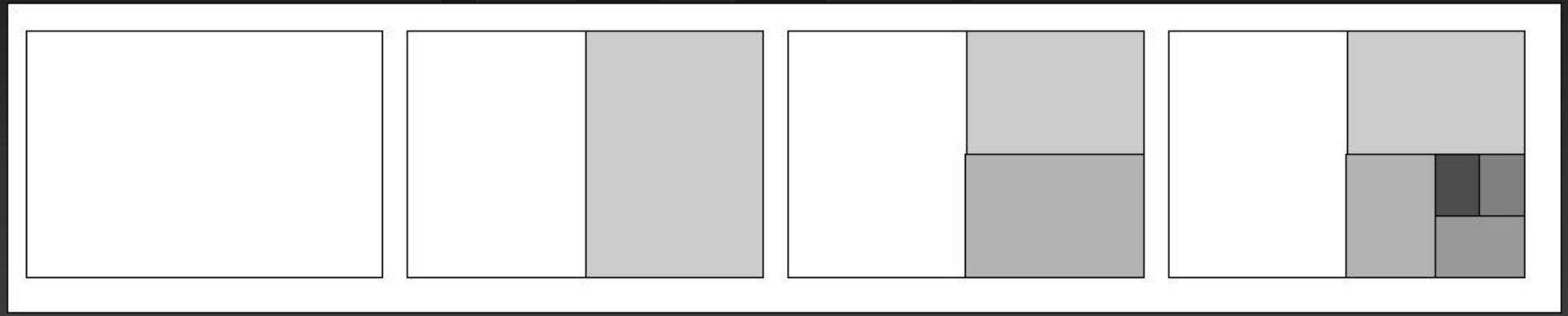
Room Placement - Animated



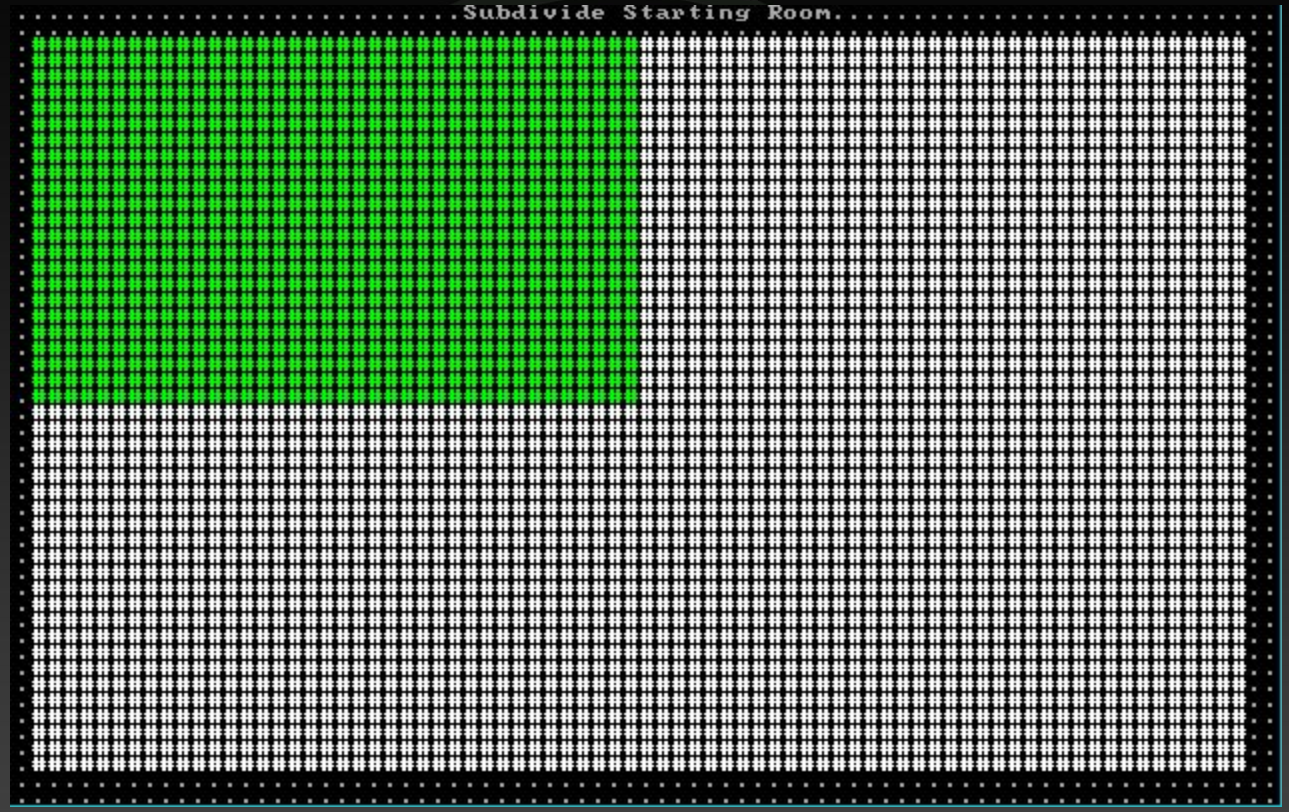
BSP Rooms

Similar results to random room placement, better spaced out. Used in *Nethack*.

Divide map into two. Divide area into two. Repeat. Use divided space for room.



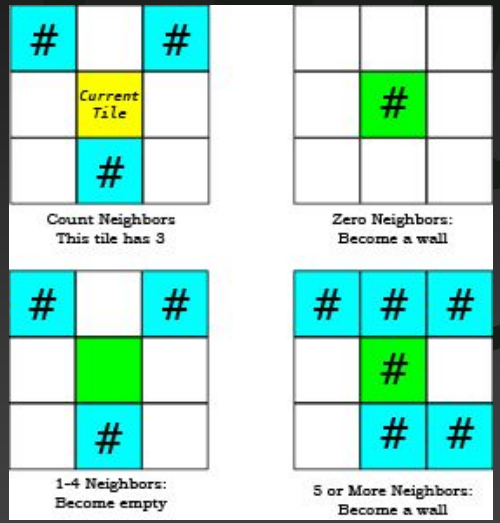
Animated BSP



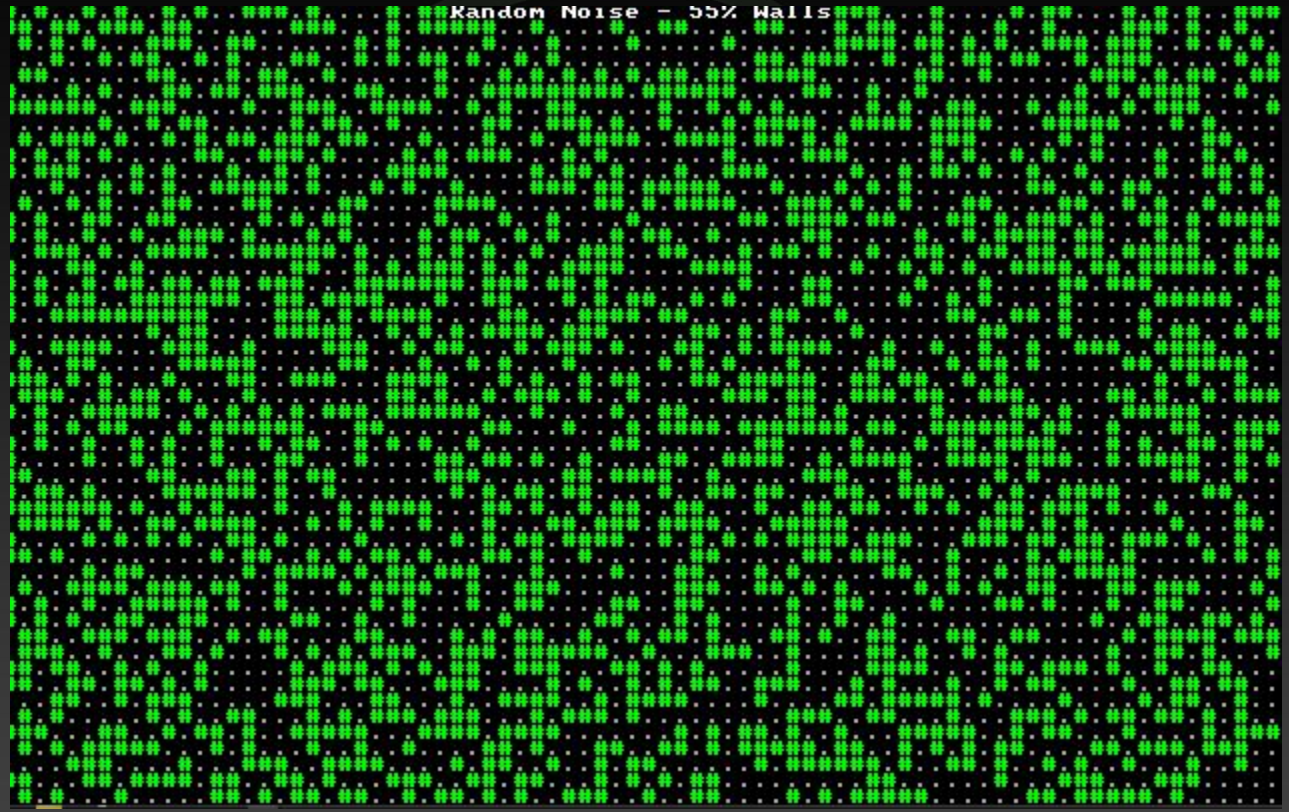
Cellular Automata

Evolve order from chaos.
Popularized in *Conway's Game of Life*.

Make a random map. Apply cell life rules to each tile. Repeat.



Animated Cellular Automata



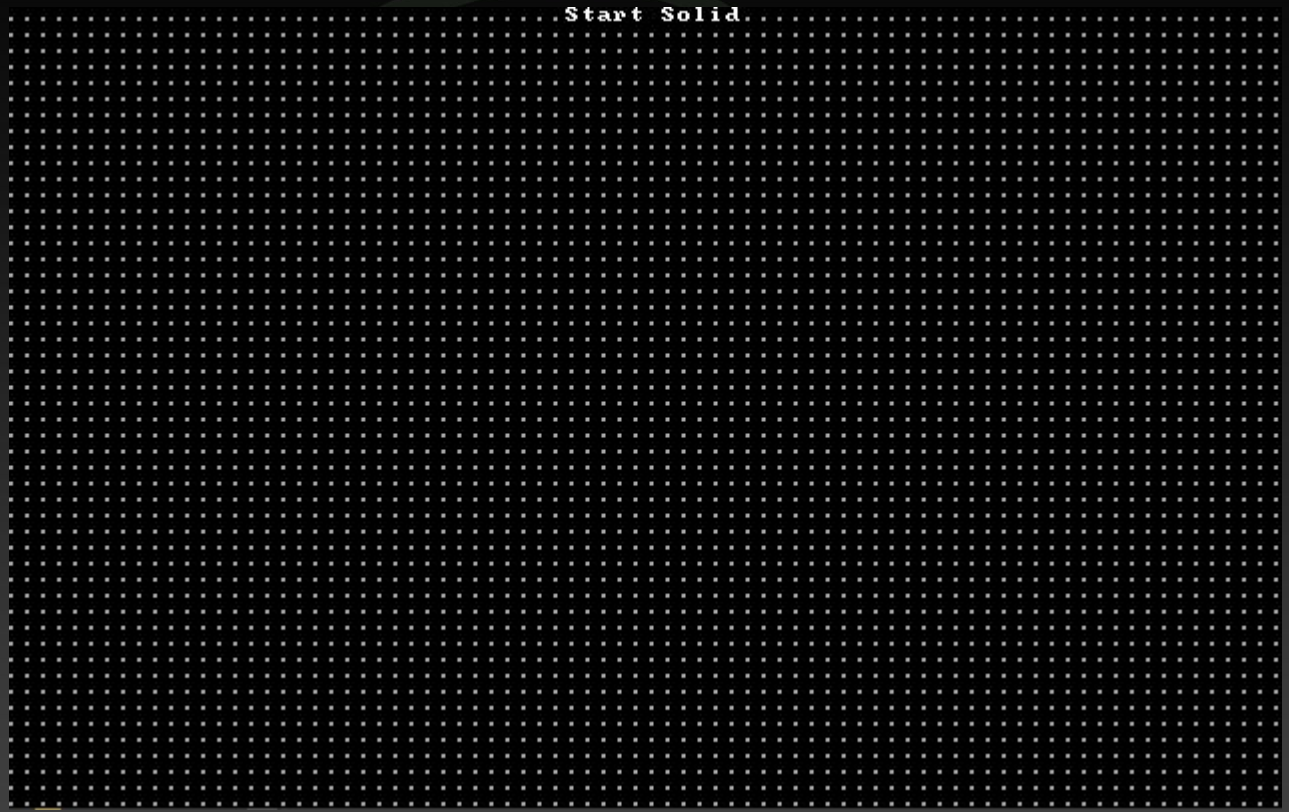
Drunkard’s Walk

Find Umber Hulk. Insert beer.

Place Hulk randomly on solid map. See what he smashes.

Hulk’s stop when they leave the map, or pass out after *n* steps.

Animated Drunken Umber Hulks



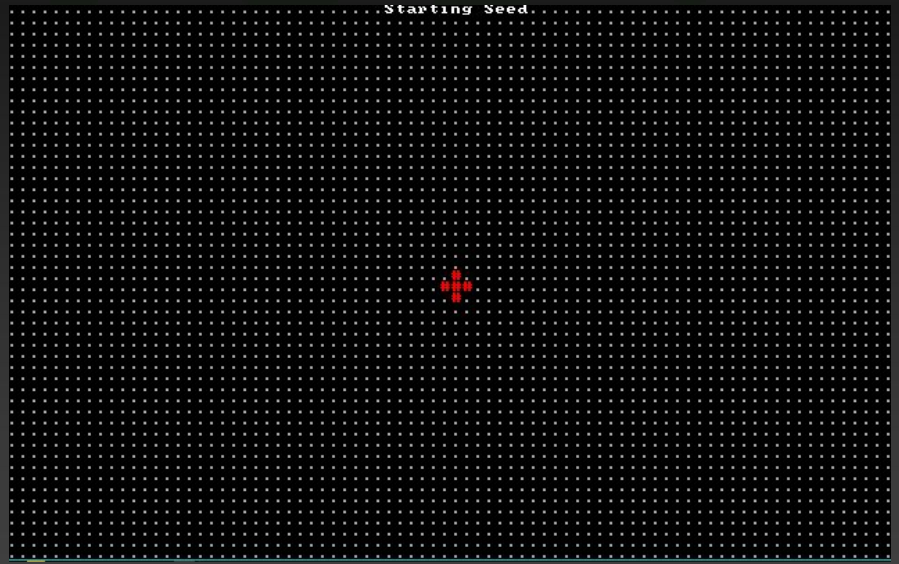
Roguelike Celebration 2020

Diffusion Limited Aggregation

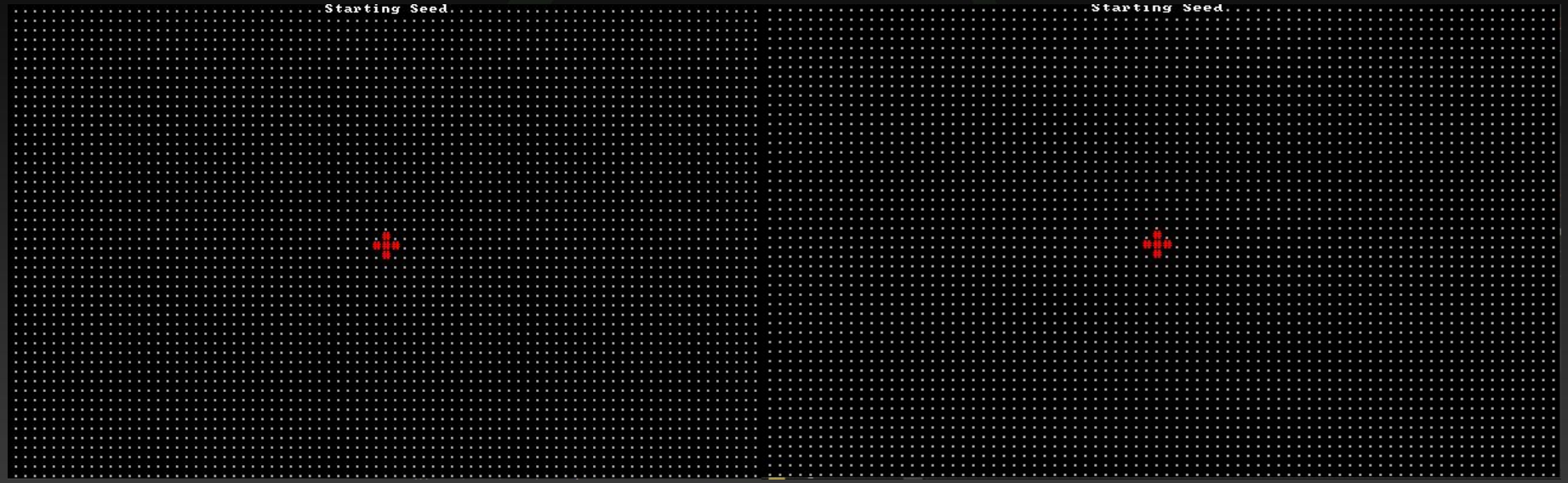
Start with a target seed.

Randomly - or not - fire particles at it.

Dig out the last edge the particle hit.

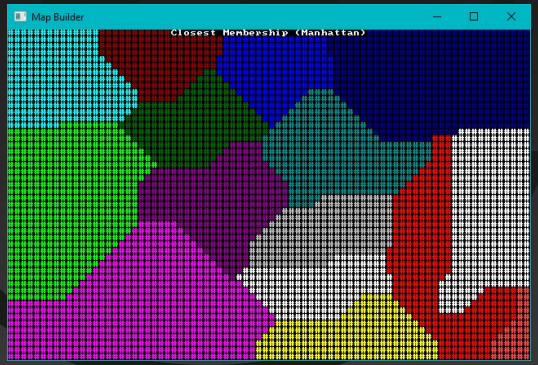


DLA with a Central Attractor

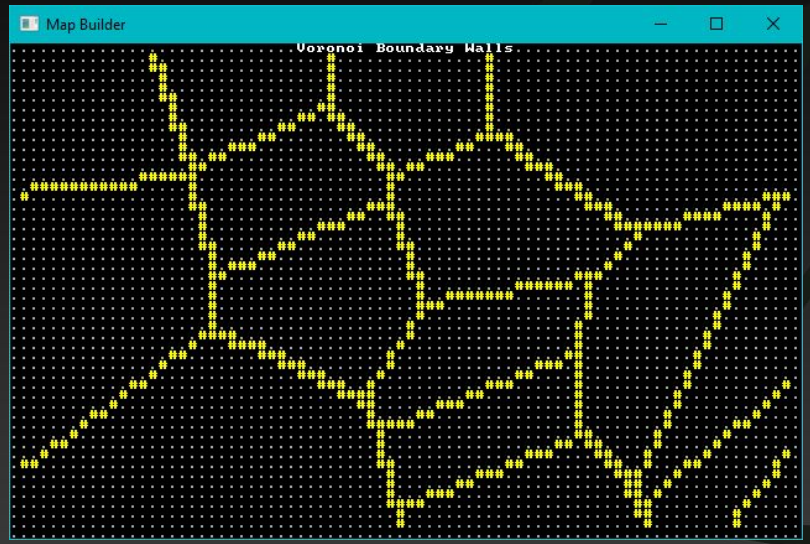


Voronoi Diagrams

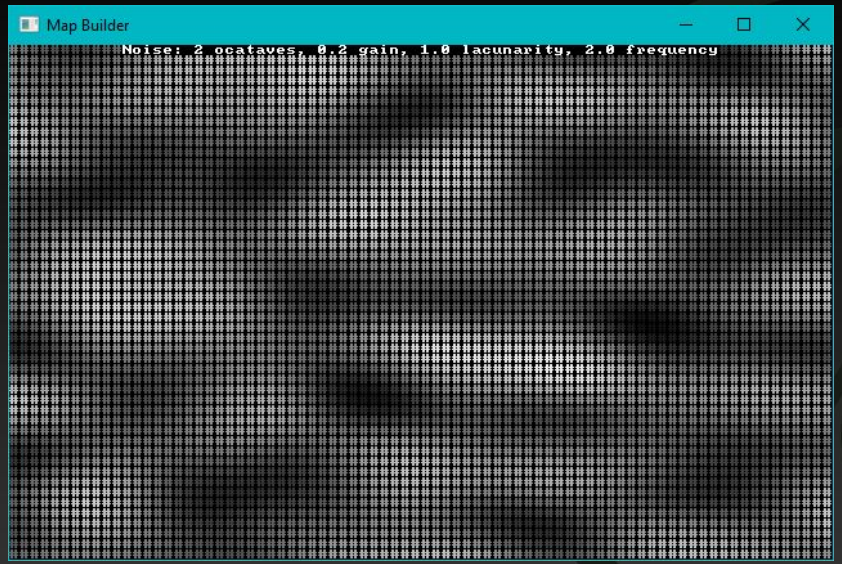
Randomly (or deliberately) place seeds.
Each tile joins the closest seed.
Vary distance heuristic for different effects.

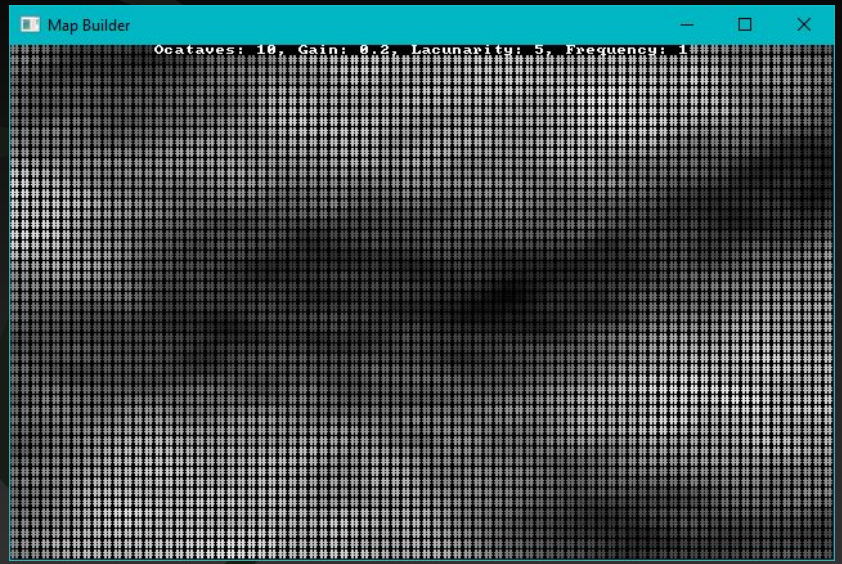


Find the edges

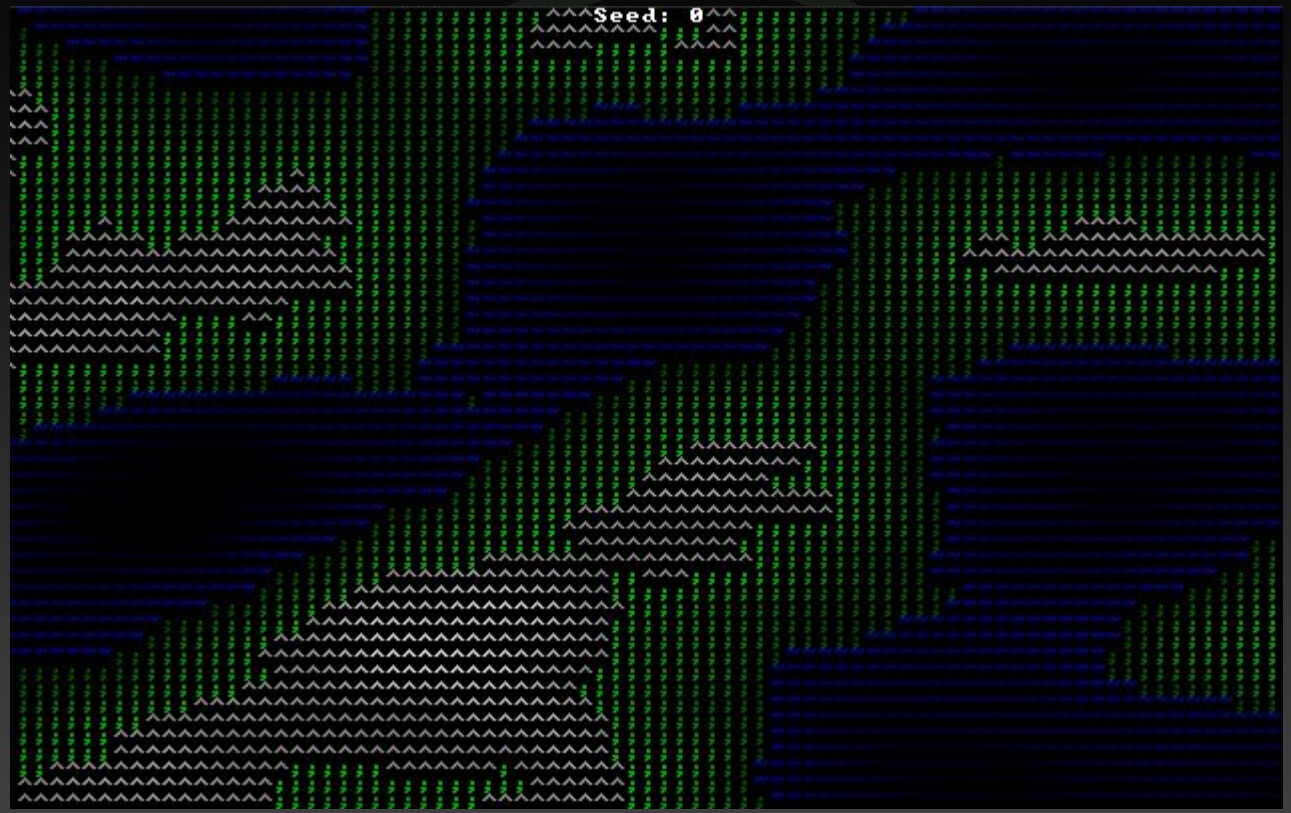


Perlin/Simplex Noise



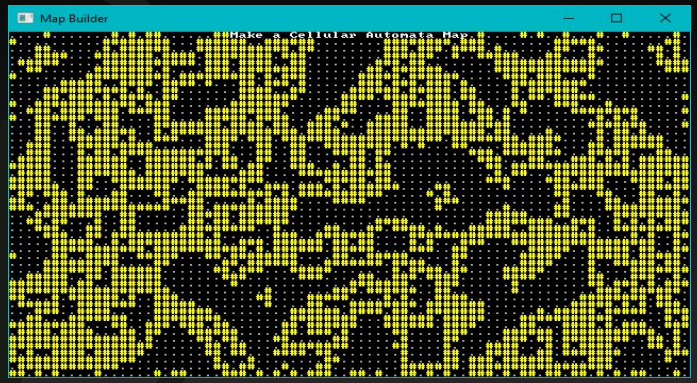


Overworld by Altitude

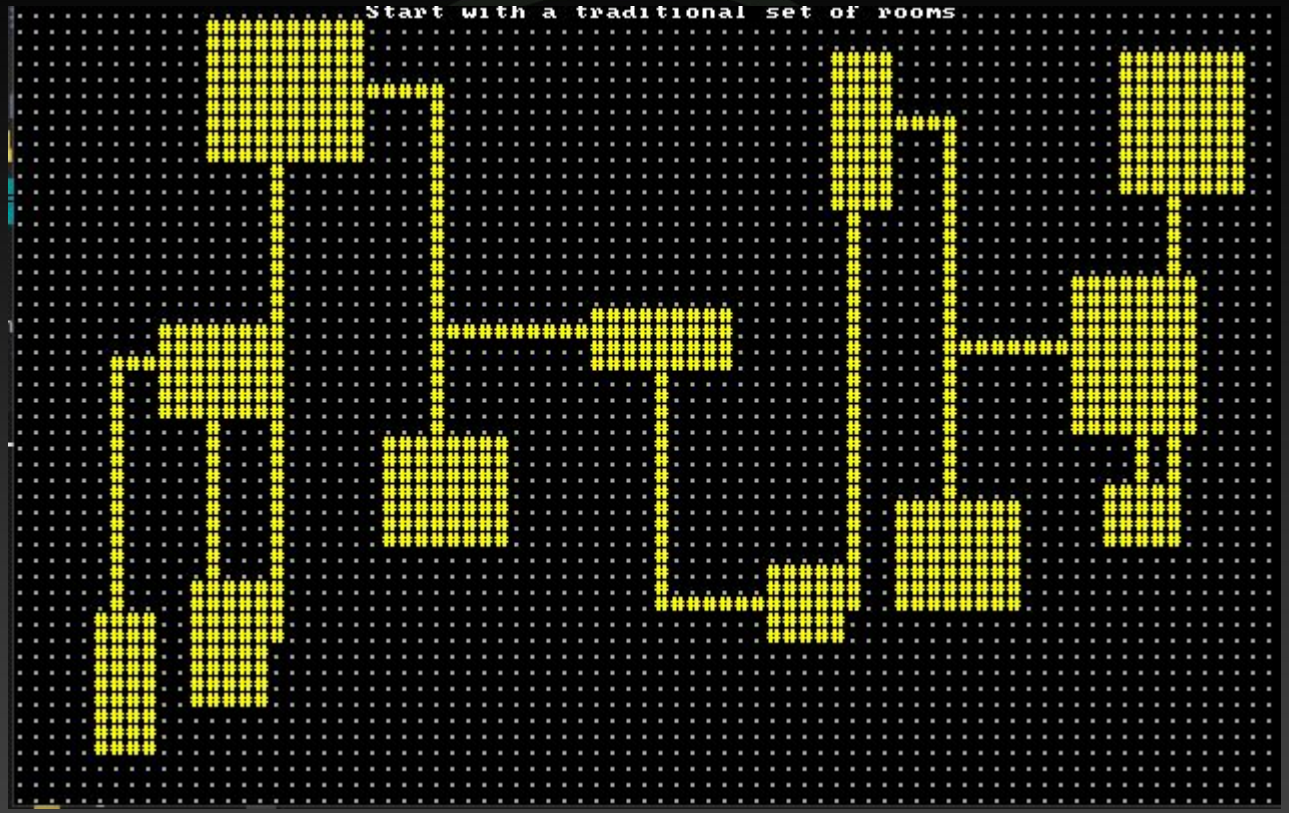


Mix in a second noise layer

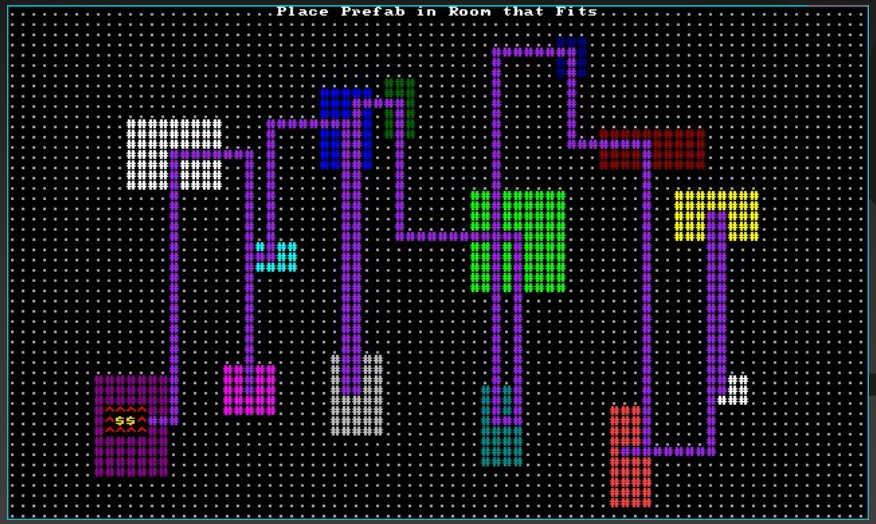
You can use more than one technique

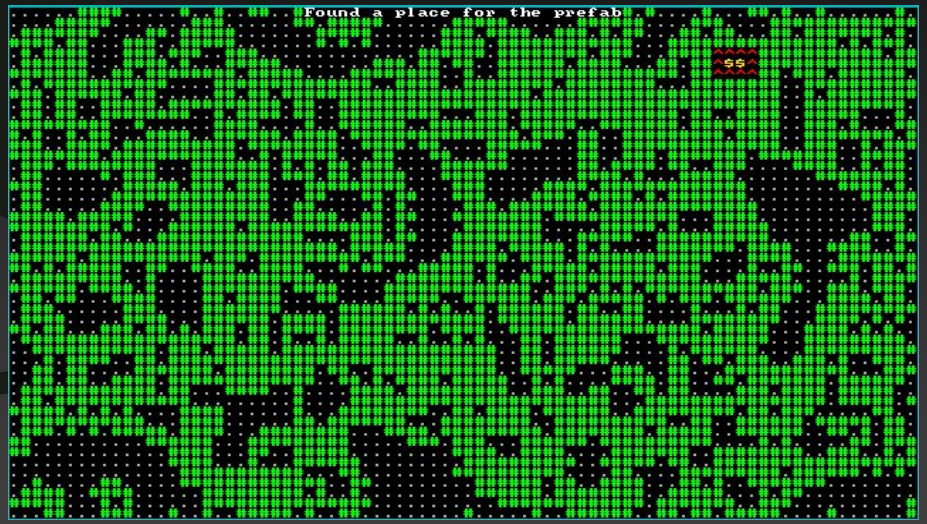


Combine Techniques - DLA for erosion



Placing Prefabs





Algorithm: Dijkstra Maps

1. Start with 1 or more starting points.
2. Rest of the map “sentinel” value - unreachable.
3. Set points adjacent to start to 1.
4. Points adjacent to those 2.
5. Etc. until whole map walked.

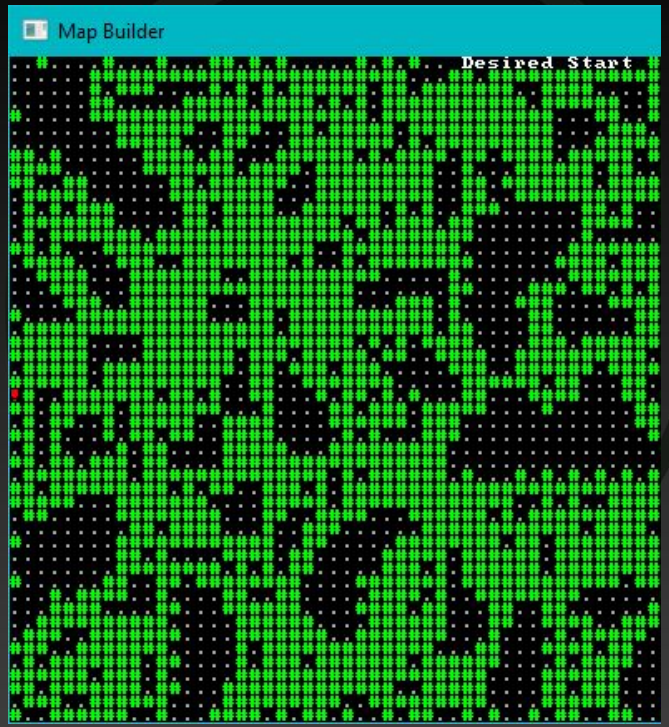
Removing Unreachable Areas

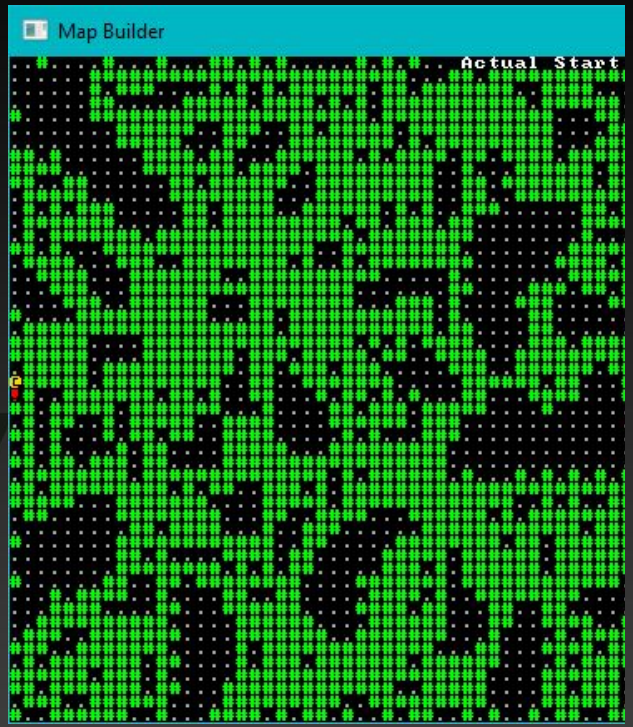
1. Find Central Start
2. Run Dijkstra
3. Cull tiles without a valid distance.

Finding a Starting Point

Find desired starting point.

Find closest open tile for actual start.





Find an Endpoint - using distance to target

Endpoint - Dijkstra to find farthest point

The “Hot Path”

Path-find from start to end.

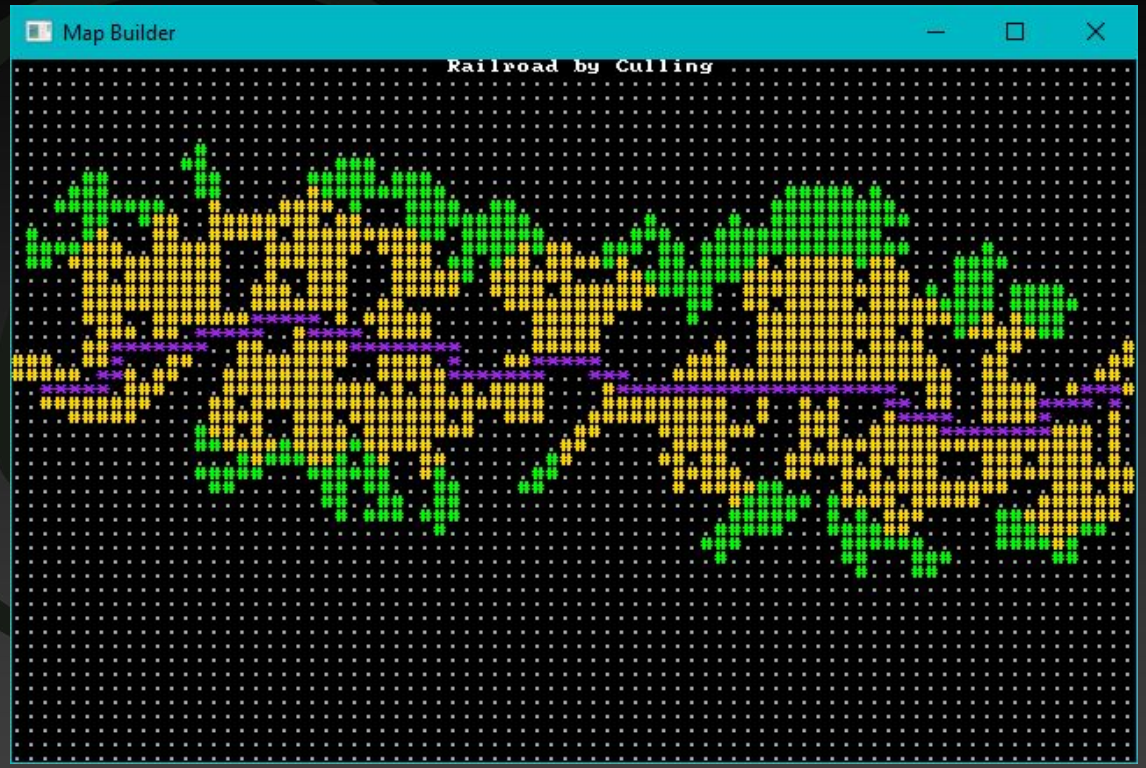
Dijkstra Map with the path as starting points.

<*n* distance is “hot path”

Culling based on the hot path

Hate branching?

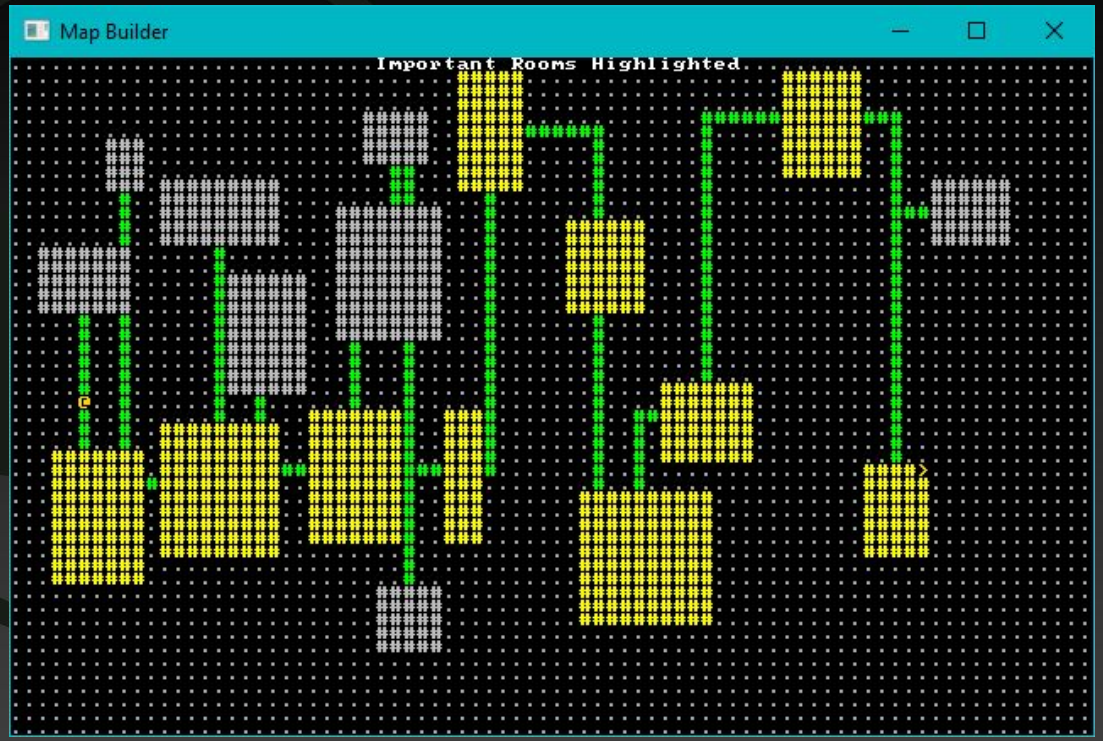
Cull map regions outside of the hot path.



Hot Path for Rooms

Now you know where to put important encounters.

Or “bonus” content to reward branch exploration.

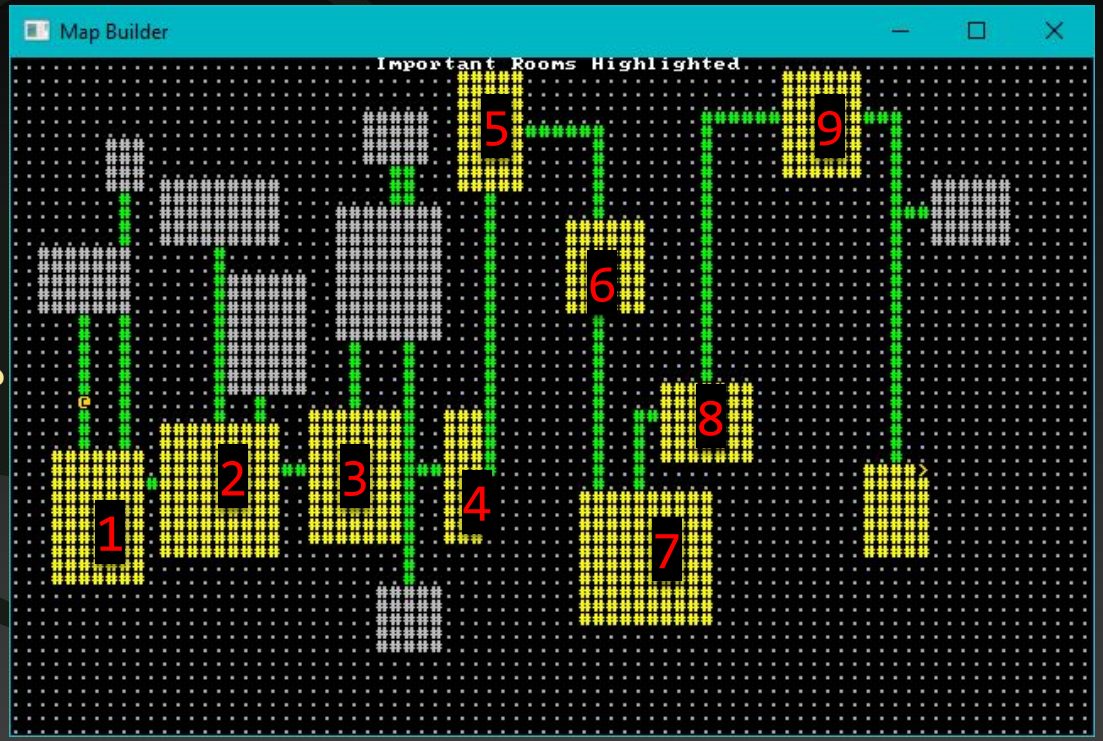


Telling a Story

Rooms are ordered.

Story progression is in order, but RNG retained.

Maybe 5 has a locked door?
Key must be in 1-4!



Questions?

Source code for this talk:
https://github.com/thebracket/roguelike-celebration-2020

Hands-on Rust: Effective Learning through 2D Game Development and Play
Look for it soon on https://PragProg.com/