

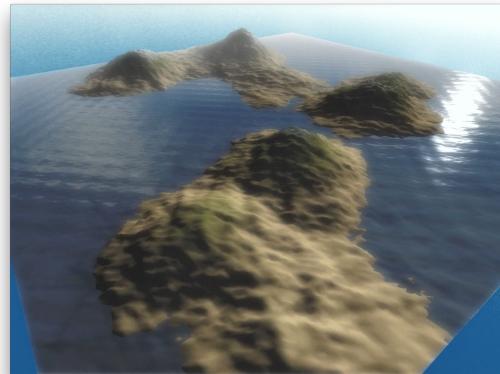
# Procedural Generation of Mountain Ranges based on Geology

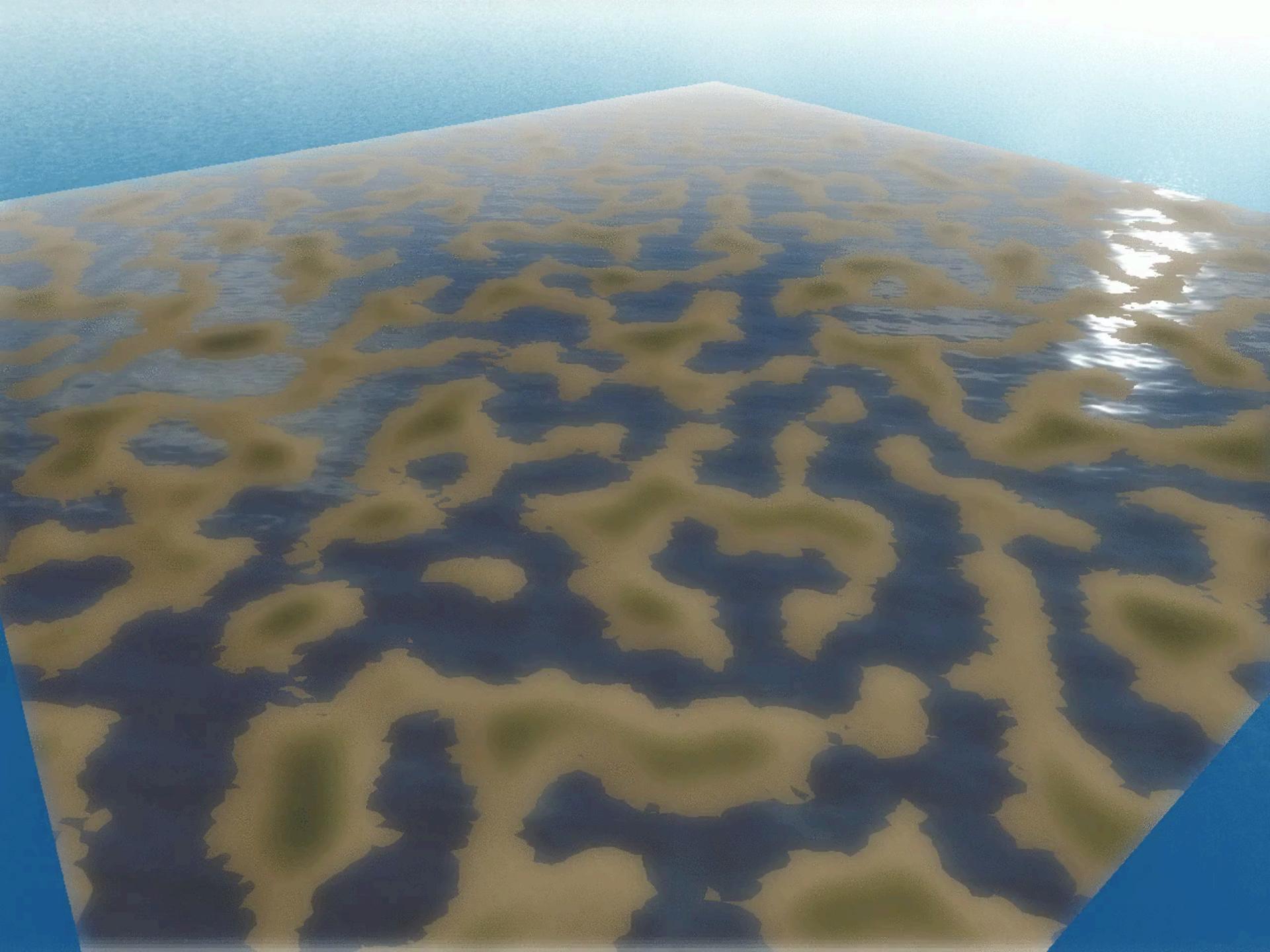
Final presentation

Presented by: Bernhard Fritz

Supervisor: Univ.-Prof. Dr. Matthias Harders

Group: Interactive Graphics and Simulation





# Outline

---

- Task description
- Challenges
- Essential data structures
- Basic, non-geology-based terrain generation algorithms
- Advanced, geology-based terrain generation and degradation algorithms
- Visualization techniques

# Task description

---

- Implement and test several existing terrain generation algorithms
- Develop an algorithm to procedurally generate mountain ranges
- Learn about how mountains originally came into existence on Earth
- Develop a simulation that accounts for phenomena involved in mountain generation and degradation

# Challenges

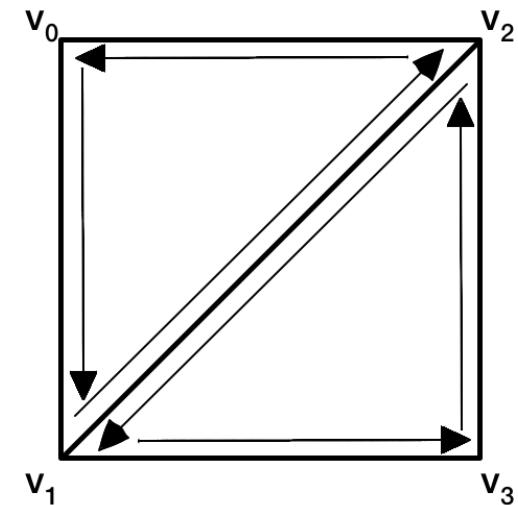
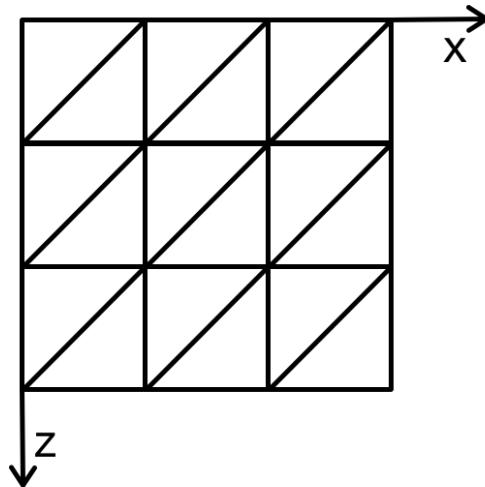
---

- Papers often only vaguely describe the techniques used to produce the showcased results
- Computer graphics algorithms and simulations are known to require a lot of computational resources. Performance optimizations were often necessary and added another layer of difficulty
- Creative thinking was required to find simple, yet realistic solutions that can be used in a real-time environment

# Essential data structures (1)

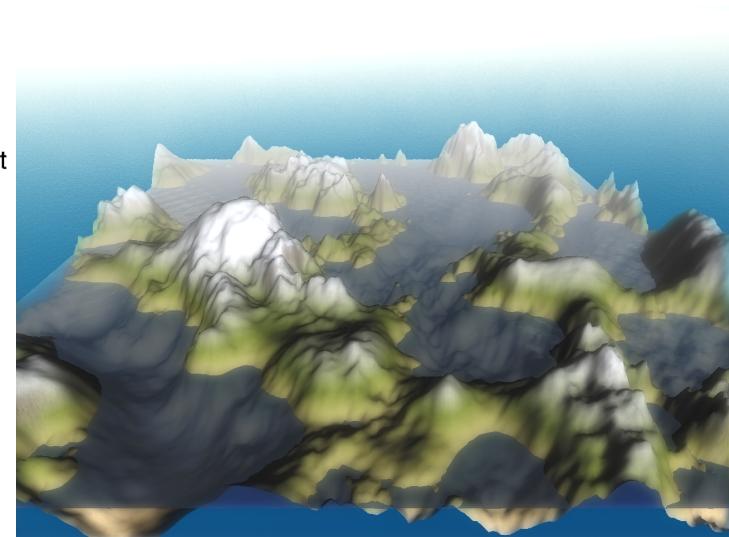
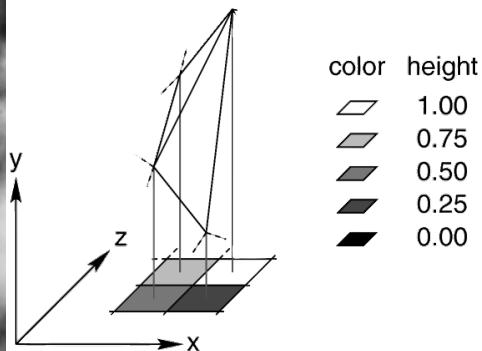
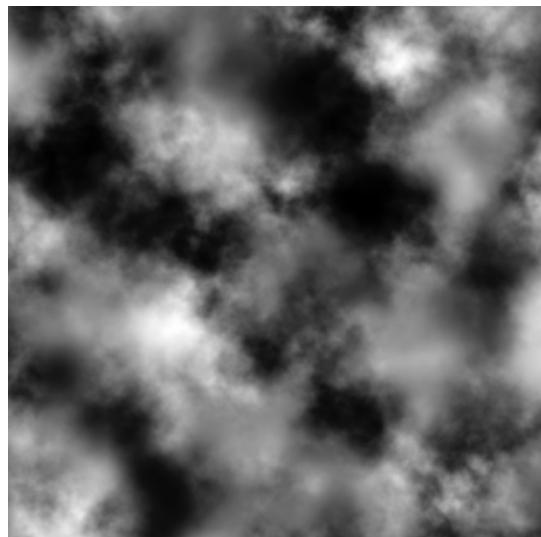
---

- Heightmaps consist of tiles
- Tiles consist of triangles
- Triangles consist of vertices
- Vertices are reused to save space



# Essential data structures (2)

- Height is well defined for any tuple  $(x, z)$ 
  - This implies no overhanging landscape
- Heightmaps can be saved as grayscale images
  - 1 byte per pixel = 256 possible height values



# Basic, non-geology-based terrain generation algorithms

---

- Random, hardly controllable terrain generation algorithms
  - Diamond-Square algorithm
  - Fault algorithm
- Parametrically controllable terrain generation algorithms
  - Repeated Magnification and Probing algorithm

# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

Diamond-Square algorithm is an enhancement of Midpoint-Displacement algorithm

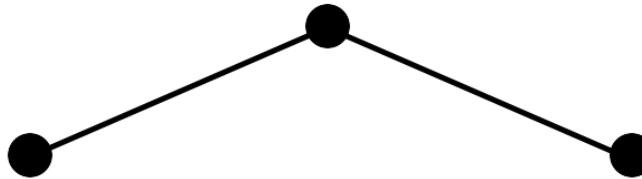


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

Diamond-Square algorithm is an enhancement of Midpoint-Displacement algorithm

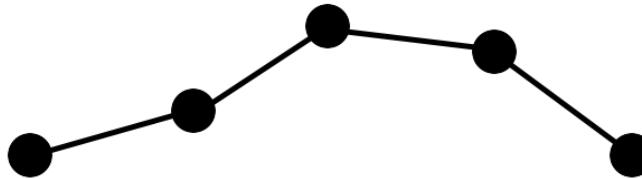


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

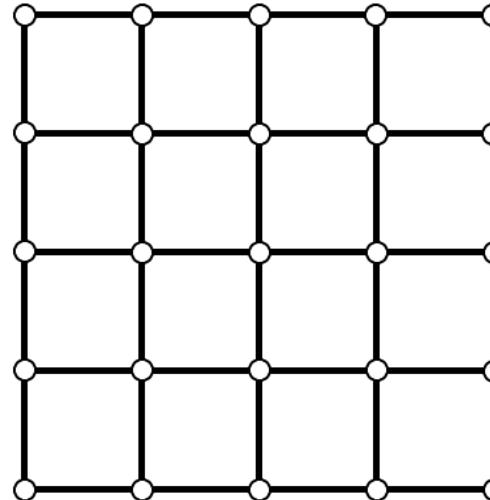
Diamond-Square algorithm is an enhancement of Midpoint-Displacement algorithm



# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

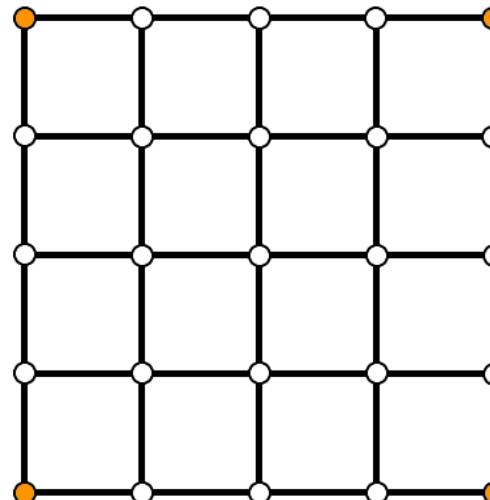


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

Initialization of corner vertices

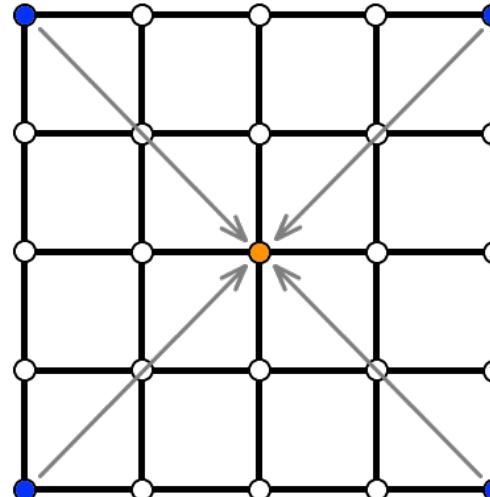


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

## Diamond-Step

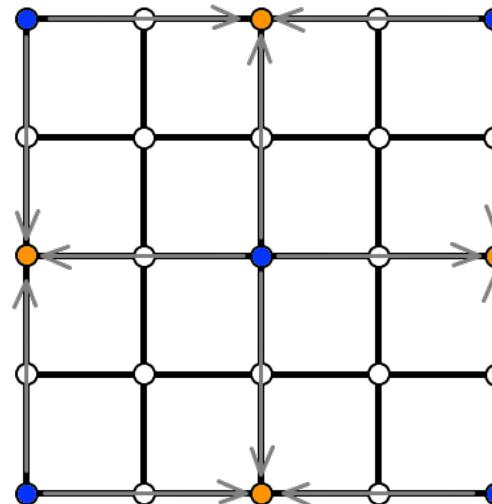


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

## Square-Step

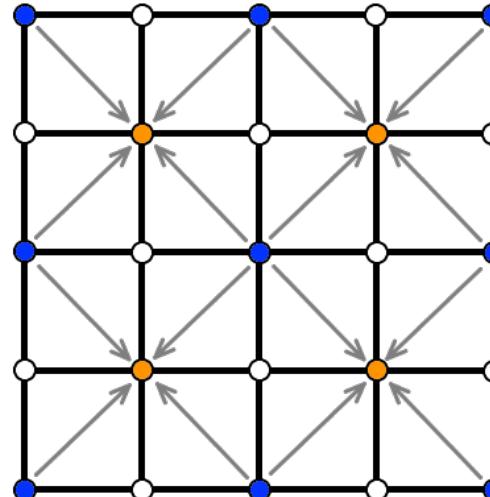


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

## Diamond-Step

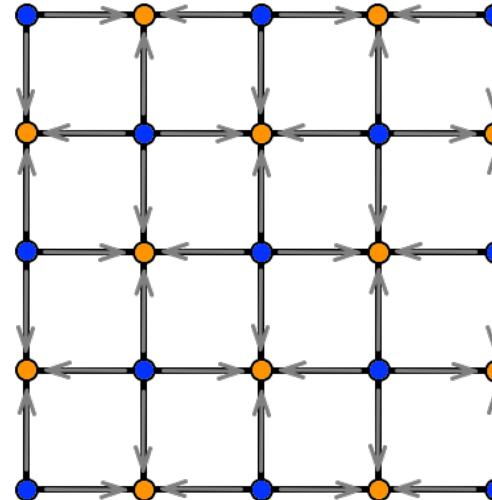


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
  - Fault algorithm

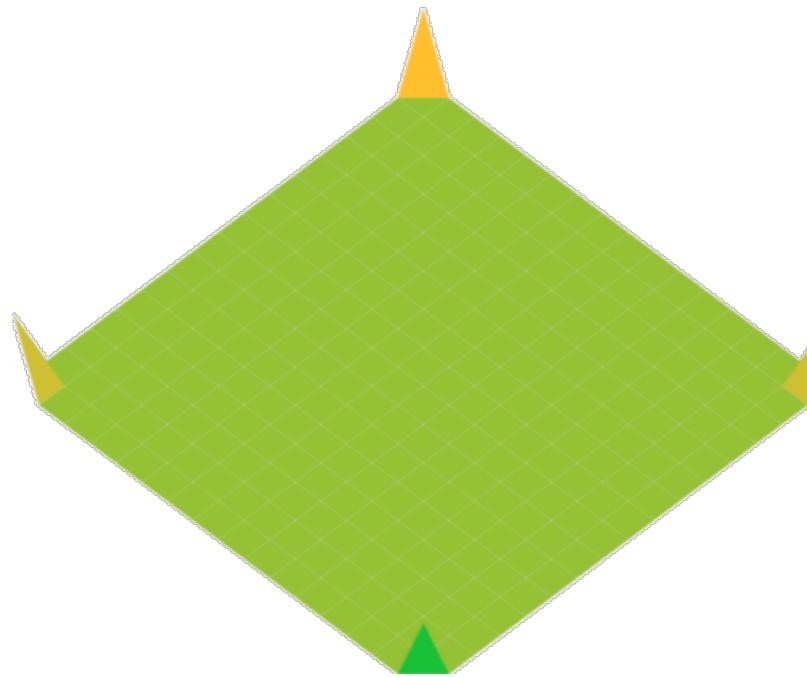
## Square-Step



# Random, hardly controllable terrain generation algorithms

---

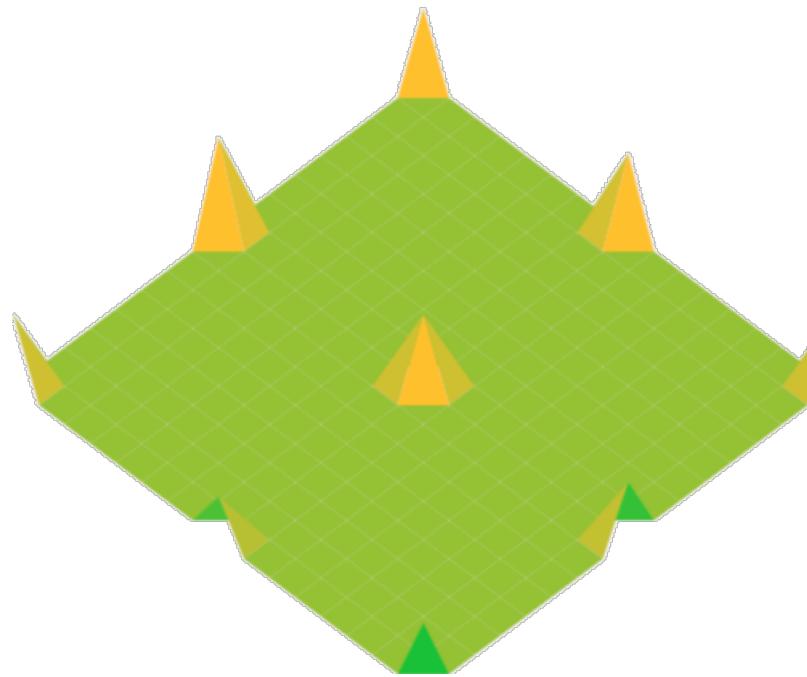
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

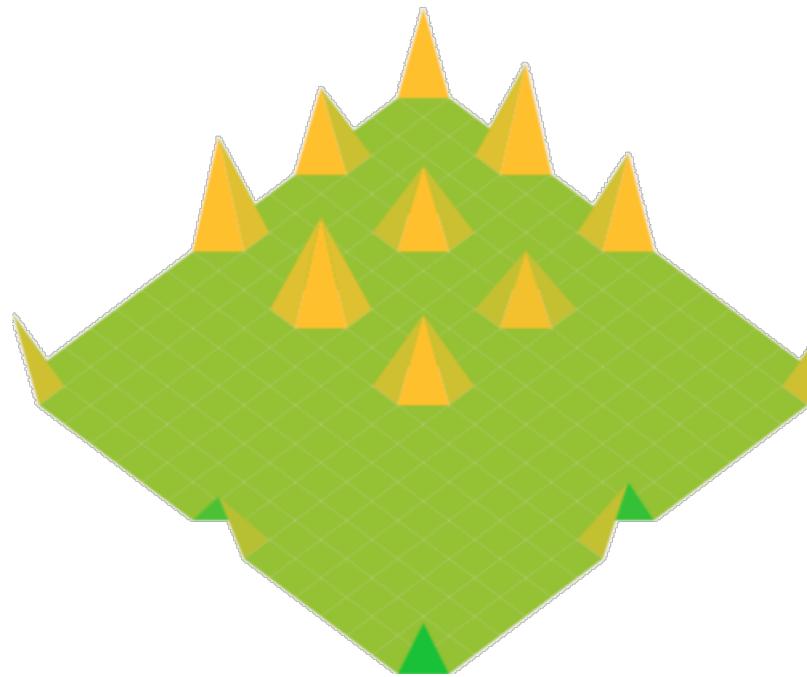
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

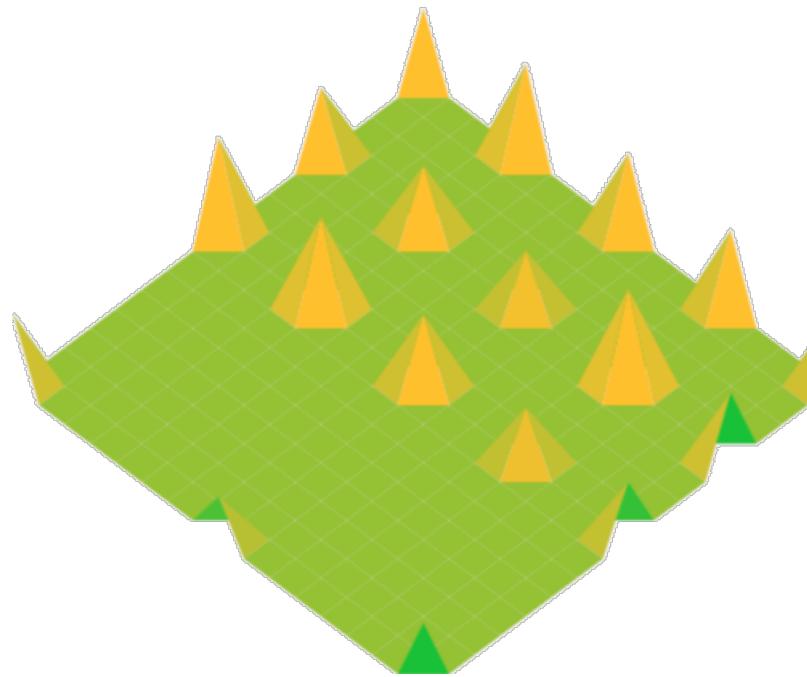
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

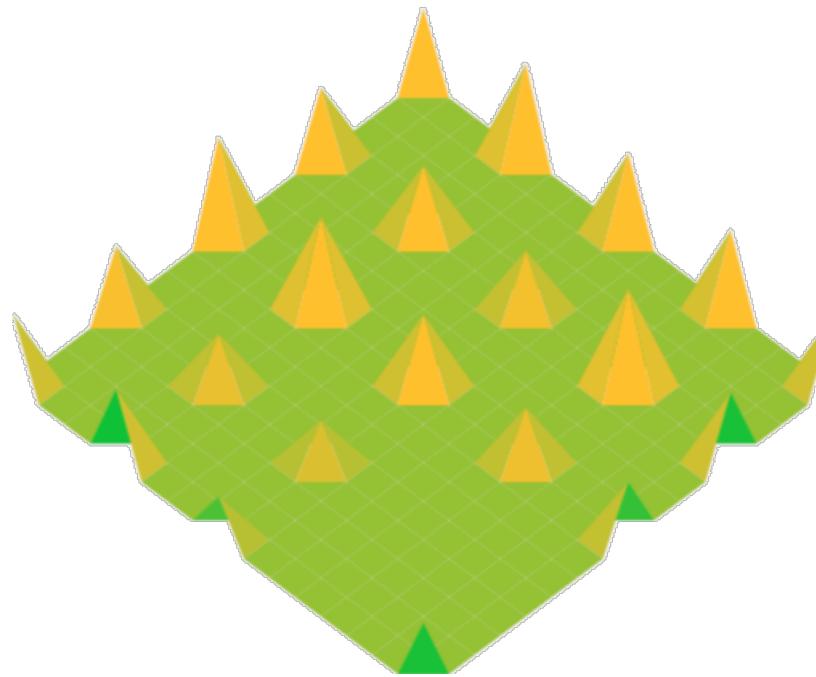
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

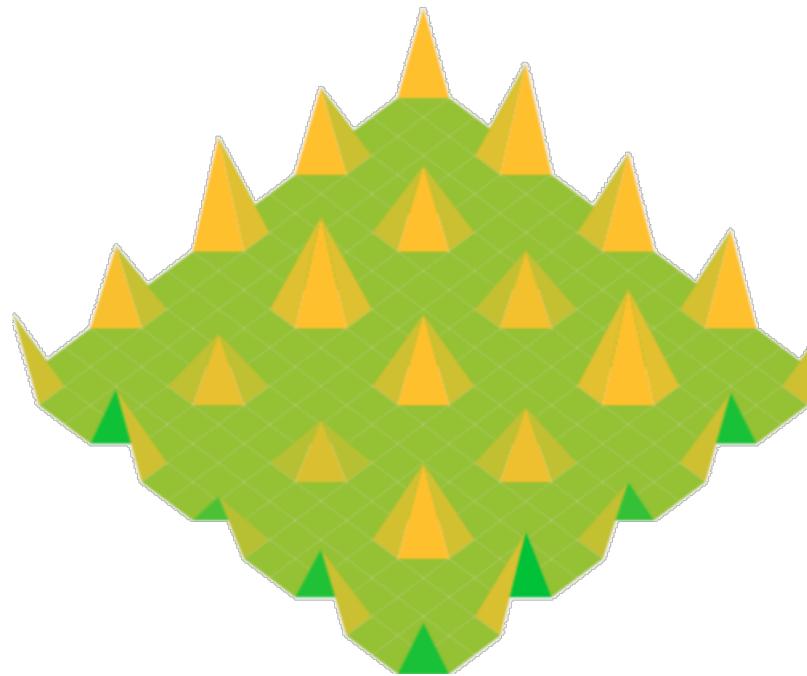
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

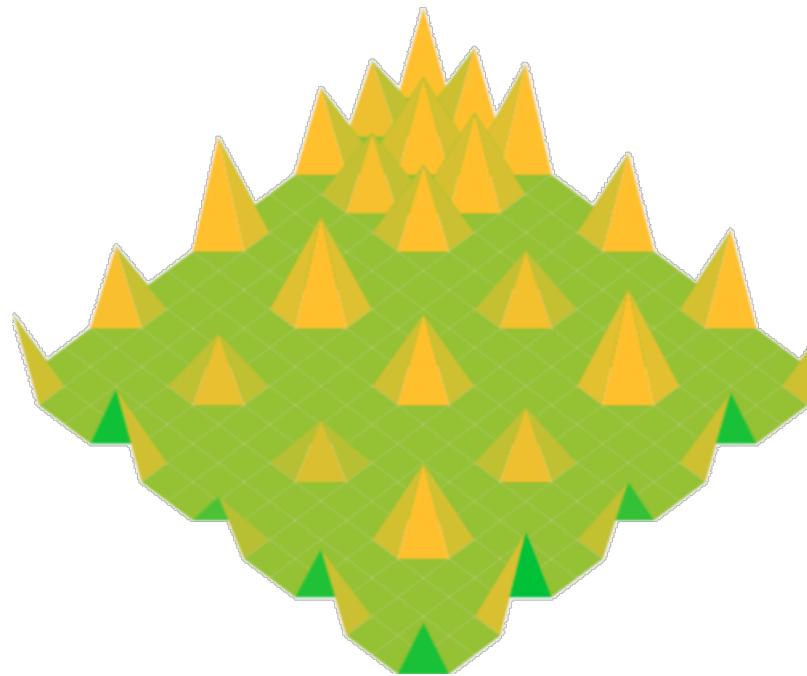
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

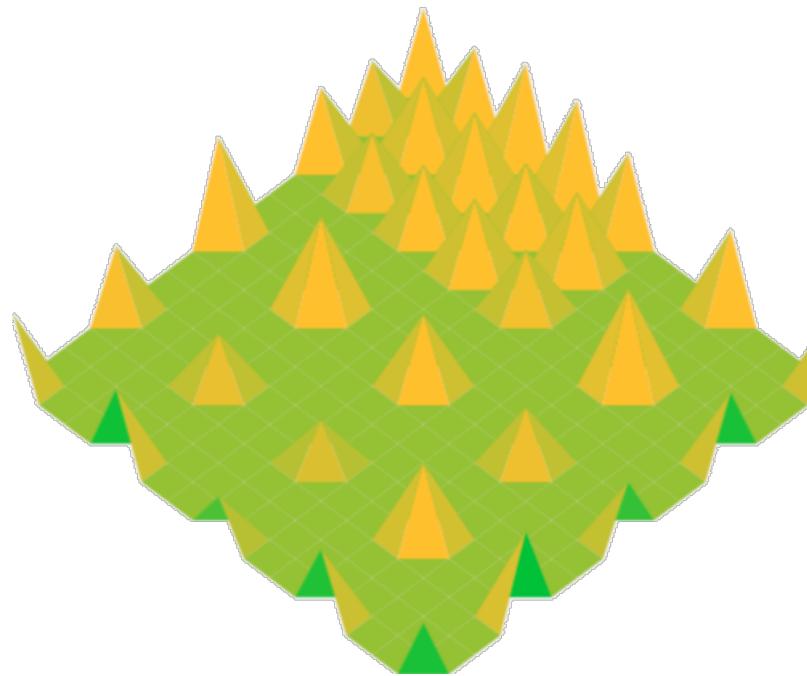
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

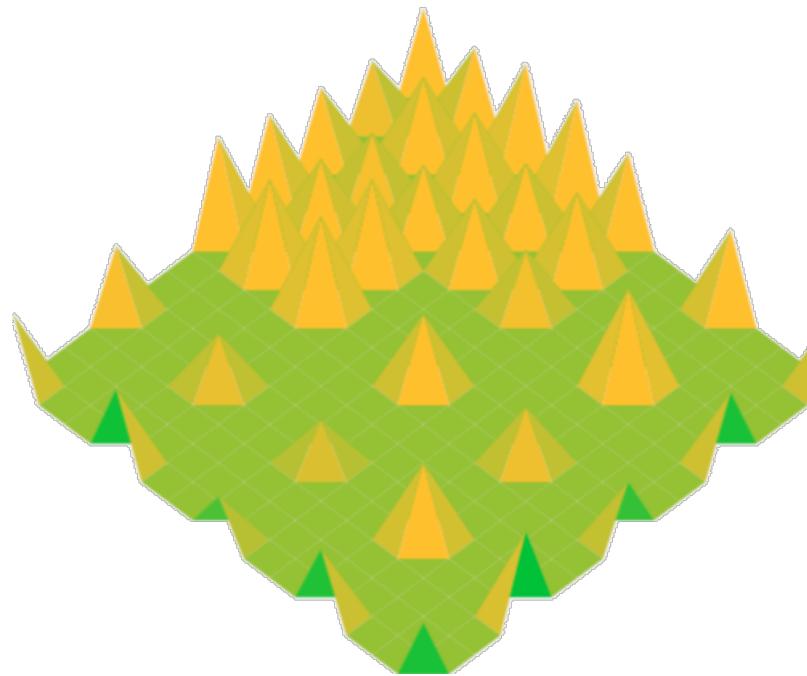
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

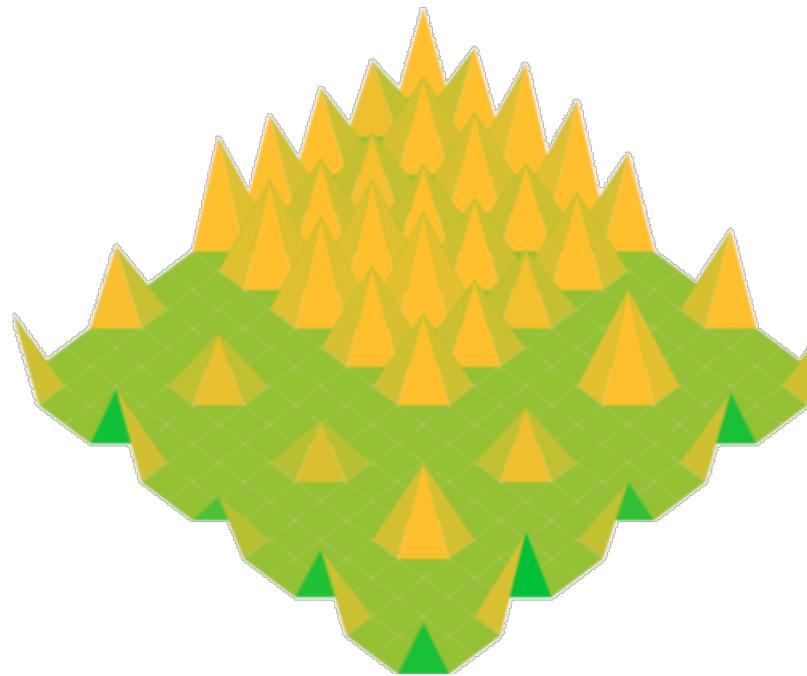
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

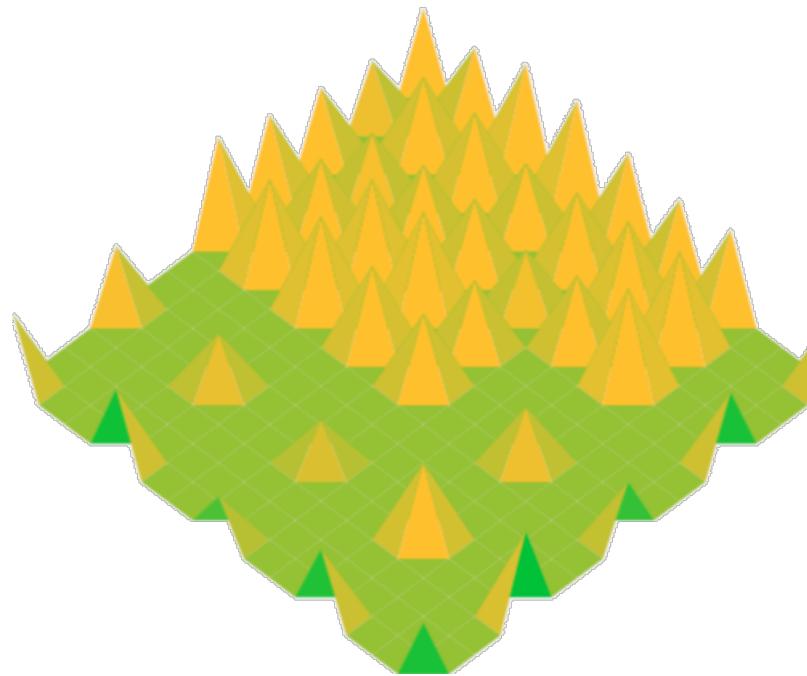
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

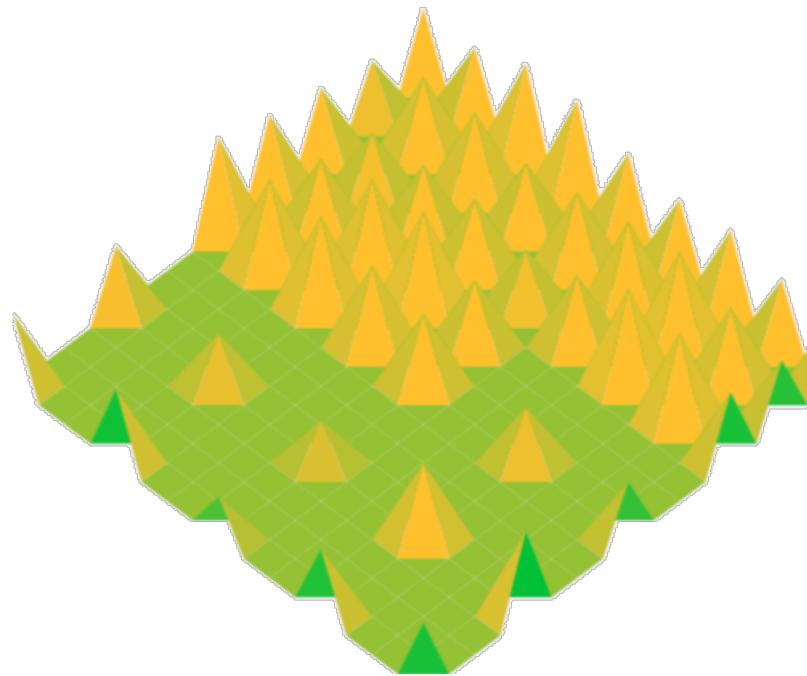
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

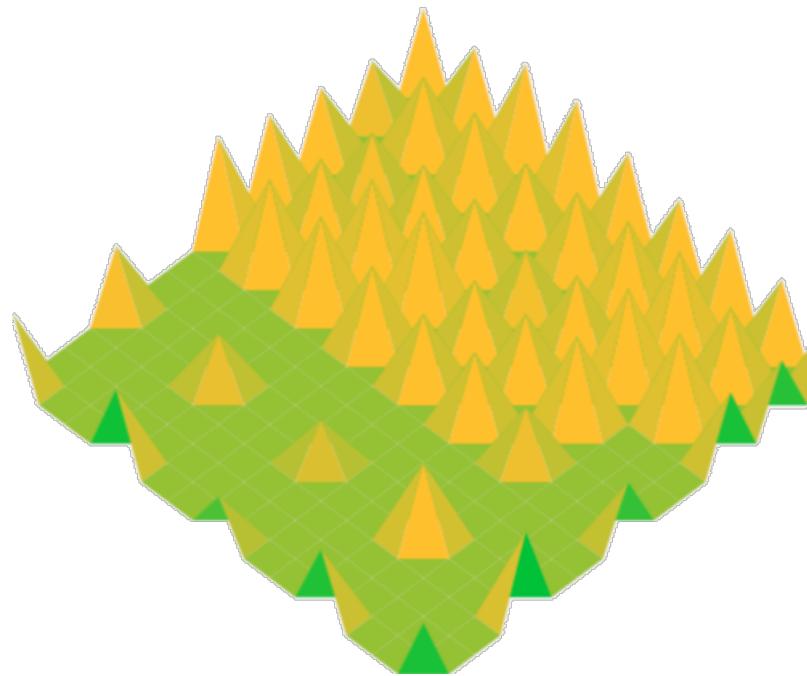
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

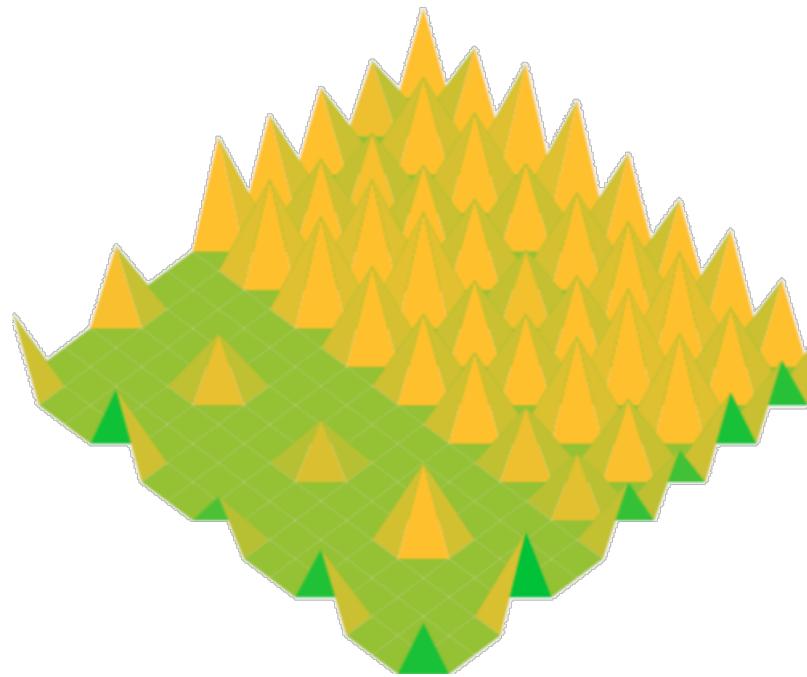
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

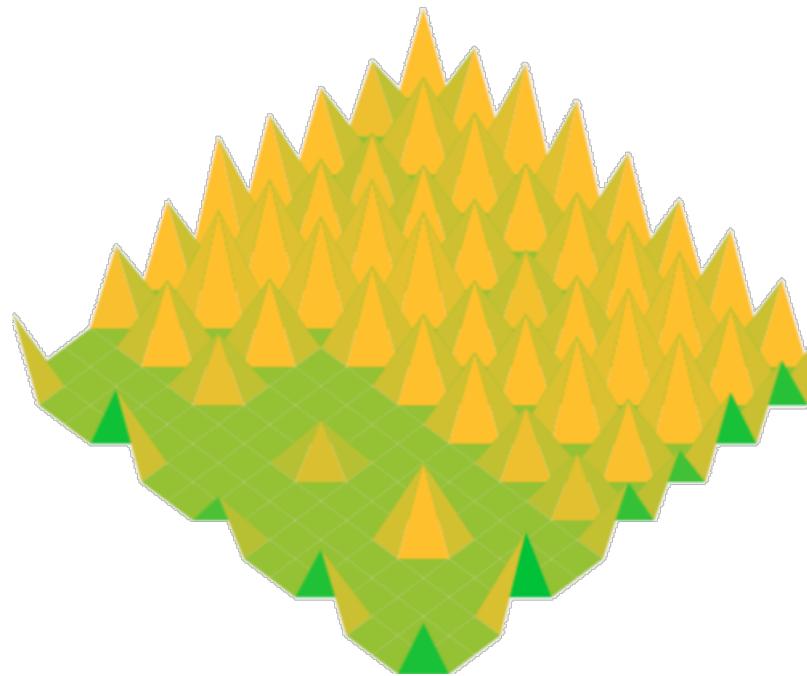
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

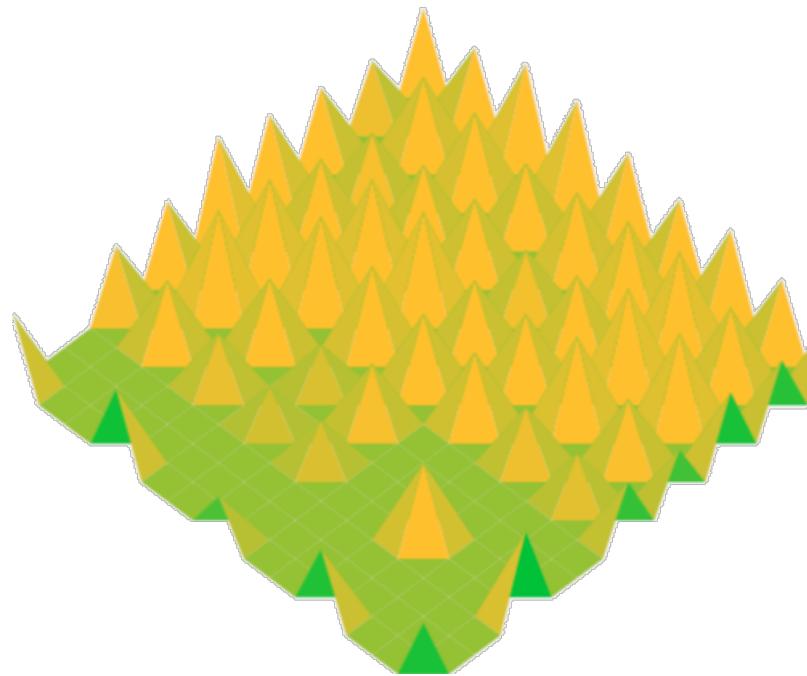
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

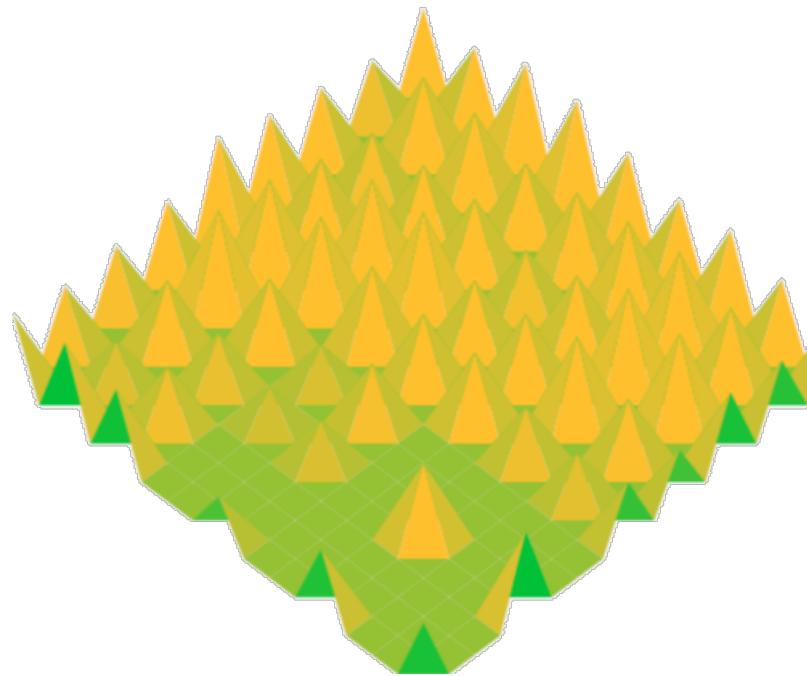
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

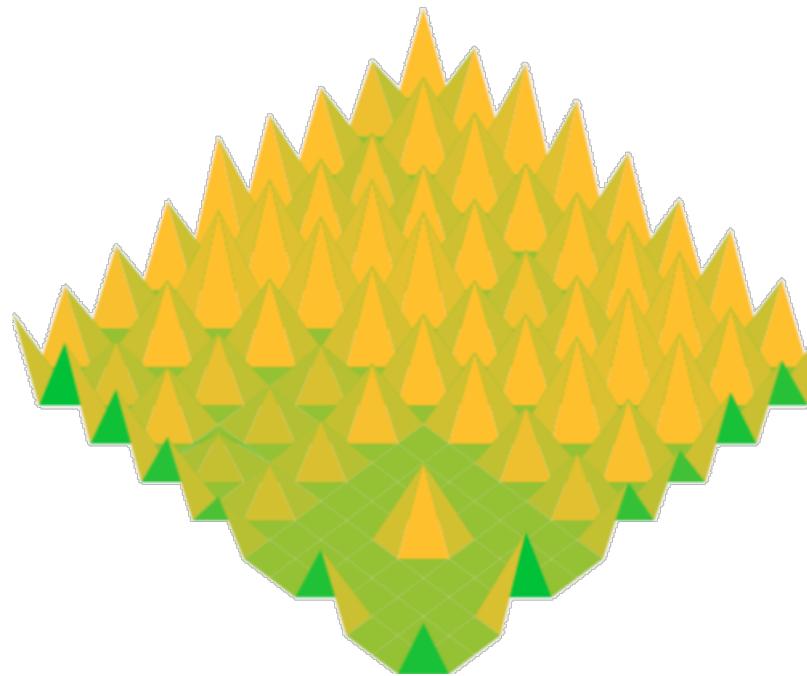
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

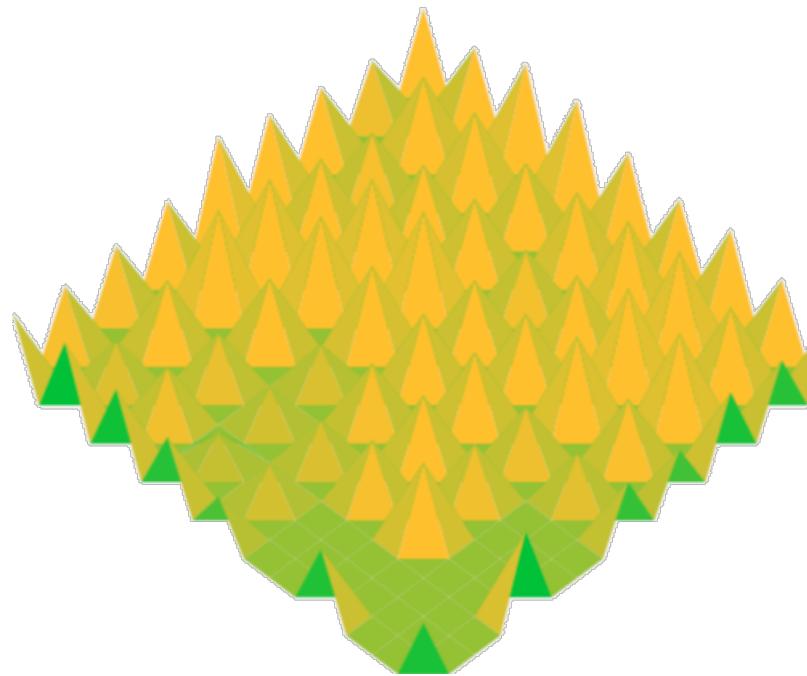
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

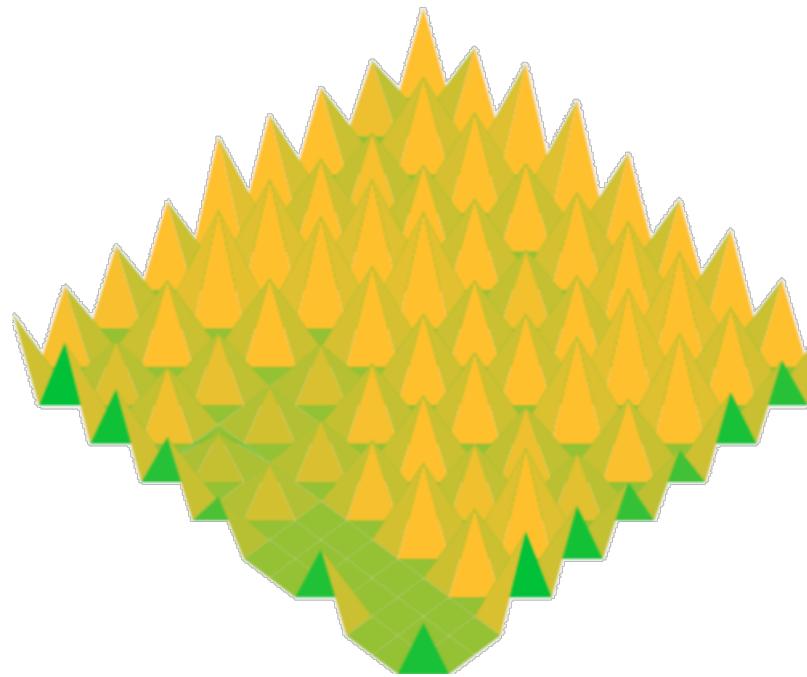
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

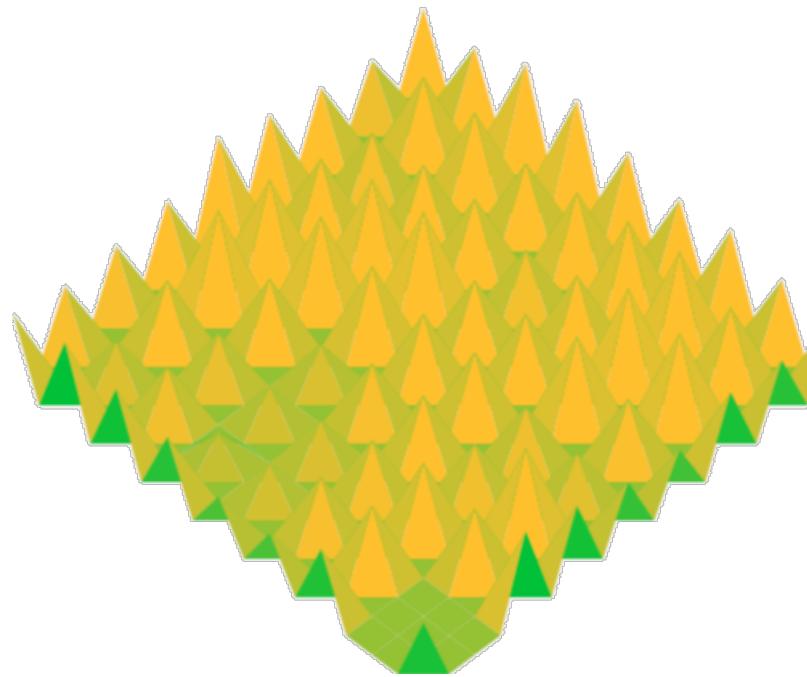
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

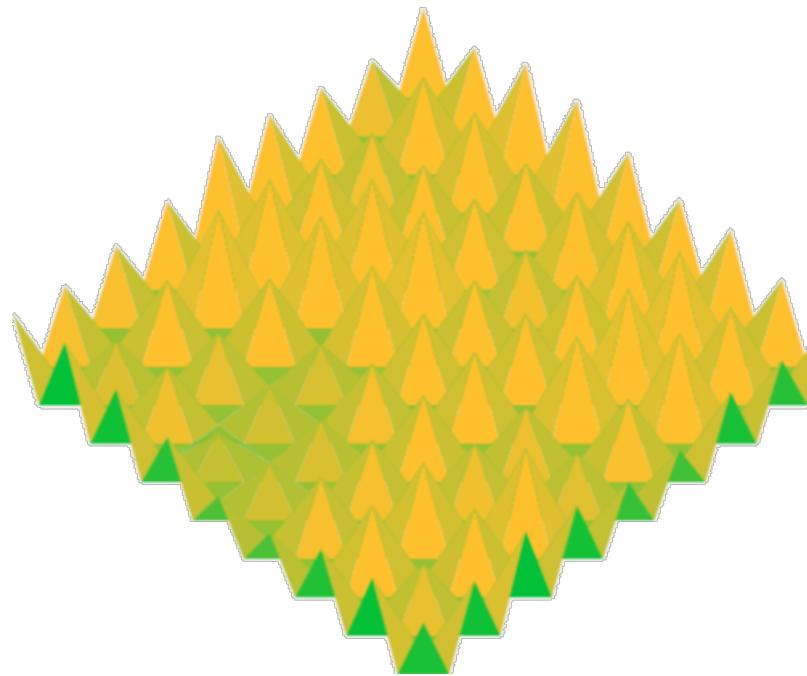
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

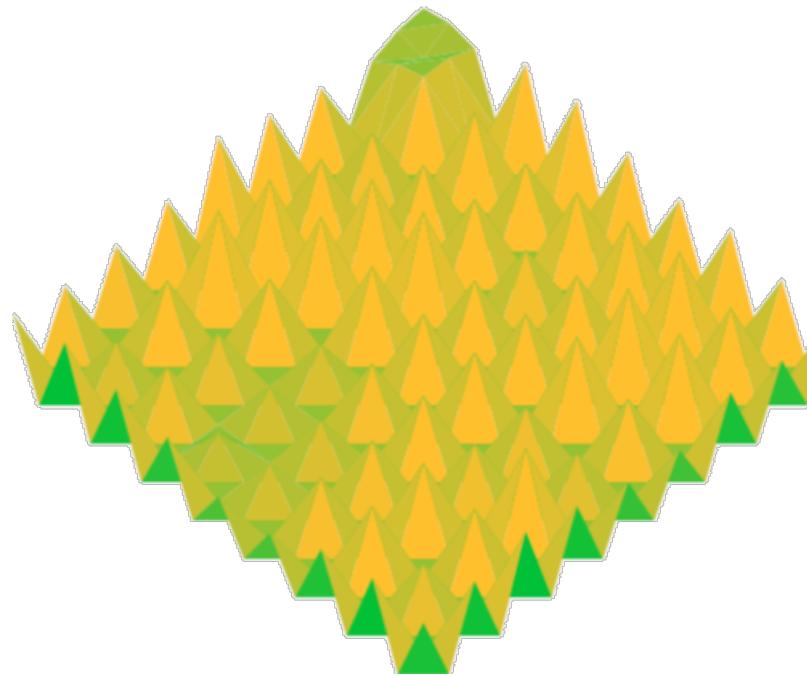
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

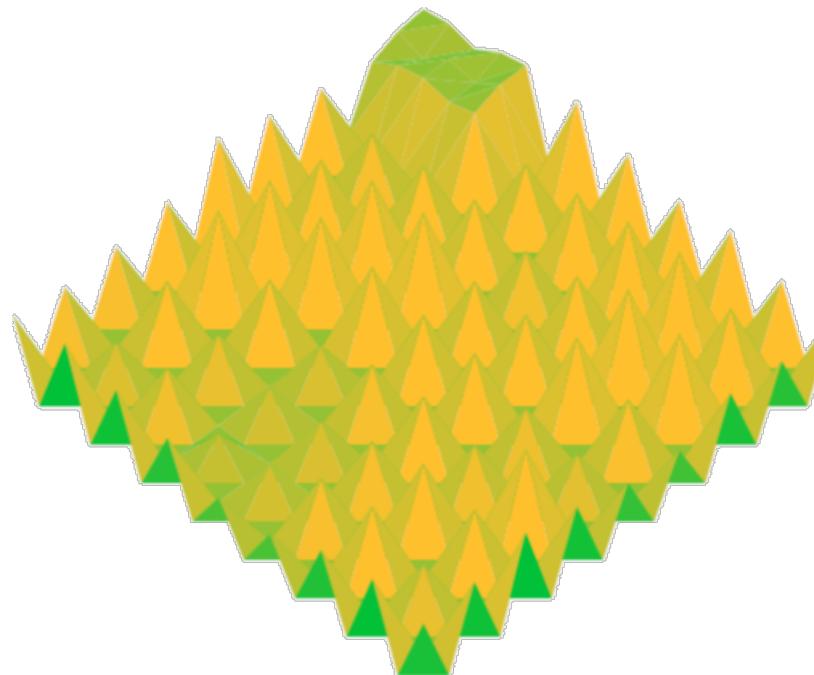
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

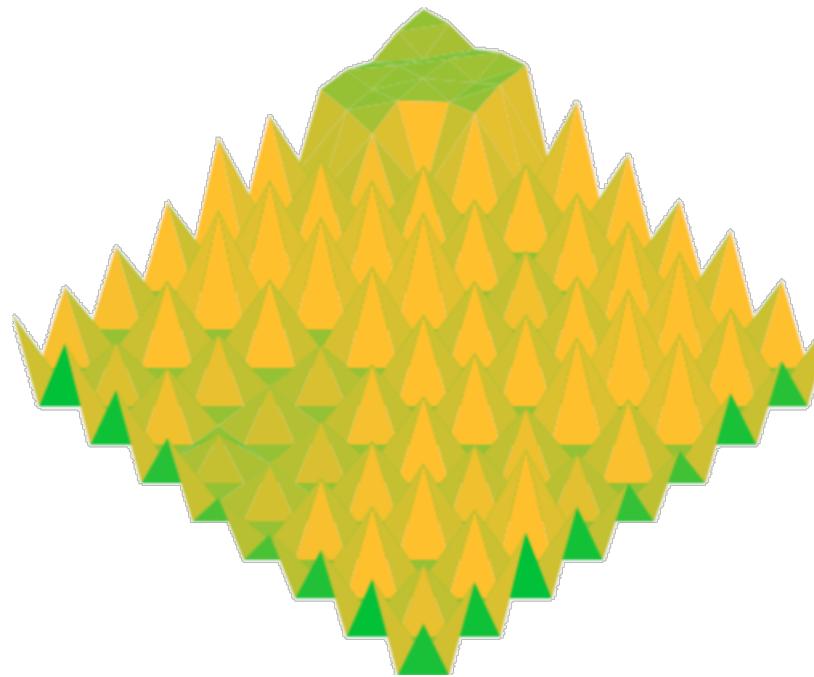
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

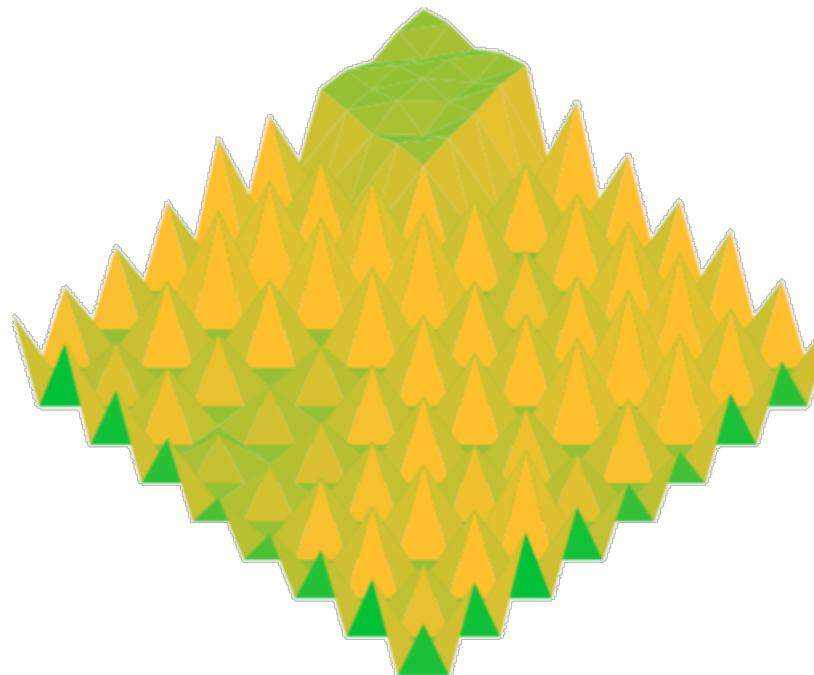
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

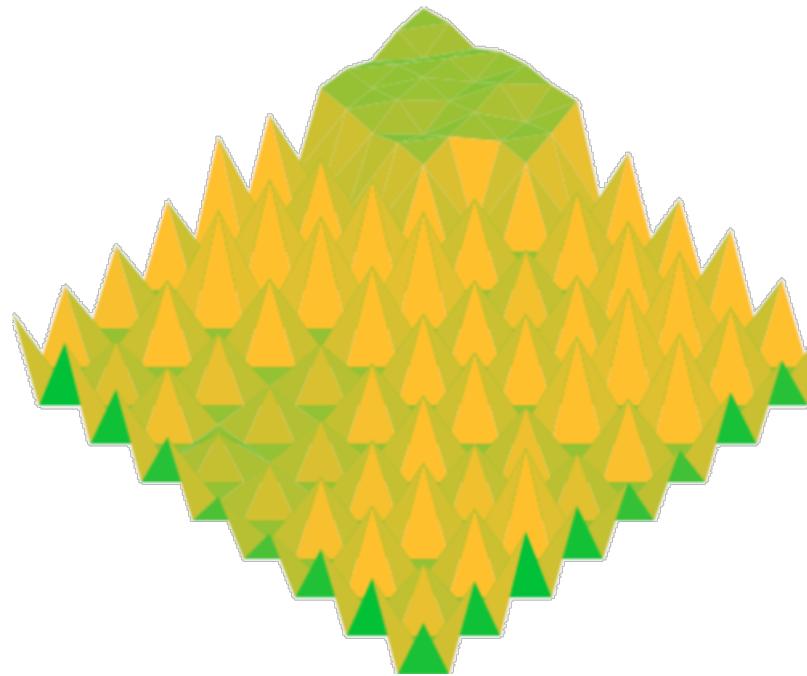
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

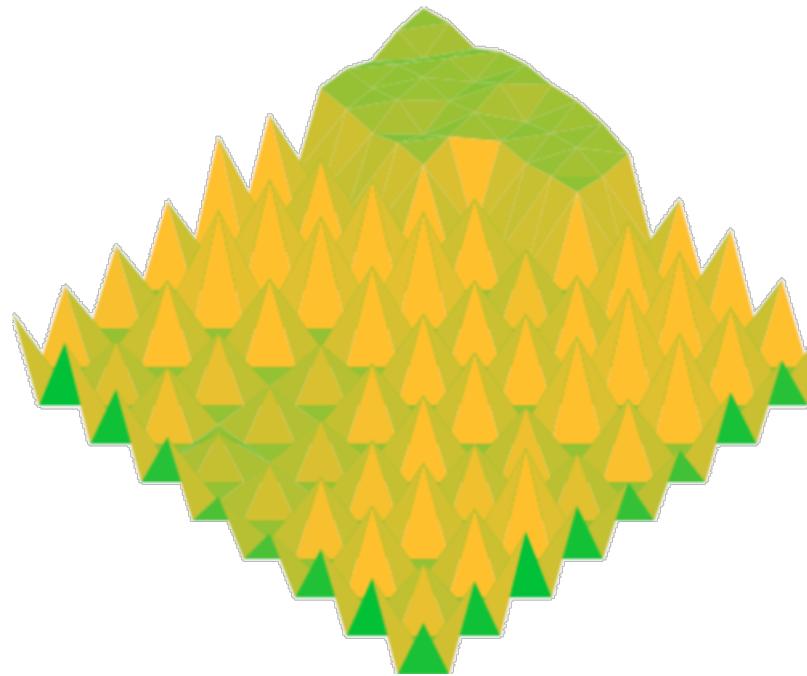
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

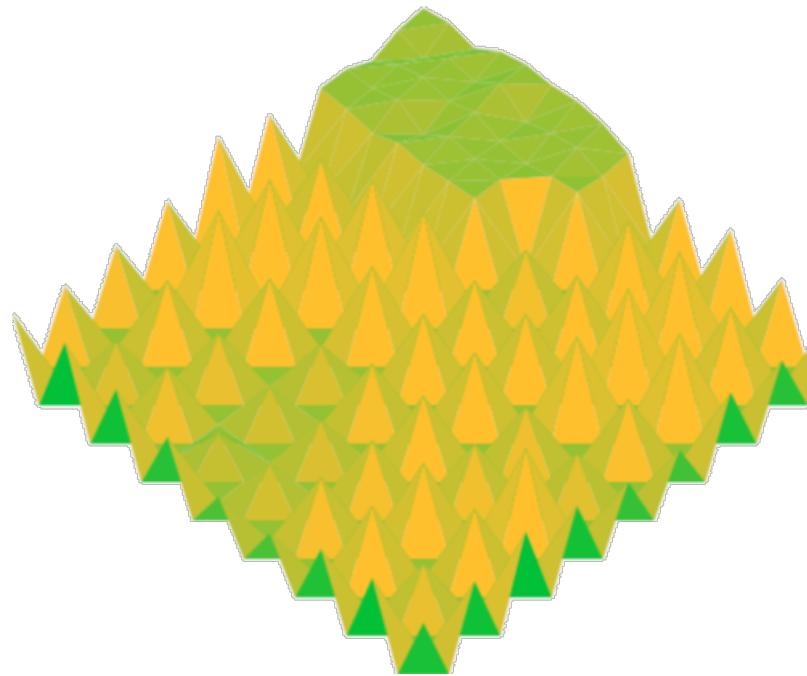
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

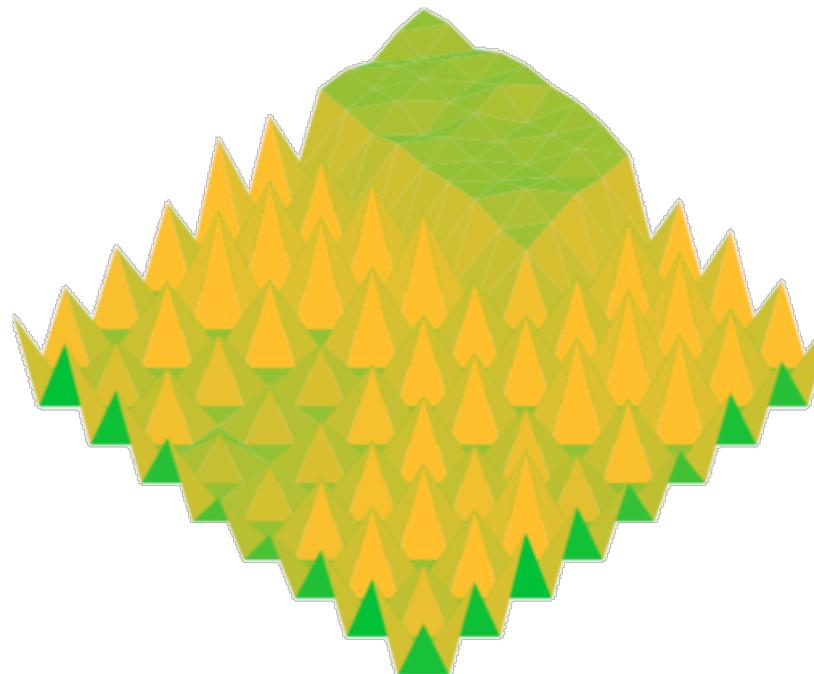
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

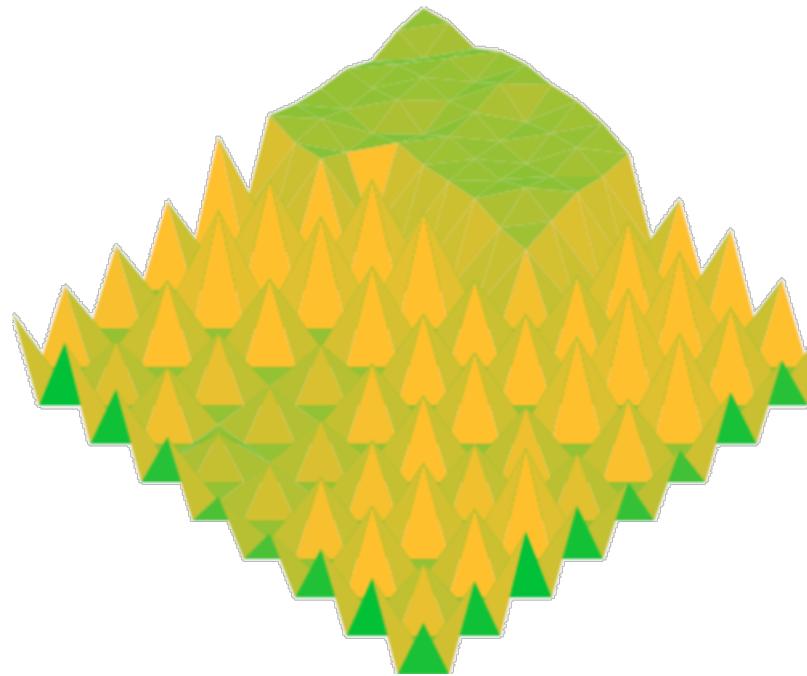
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

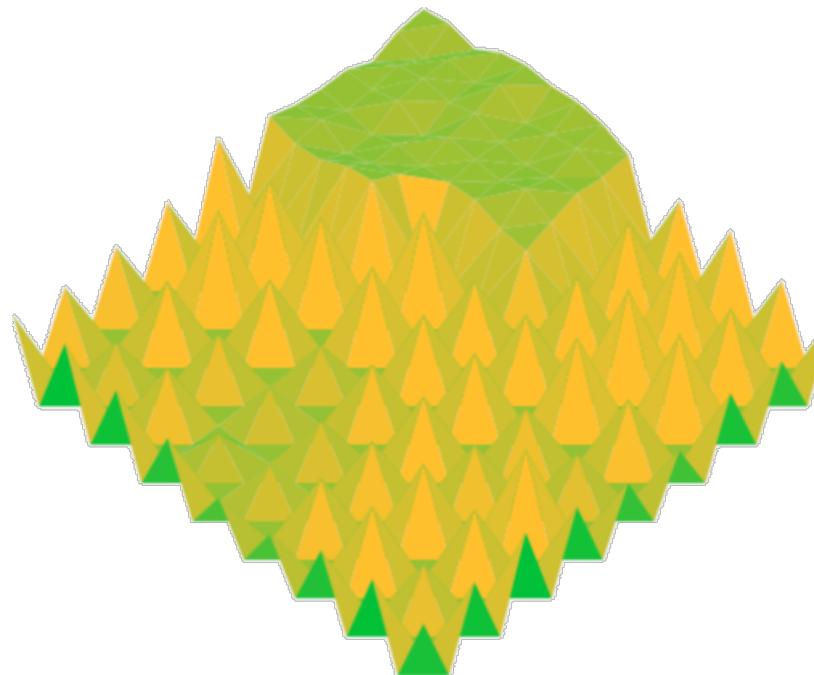
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

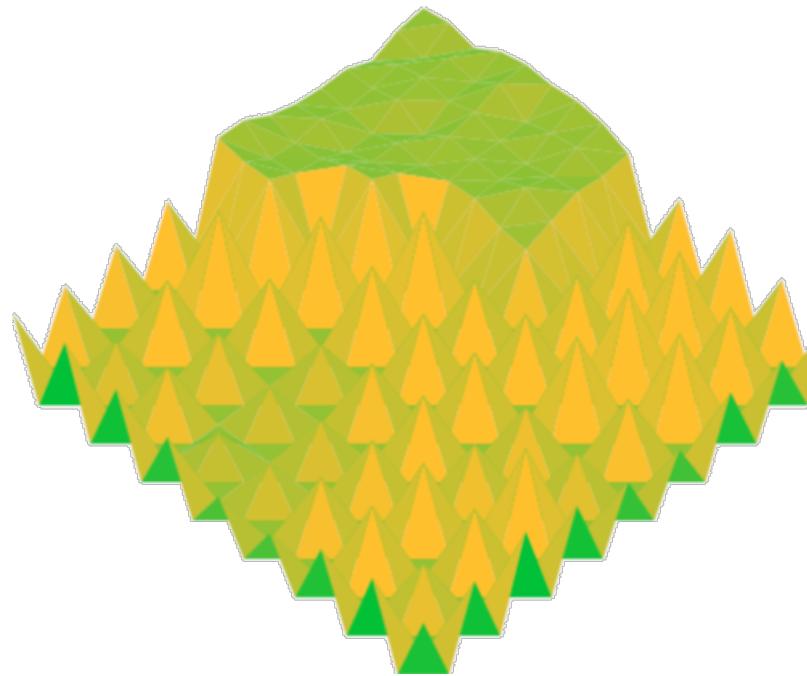
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

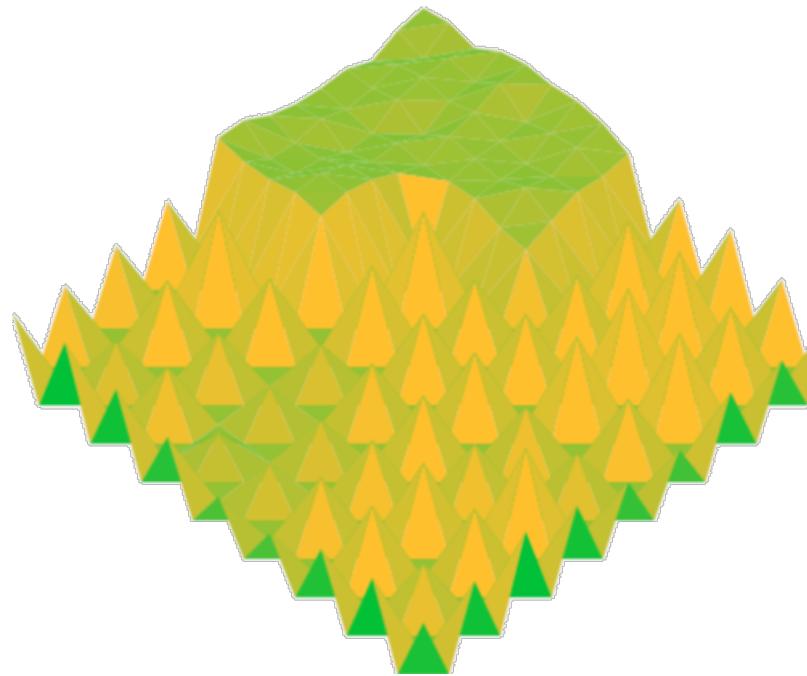
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

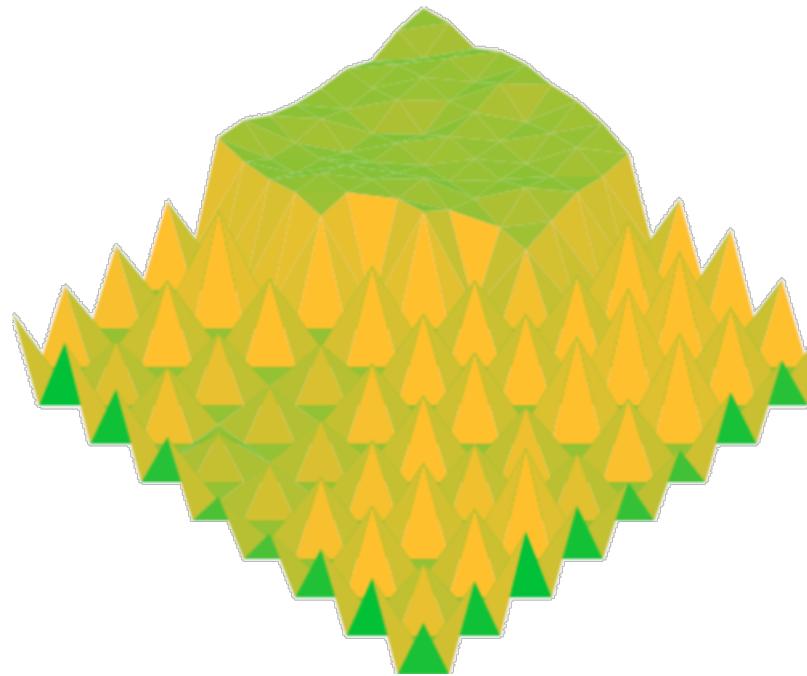
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

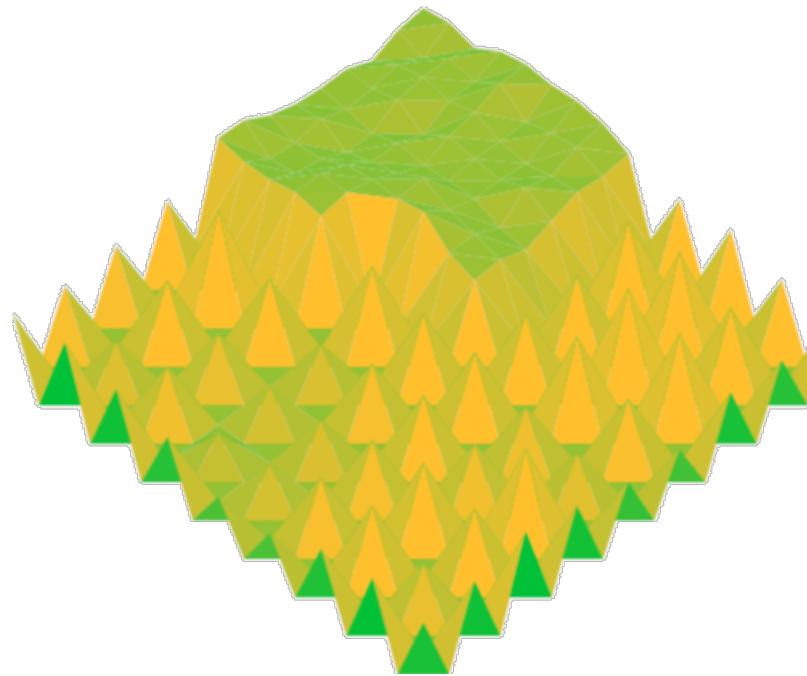
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

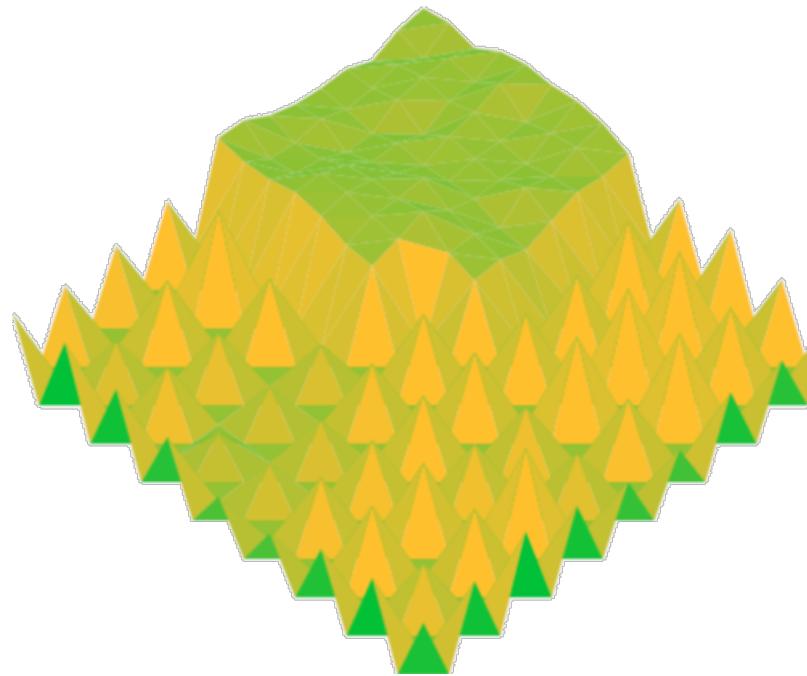
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

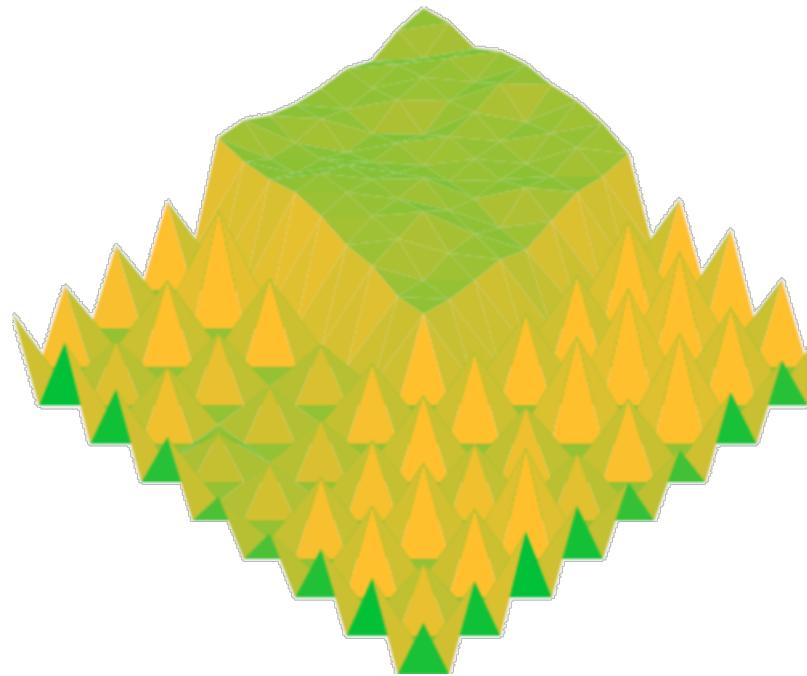
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

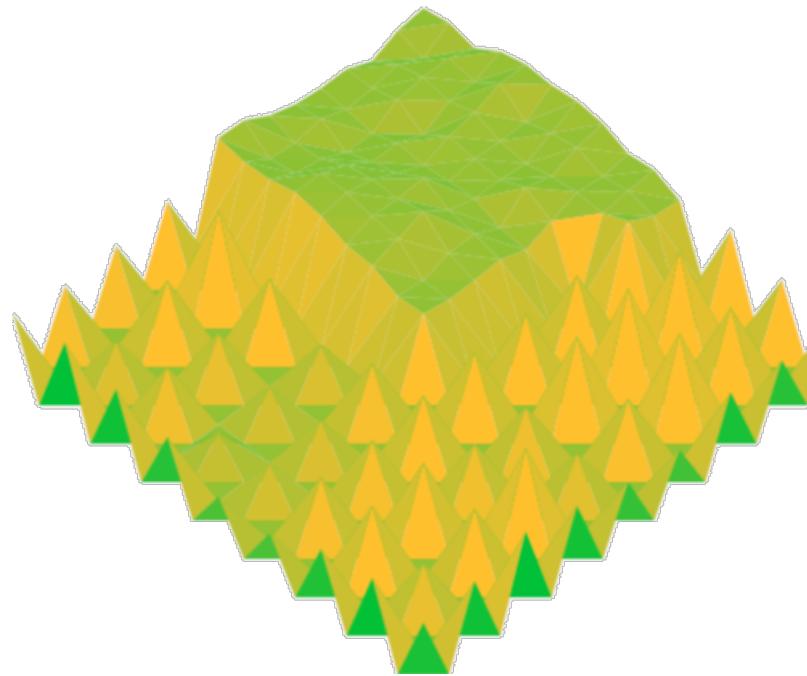
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

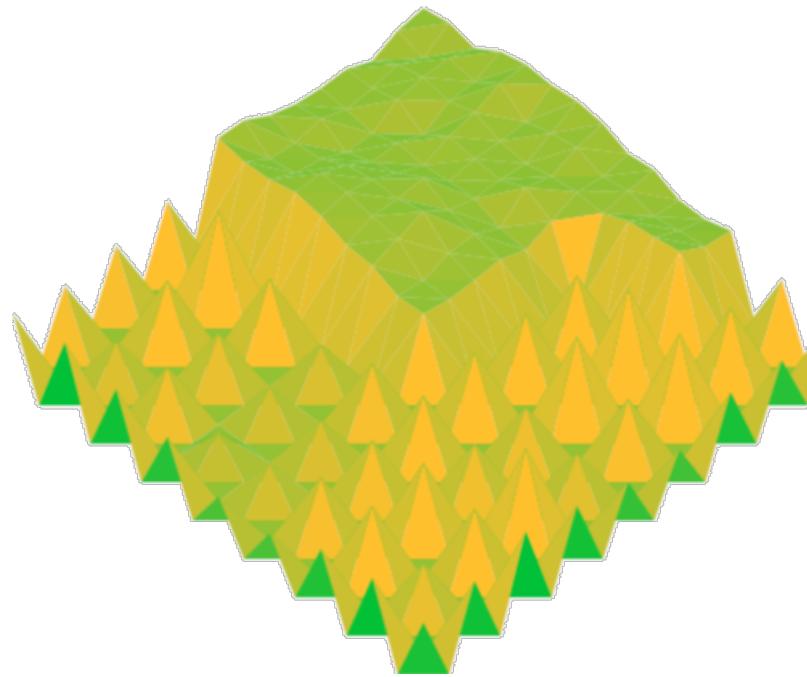
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

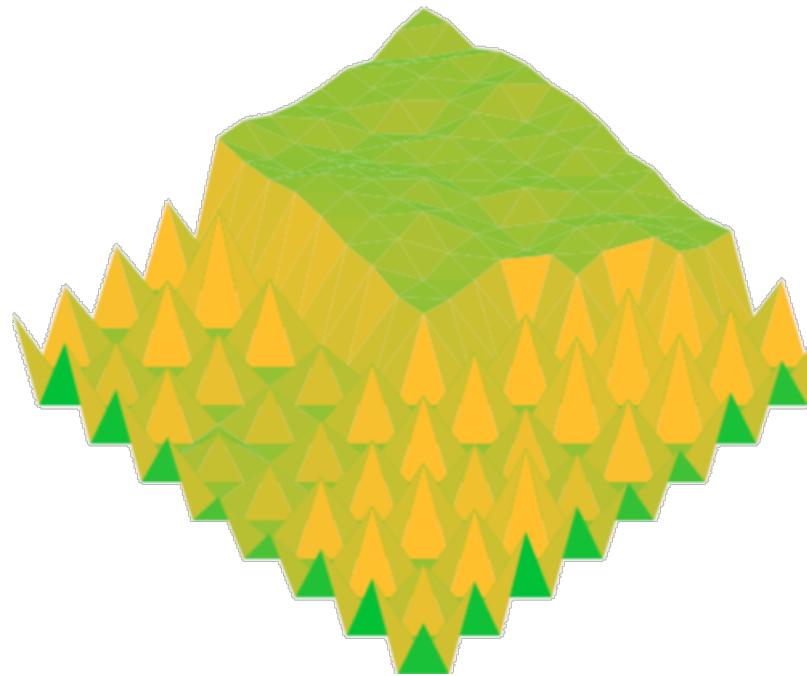
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

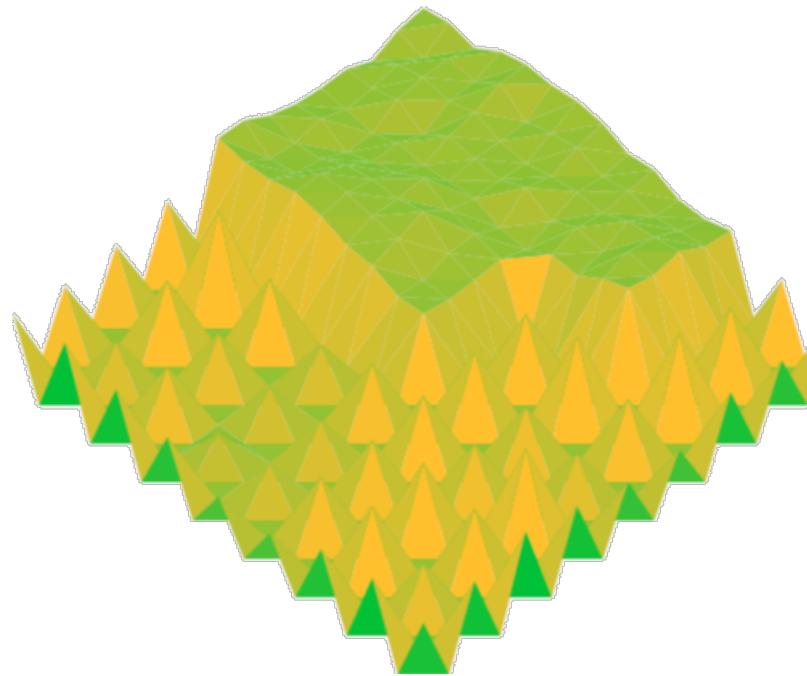
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

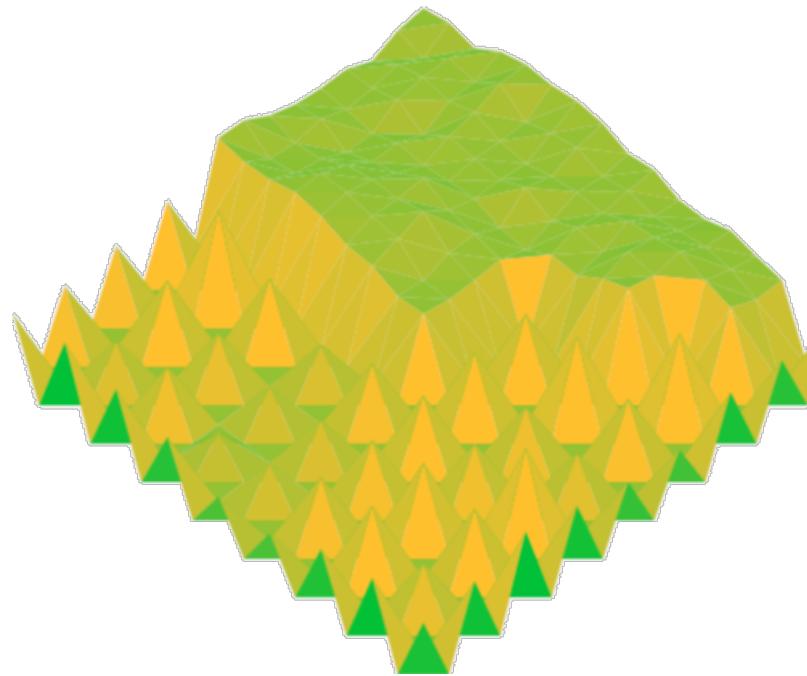
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

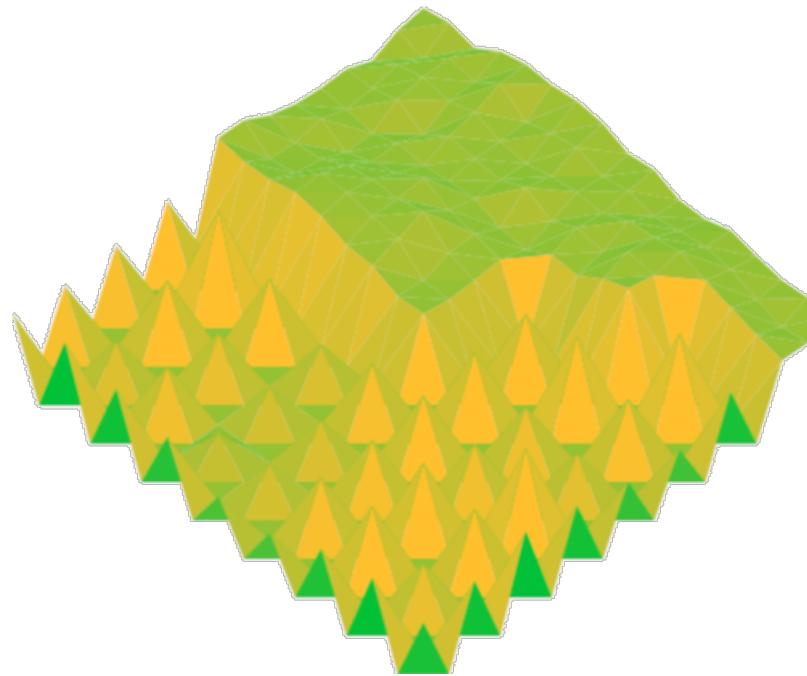
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

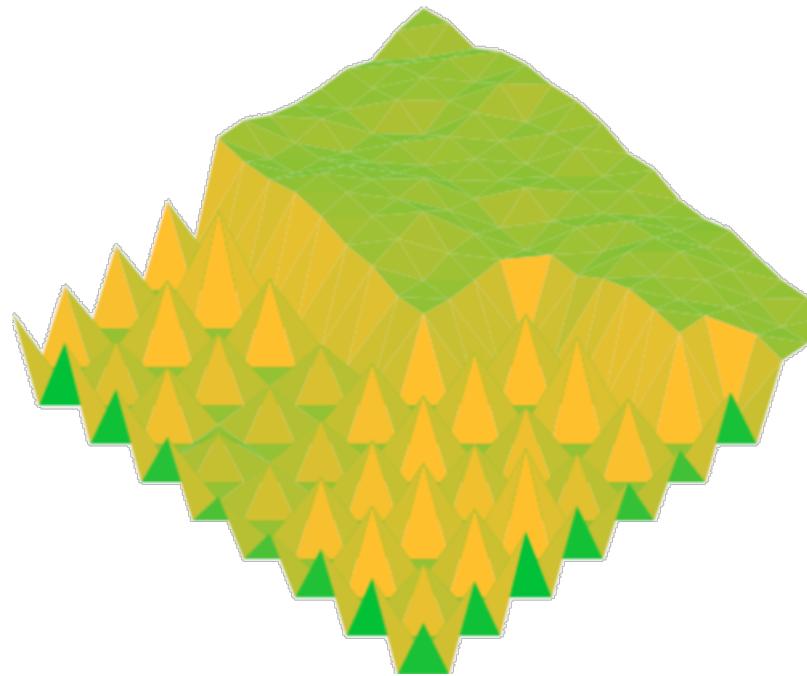
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

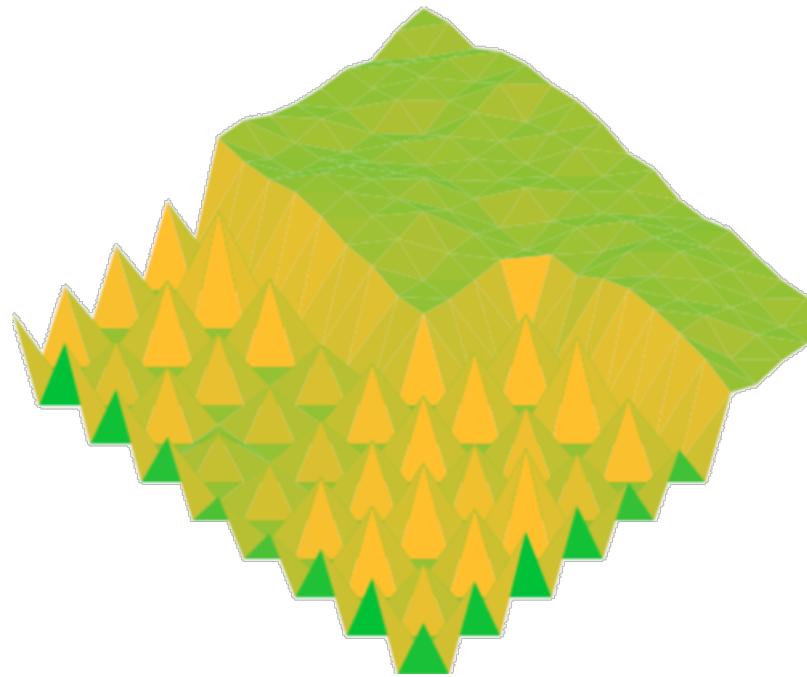
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

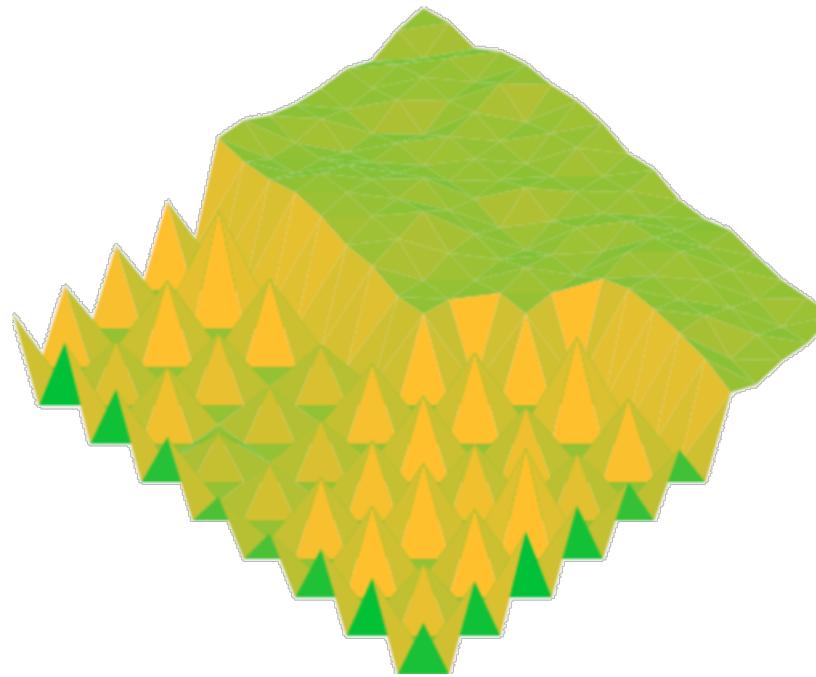
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

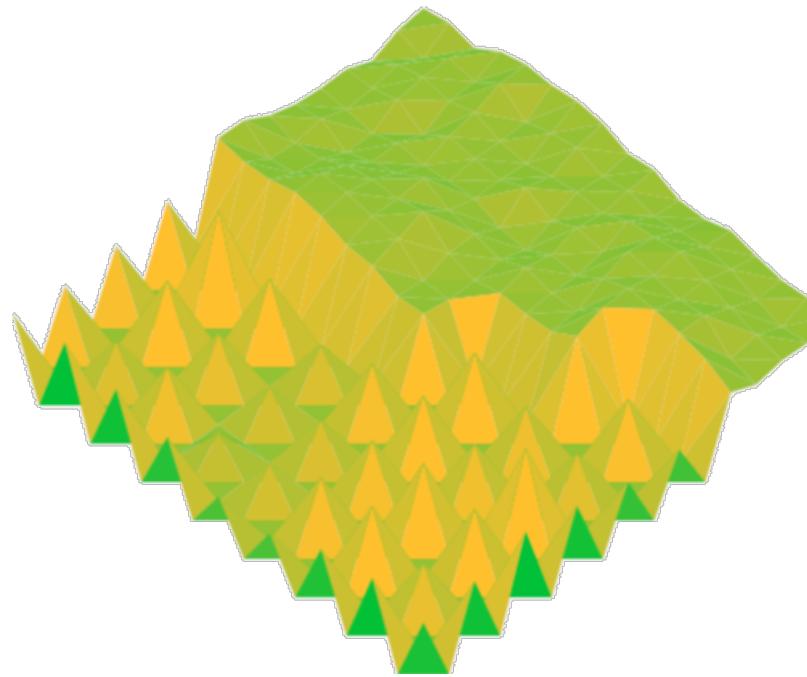
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

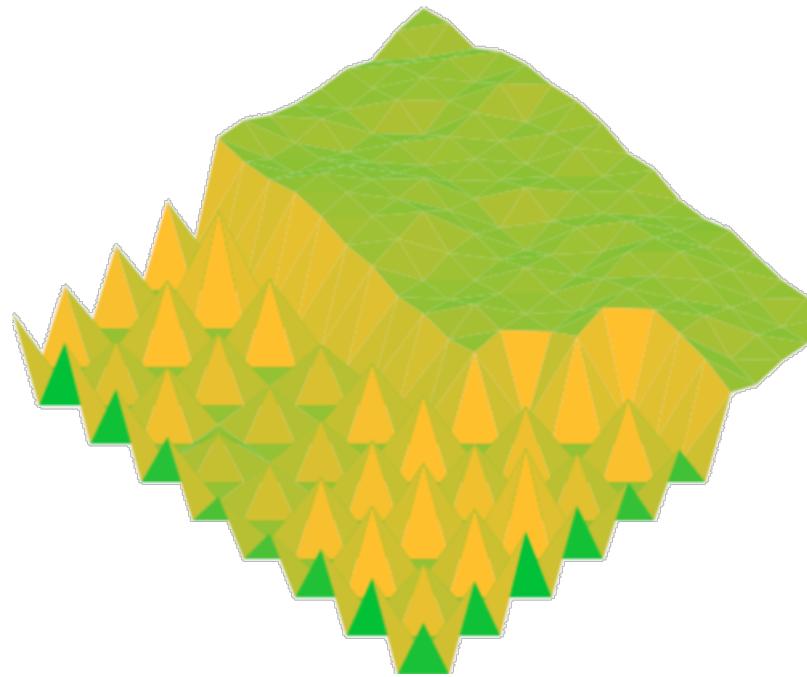
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

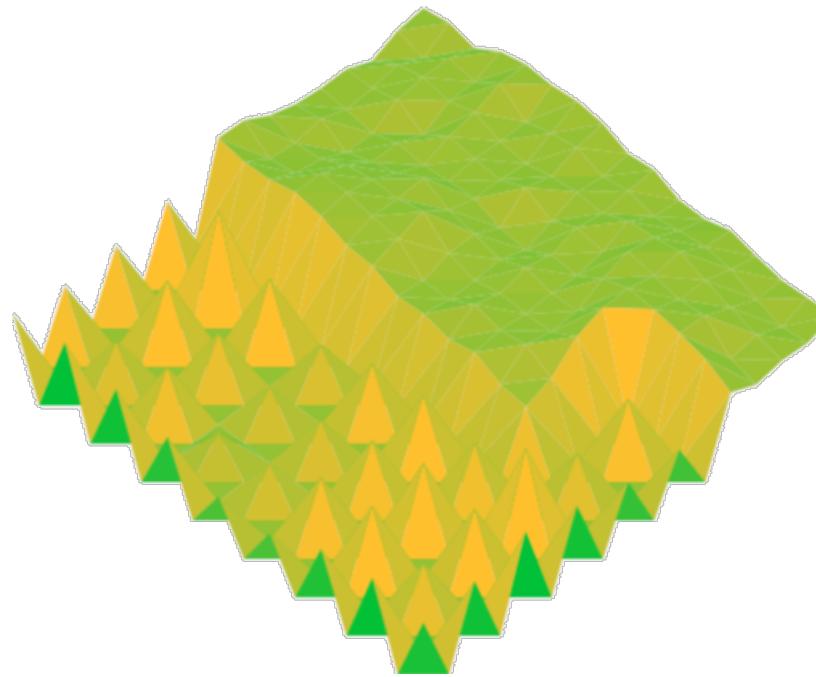
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

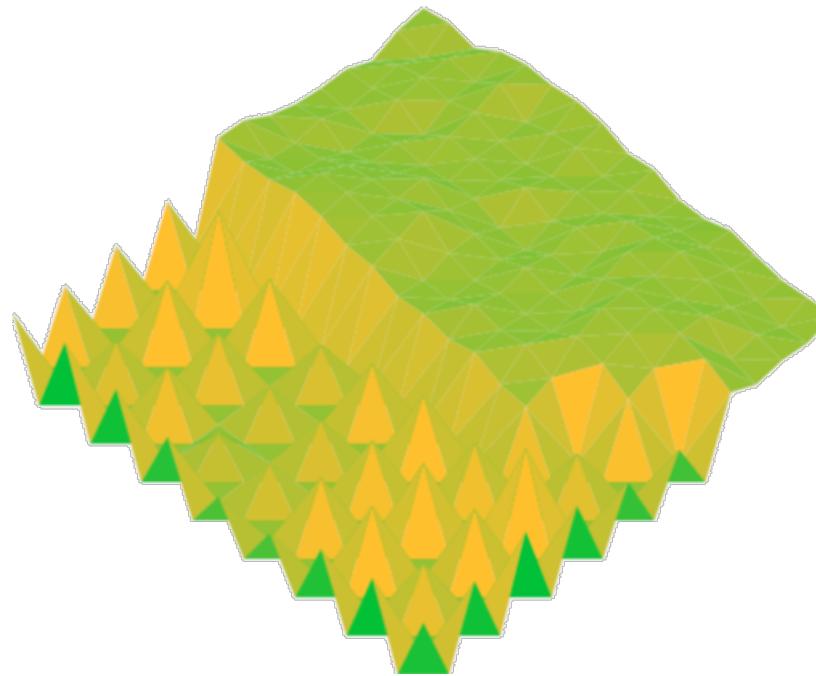
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

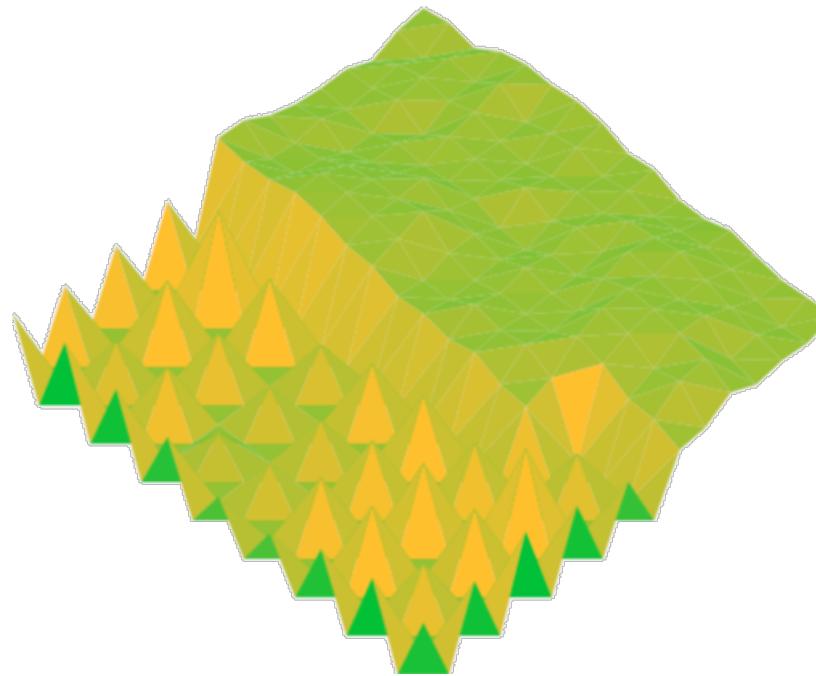
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

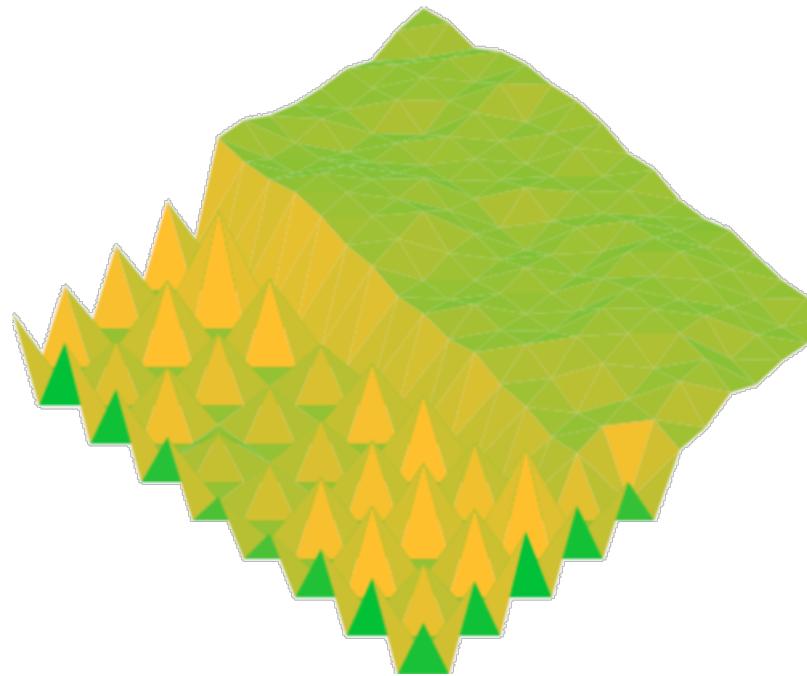
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

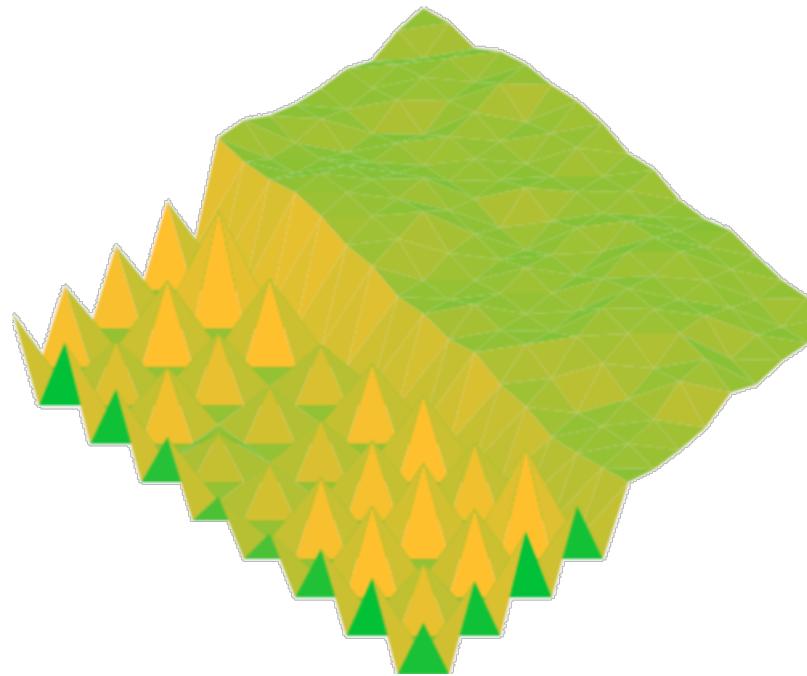
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

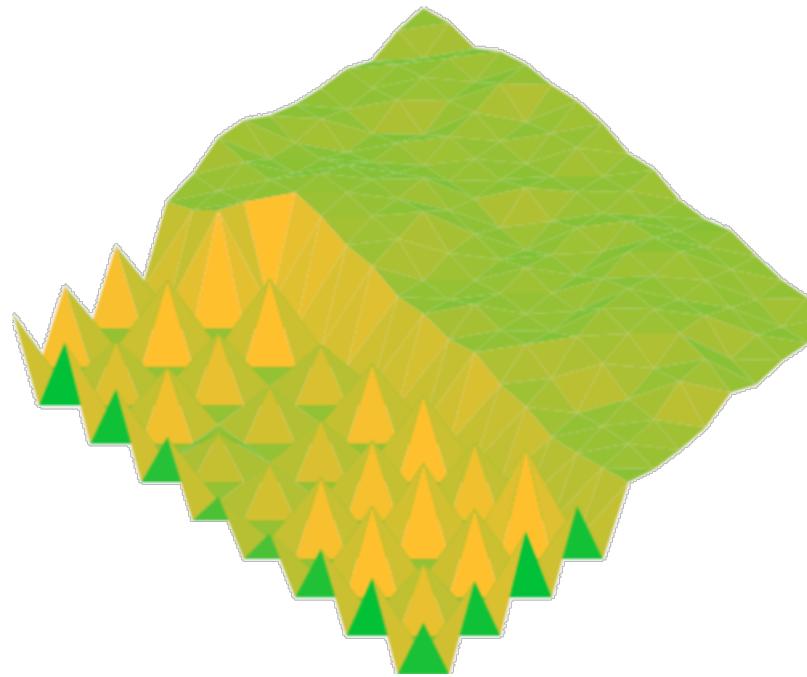
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

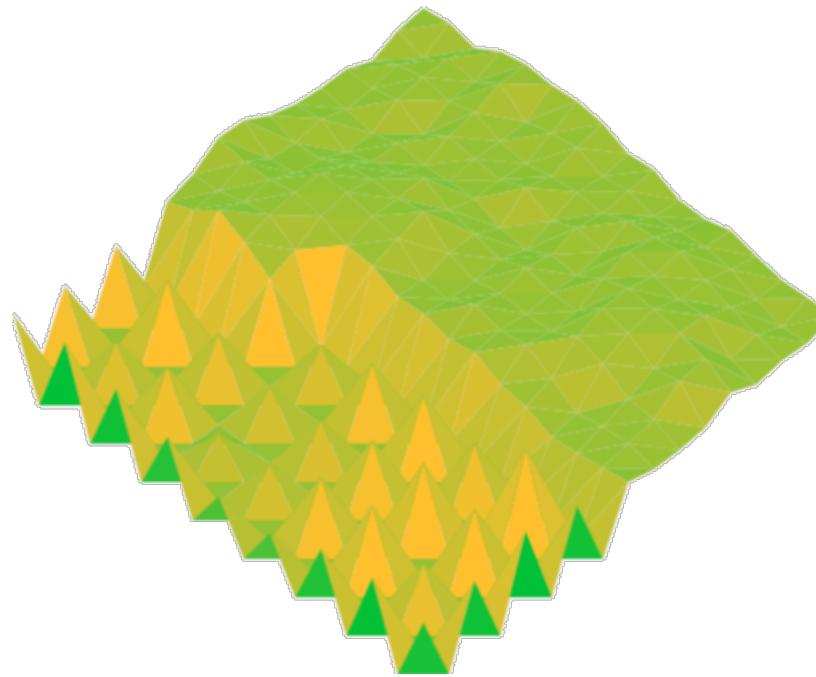
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

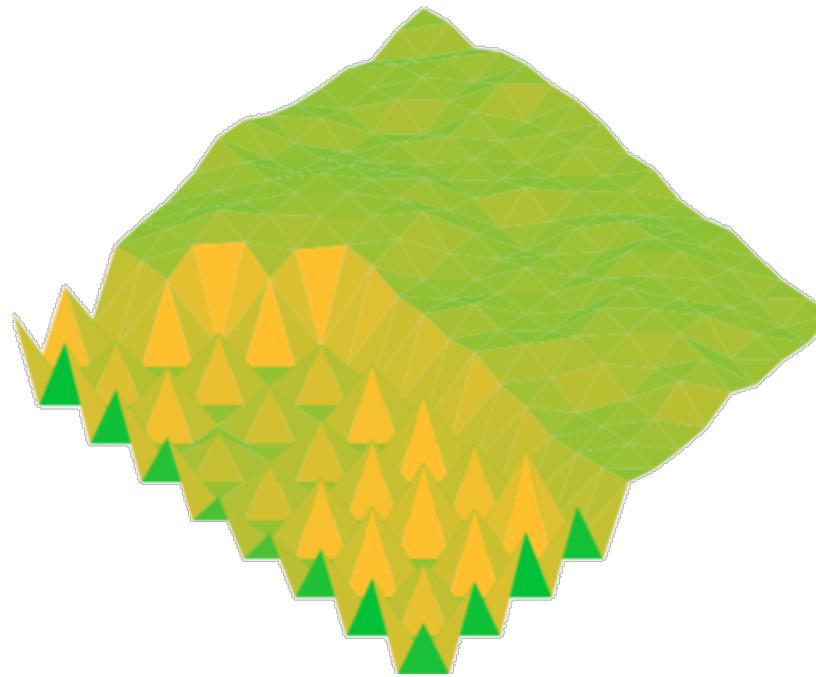
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

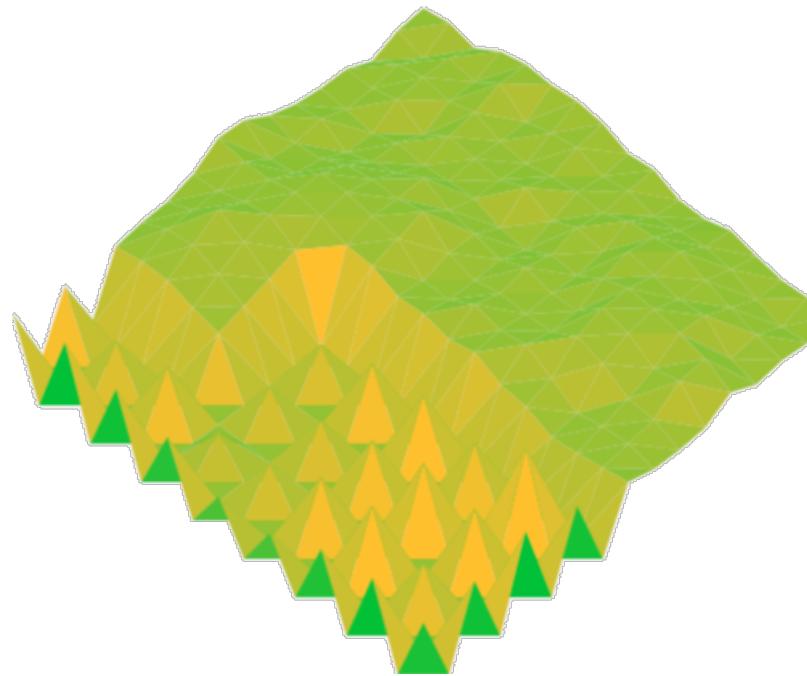
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

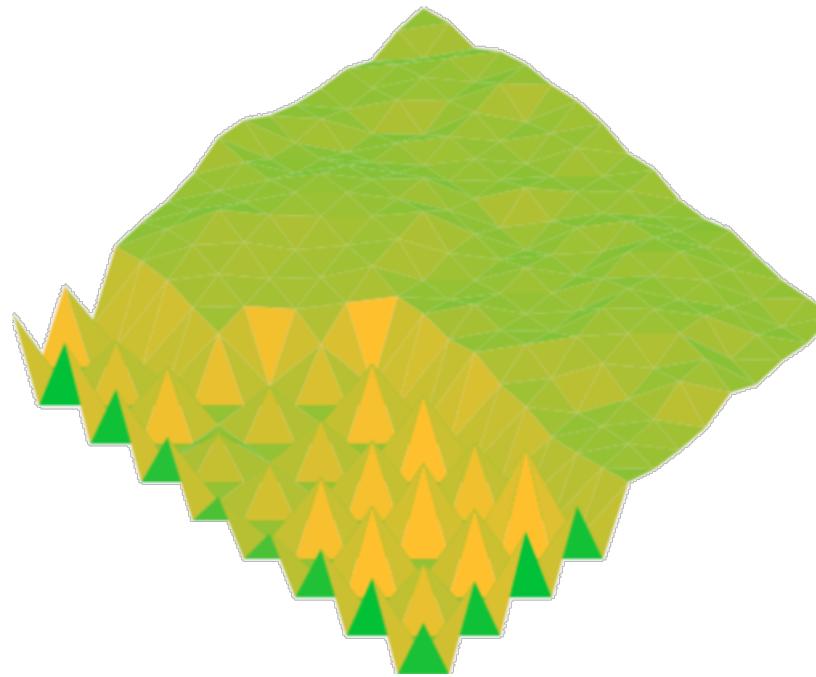
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

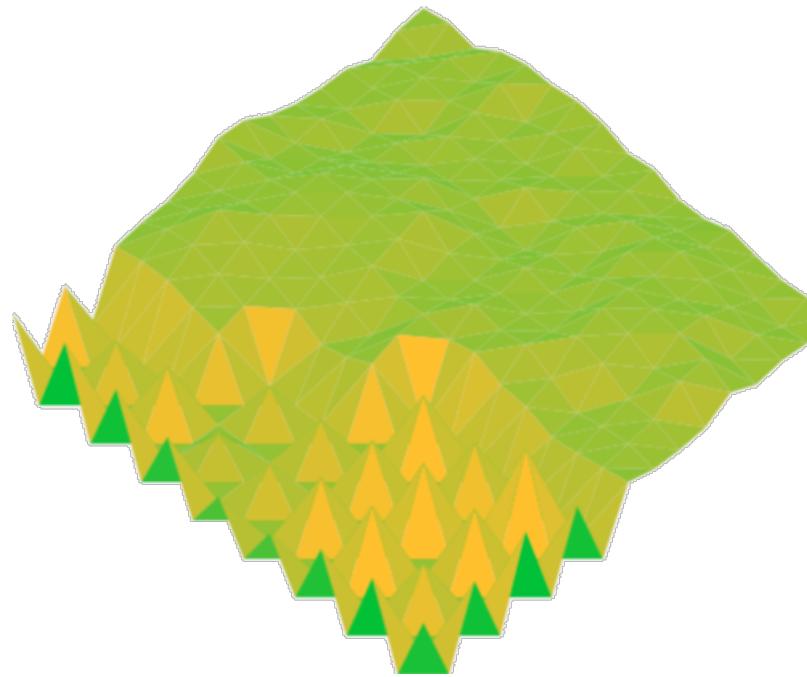
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

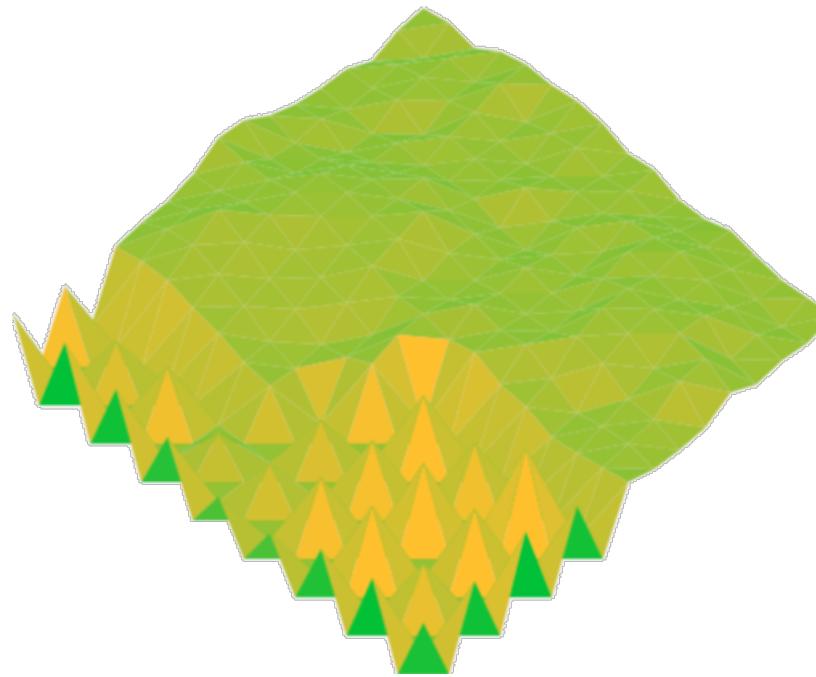
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

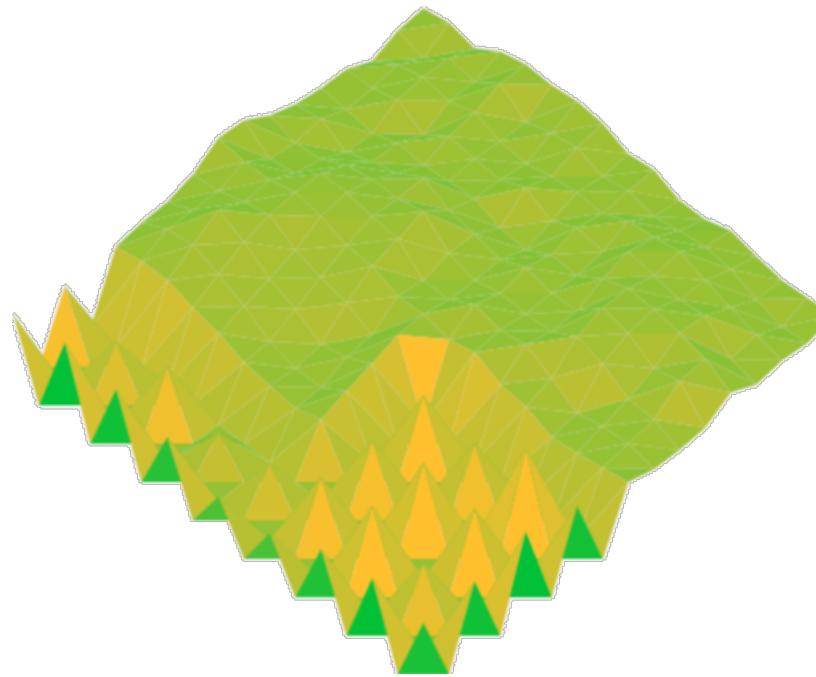
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

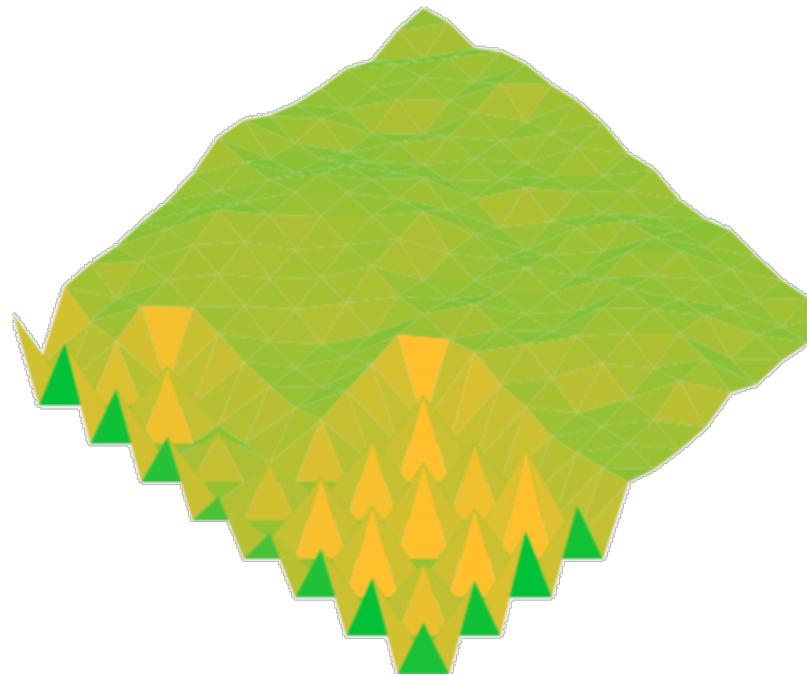
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

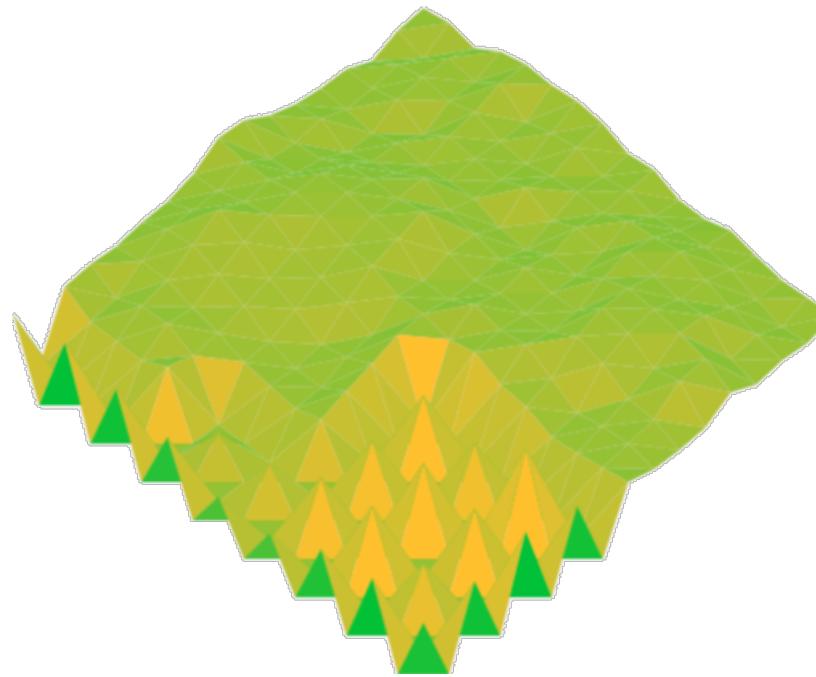
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

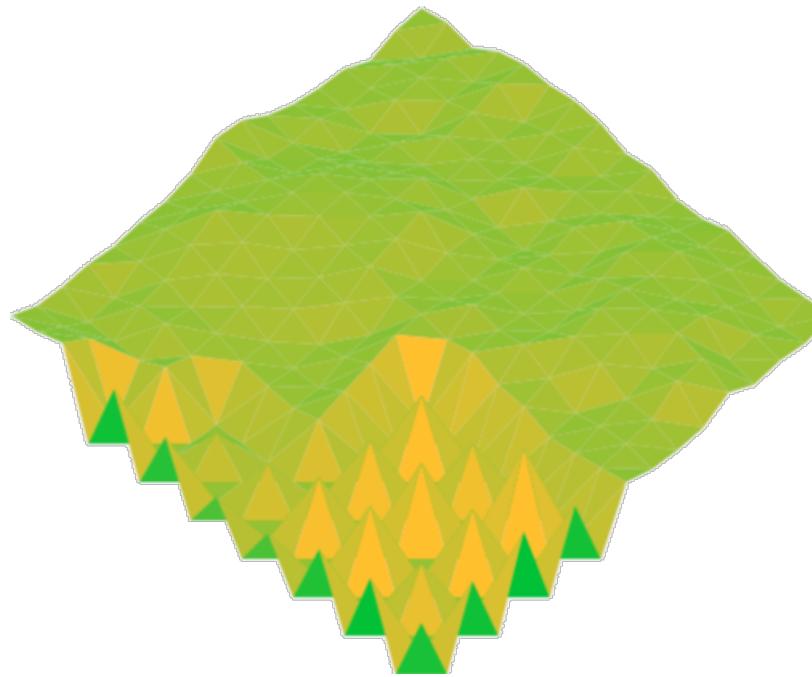
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

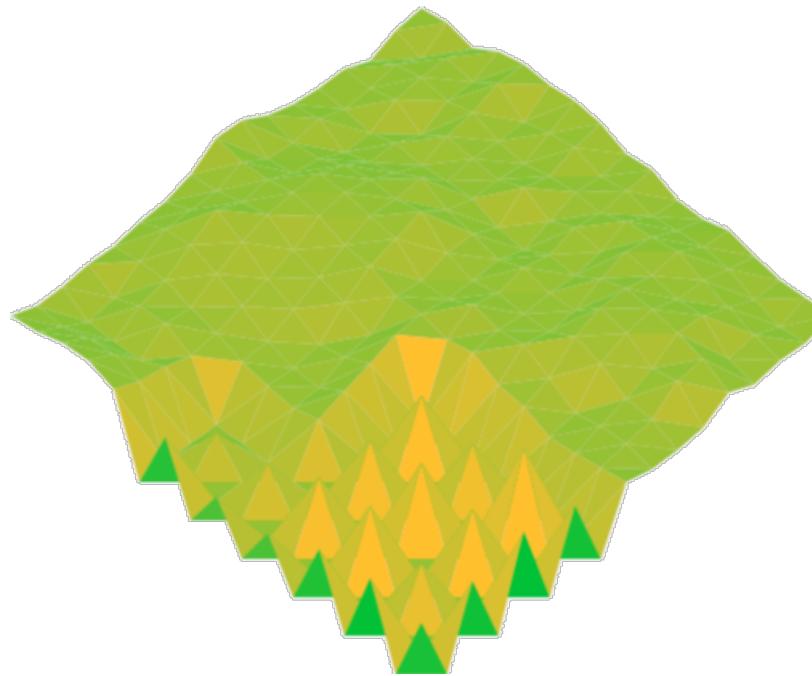
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

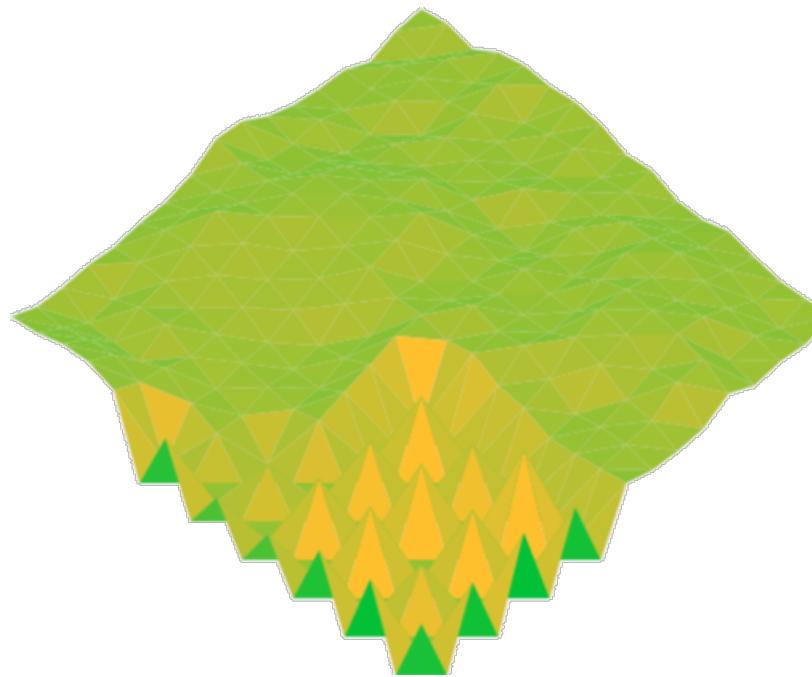
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

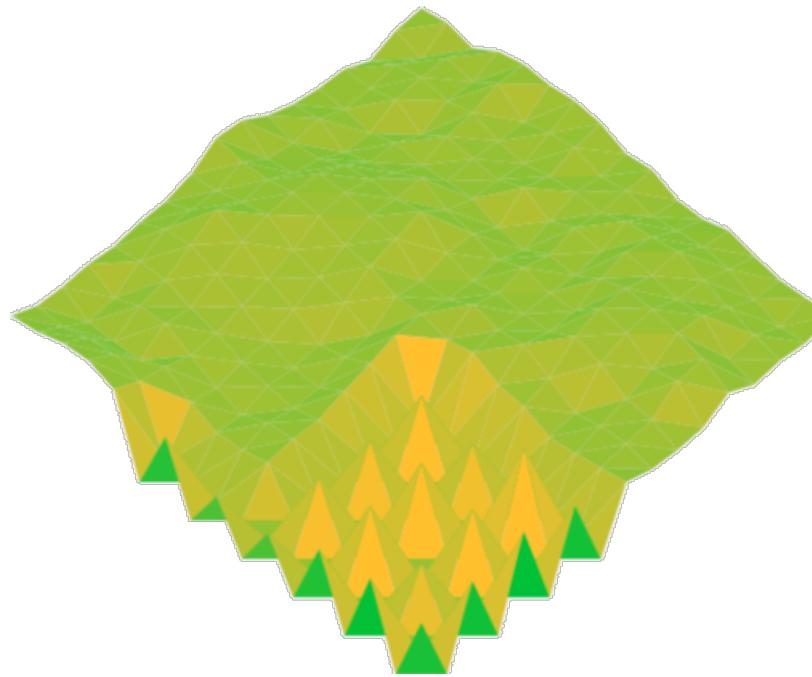
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

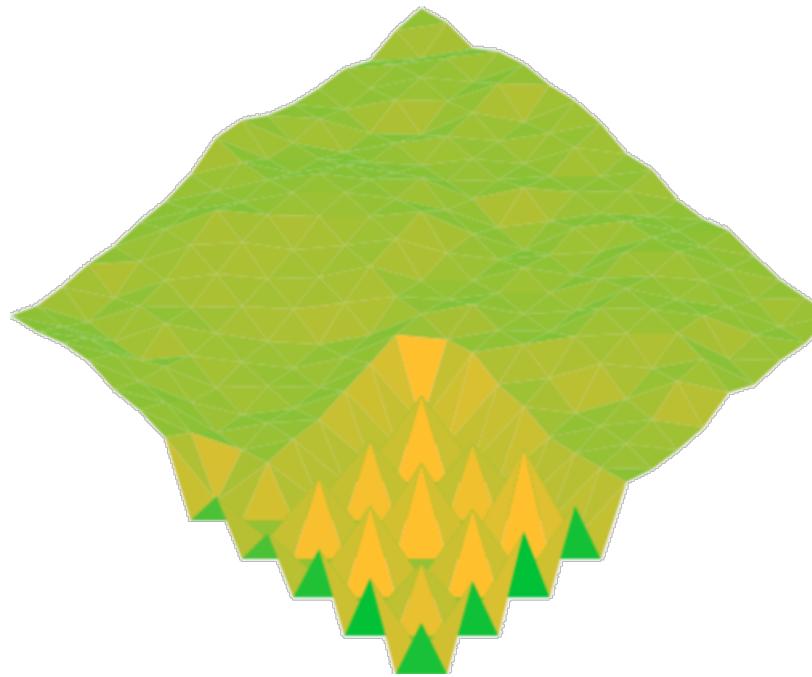
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

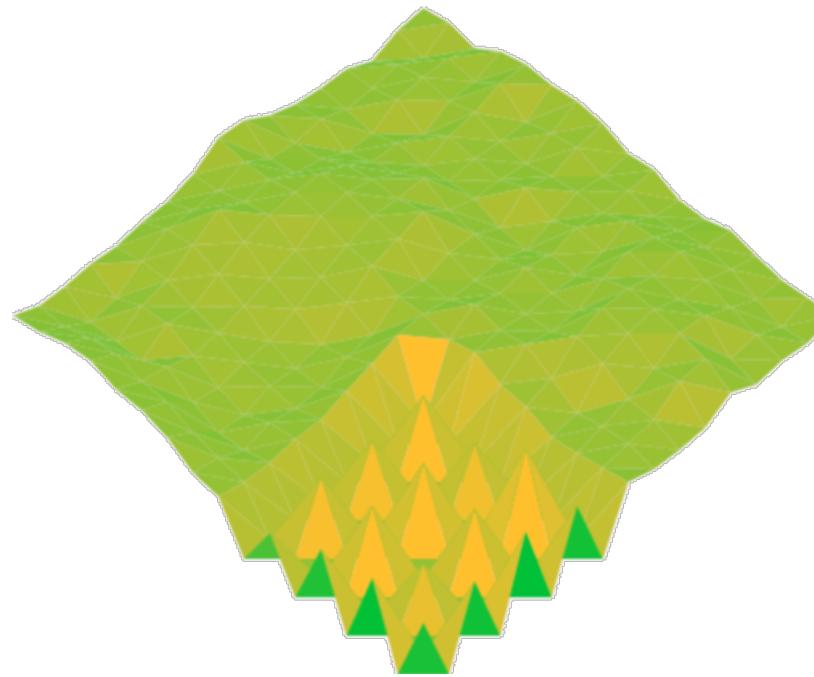
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

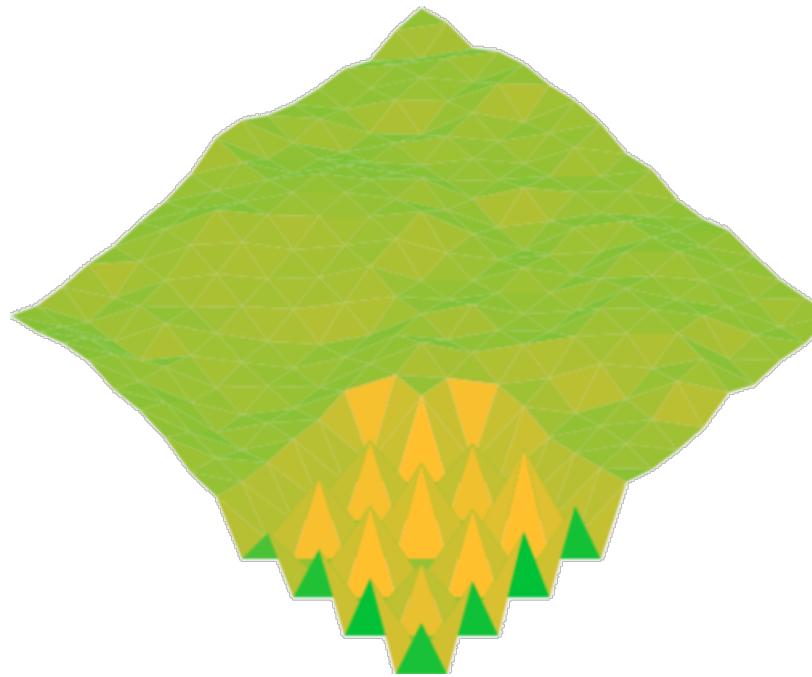
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

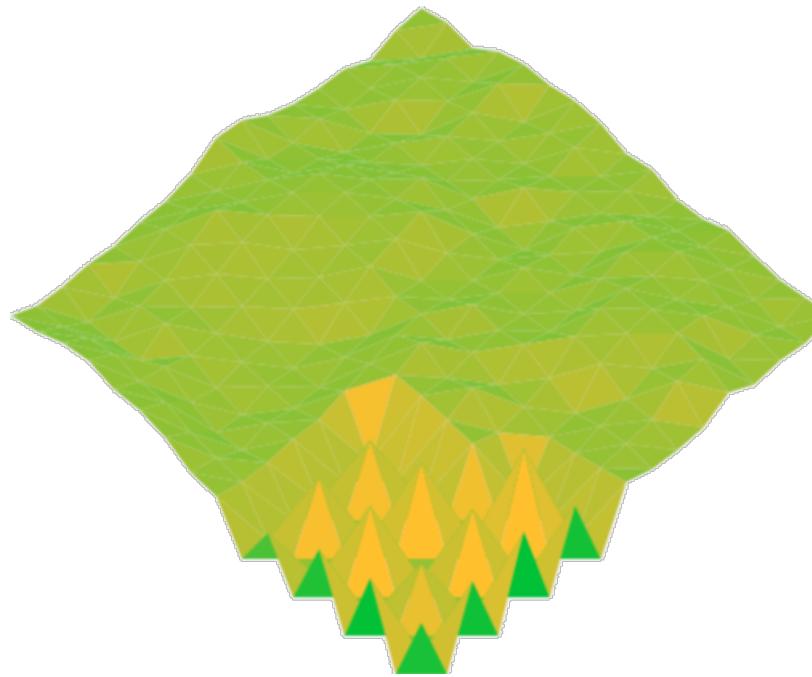
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

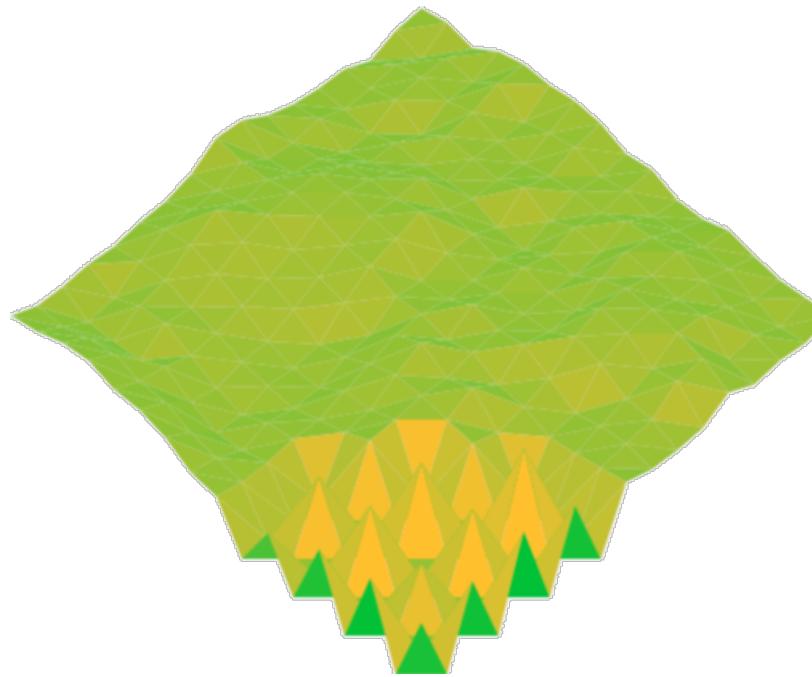
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

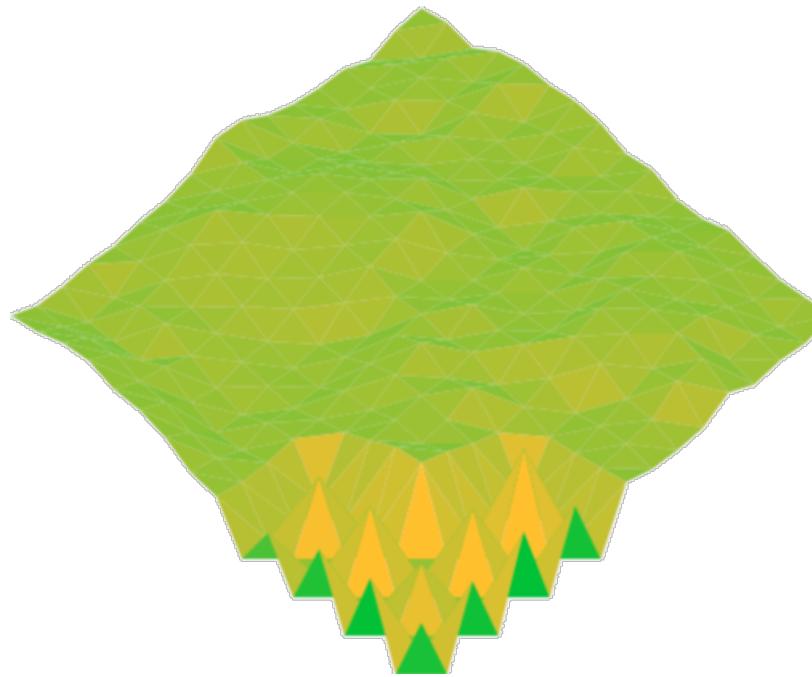
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

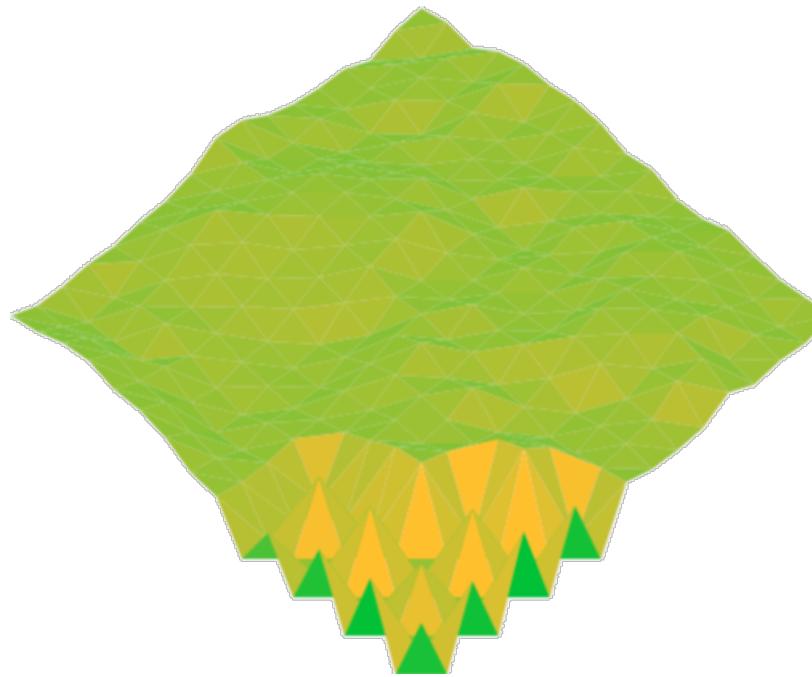
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

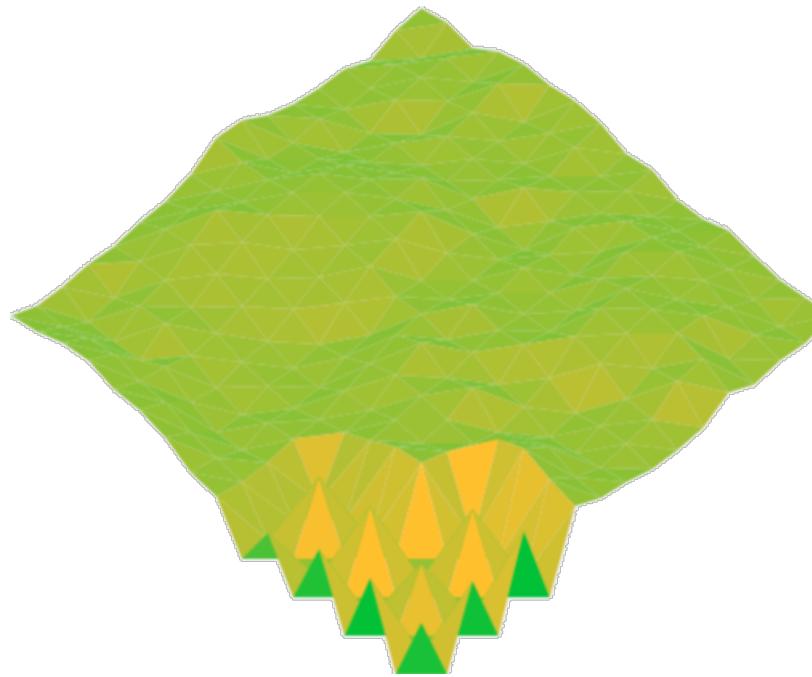
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

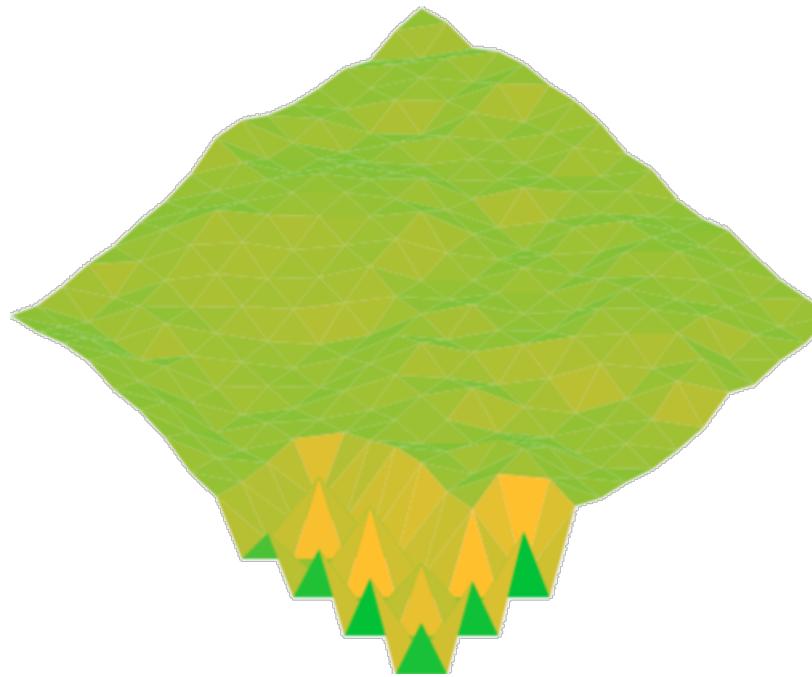
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

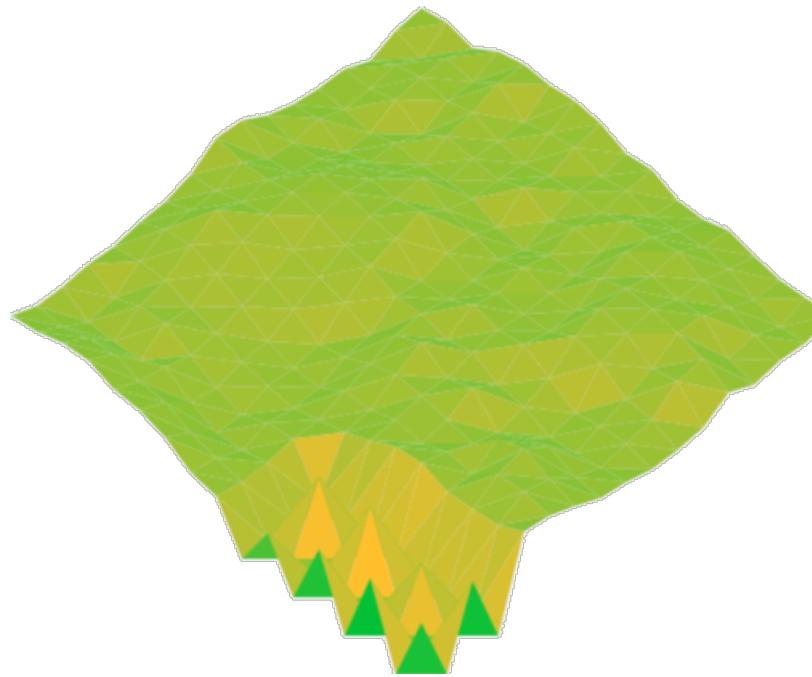
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

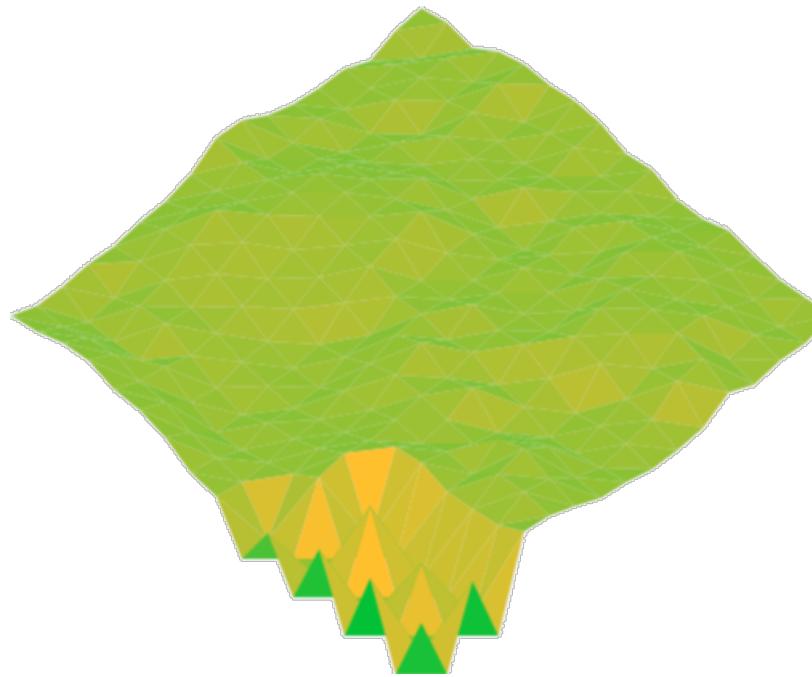
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

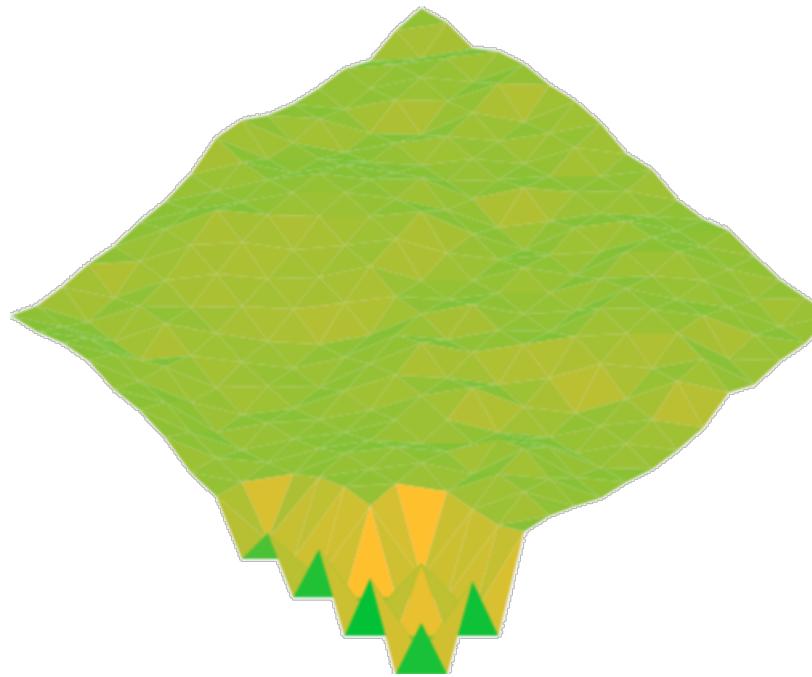
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

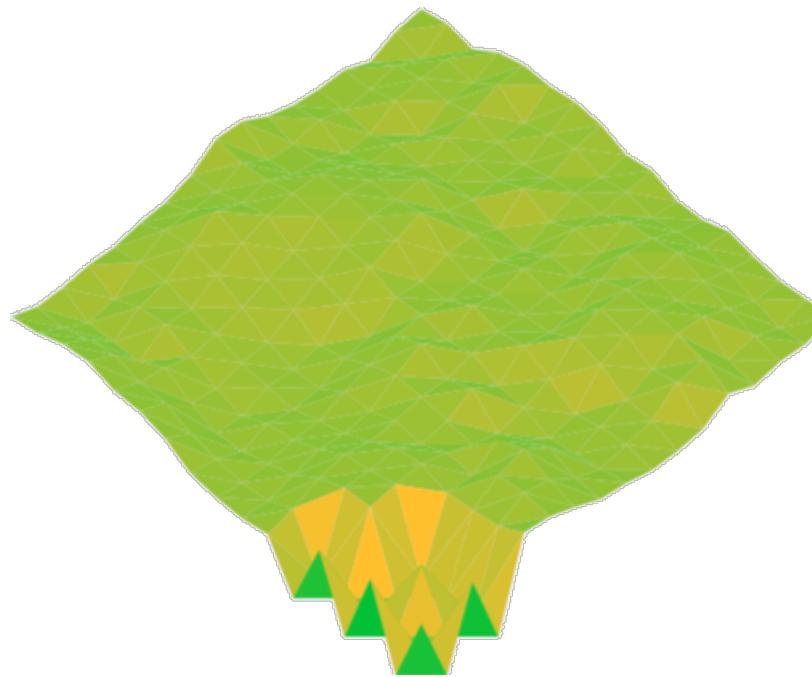
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

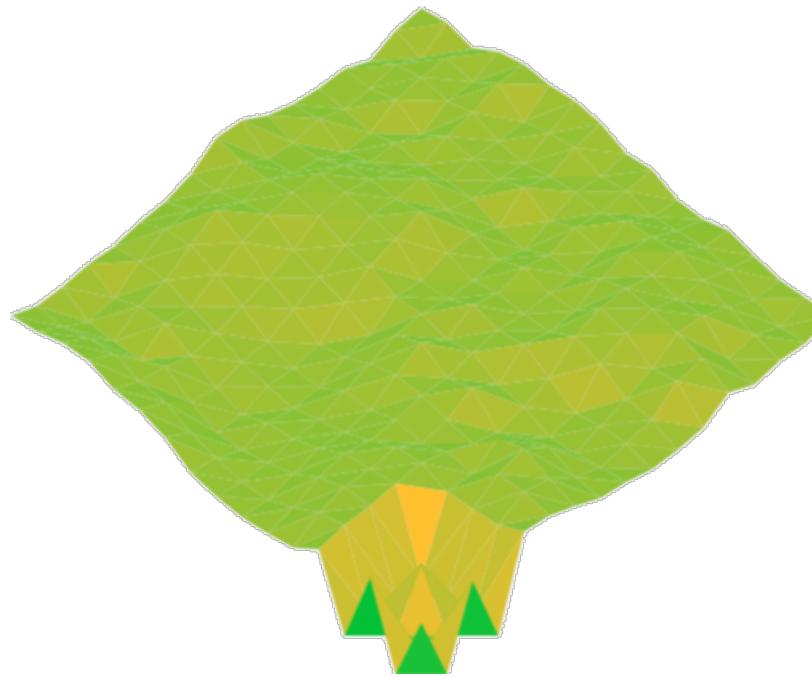
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

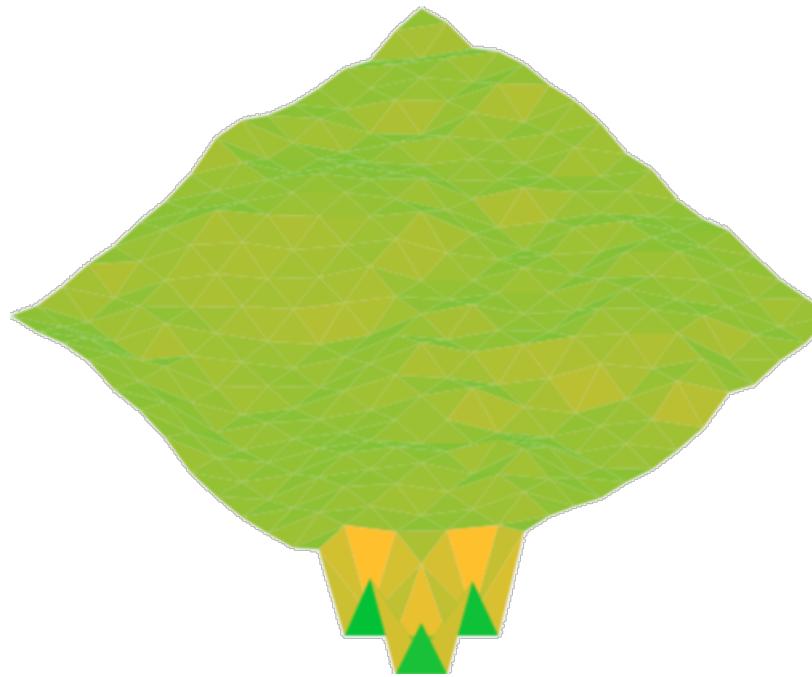
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

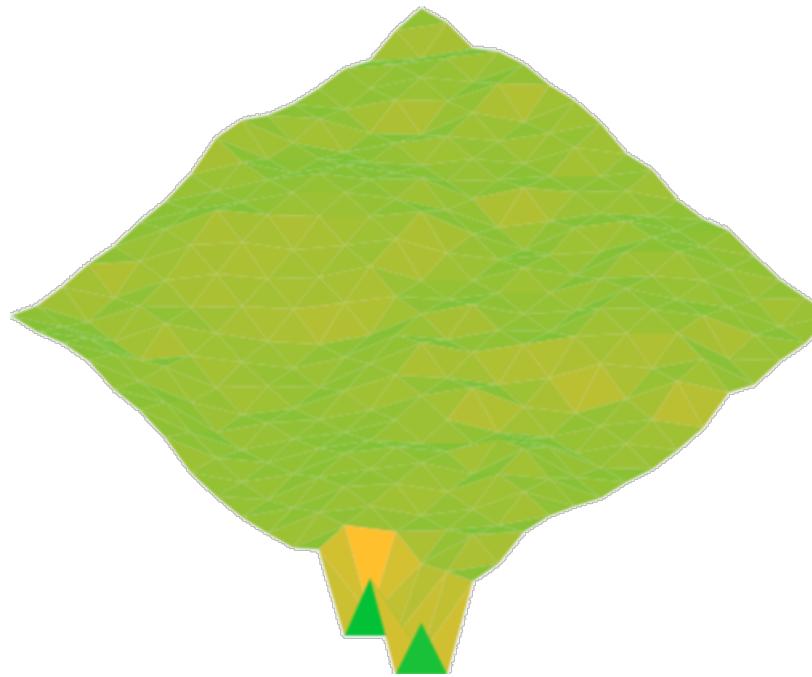
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

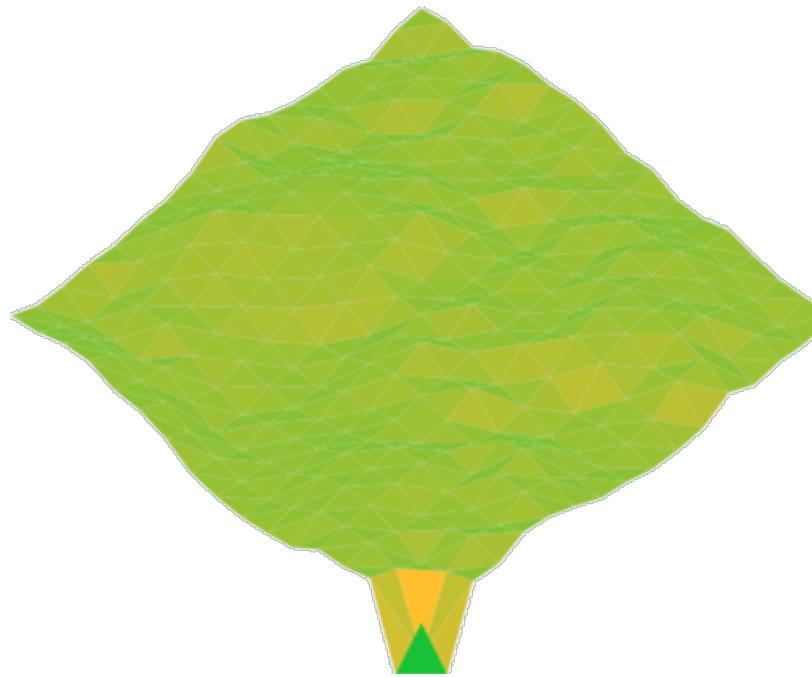
- Diamond-Square algorithm
  - Fault algorithm



# Random, hardly controllable terrain generation algorithms

---

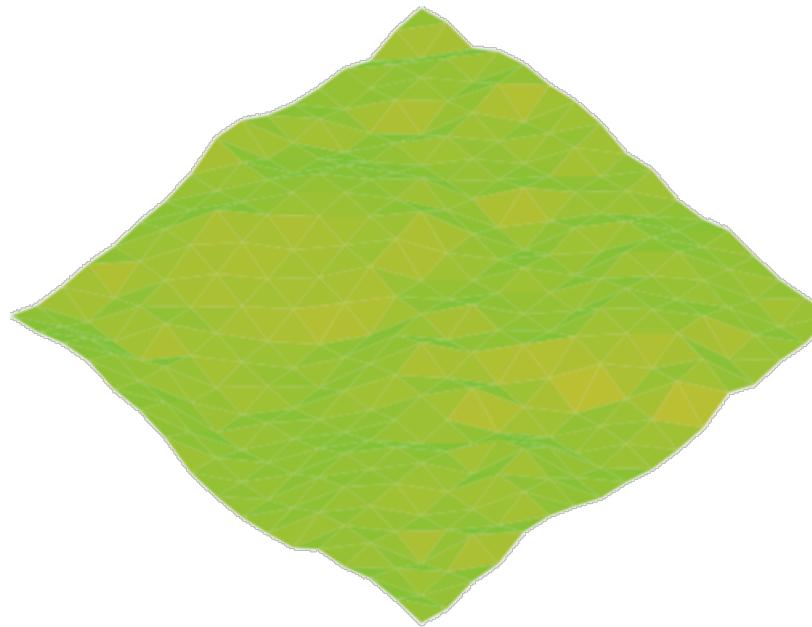
- Diamond-Square algorithm
  - Fault algorithm



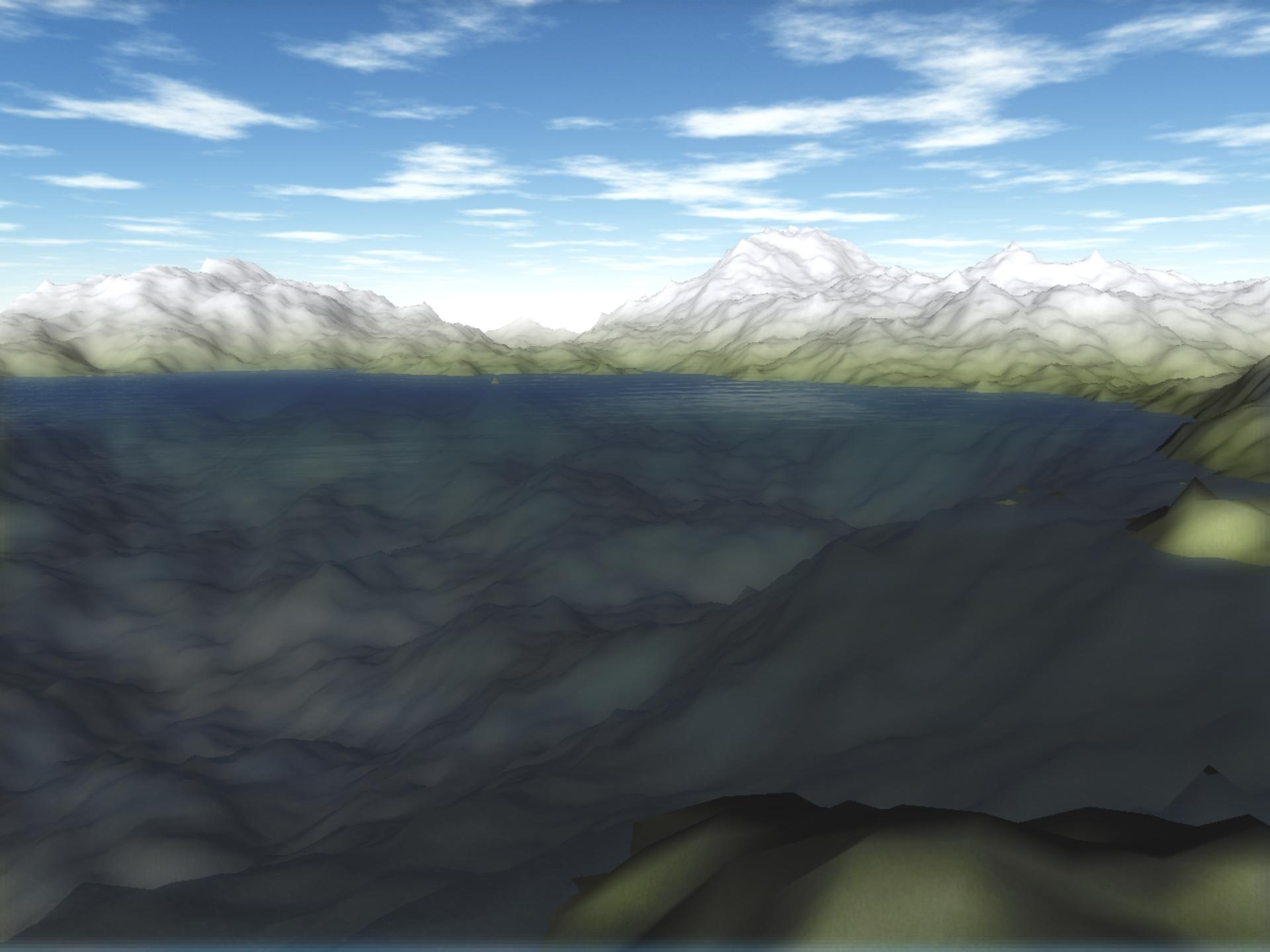
# Random, hardly controllable terrain generation algorithms

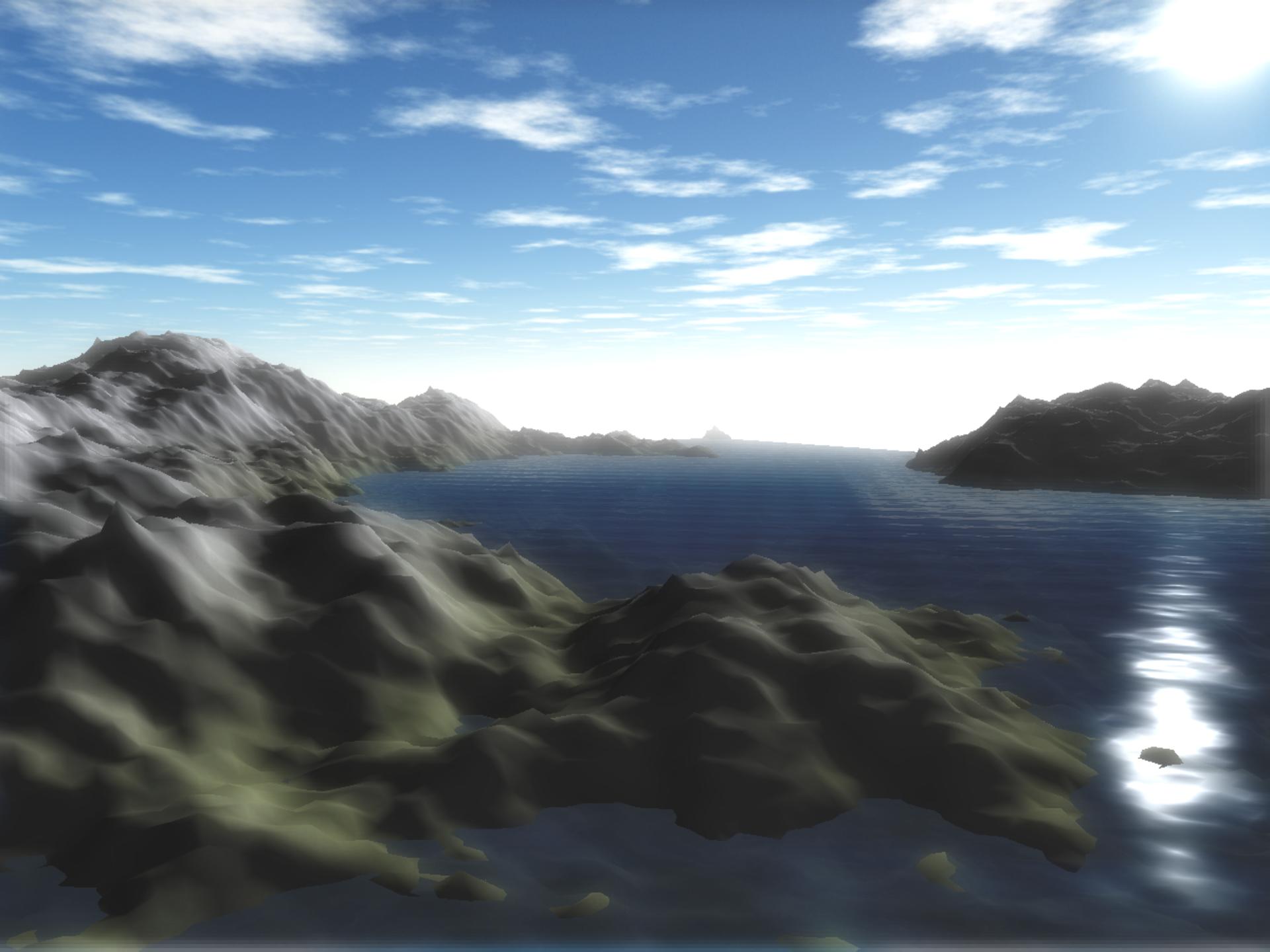
---

- Diamond-Square algorithm
  - Fault algorithm



Step visualization frames were generated using Paul Boxley's CoffeeScript script





# Random, hardly controllable terrain generation algorithms

---

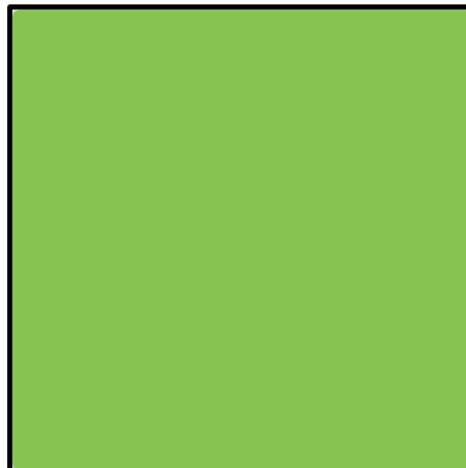
- Diamond-Square algorithm
- Fault algorithm

# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

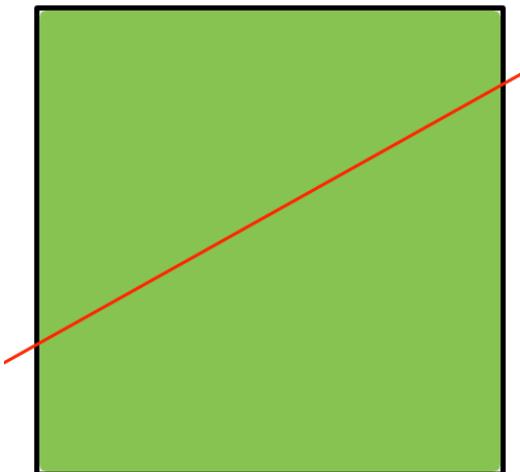


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

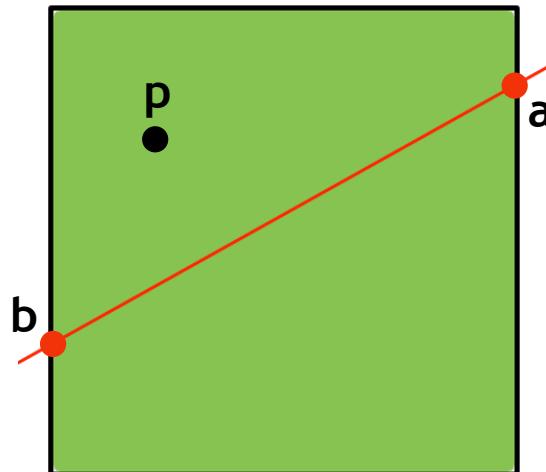


# Random, hardly controllable terrain generation algorithms

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

a ... entry point  
b ... exit point  
p ... heightmap vertex

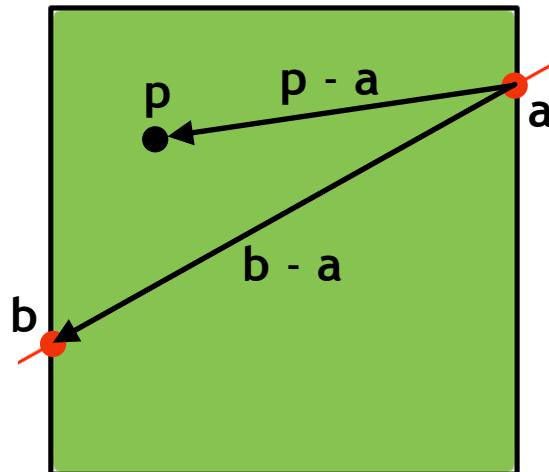


# Random, hardly controllable terrain generation algorithms

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

a ... entry point  
b ... exit point  
p ... heightmap vertex



# Random, hardly controllable terrain generation algorithms

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

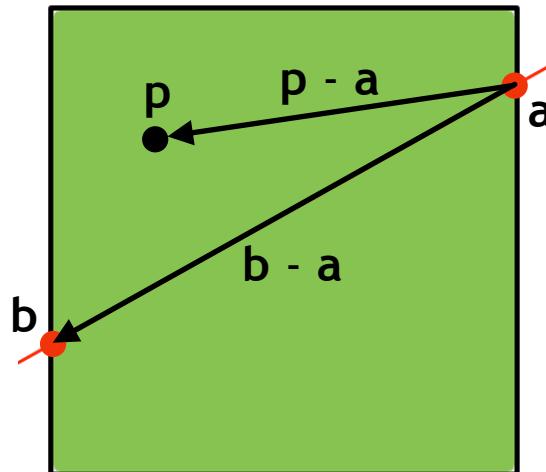
a ... entry point

b ... exit point

p ... heightmap vertex

$$(p - a) \times (b - a) =$$

vector perpendicular  
to  $(p - a)$  and  $(b - a)$



# Random, hardly controllable terrain generation algorithms

- Diamond-Square algorithm
- Fault algorithm

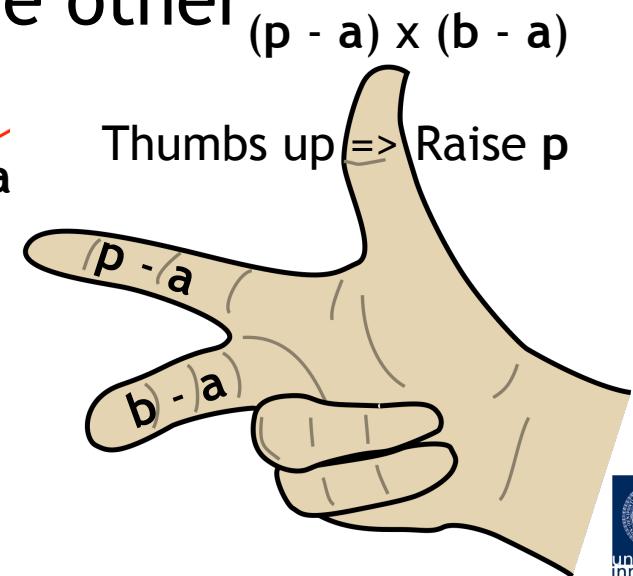
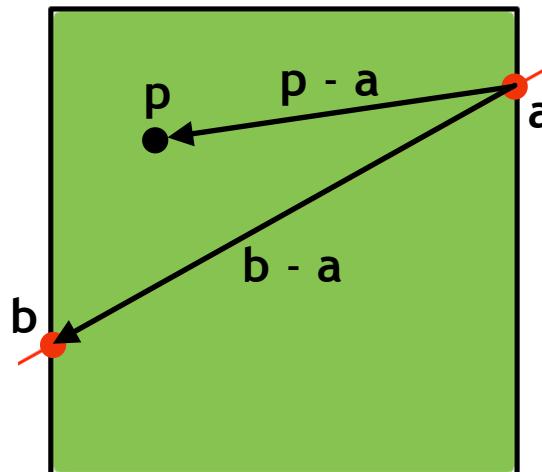
Continuously split the height map in two regions, raise one and lower the other

a ... entry point

b ... exit point

p ... heightmap vertex

$(p - a) \times (b - a) =$   
vector perpendicular  
to  $(p - a)$  and  $(b - a)$

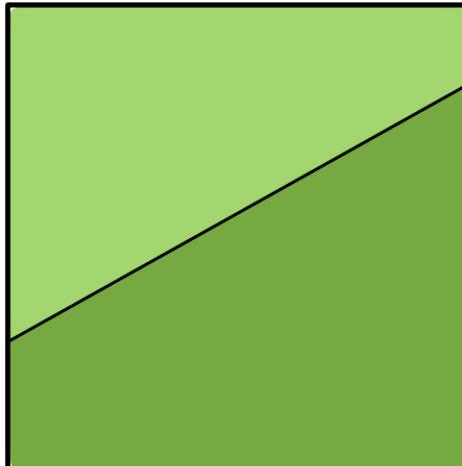


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

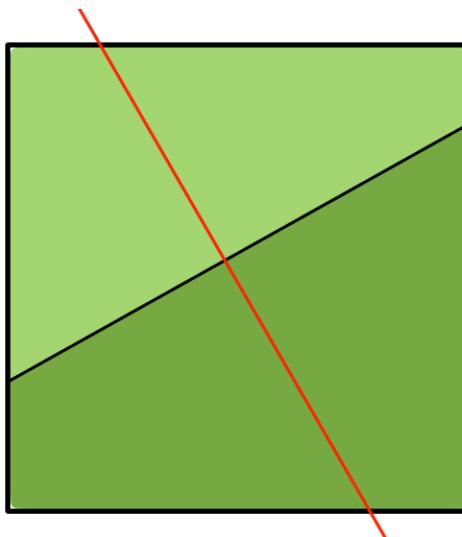


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

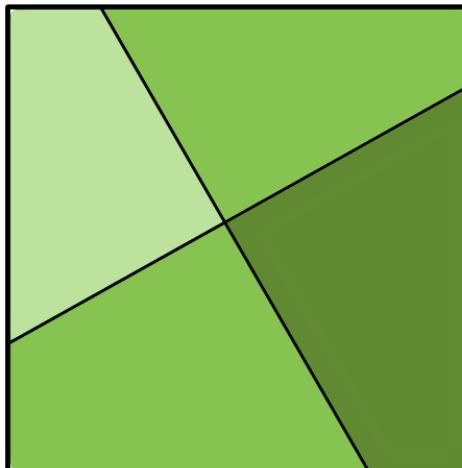


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

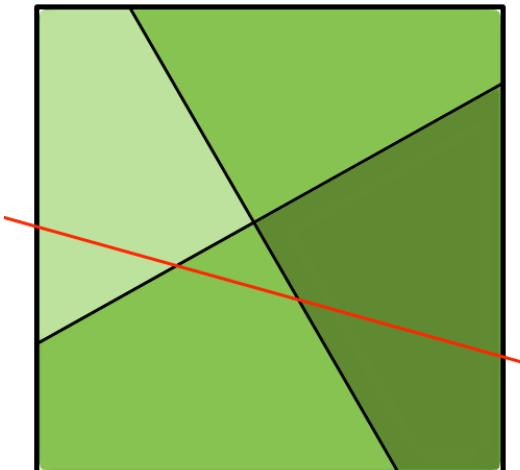


# Random, hardly controllable terrain generation algorithms

---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other

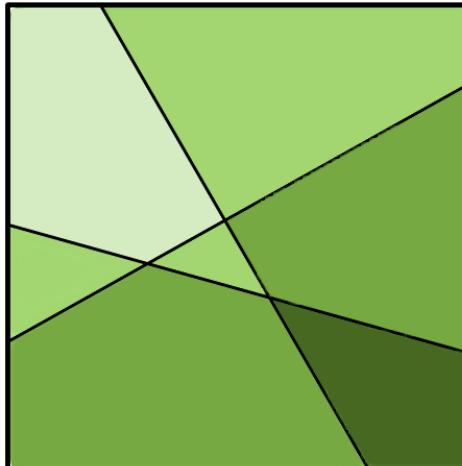


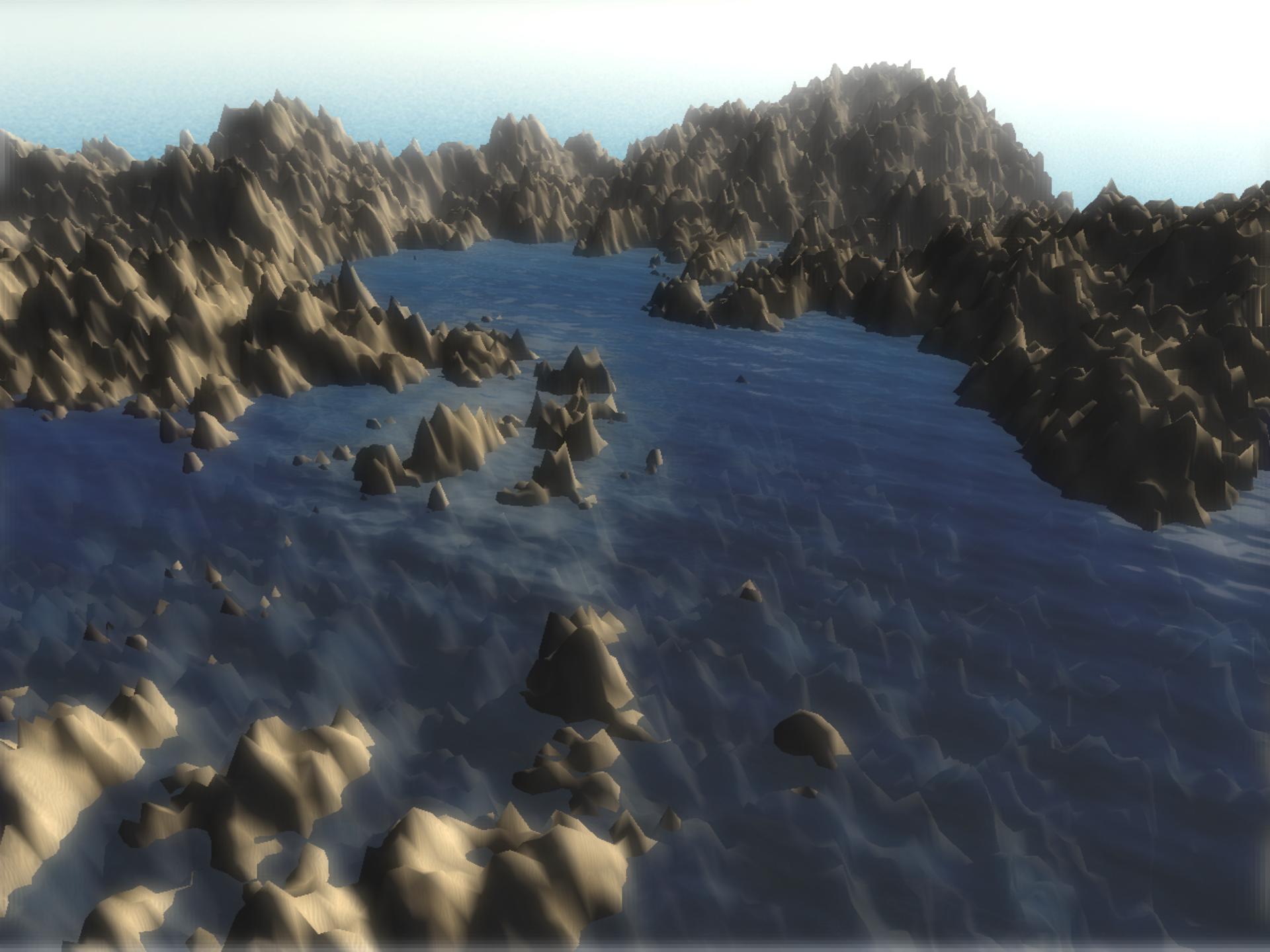
# Random, hardly controllable terrain generation algorithms

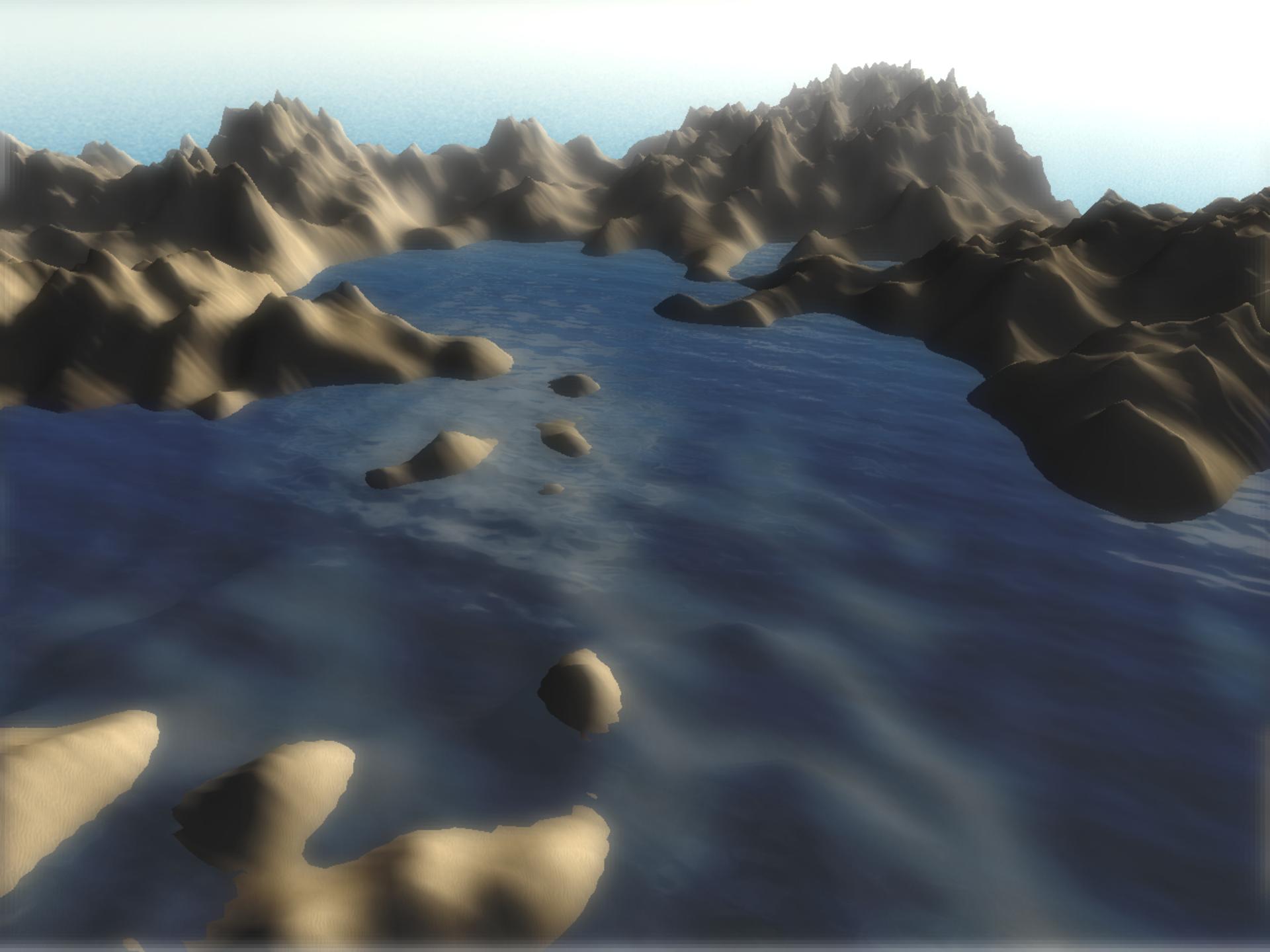
---

- Diamond-Square algorithm
- Fault algorithm

Continuously split the height map in two regions, raise one and lower the other







# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP

# Parametrically controllable terrain generation algorithms

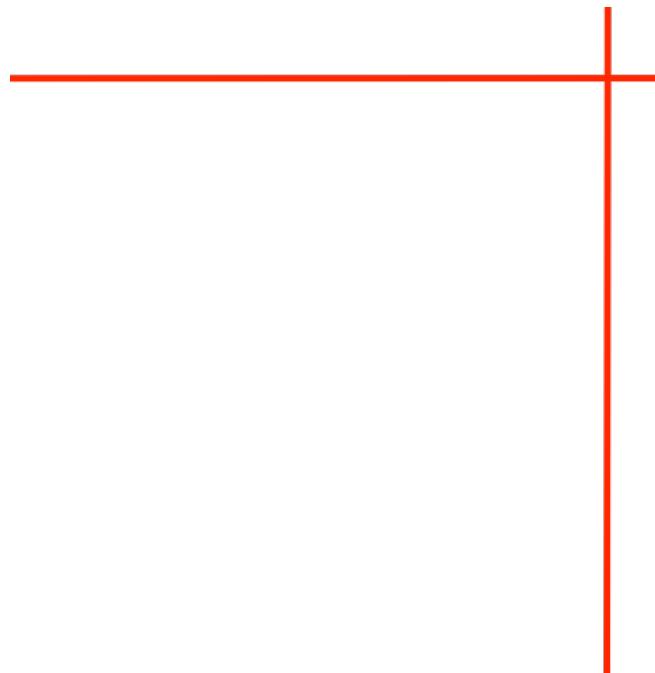
---

- Repeated Magnification and Probing algorithm
    - ➡ Graph-based RMP
    - Voronoi-based RMP
-

# Parametrically controllable terrain generation algorithms

---

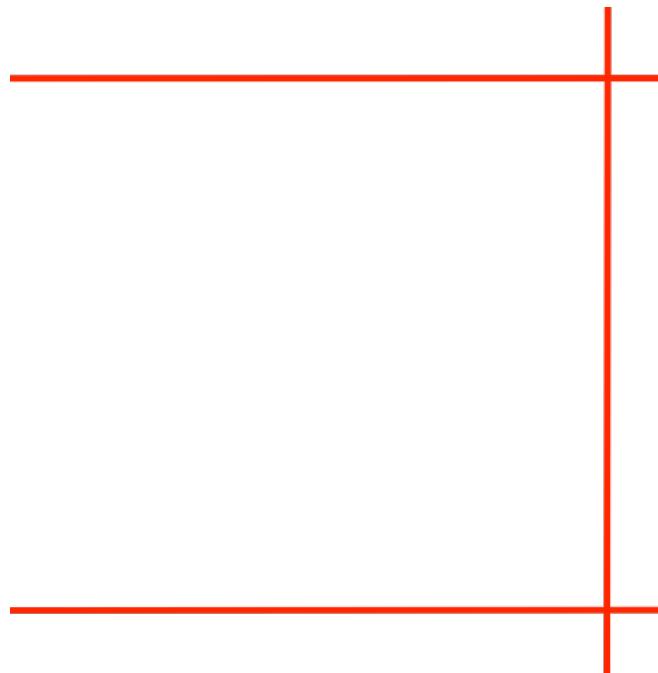
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

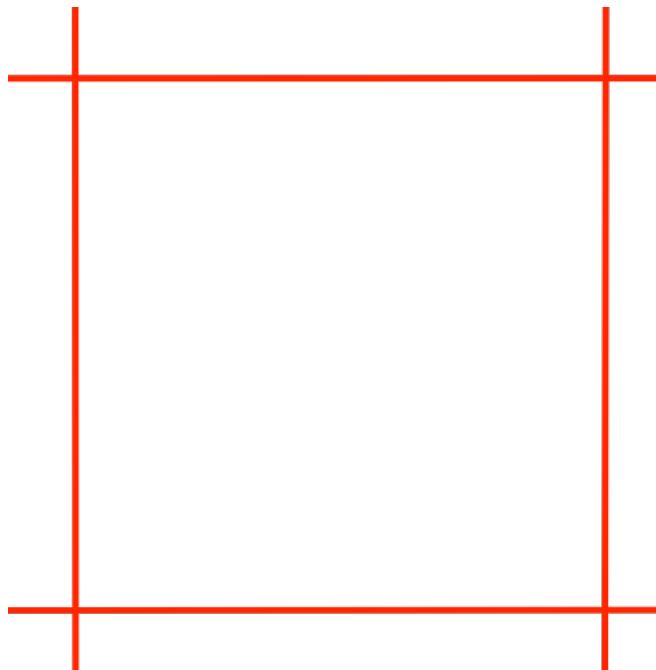
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

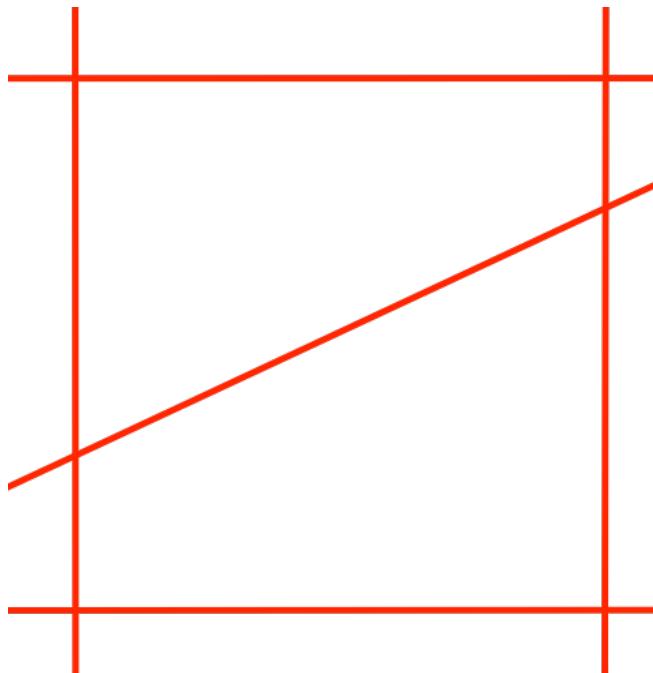
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

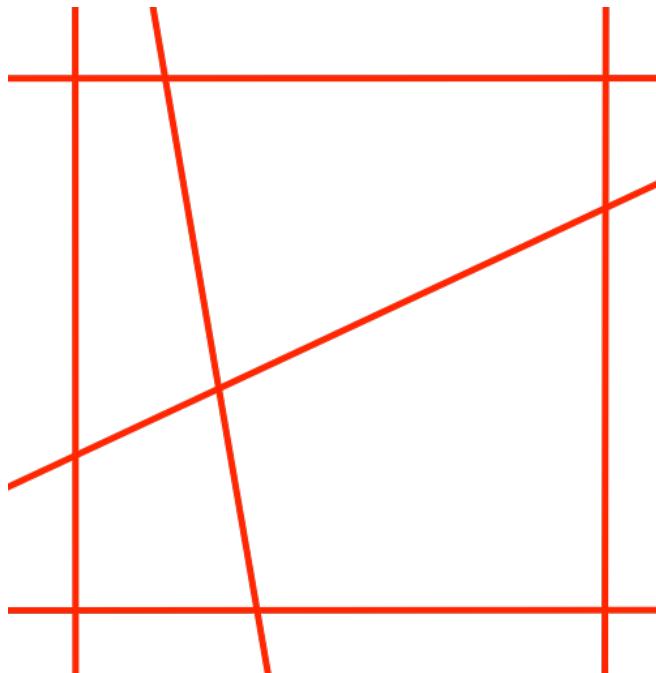
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

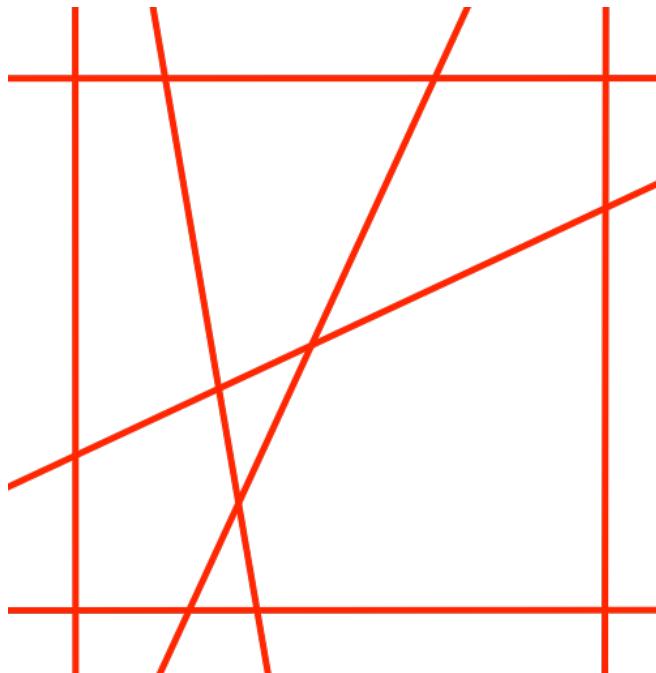
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

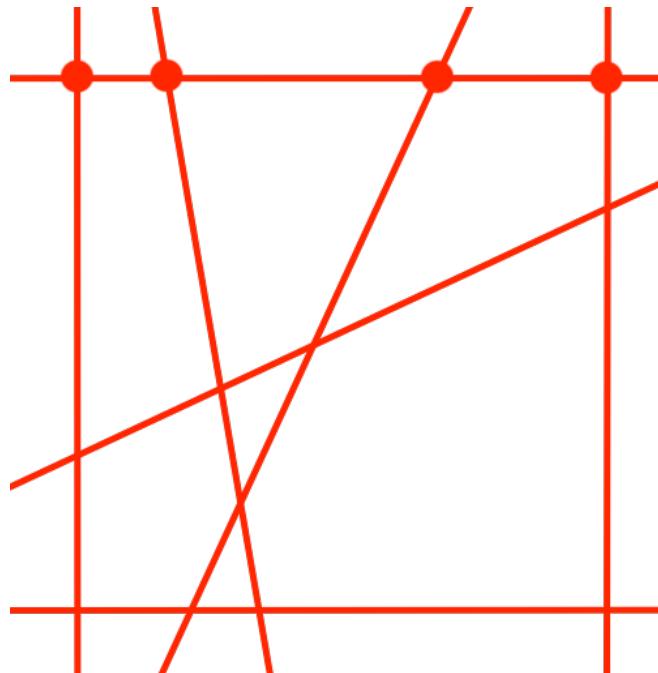
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

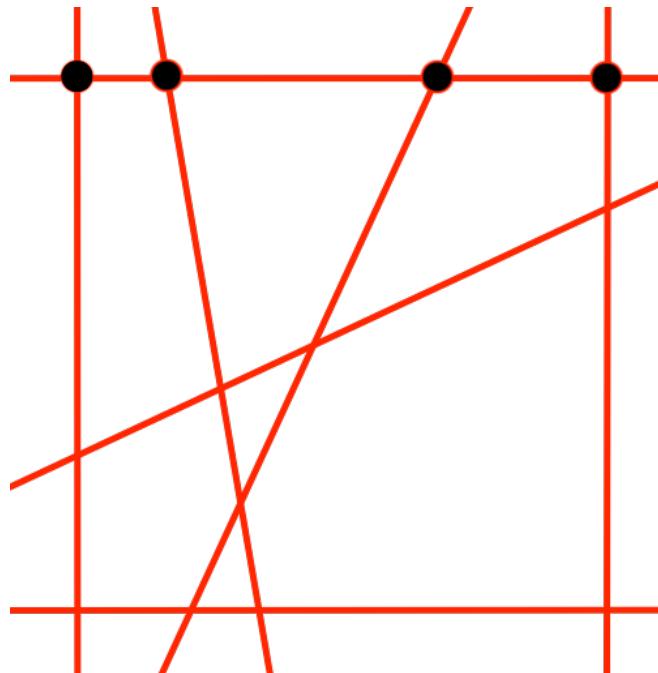
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

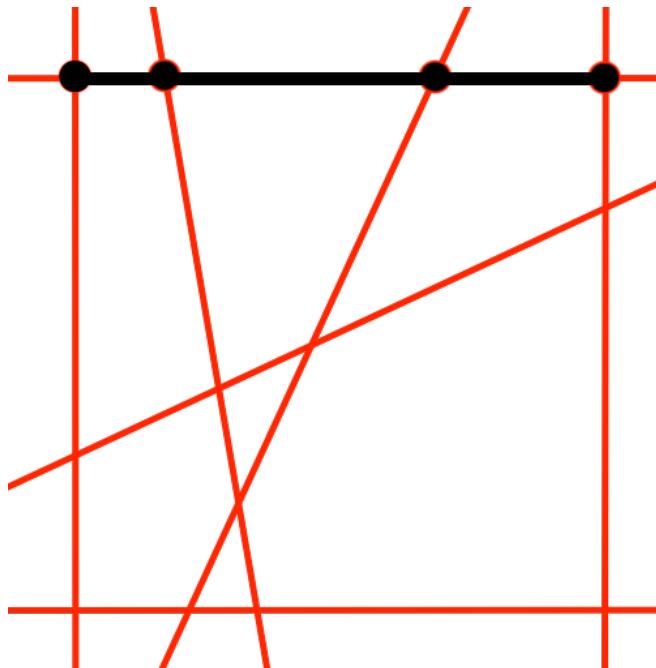
---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



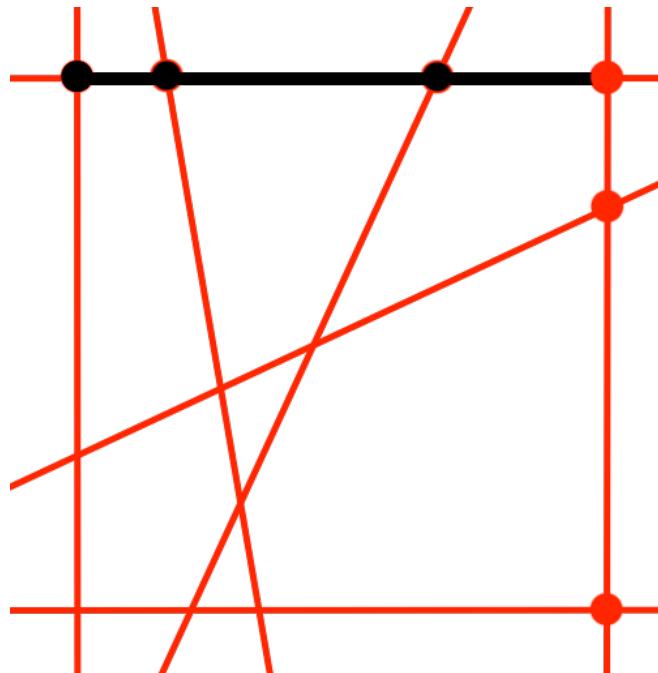
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



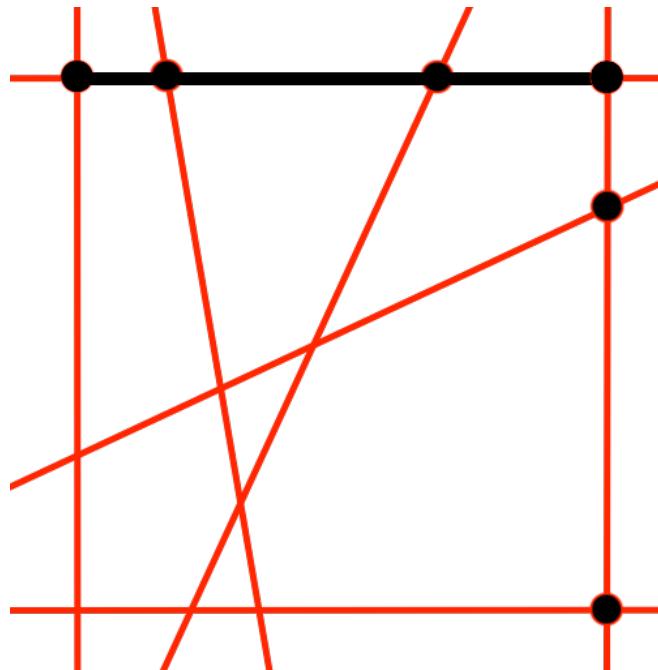
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



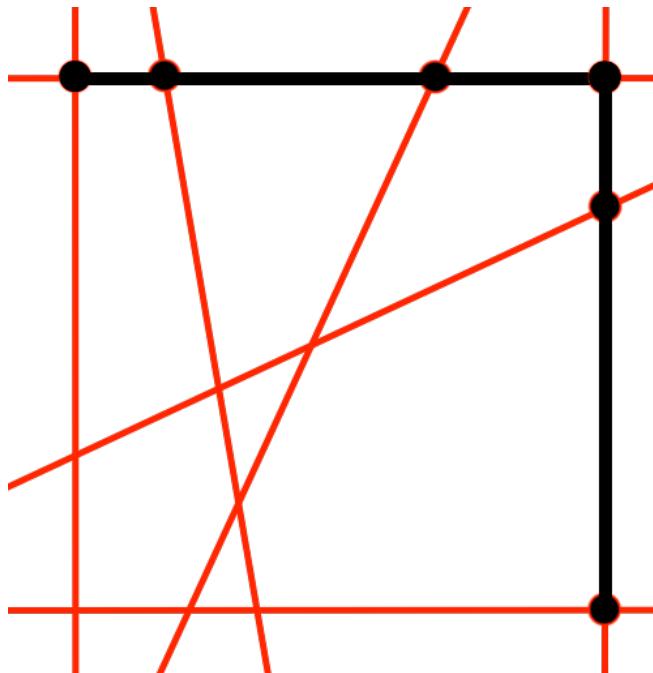
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



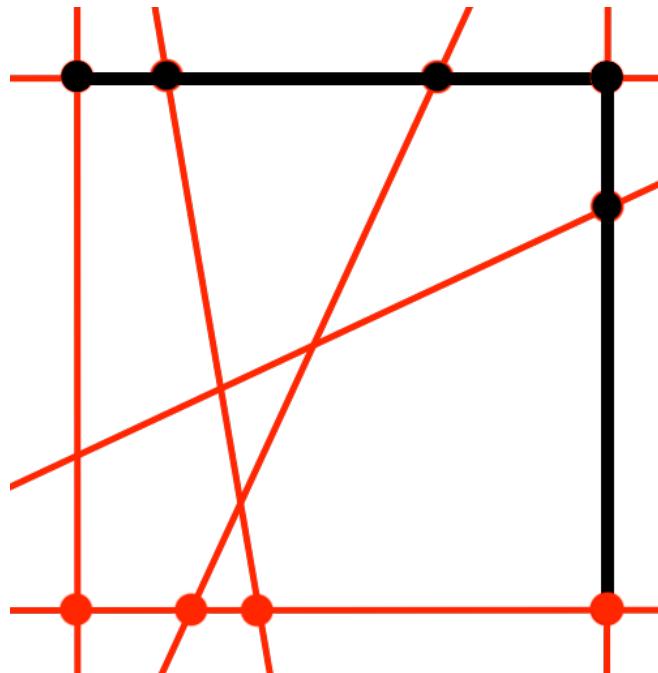
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



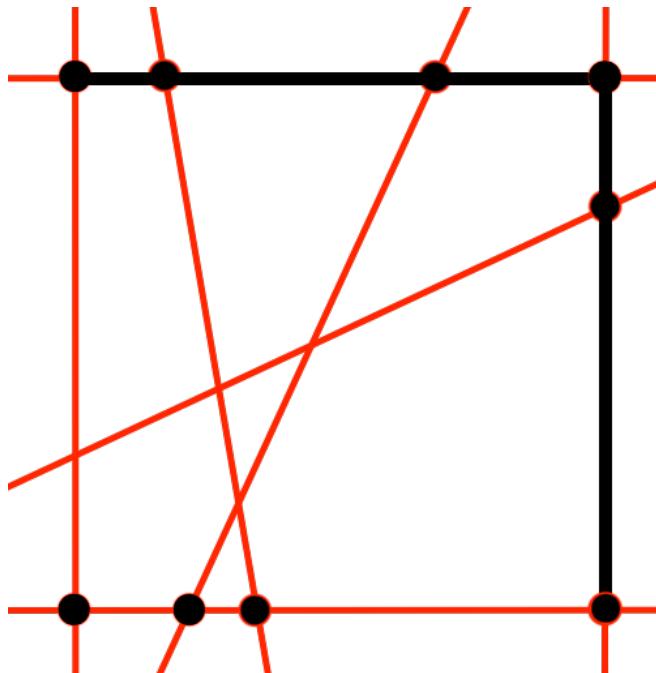
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



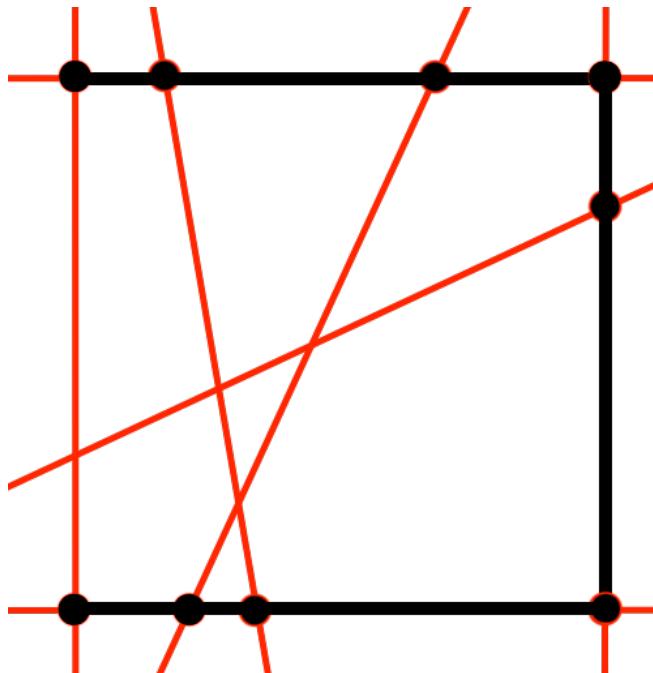
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



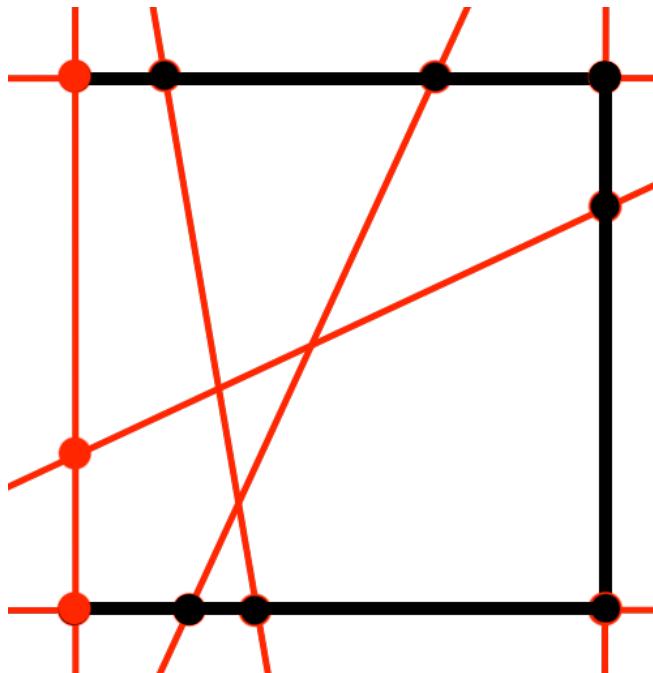
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



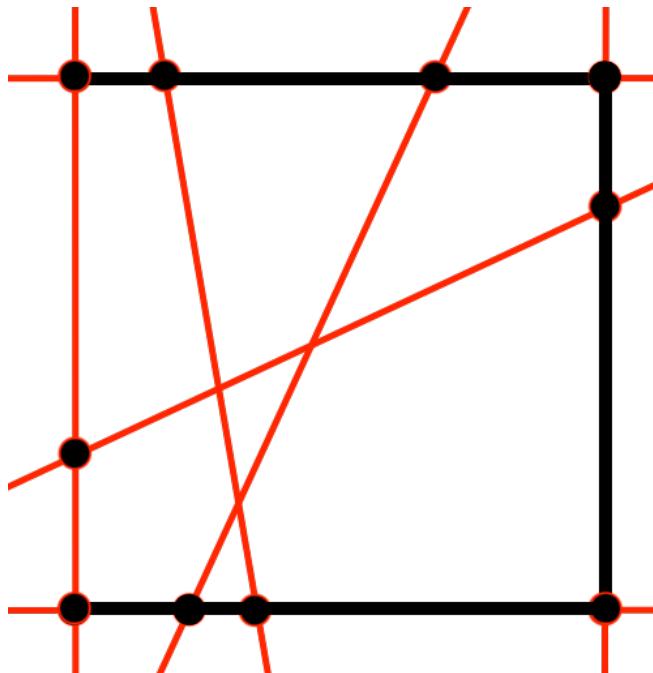
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



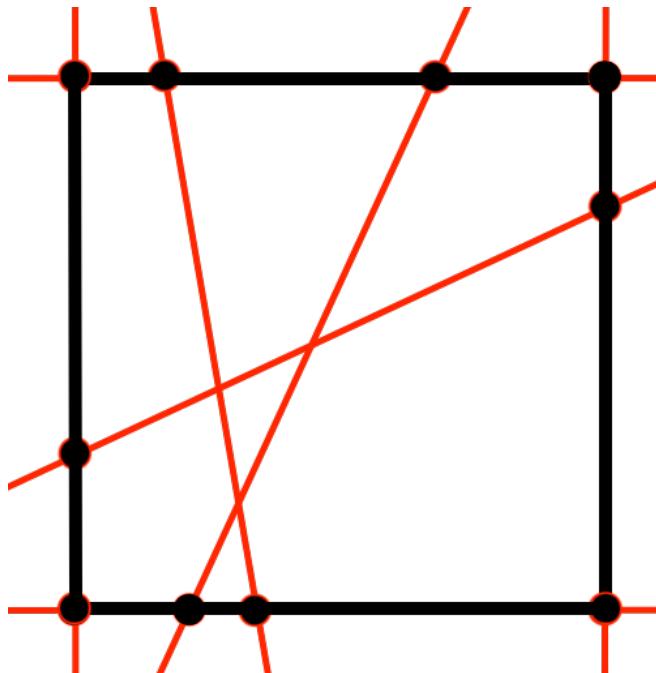
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



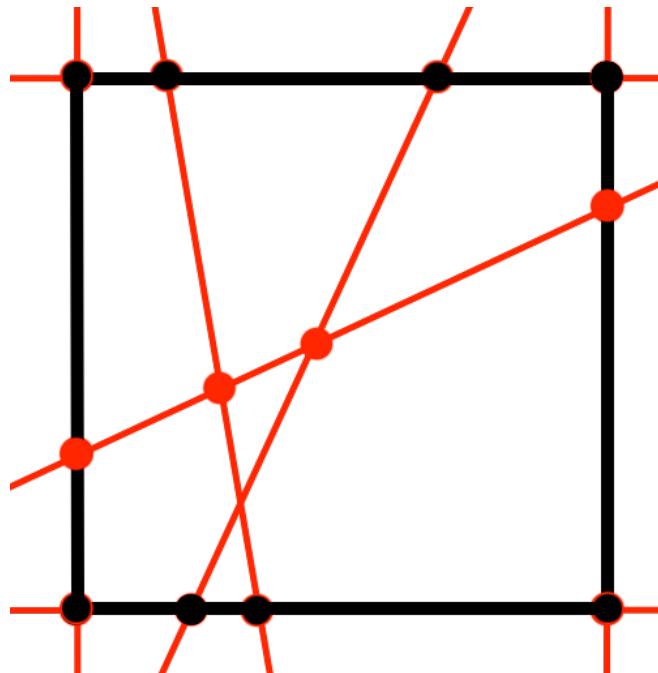
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



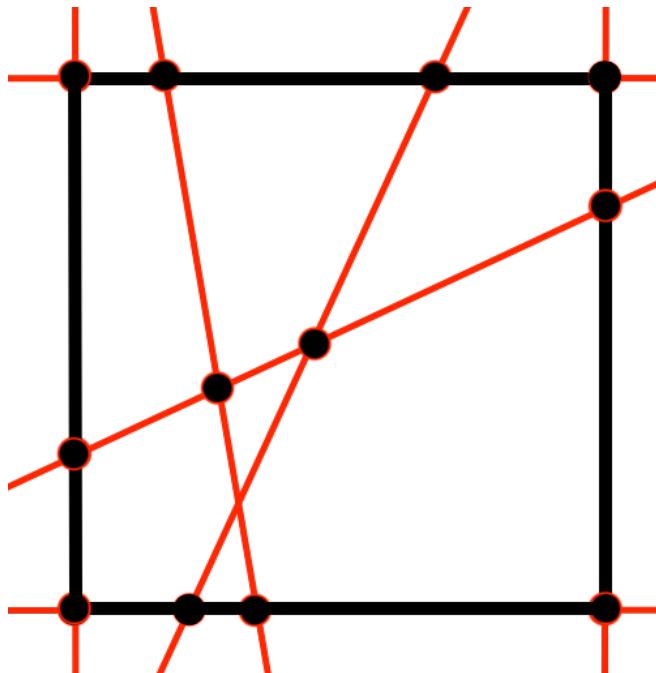
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



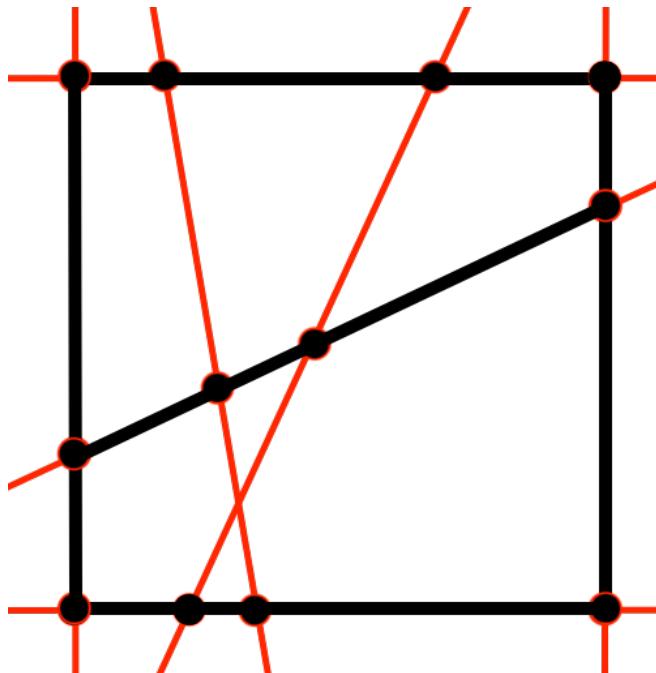
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



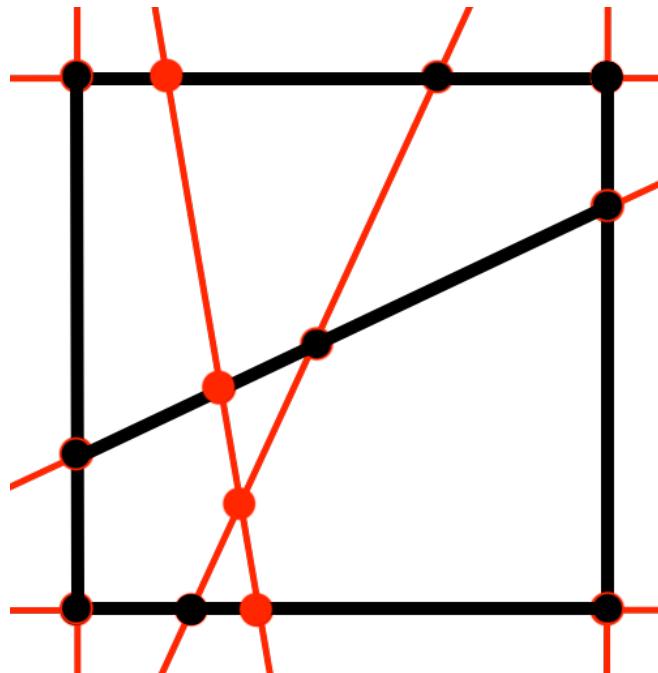
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



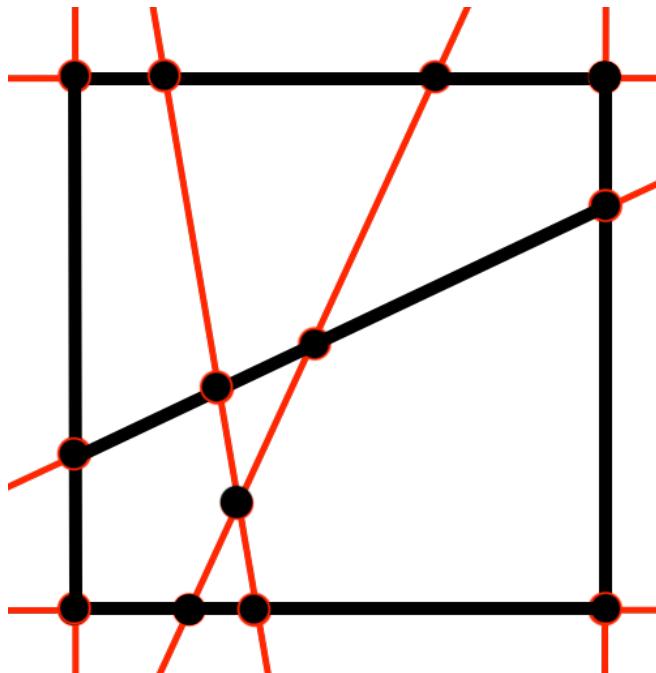
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



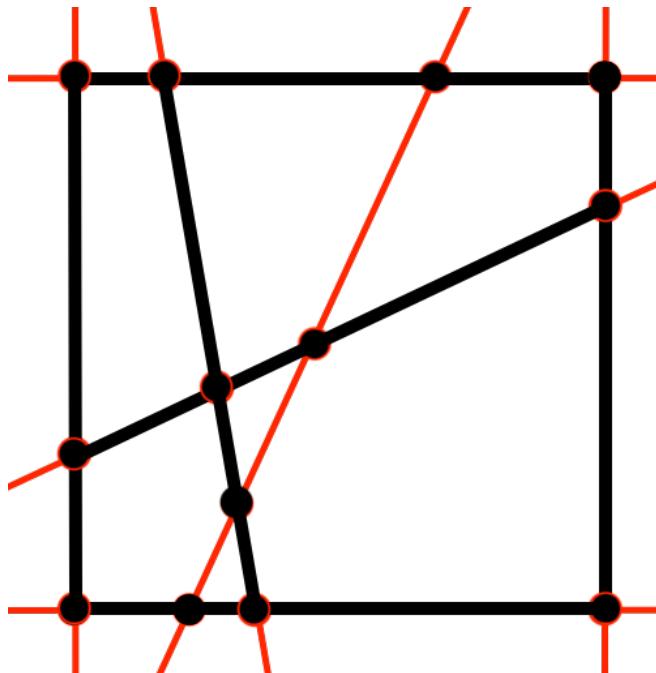
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



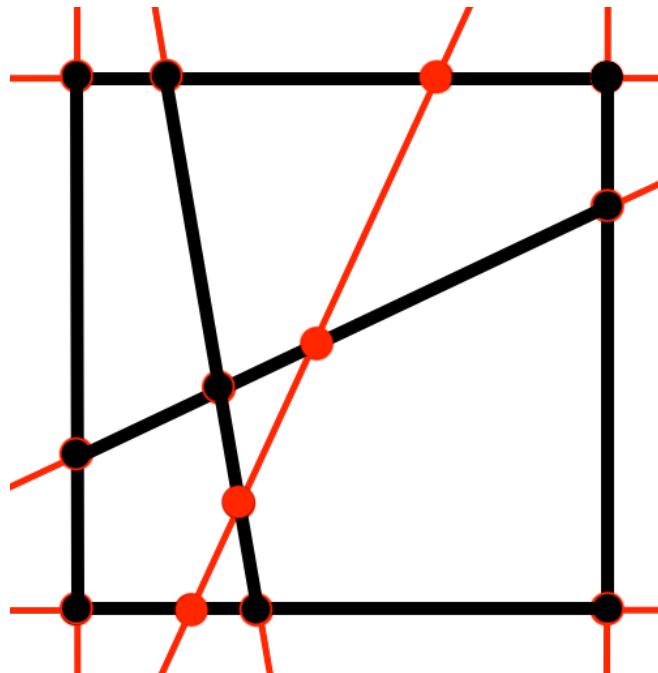
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



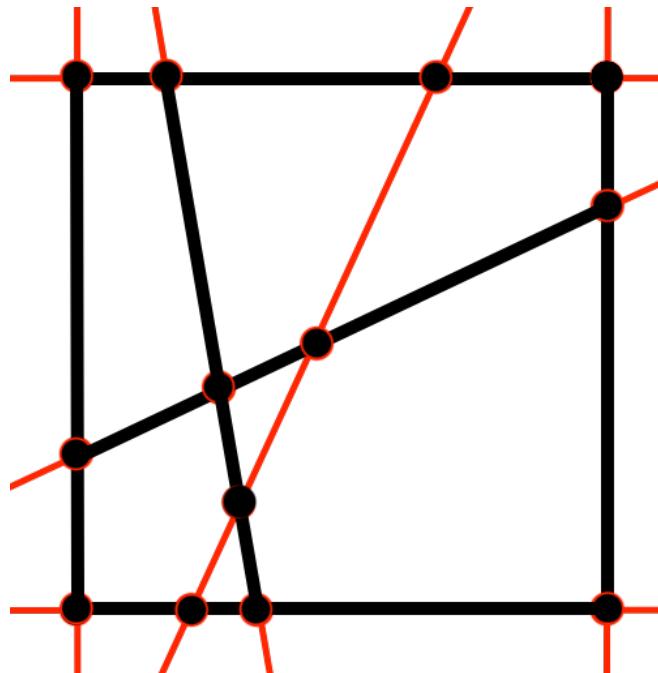
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



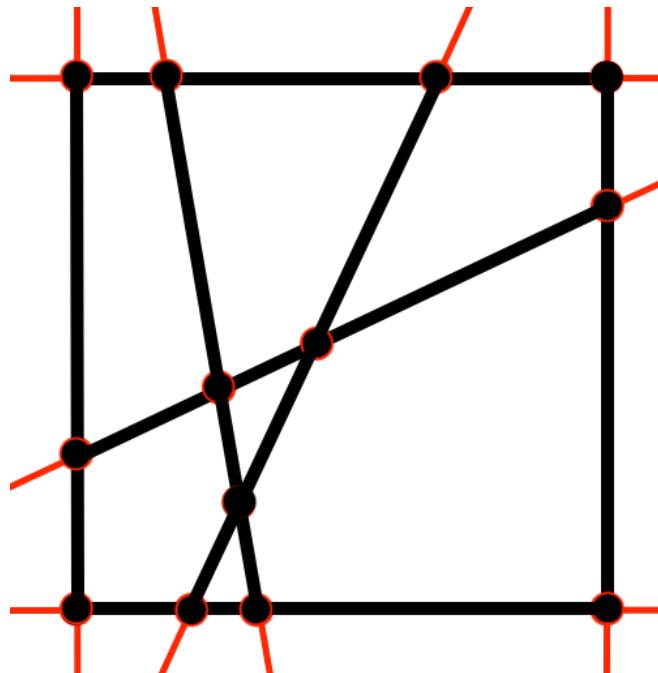
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

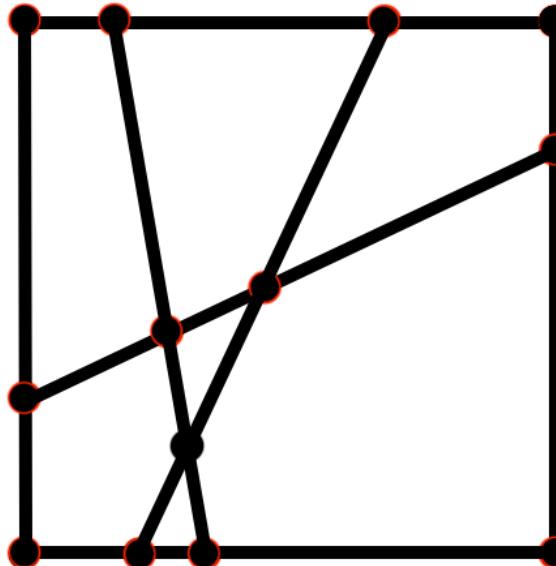
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

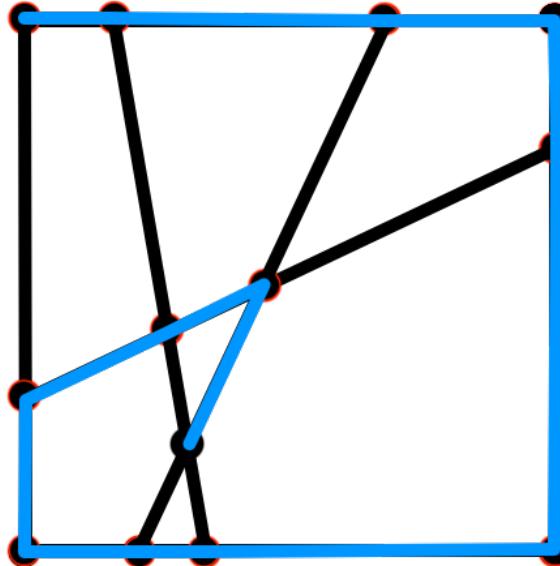
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

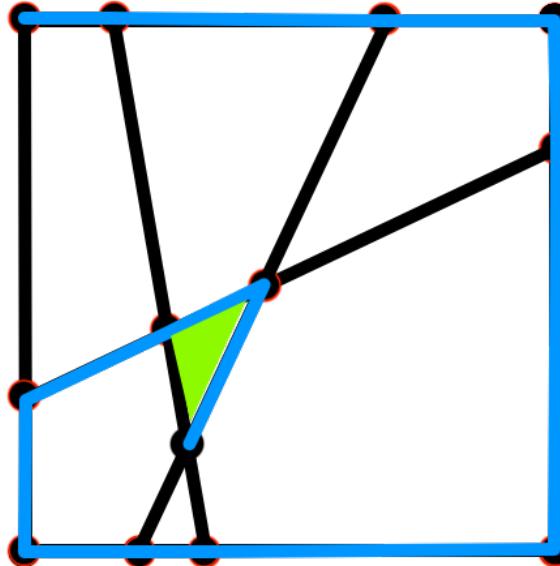
---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

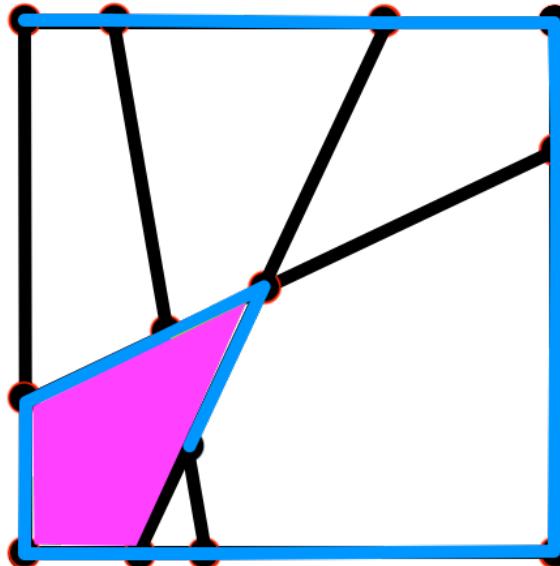
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

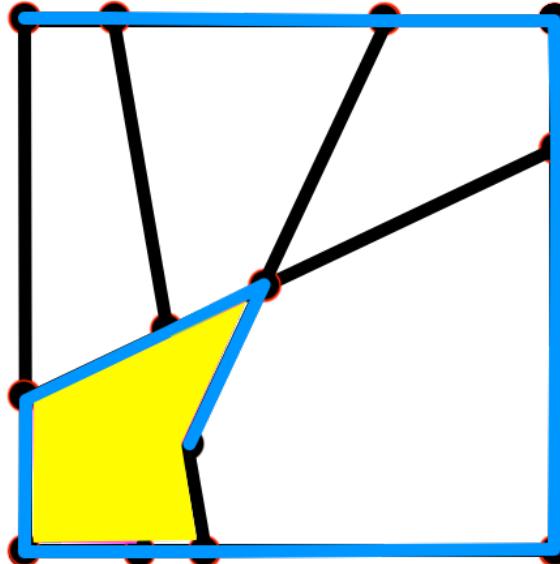
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

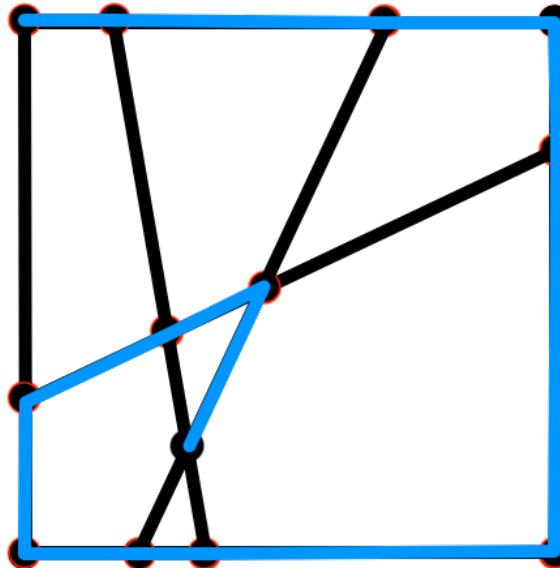
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

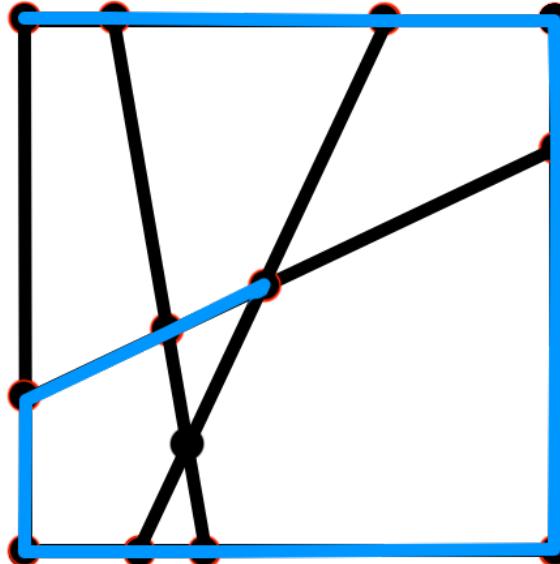
---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

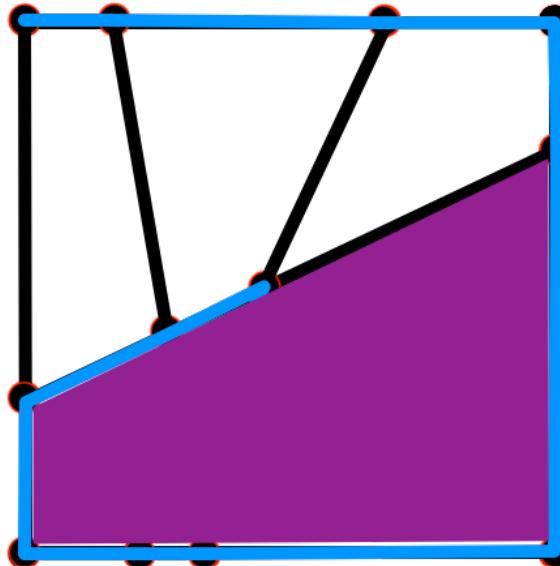
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

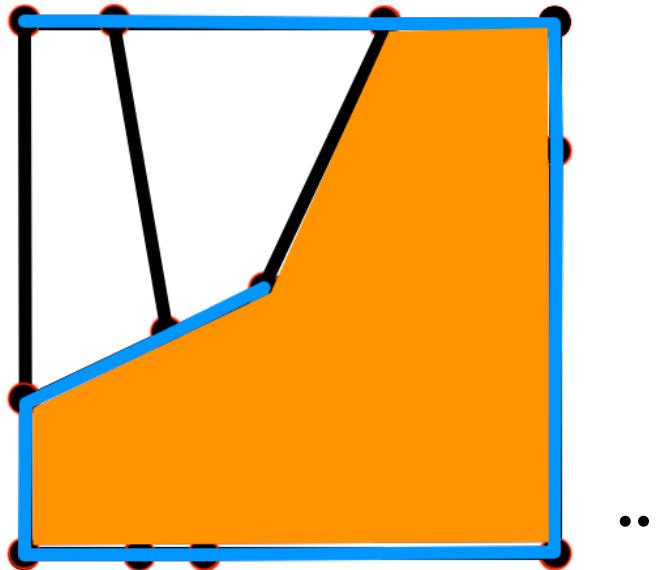
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

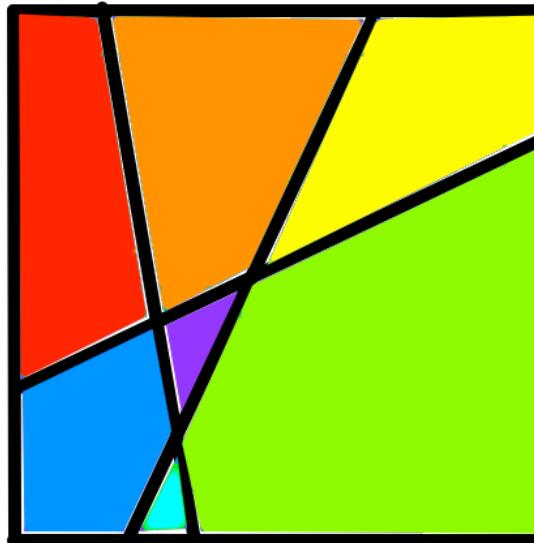
Find minimal cycles by eliminating duplicates and cycles that contain other cycles.

A cycle contains another cycle if they share a path of 3 vertices.

# Parametrically controllable terrain generation algorithms

---

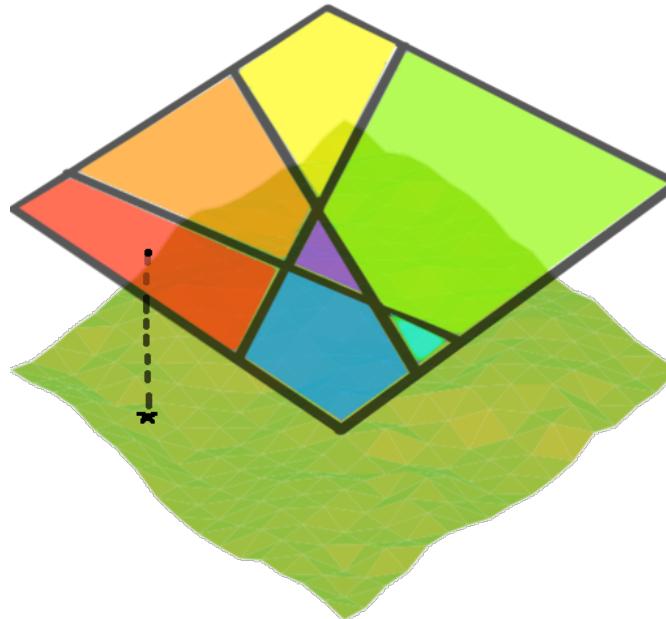
- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

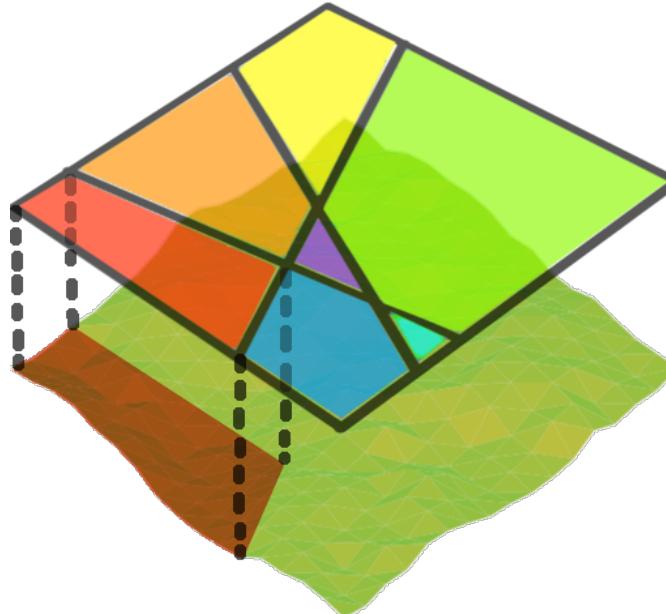
---

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

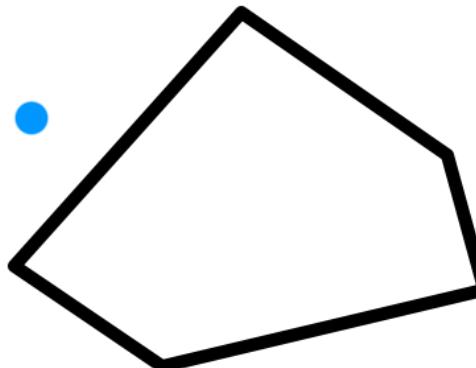


# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

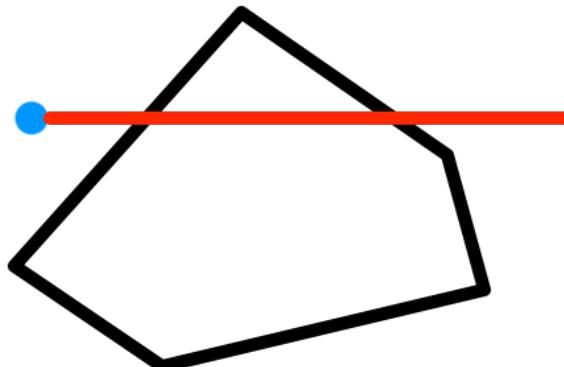


# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

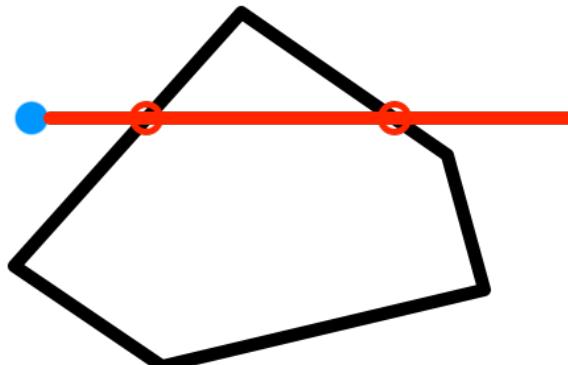


# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

even number of intersections  
=> outside

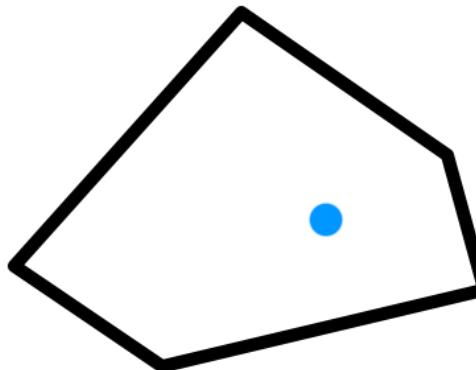


# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

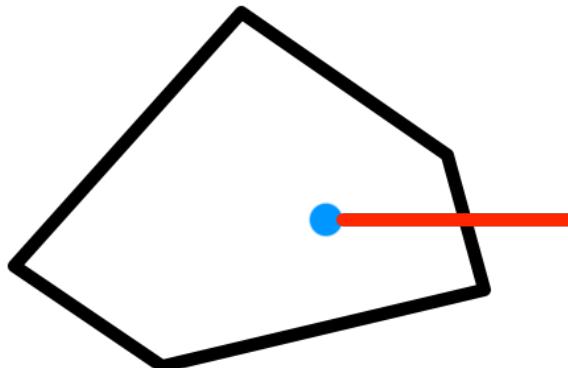


# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

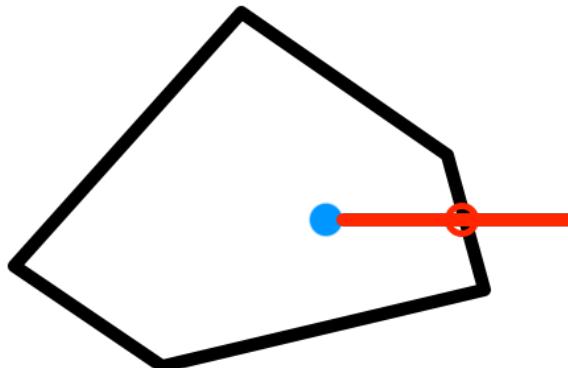


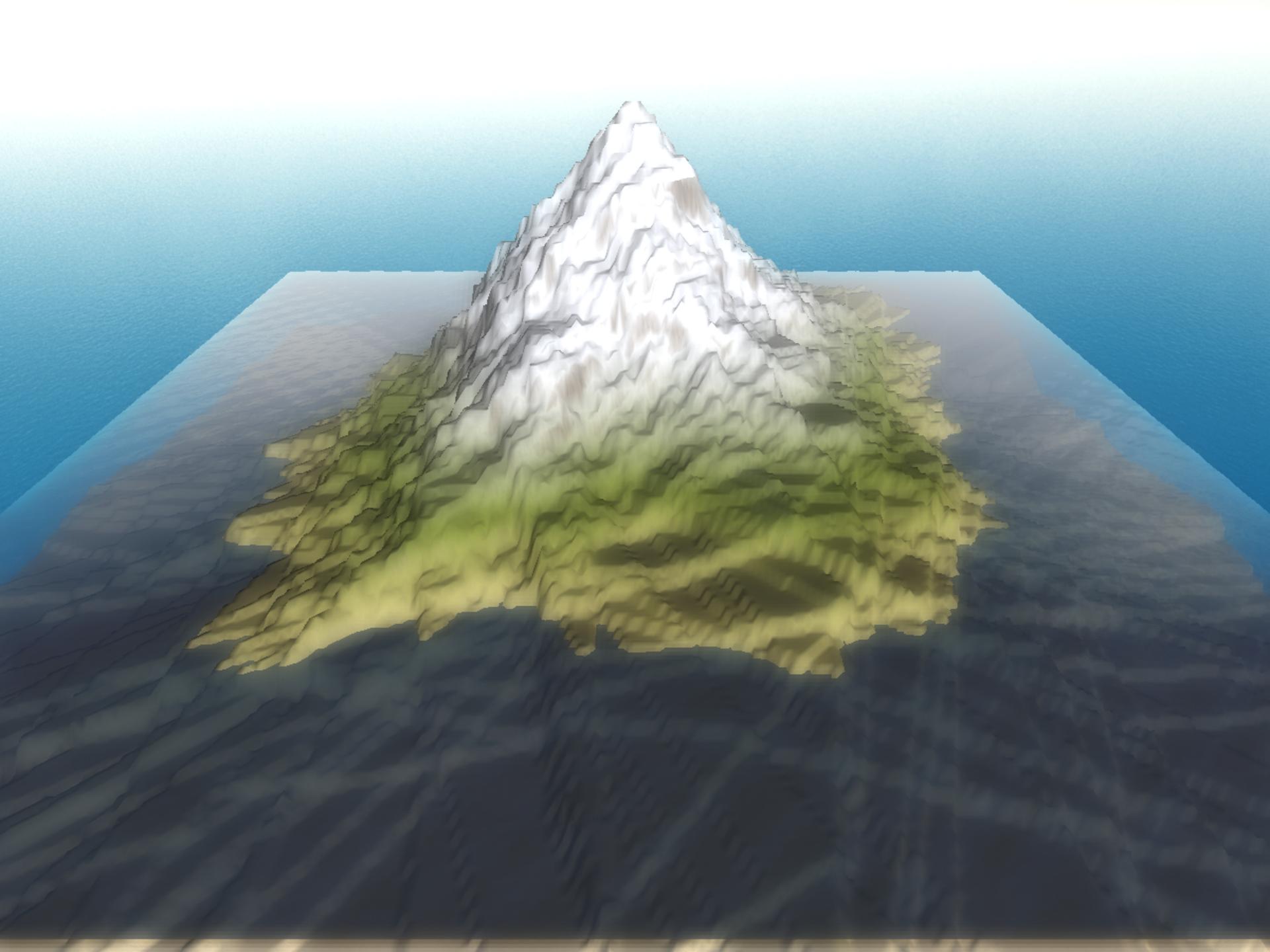
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP

An algorithm called *pnpoly* was used to determine if a point is in a polygon or not.

odd number of intersections  
=> inside





# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - ➡ Graph-based RMP
  - Voronoi-based RMP

Unfortunately, Graph-based RMP is not very efficient.

A different approach was needed!

# Parametrically controllable terrain generation algorithms

---

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - Voronoi-based RMP

# Parametrically controllable terrain generation algorithms

---

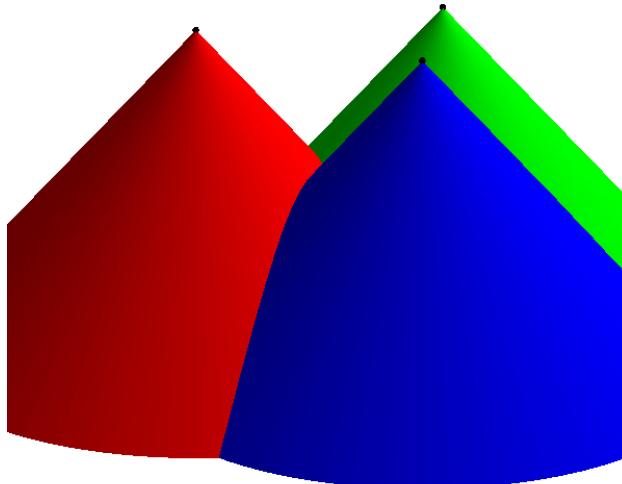
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP

The idea is to utilize the GPU to generate random polygons similar to graph-based RMP.

# Parametrically controllable terrain generation algorithms

---

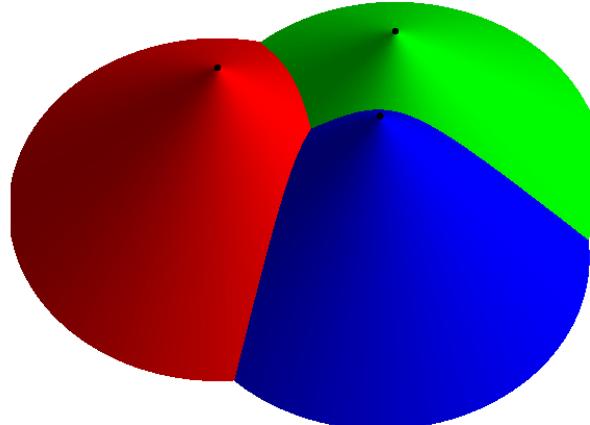
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

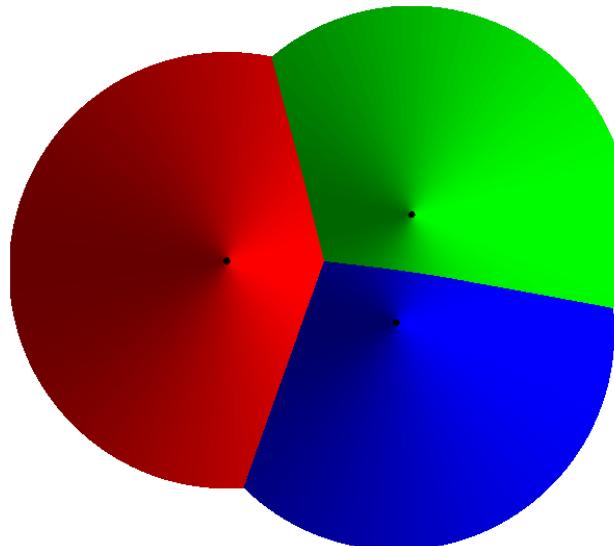
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

---

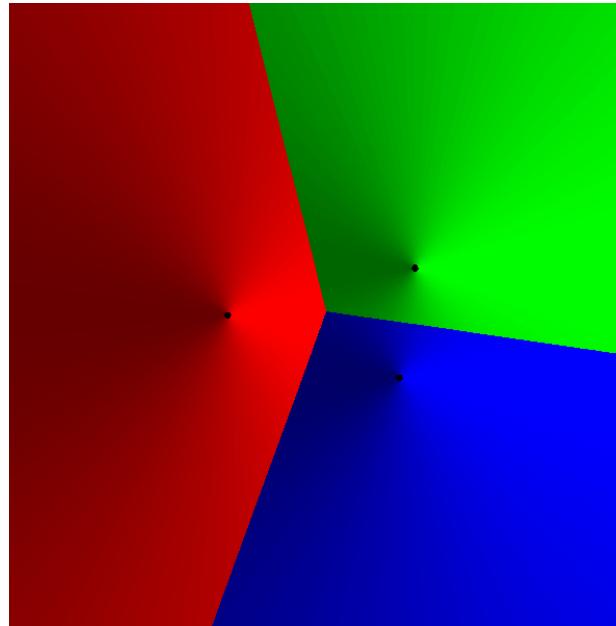
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP



# Parametrically controllable terrain generation algorithms

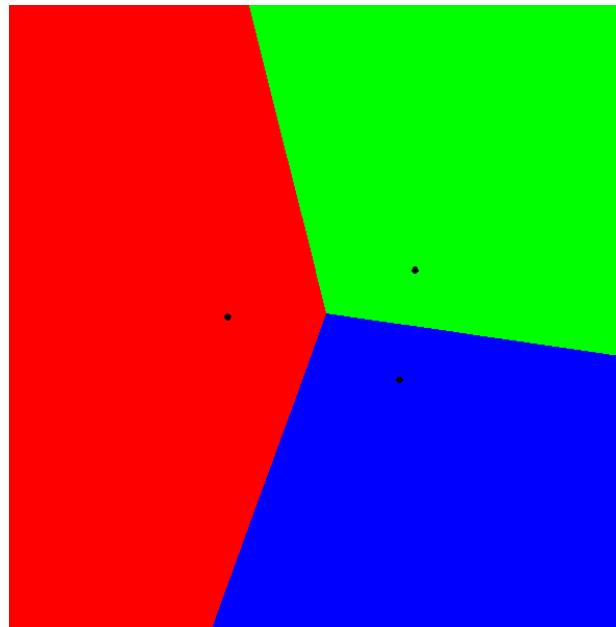
---

- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP

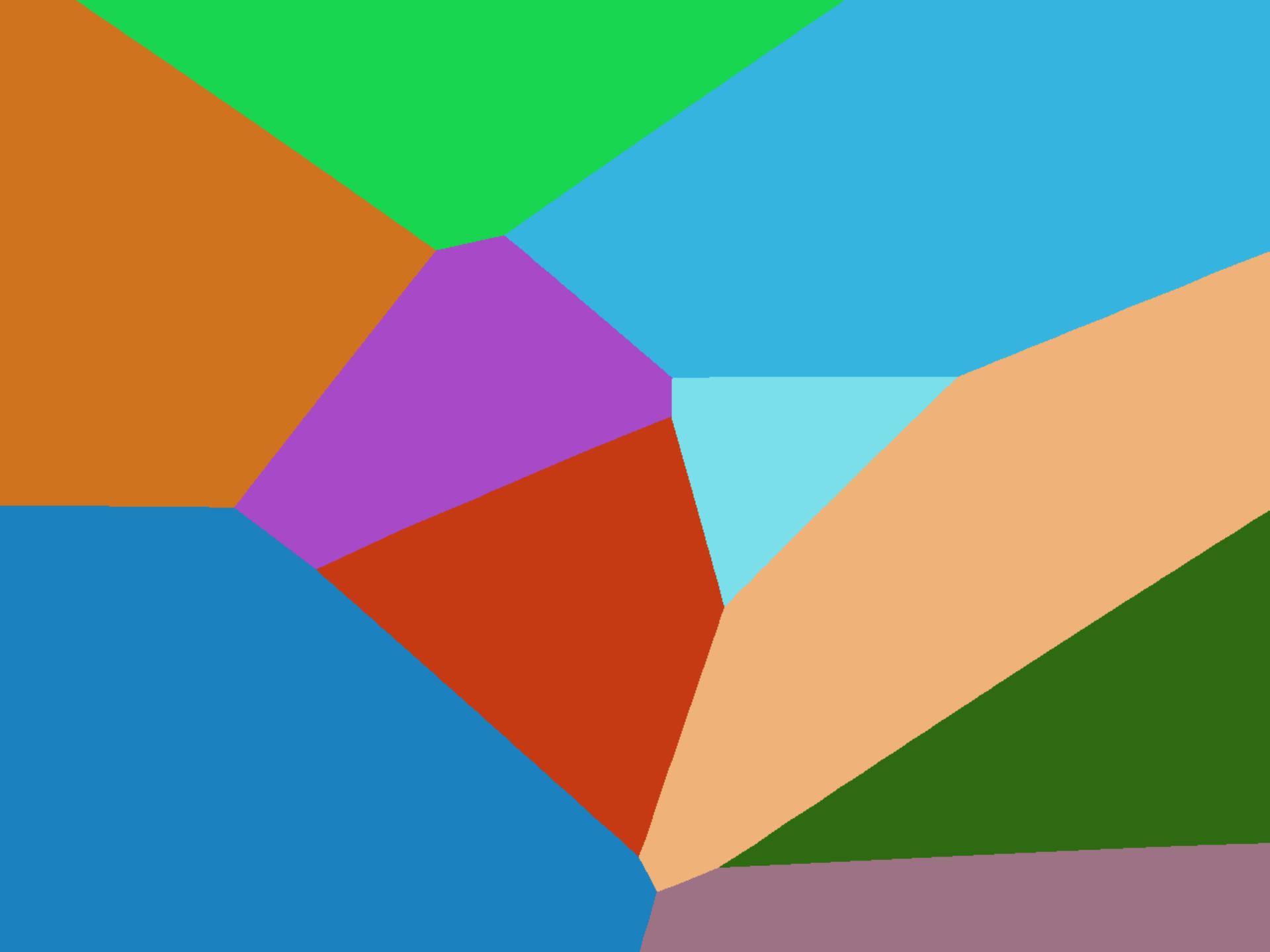


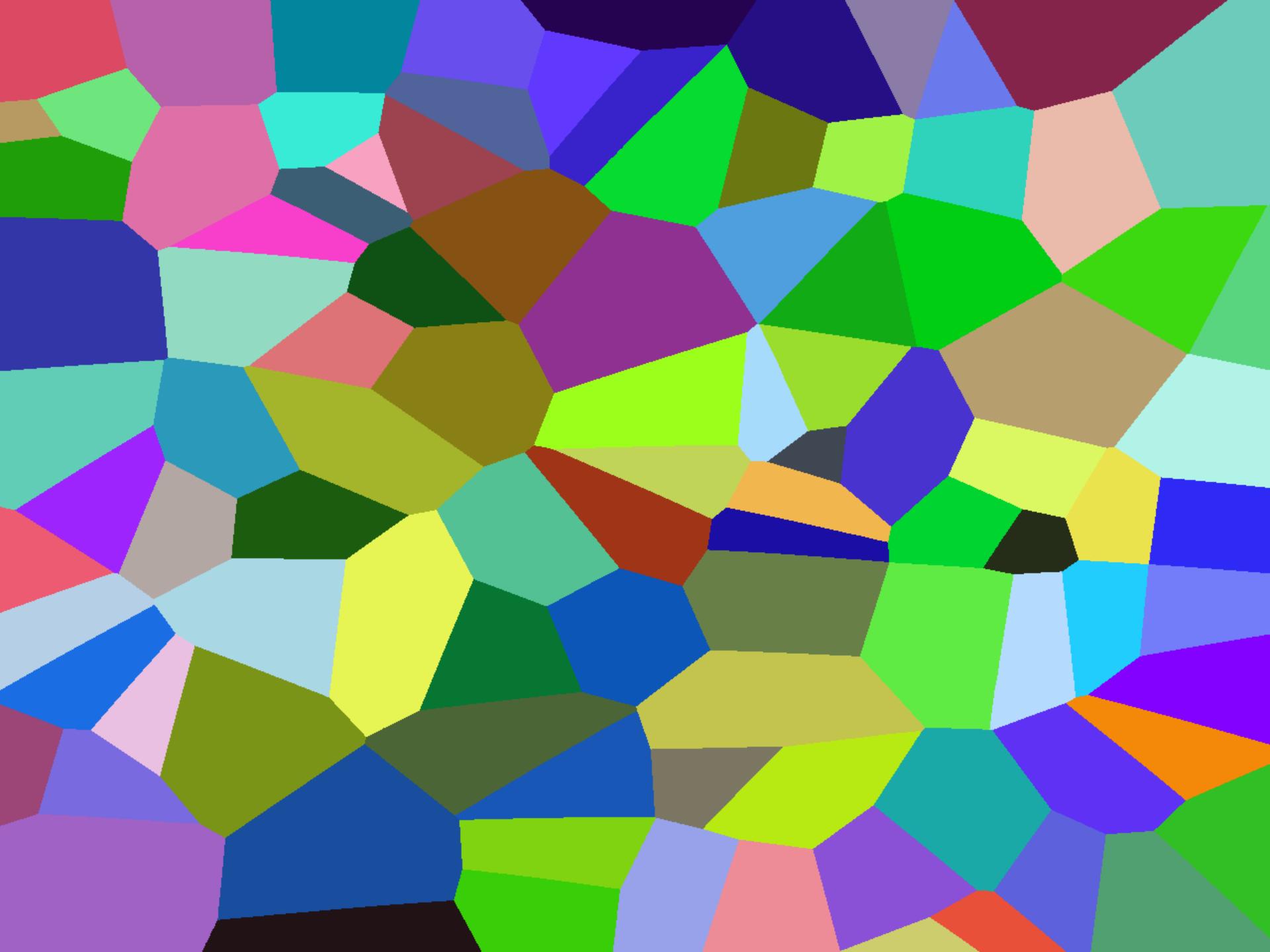
# Parametrically controllable terrain generation algorithms

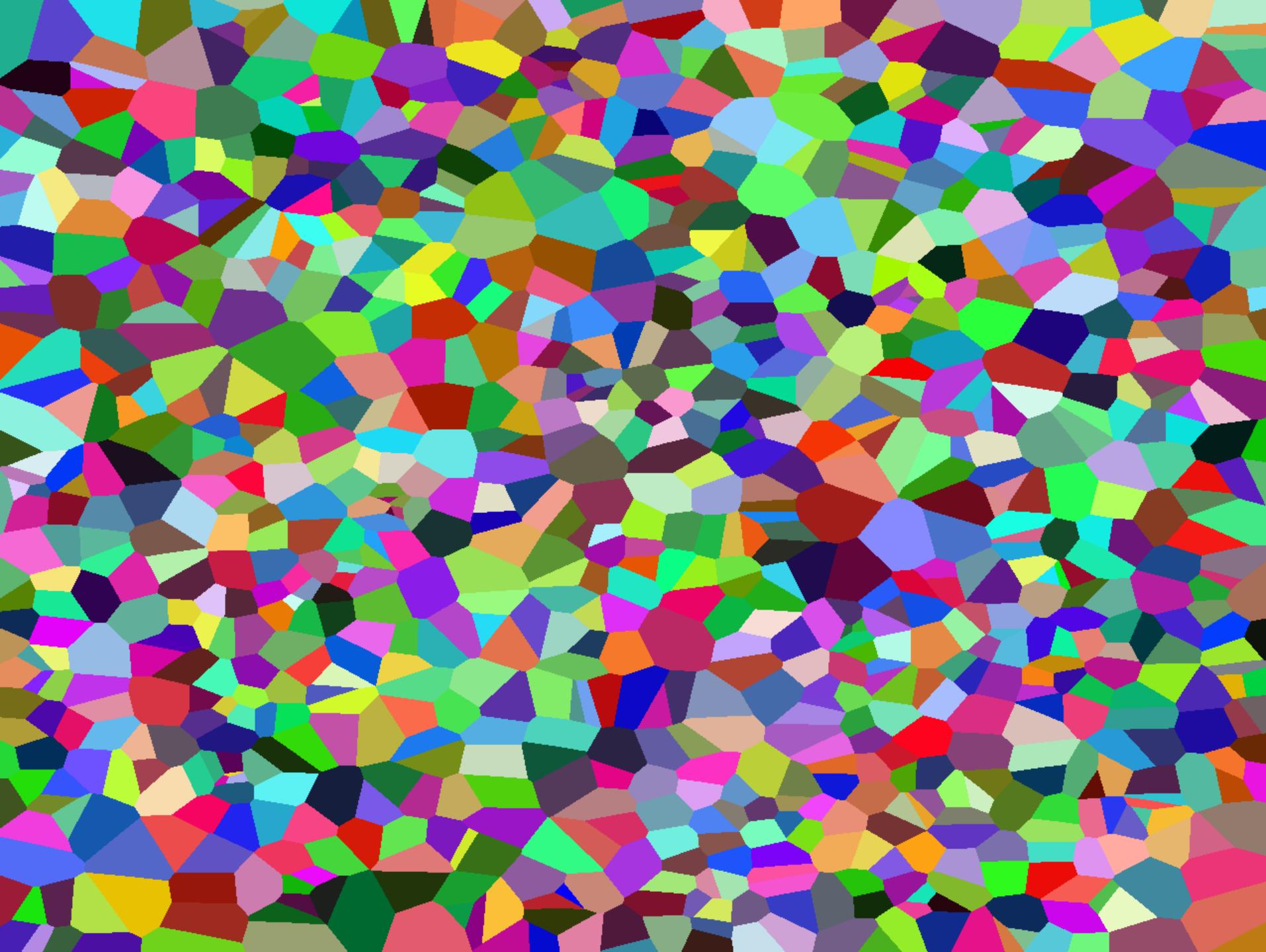
- Repeated Magnification and Probing algorithm
  - Graph-based RMP
  - ➡ Voronoi-based RMP

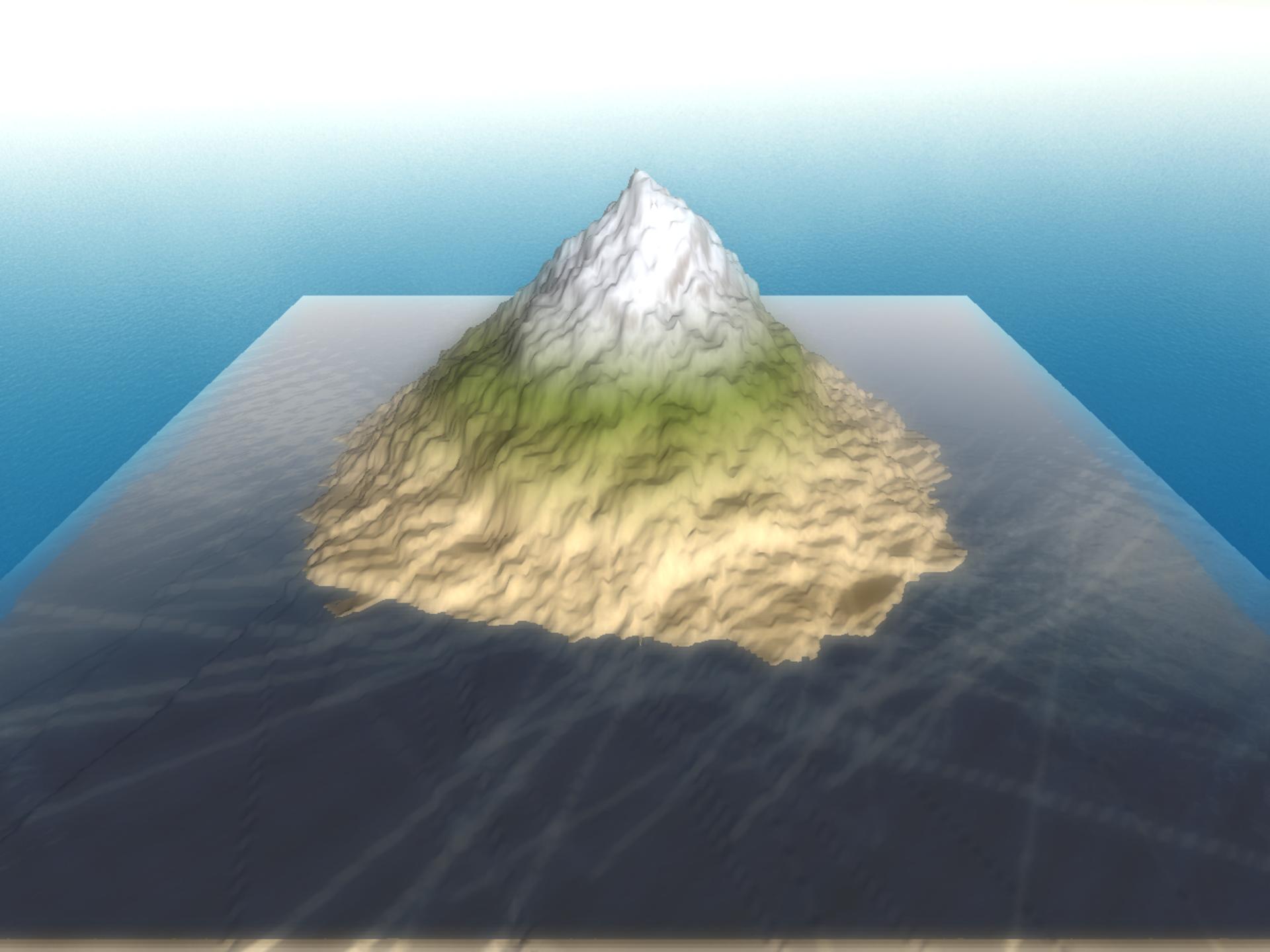


Cone animation frames were extracted from Chris Wellon's voronoi-cones.gif









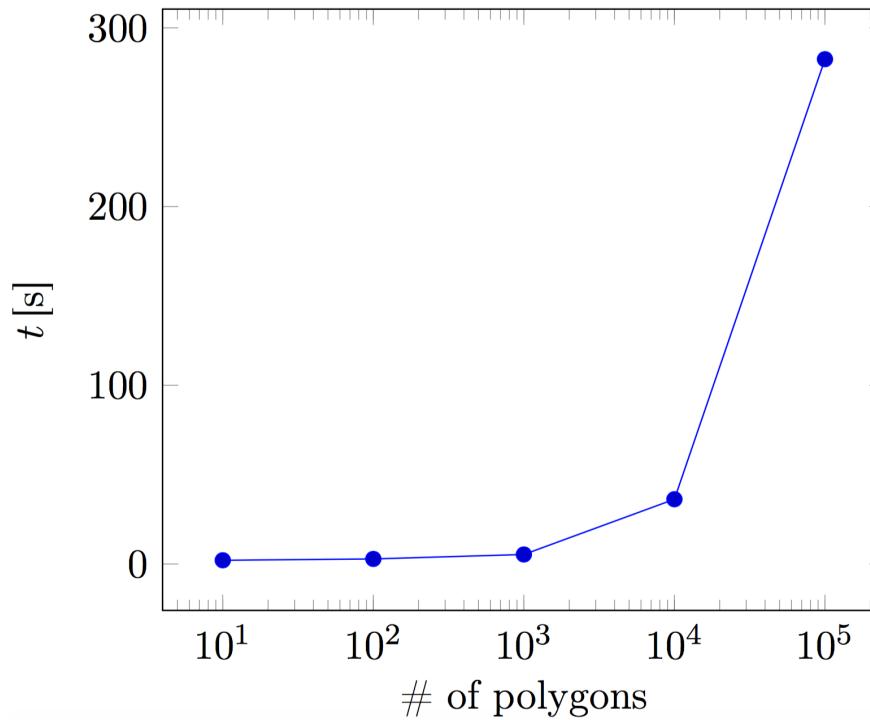
# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - Performance comparison

	Task 1	Task 2	Task 3
Number of polygons per iteration	7	11	16
Number of iterations	100	100	100
Graph-based RMP duration	2.975s	46.729s	5937.348s
Voronoi-based RMP duration	1.989s	2.004s	2.025s

# Parametrically controllable terrain generation algorithms

- Repeated Magnification and Probing algorithm
  - Voronoi performance stress test



# Advanced, geology-based terrain generation and degradation algorithms

---

- Geology-based terrain generation
  - Tectonic plate simulation
- Geology-based terrain degradation
  - Cellular automata-based thermal and hydraulic erosion algorithm
  - Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Advanced, geology-based terrain generation and degradation algorithms

---

- Geology-based terrain generation
  - Tectonic plate simulation
- Geology-based terrain degradation
  - Cellular automata-based thermal and hydraulic erosion algorithm
  - Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain generation

---

- Tectonic plate simulation

# Geology-based terrain generation

---

## → Tectonic plate simulation

- Earth consists of two essential layers:
  - lithosphere (solid)
  - asthenosphere (liquid)
- The lithosphere is divided into distinct plates which float on top of the asthenosphere

# Geology-based terrain generation

---

## → Tectonic plate simulation

- There are several ways lithospheric plates can interact with each other:
  - They can grind past each other
  - They can slide apart from each other
  - They can slide toward each other

# Geology-based terrain generation

---

## → Tectonic plate simulation

- Due to the complexity of real tectonic plate movement, it was necessary to reduce the problem to two dimensions
- Polygons, representing tectonic plates, were used to simulate the geological phenomena involved in tectonic plate movement

# Geology-based terrain generation

---

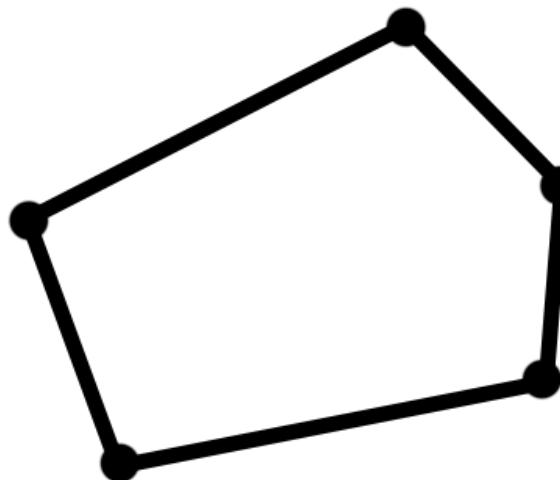
## → Tectonic plate simulation

- Box2D, a 2D physics engine was used to simulate polygon movement and detect collisions
- Splitting of polygons was not supported by Box2D and had to be implemented by hand

# Geology-based terrain generation

---

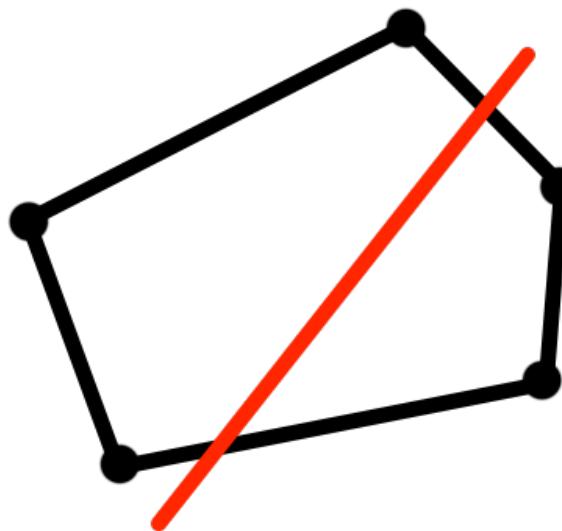
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

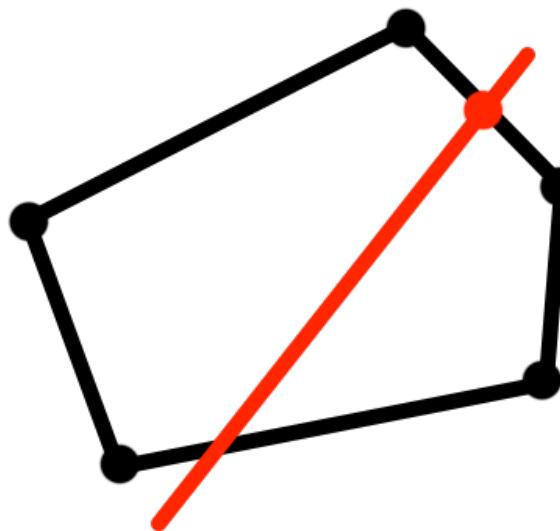
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

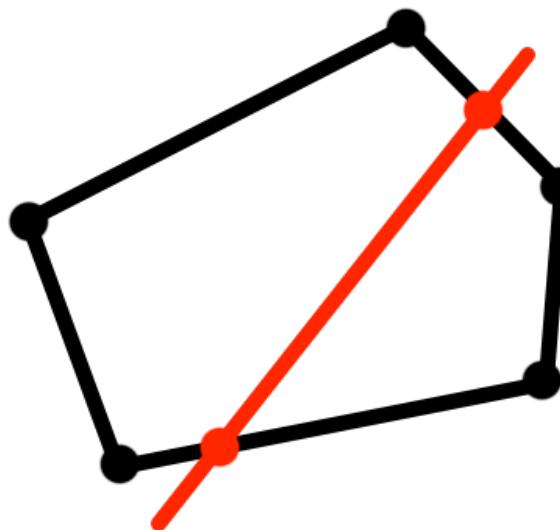
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

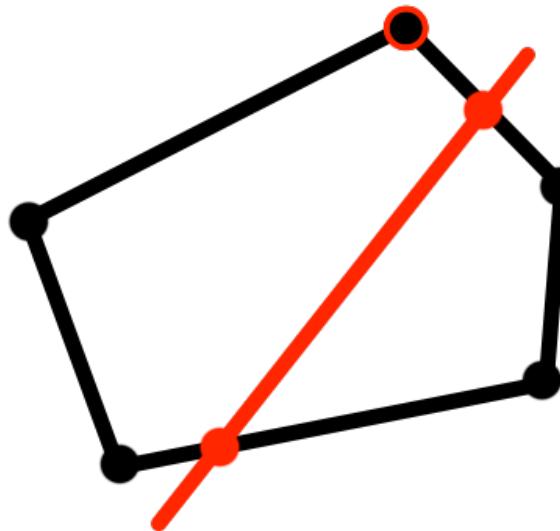
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

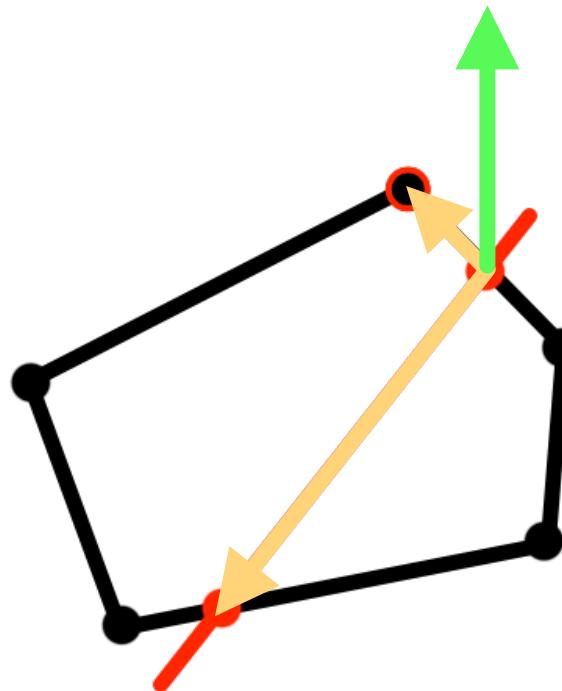
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

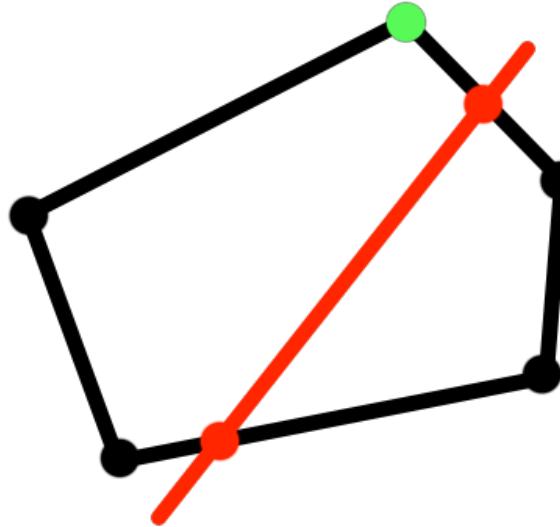
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

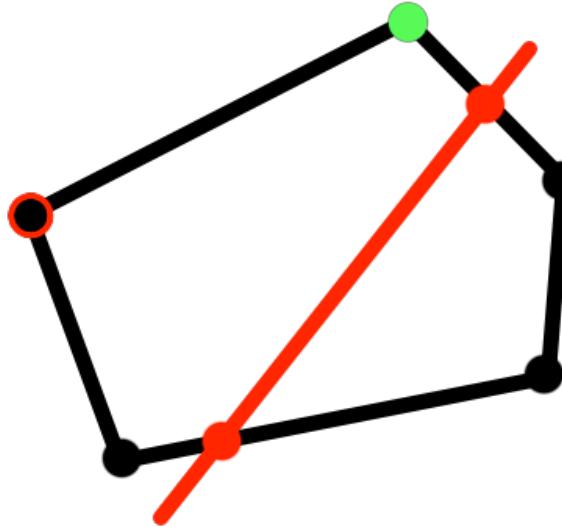
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

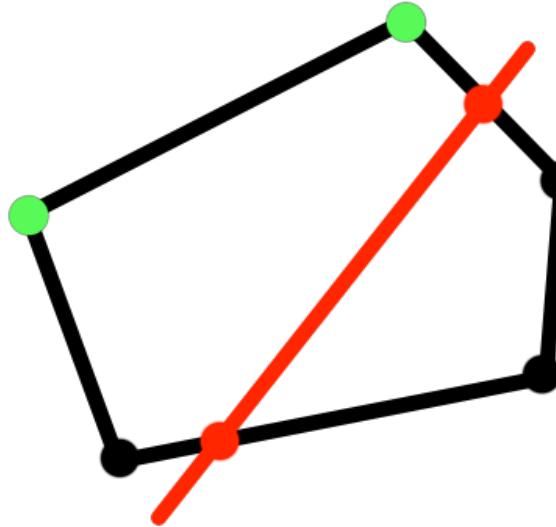
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

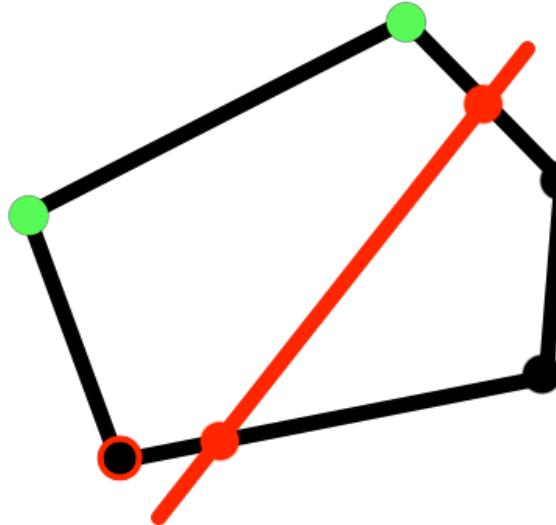
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

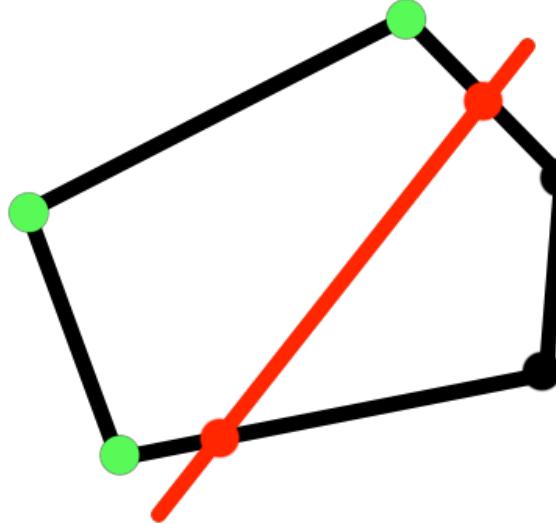
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

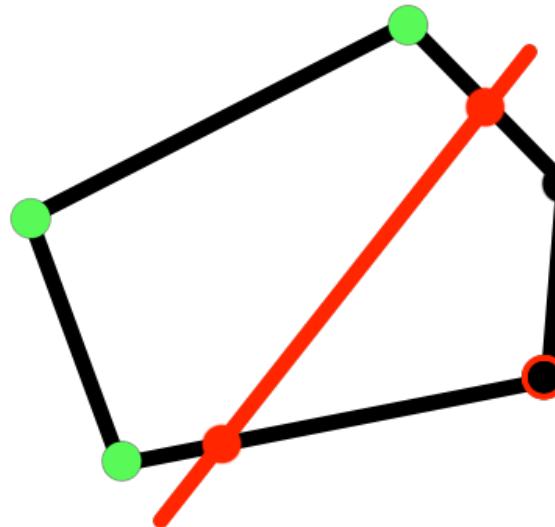
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

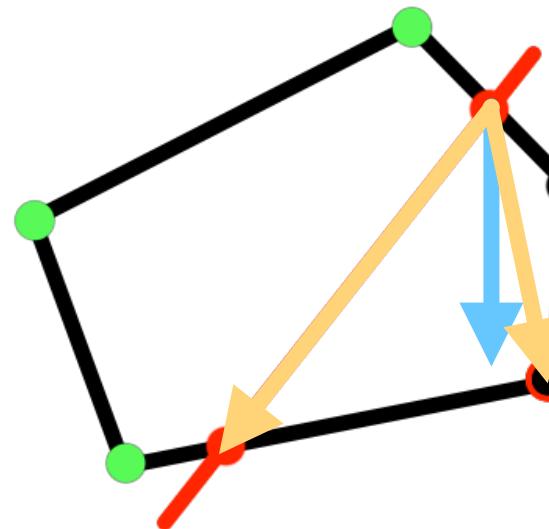
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

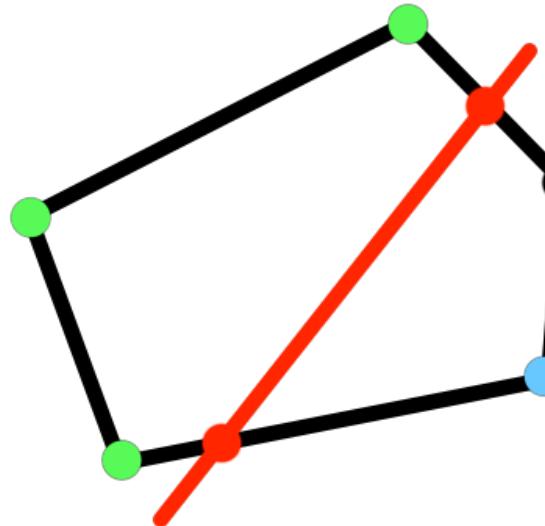
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

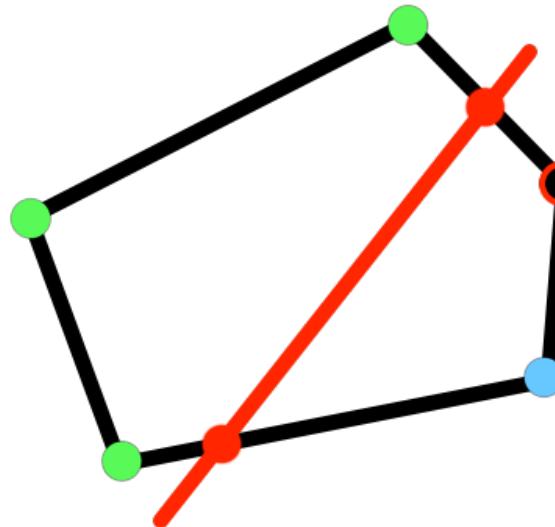
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

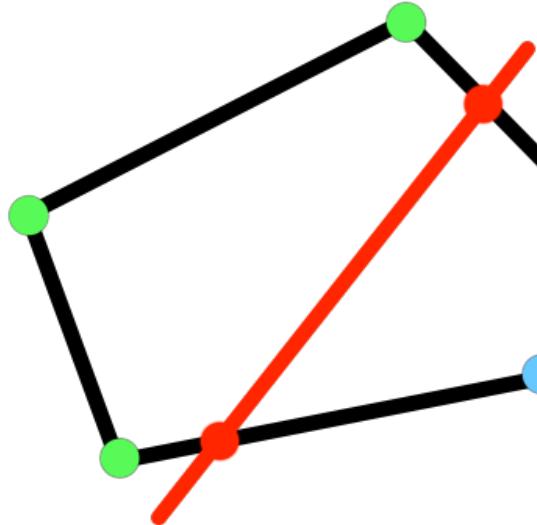
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

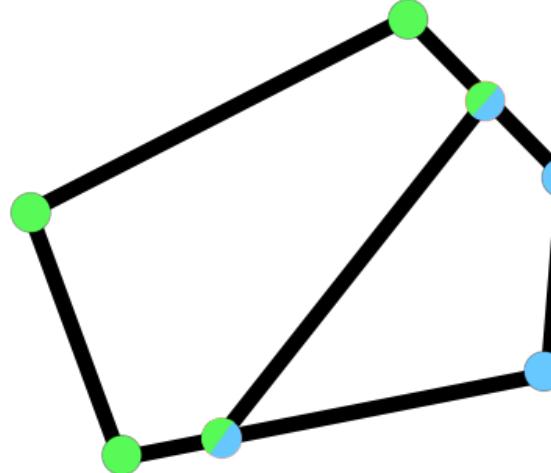
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

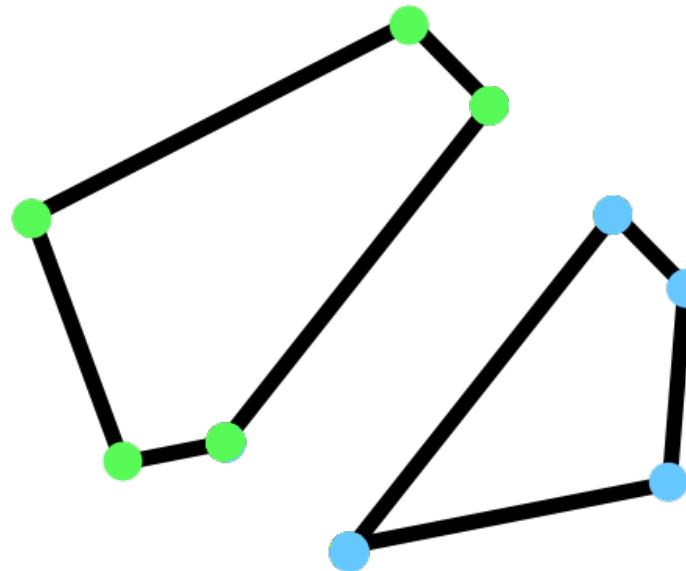
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

---

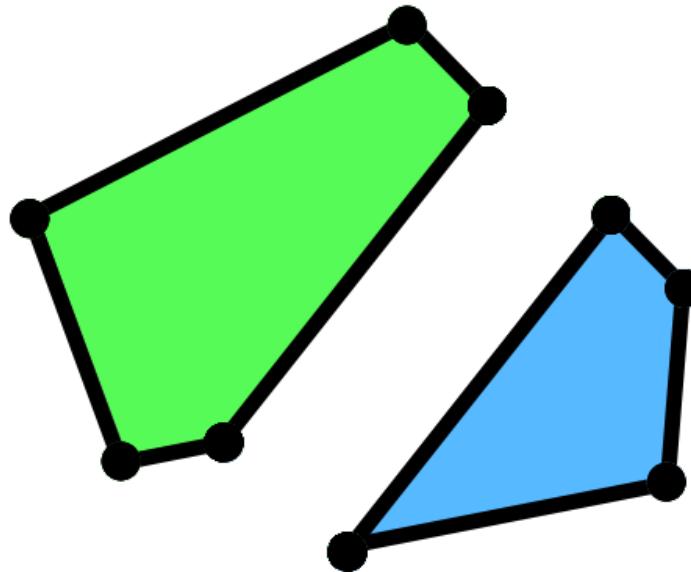
- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

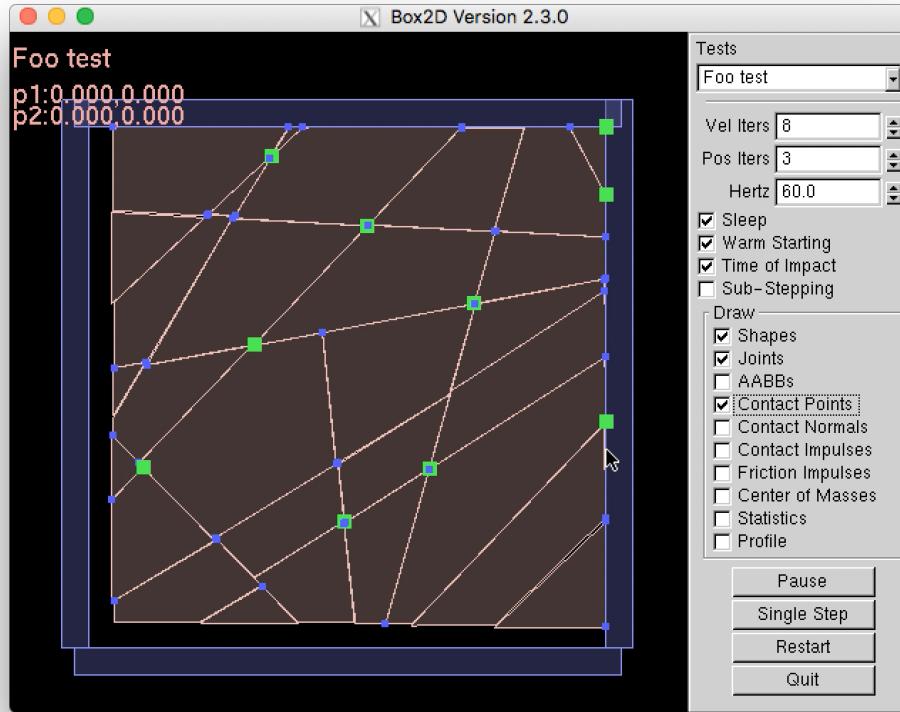
---

- Tectonic plate simulation
  - Splitting of polygons:



# Geology-based terrain generation

- Tectonic plate simulation
  - DEMO Box2D



# Geology-based terrain generation

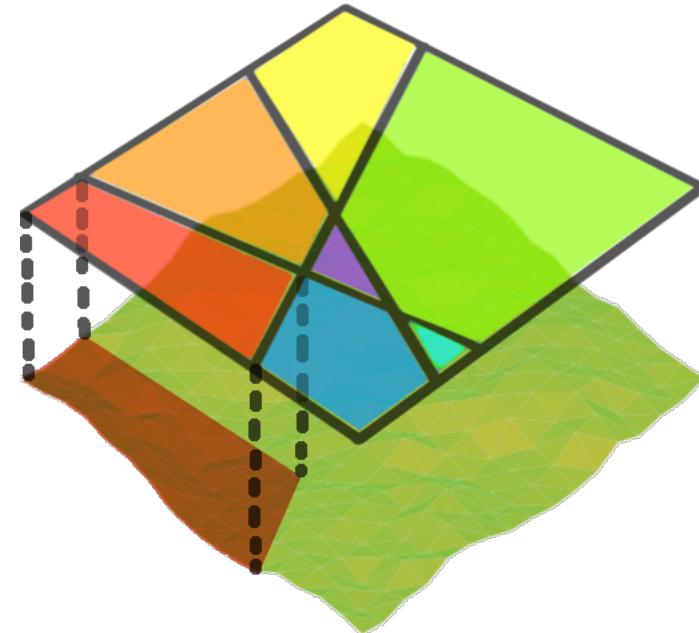
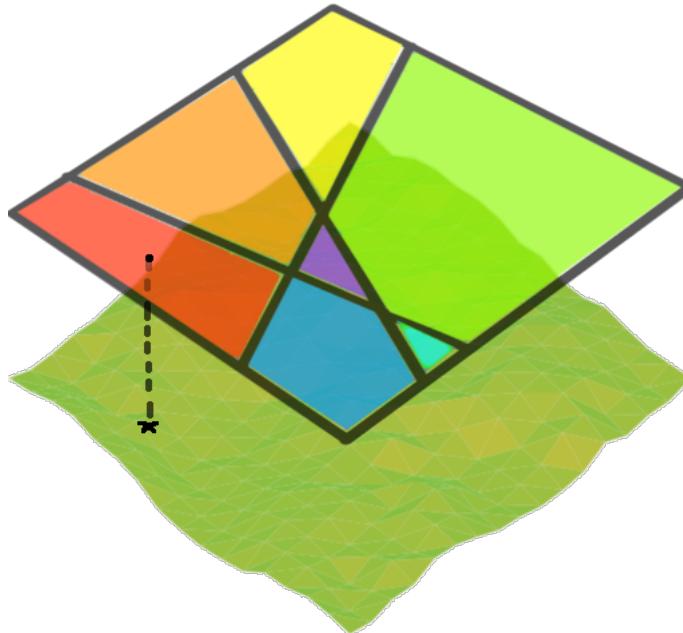
---

## → Tectonic plate simulation

- What to do with collision data?  
Generate mountain ranges between two colliding plates.
- How?  
Use an advanced version of Voronoi-based RMP algorithm.

# Geology-based terrain generation

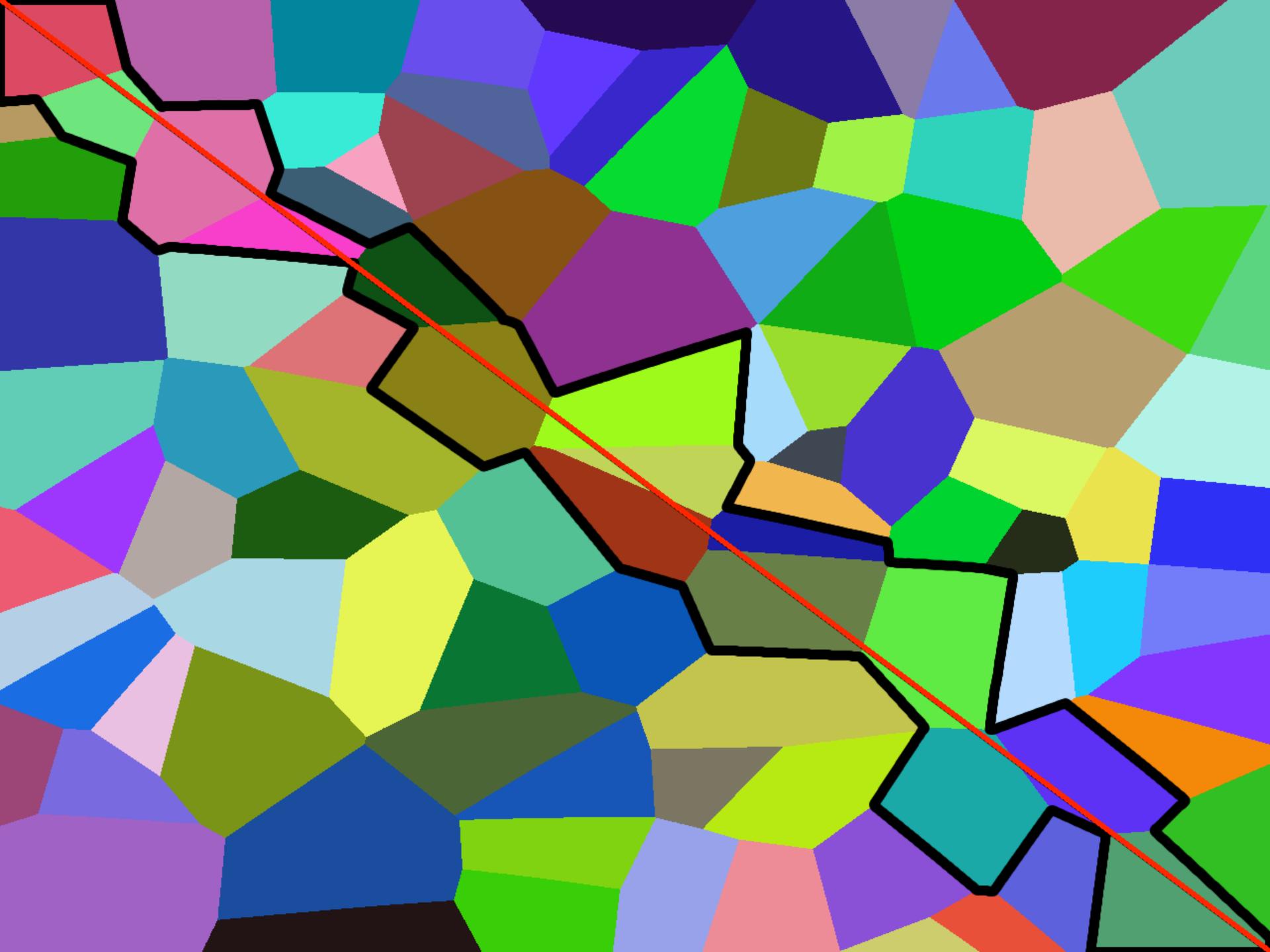
- Tectonic plate simulation
  - Recap Voronoi-based RMP algorithm

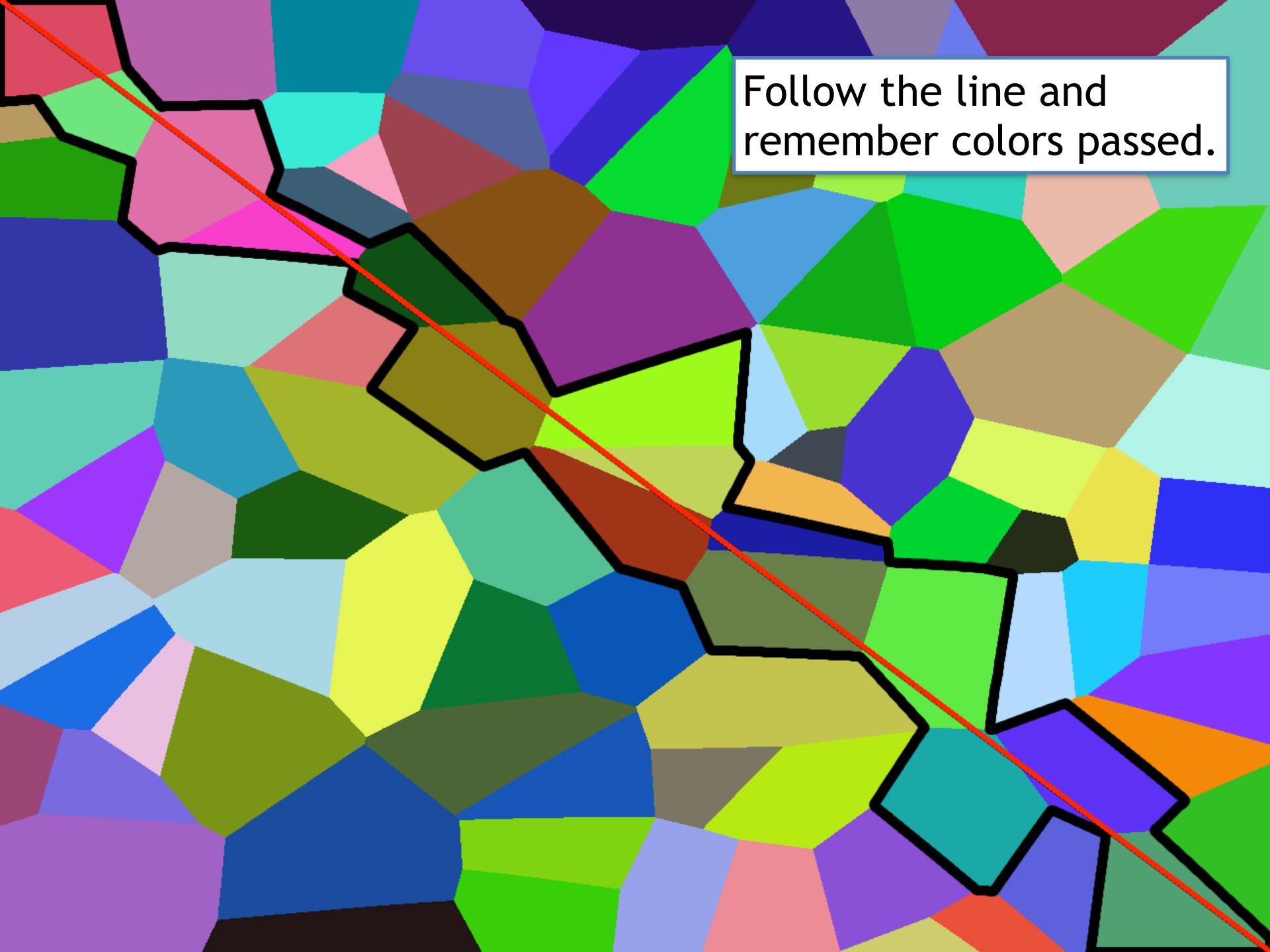


# Geology-based terrain generation

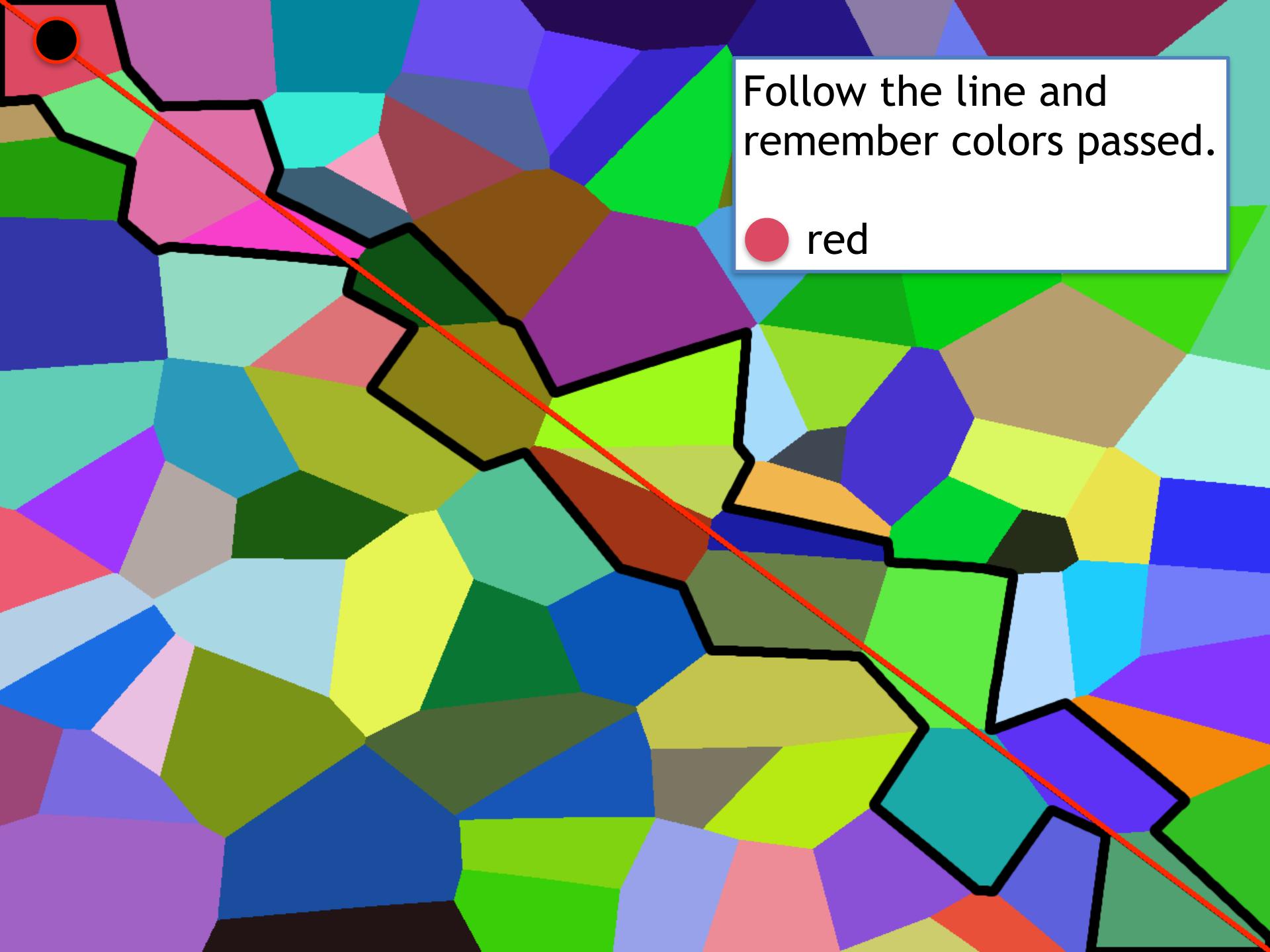
---

- Tectonic plate simulation
  - Mountain range generation algorithm
    - Difference to Voronoi-based RMP algorithm?  
An array of polygons instead of a single polygon, is raised in height.



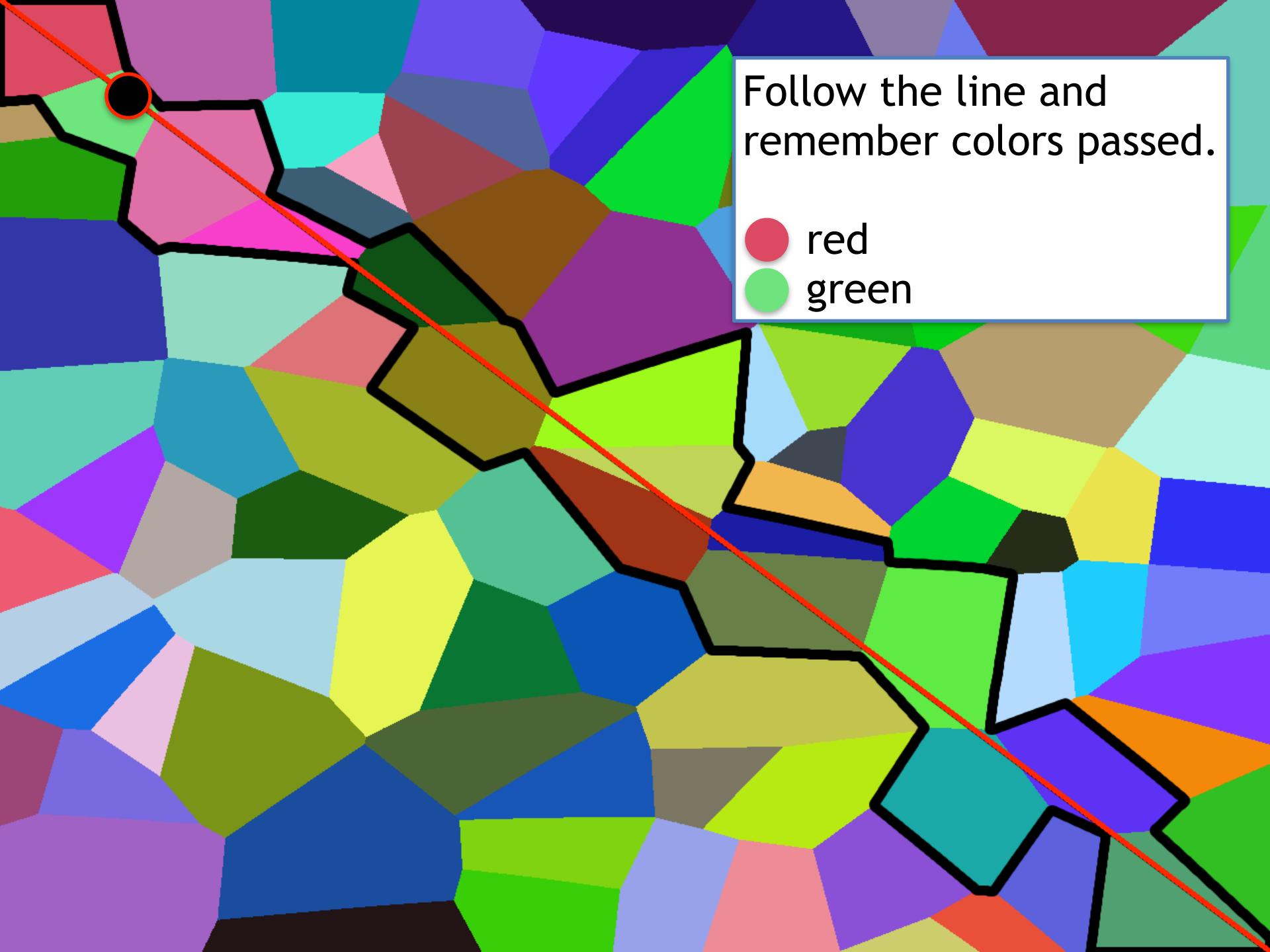


Follow the line and  
remember colors passed.



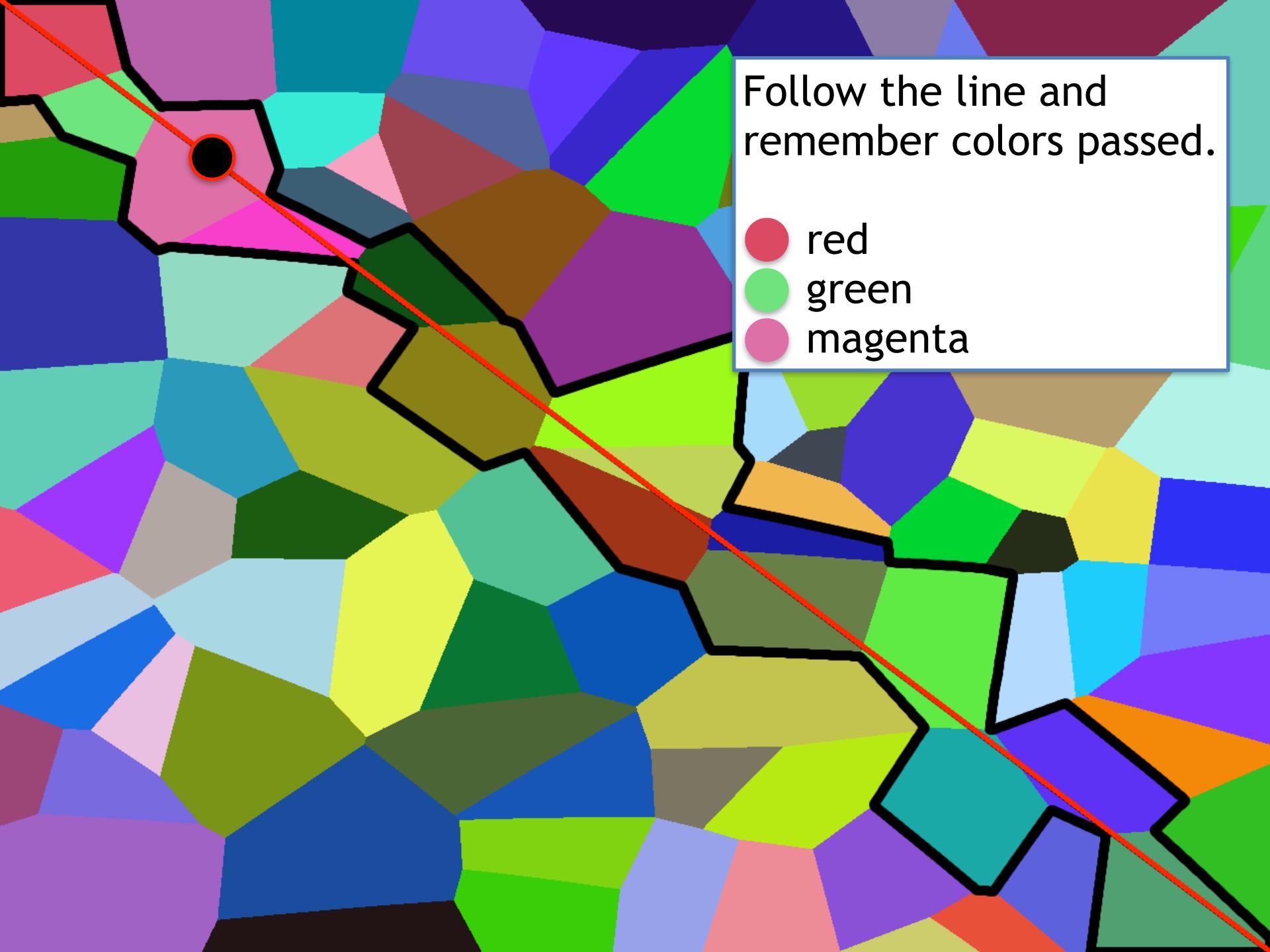
Follow the line and  
remember colors passed.

red



Follow the line and  
remember colors passed.

- red
- green



Follow the line and  
remember colors passed.

- red
- green
- magenta



Follow the line and  
remember colors passed.

- red
- green
- magenta
- pink



Follow the line and  
remember colors passed.

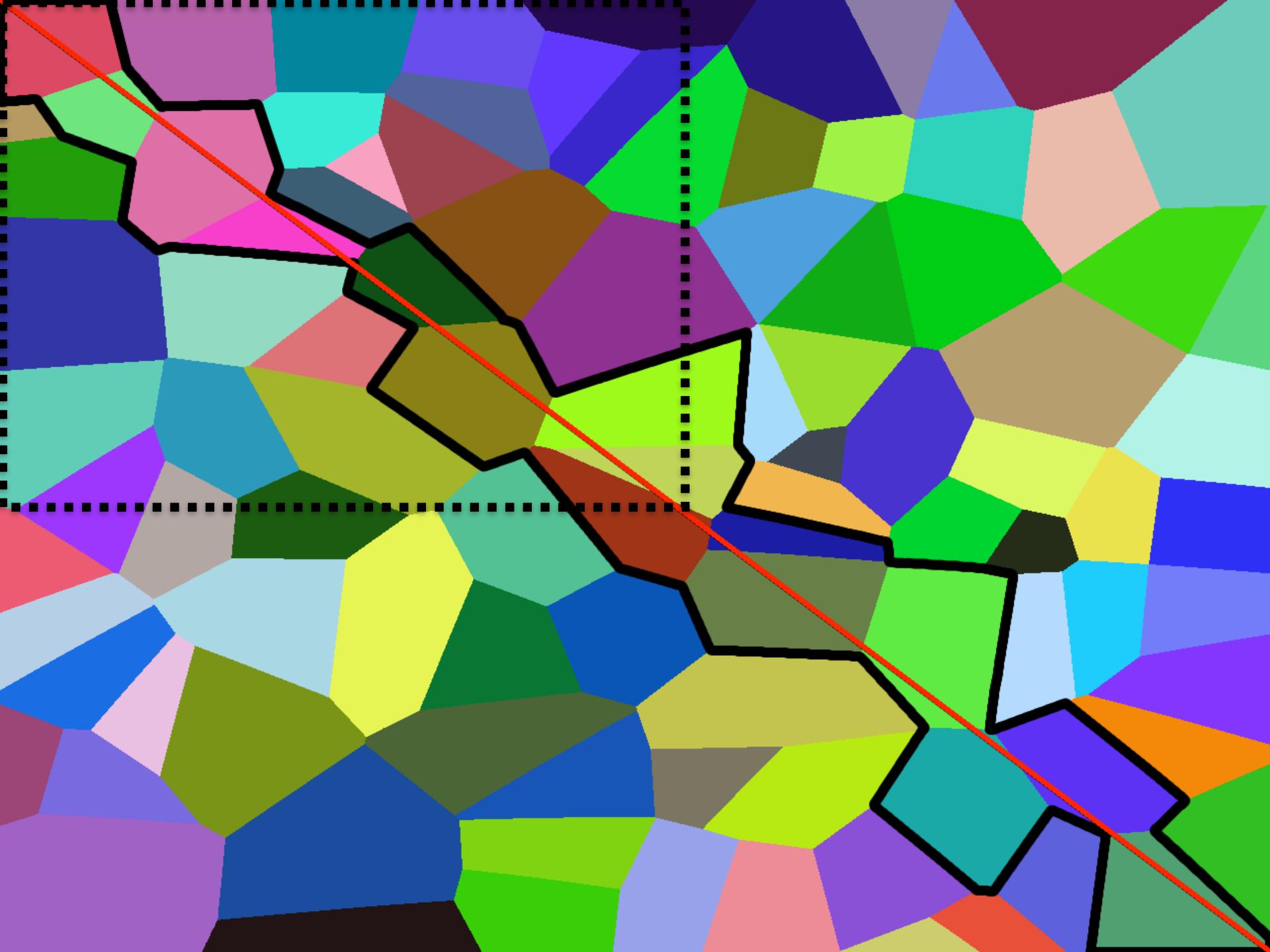
- red
- green
- magenta
- pink
- dark green

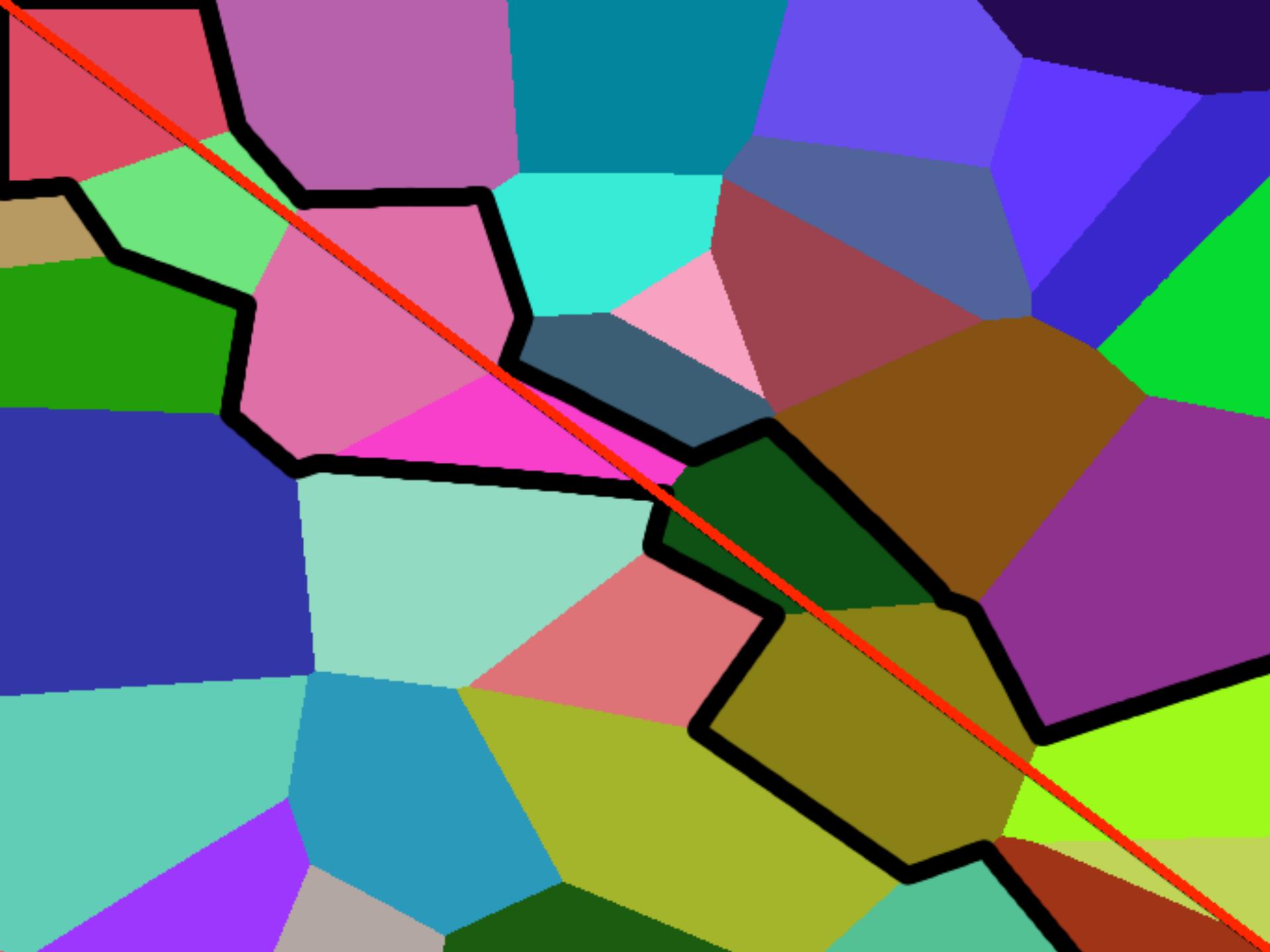


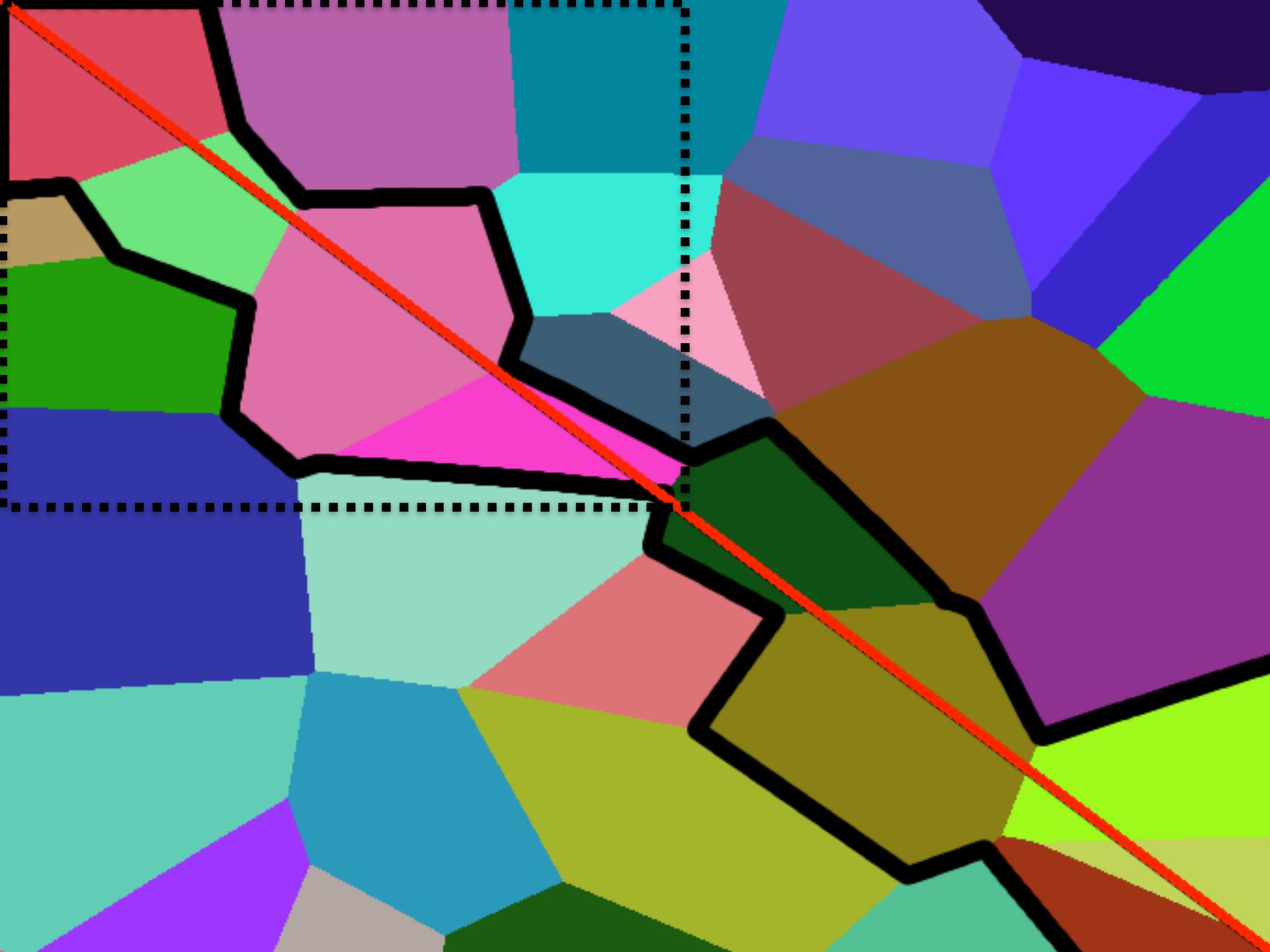
Follow the line and  
remember colors passed.

- red
- green
- magenta
- pink
- dark green
- olive

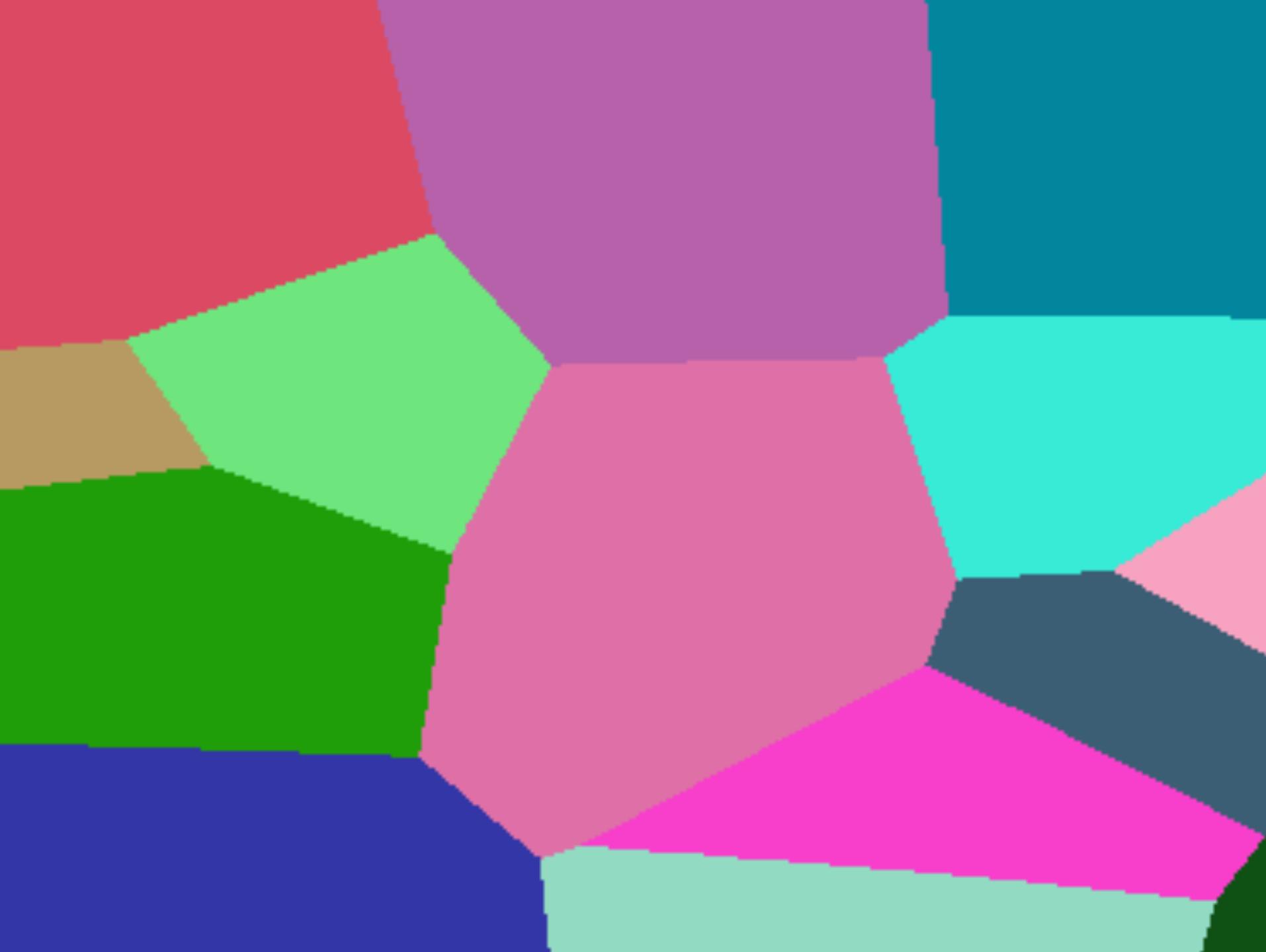
...

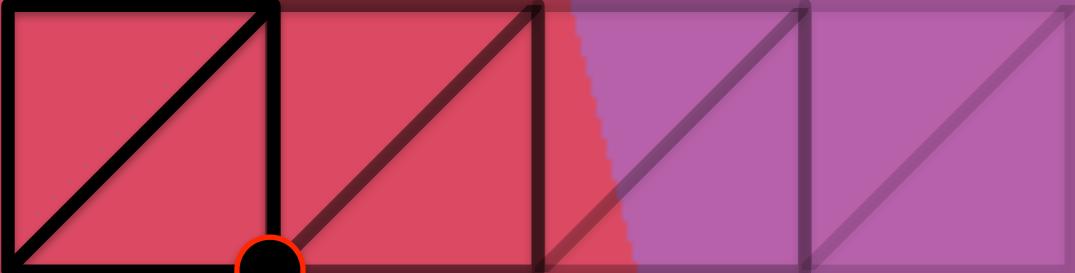








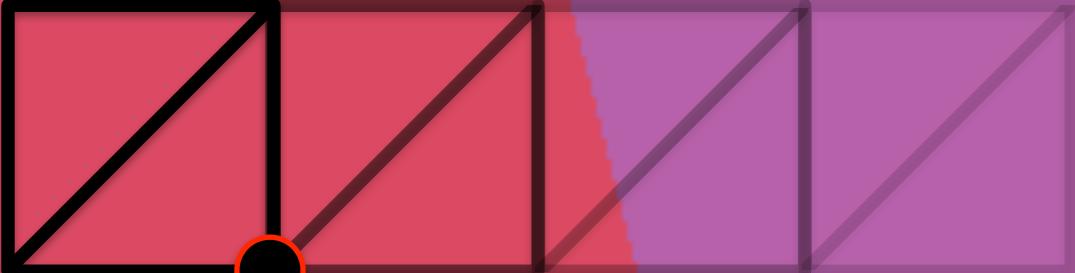




Check if any of the colors is projected onto a heightmap vertex.

- red
- green
- magenta
- pink
- dark green
- olive

...

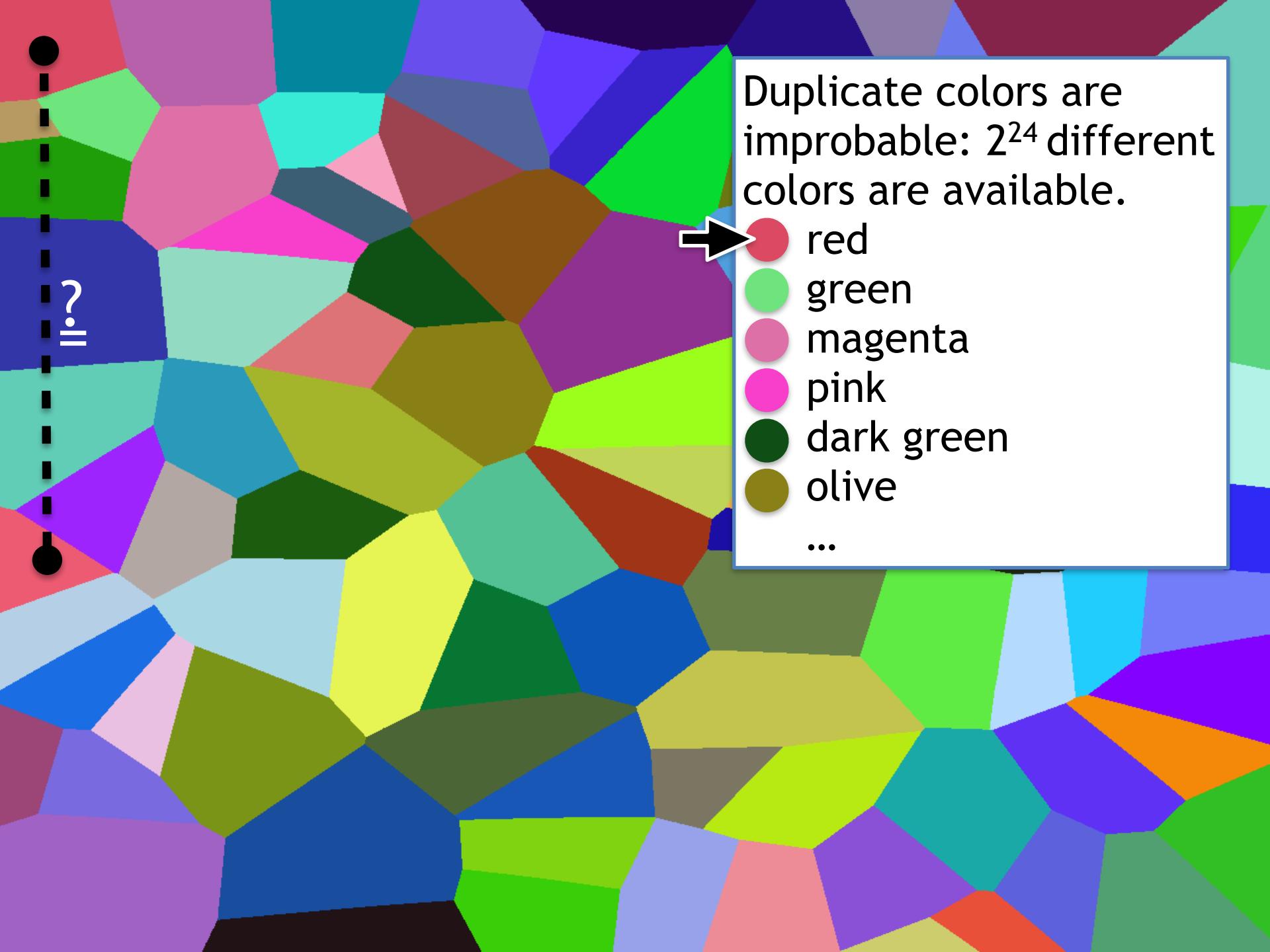


If the check is positive,  
raise the heightmap  
vertex.



- red ✓
- green
- magenta
- pink
- dark green
- olive

...

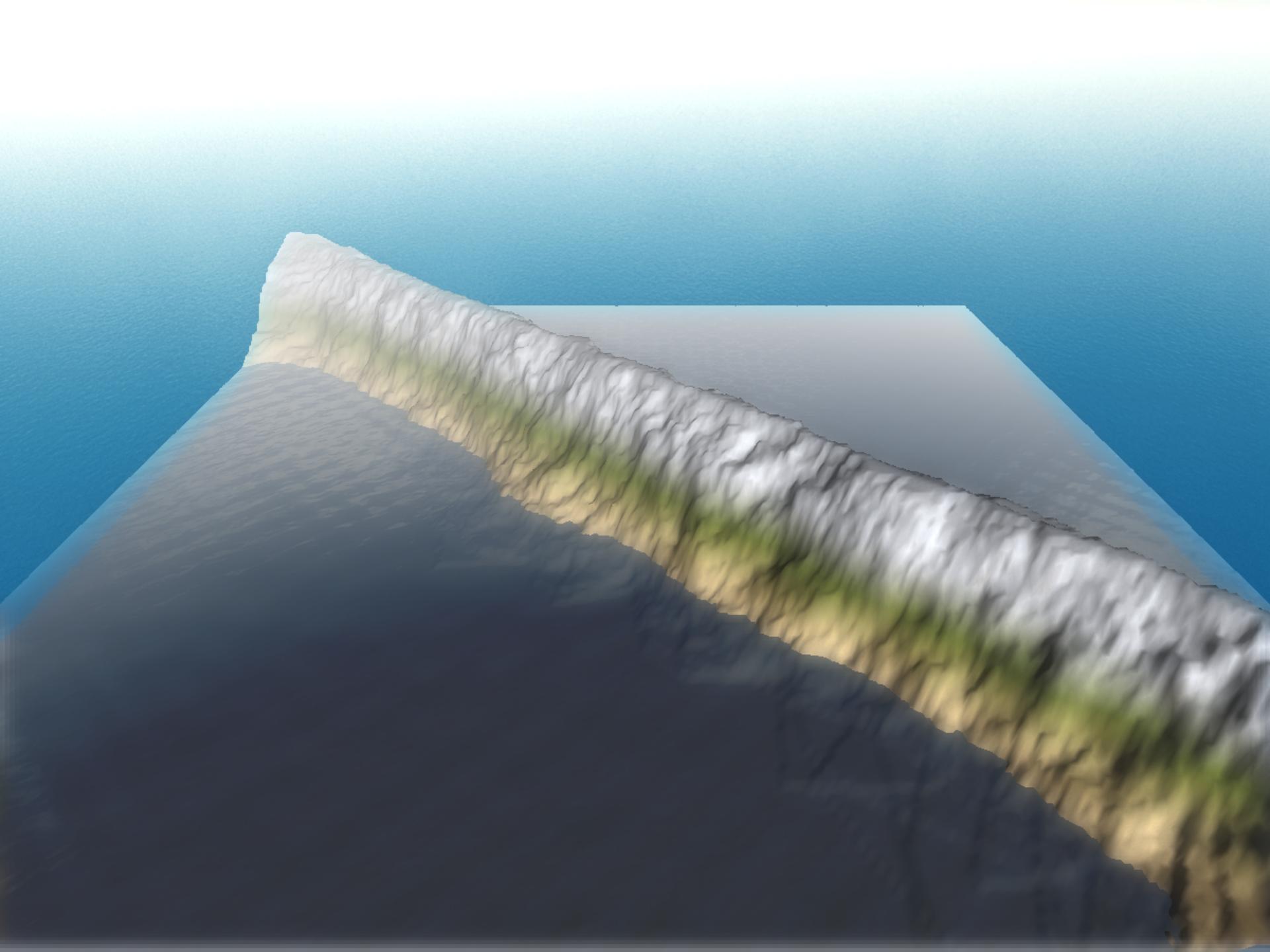


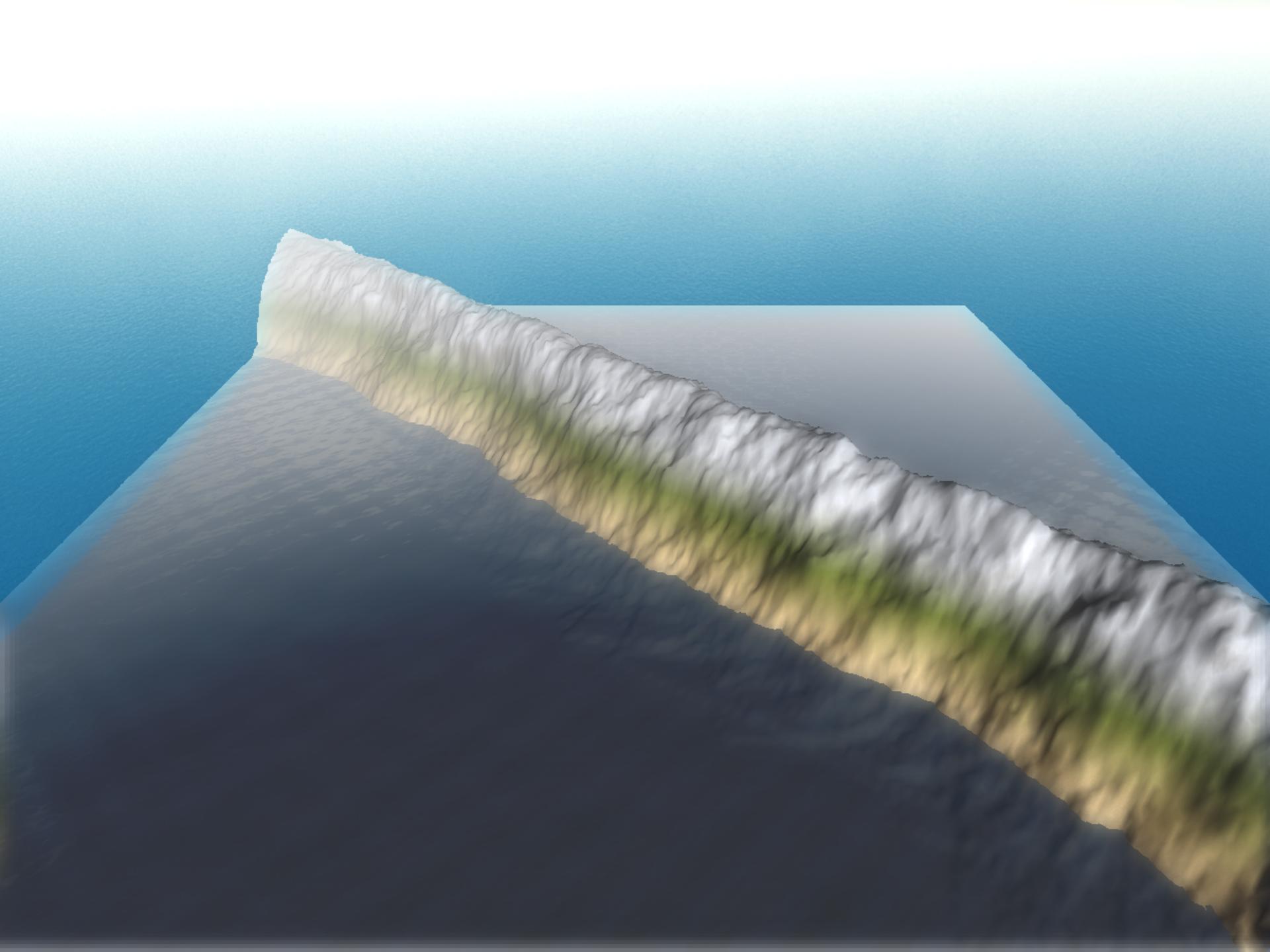
Duplicate colors are improbable:  $2^{24}$  different colors are available.



- red
- green
- magenta
- pink
- dark green
- olive

...





# Geology-based terrain degradation

---

- Cellular automata-based thermal and hydraulic erosion algorithm
- Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain degradation

---

- Cellular automata-based thermal and hydraulic erosion algorithm
- Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain degradation

---

→ CA-based thermal and hydraulic erosion algorithm

- What is a cellular automaton?
  - It consists of a grid of cells.
  - There are simple rules how cells interact with each other.
  - Each cell can only interact with other cells in their neighborhood.
  - A neighborhood is a set of cells.
  - One cell can be part of more than one neighborhood.
  - Cellular automata are step-based.

# Geology-based terrain degradation

---

→ CA-based thermal and hydraulic erosion algorithm

- 1D Example

- Initial state: 

0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---

- Neighborhood: 

--	--	--

- Rules:

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

# Geology-based terrain degradation

---

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0								

handle edge cases by wrapping around

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	1	0	1

0	0	0	0	1	0	0	0	0
0	0							

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	1	0	1

0	0	0	0	1	0	0	0	0
0	0	0						

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1					

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0				

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1			

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0		

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	1	0	1

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	1	0	1

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0

# Geology-based terrain degradation

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1	0

# Geology-based terrain degradation

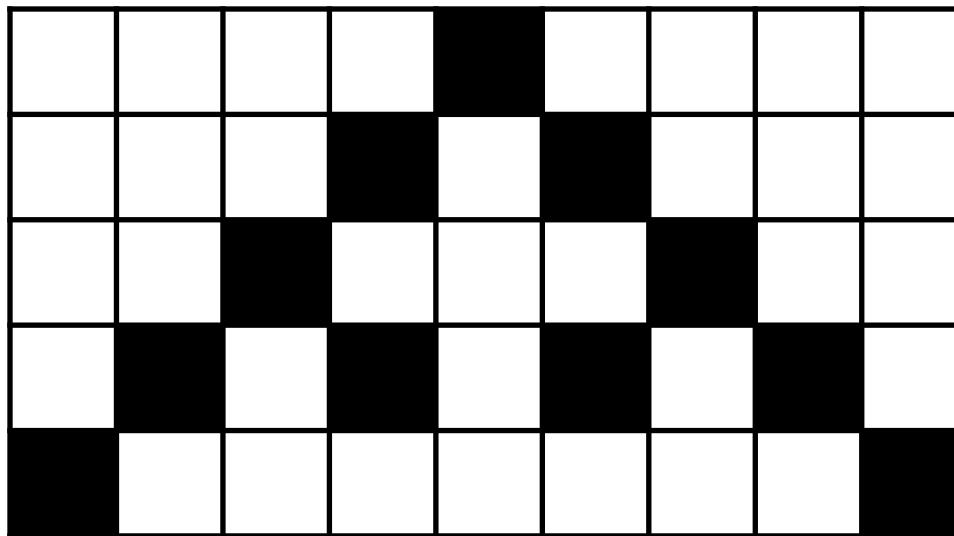
000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1	0
1	0	0	0	0	0	0	0	1

# Geology-based terrain degradation

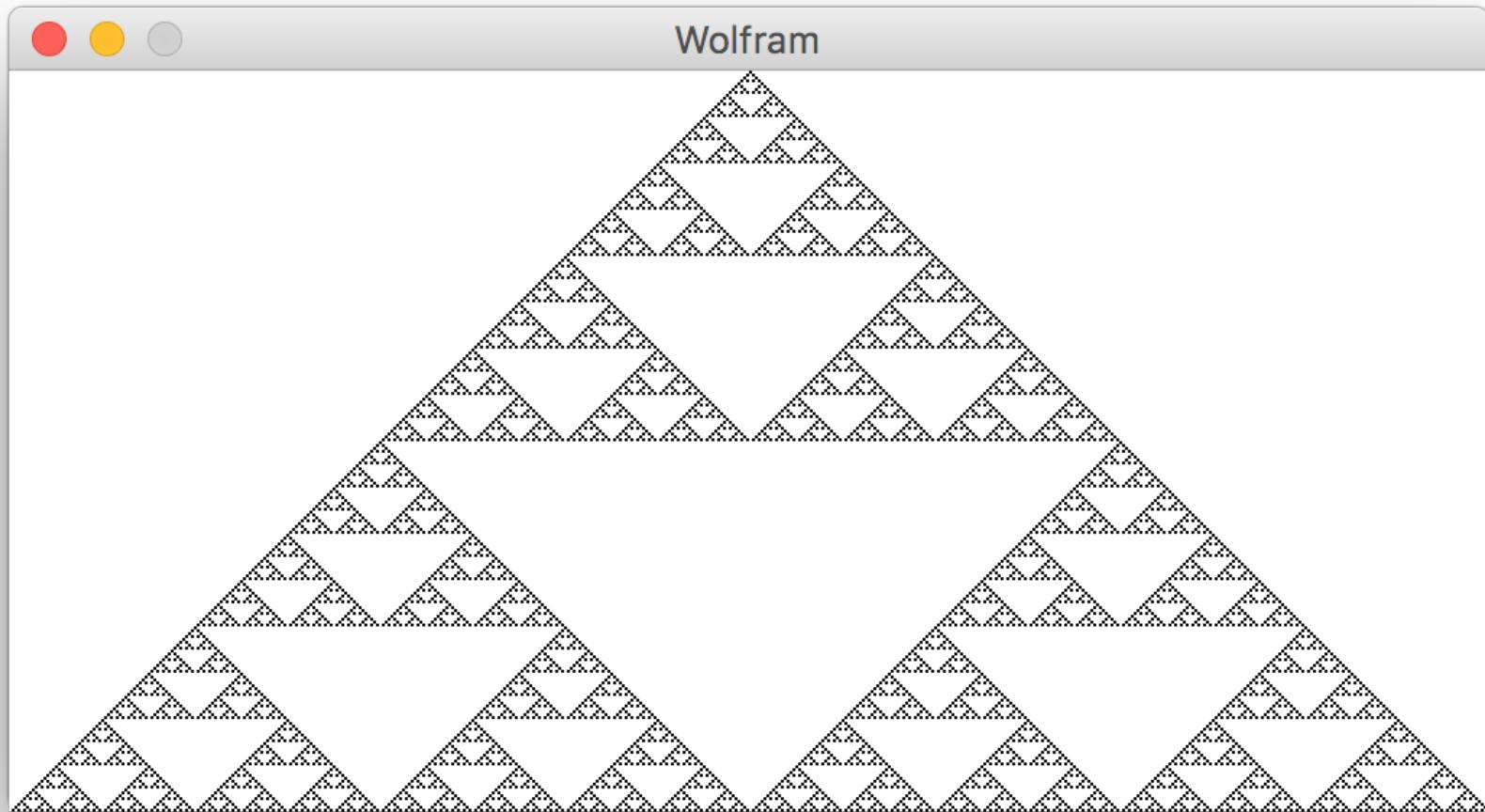
---

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0



# Geology-based terrain degradation

---

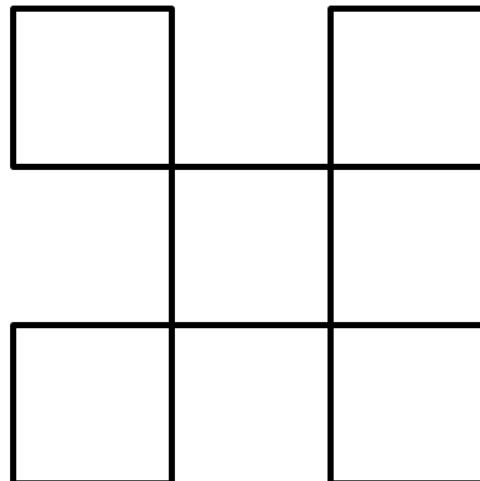


# Geology-based terrain degradation

---

→ CA-based thermal and hydraulic erosion algorithm

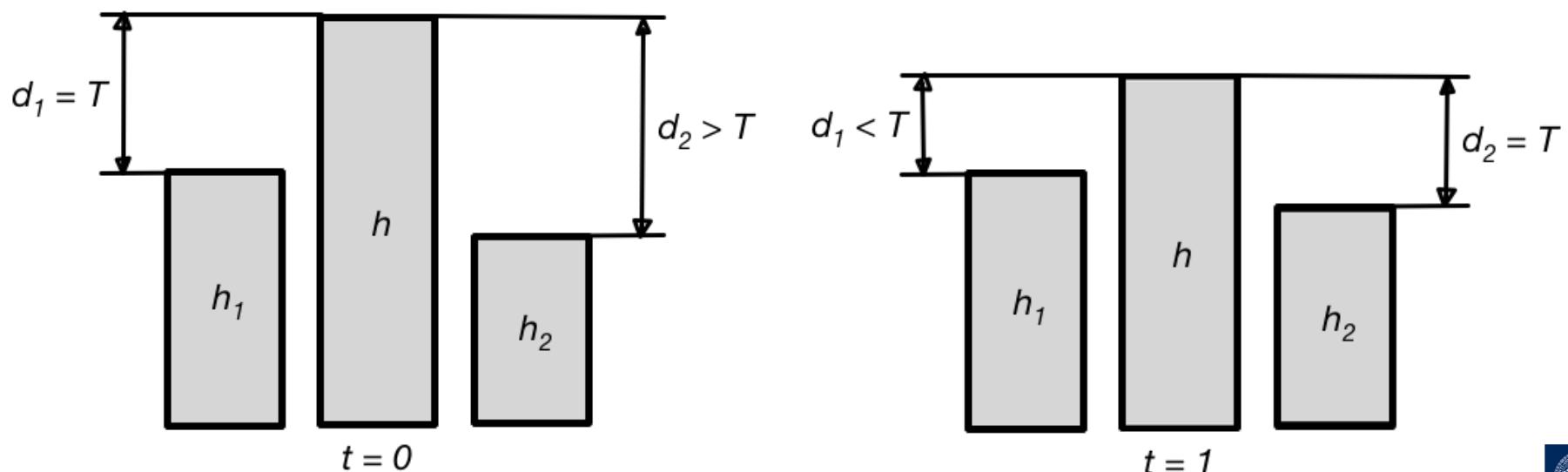
- Thermal erosion algorithm
  - 2D cellular automaton
  - Rotated Von-Neumann neighborhood

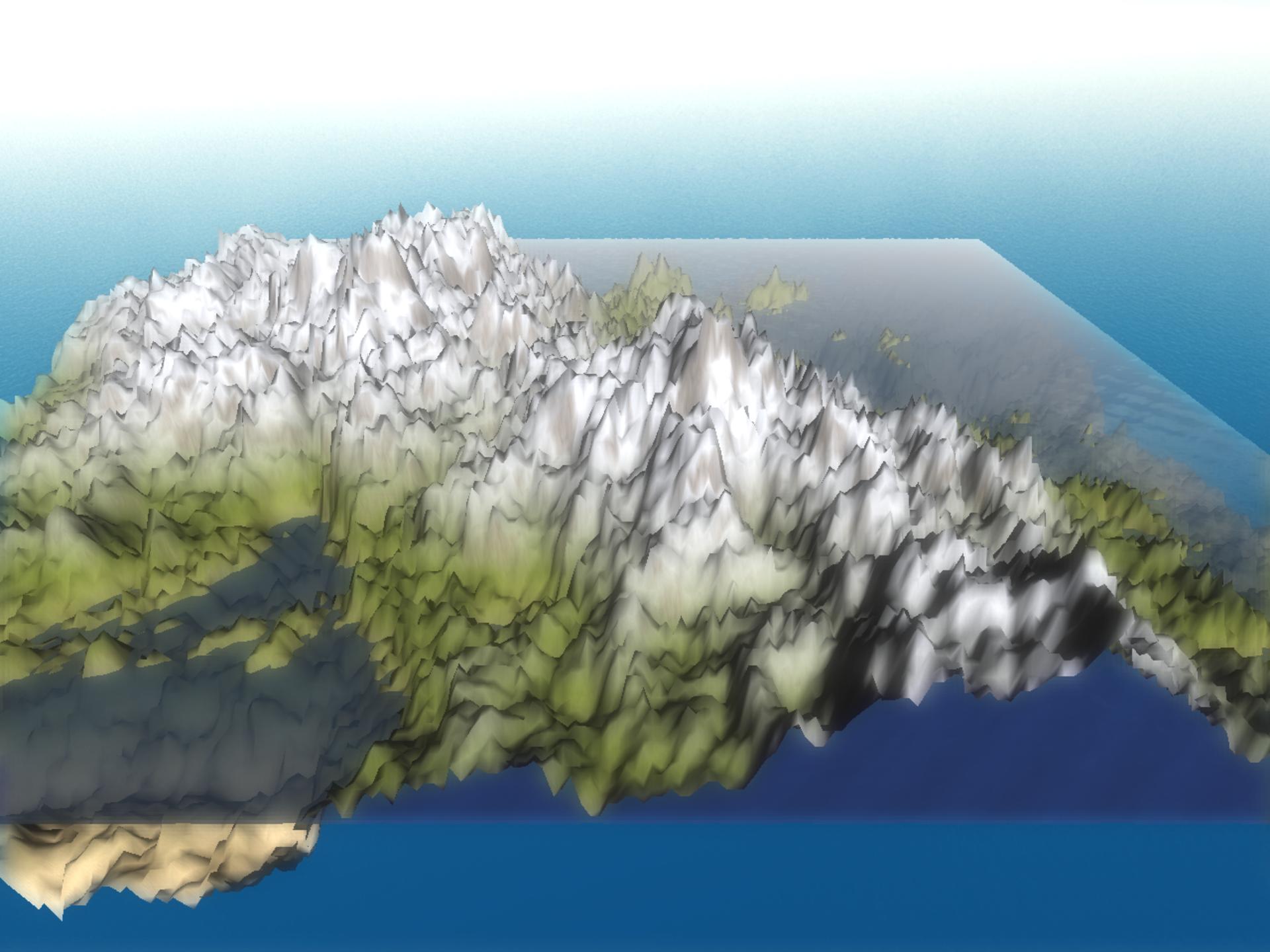


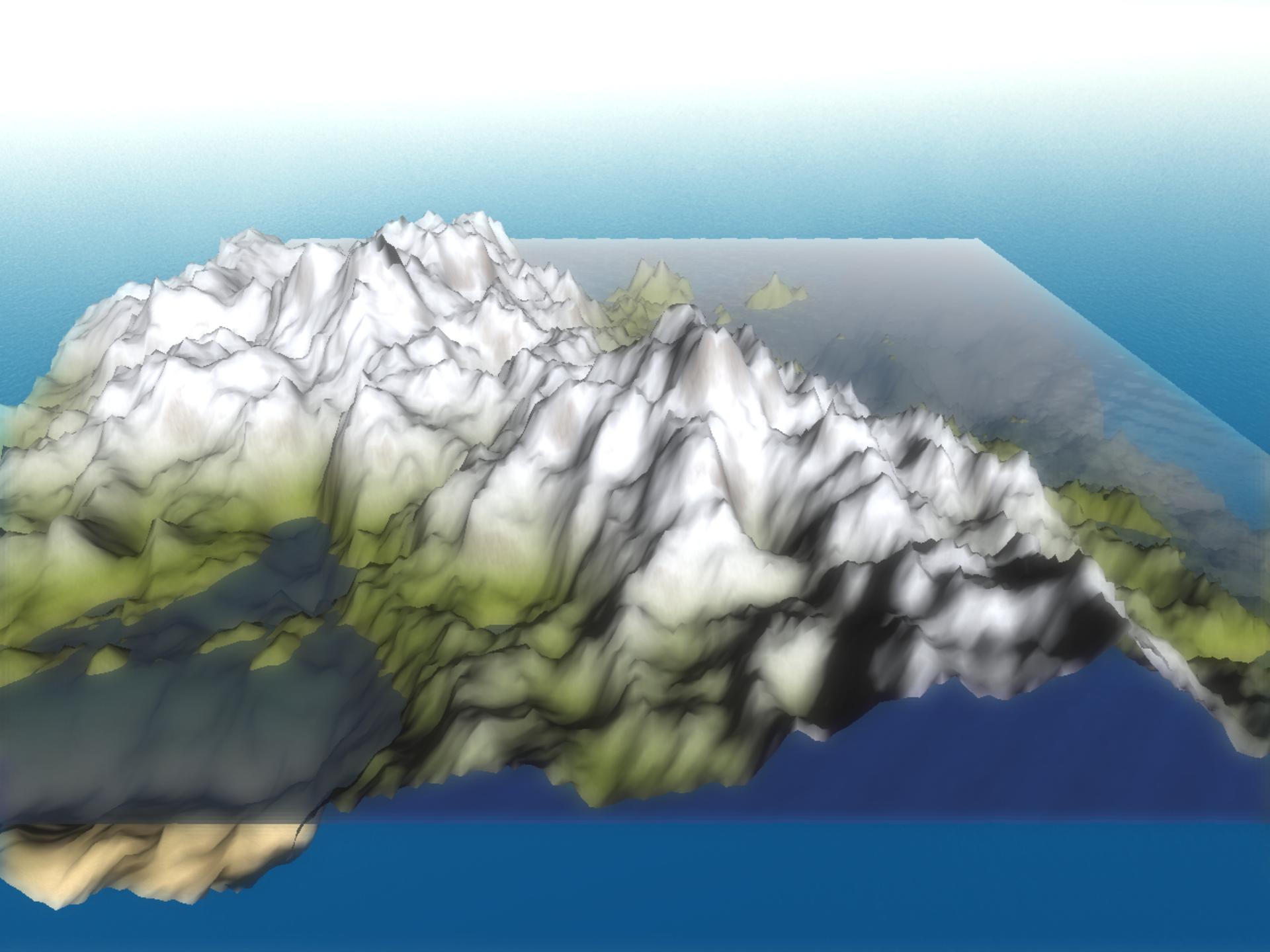
# Geology-based terrain degradation

→ CA-based thermal and hydraulic erosion algorithm

- Thermal erosion algorithm
  - Rules:



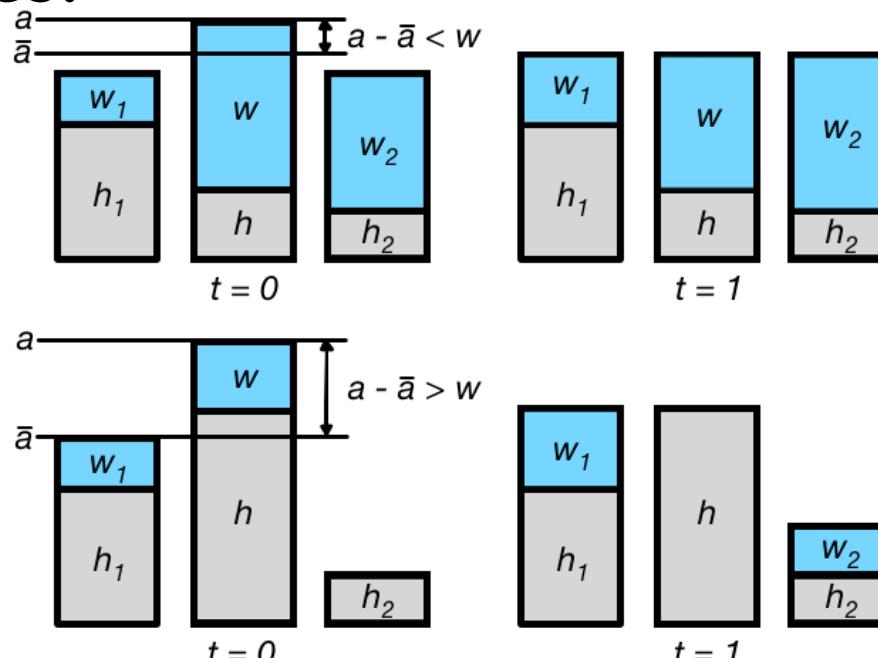




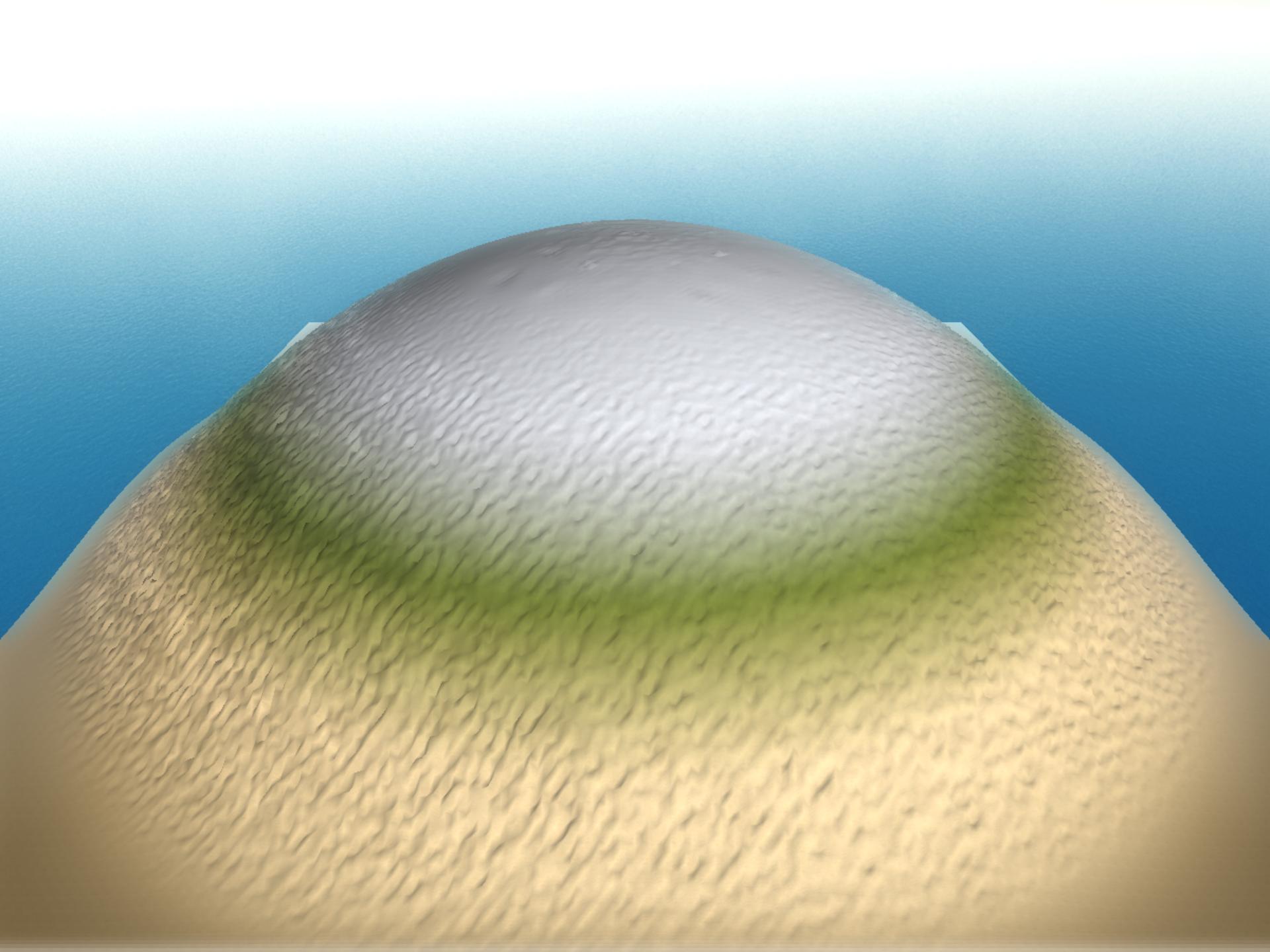
# Geology-based terrain degradation

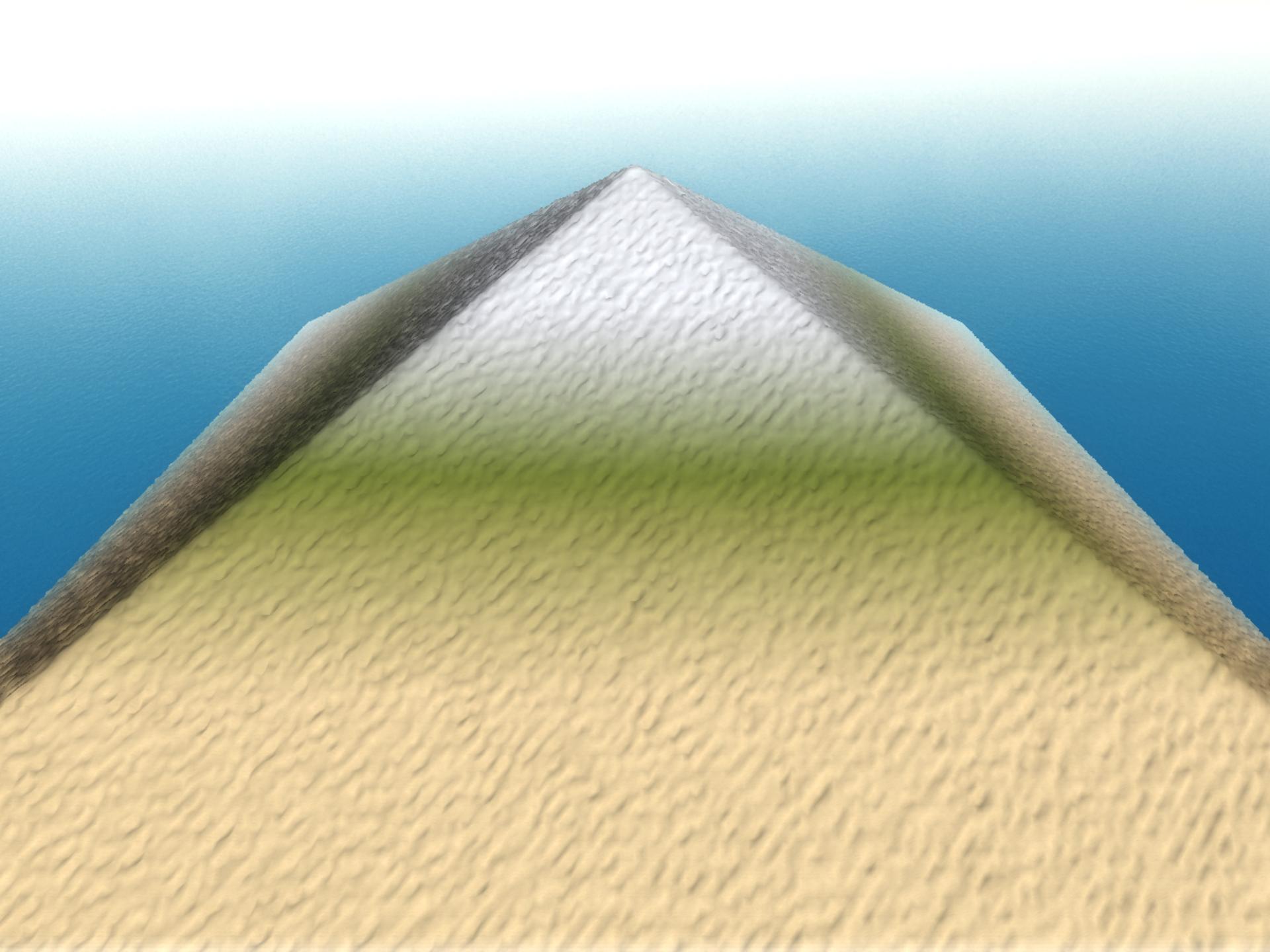
→ CA-based thermal and hydraulic erosion algorithm

- Hydraulic erosion algorithm
  - Rules:









# Geology-based terrain degradation

---

→ Cellular automata-based thermal and hydraulic erosion algorithm

- Results of CA-based hydraulic erosion algorithm are underwhelming
- Ruleset does not consider flow velocity
- Different approach: Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain degradation

---

- Cellular automata-based thermal and hydraulic erosion algorithm
- Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain degradation

---

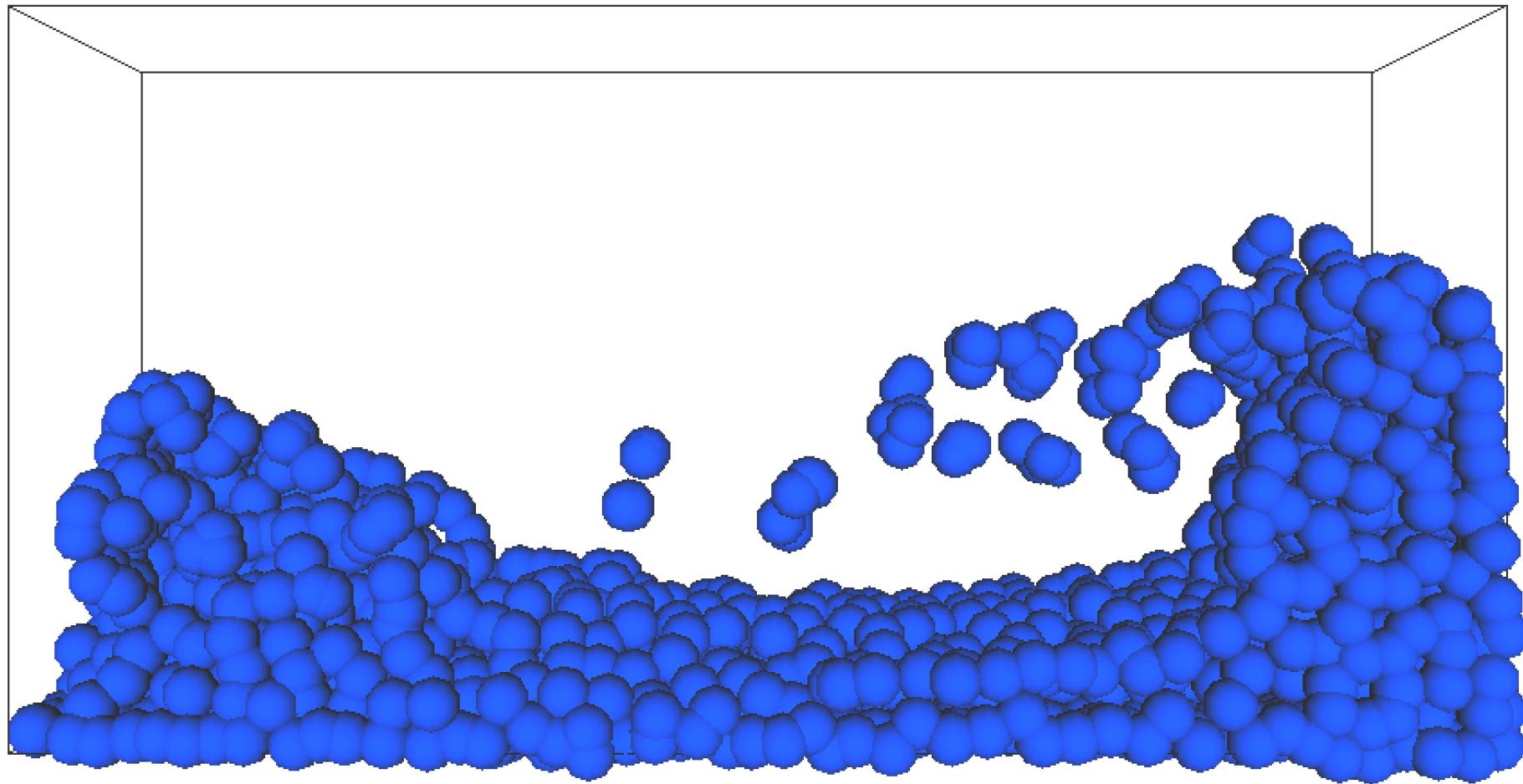
- Cellular automata-based thermal and hydraulic erosion algorithm
- Smoothed particle hydrodynamics-based hydraulic erosion simulation

# Geology-based terrain degradation

---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - What is a SPH simulation?
    - SPH is a technique used to discretize fluid motion
    - SPH is an approximation of the Navier-Stokes equations on a set of particles

# Geology-based terrain degradation



<https://github.com/saeedmahani/SPH-3D-Fluid-Simulation>

# Geology-based terrain degradation

---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - How to make particles be influenced by an arbitrary surface?
    - Use static, non-moving particles, distributed over the surface
    - Apply an impulse on collision with surface

# Geology-based terrain degradation

---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - How to make particles be influenced by an arbitrary surface?
    - ~~Use static, non moving particles, distributed over the surface~~
    - Apply an impulse on collision with surface

# Geology-based terrain degradation

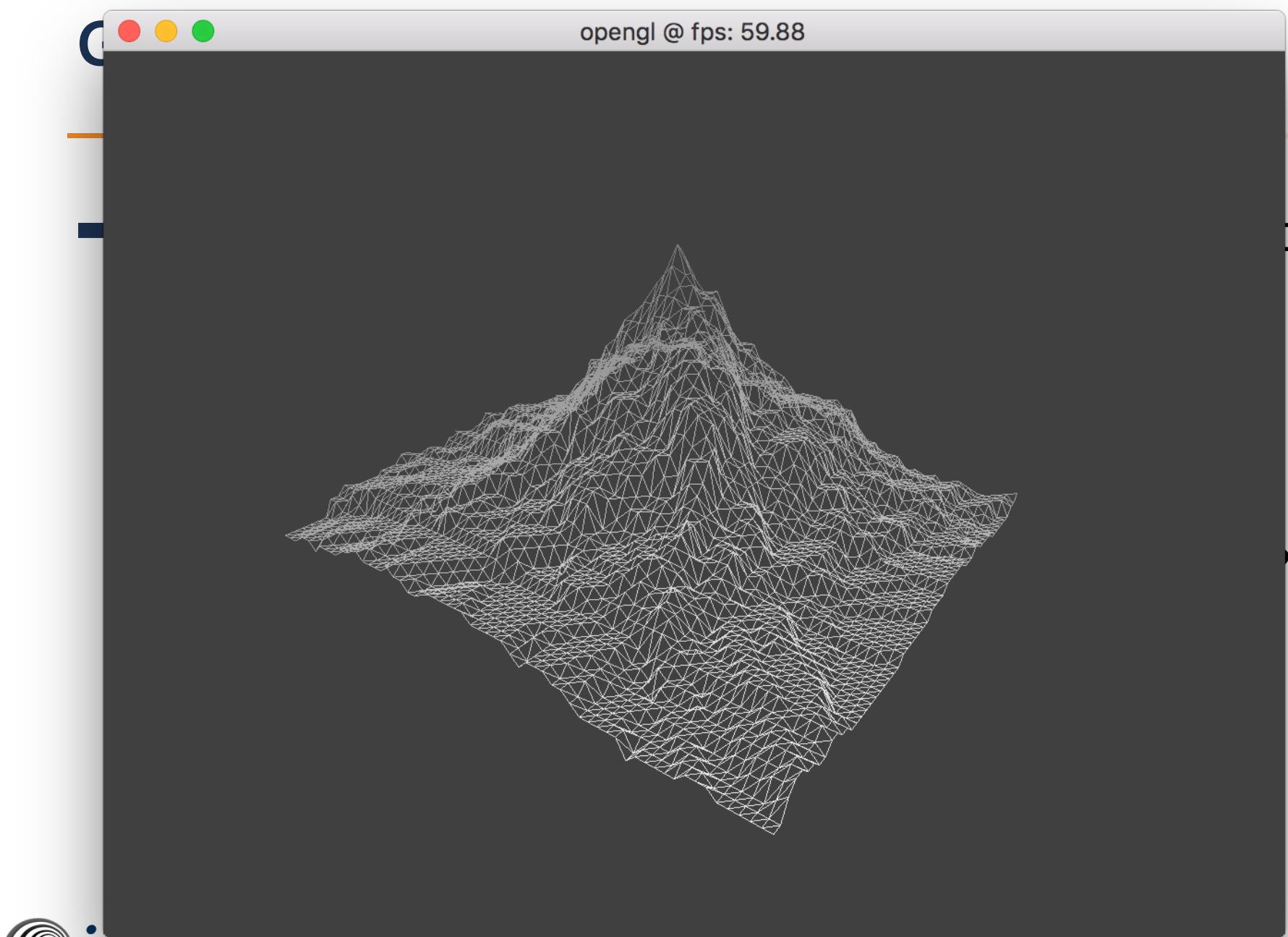
---

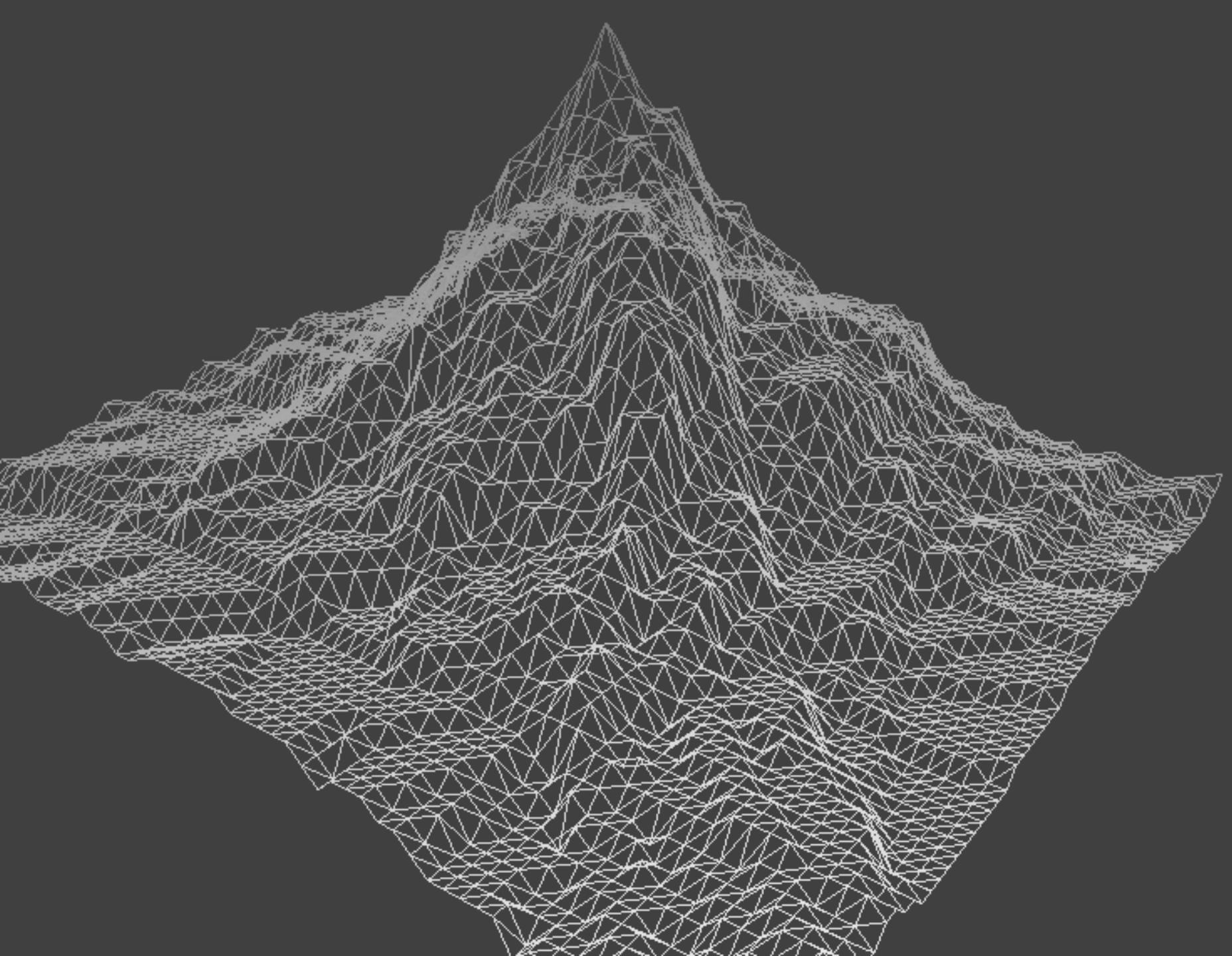
- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - The impulse has to be applied in a direction perpendicular to the surface the particle collided with
  - Two problems:
    1. How to determine the collision point?
    2. How to determine the impulse direction?

# Geology-based terrain degradation

---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - How to determine the collision point?
    - How to determine if the particle even collided with the heightmap?
      - A particle with coordinates  $(p_x, p_y, p_z)$ , collided with the heightmap if the heightmap elevation at position  $(p_x, p_z)$  is equal or larger than  $p_y$







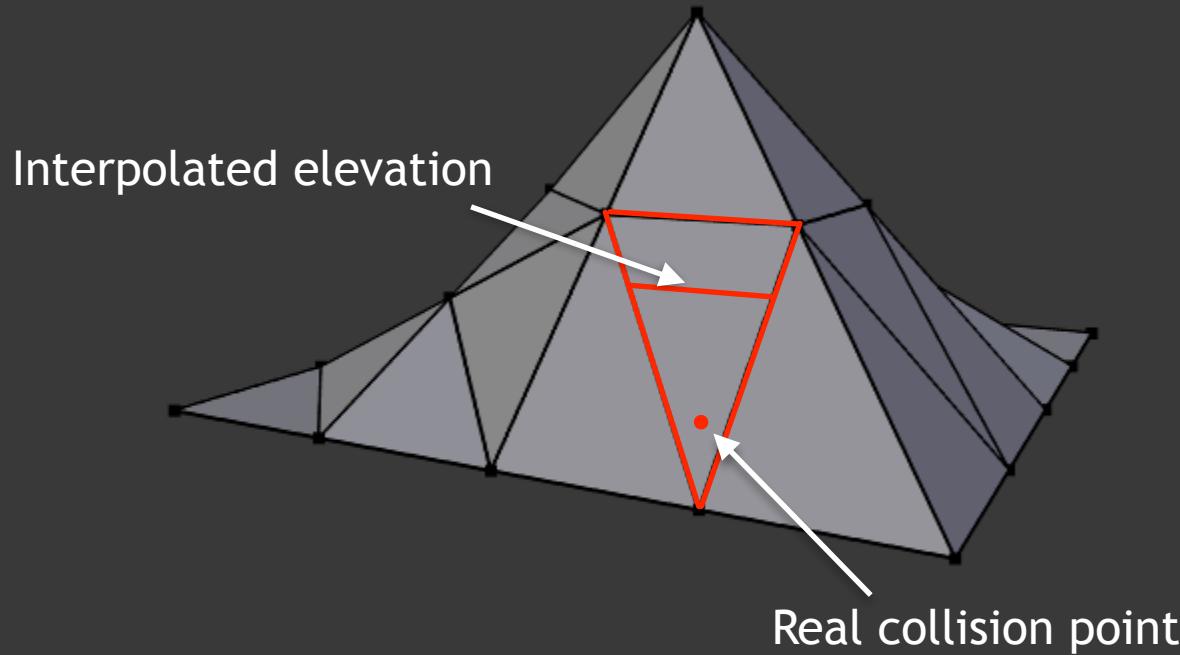
What is the elevation at this point?

# Geology-based terrain degradation

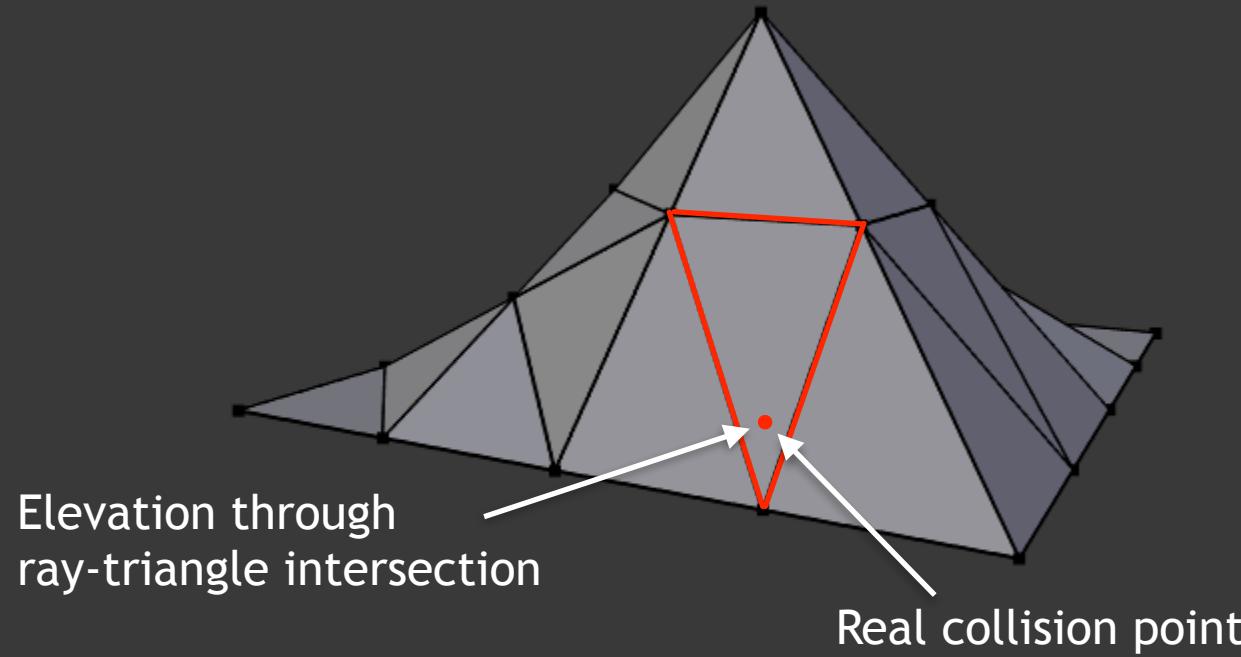
---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - How to determine the collision point?
    - Use a ray-triangle intersection algorithm (Möller-Trumbore)
  - Why not just interpolate the collision point?
    - For large heightmap resolutions this would work fine
    - For small heightmap resolutions the interpolation error would be noticeable on steep slopes

# Geology-based terrain degradation



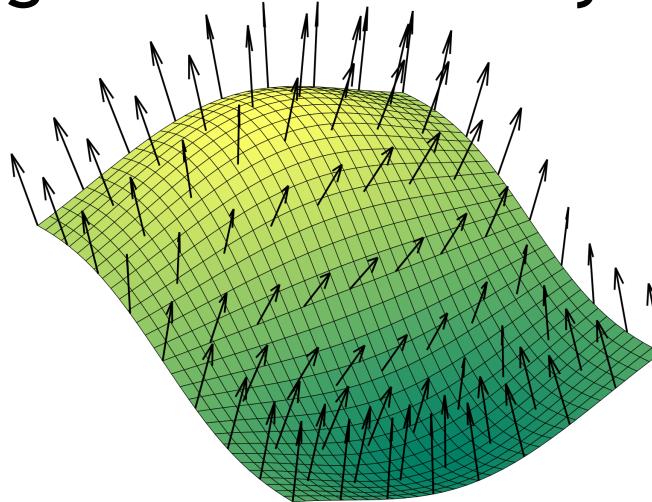
# Geology-based terrain degradation



# Geology-based terrain degradation

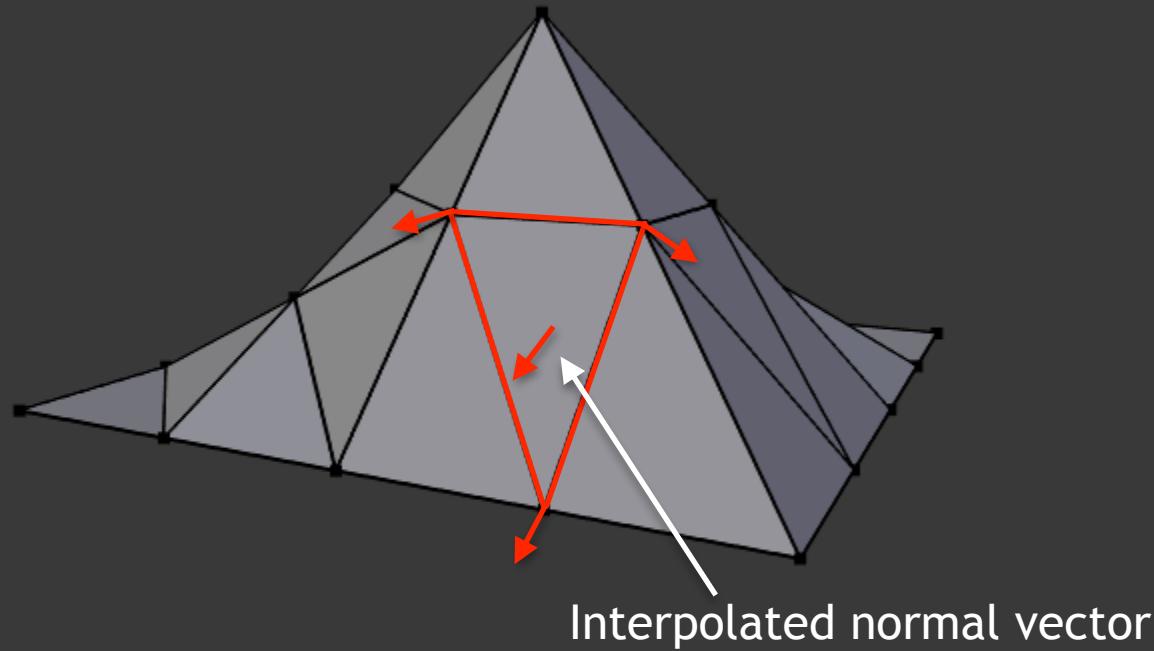
---

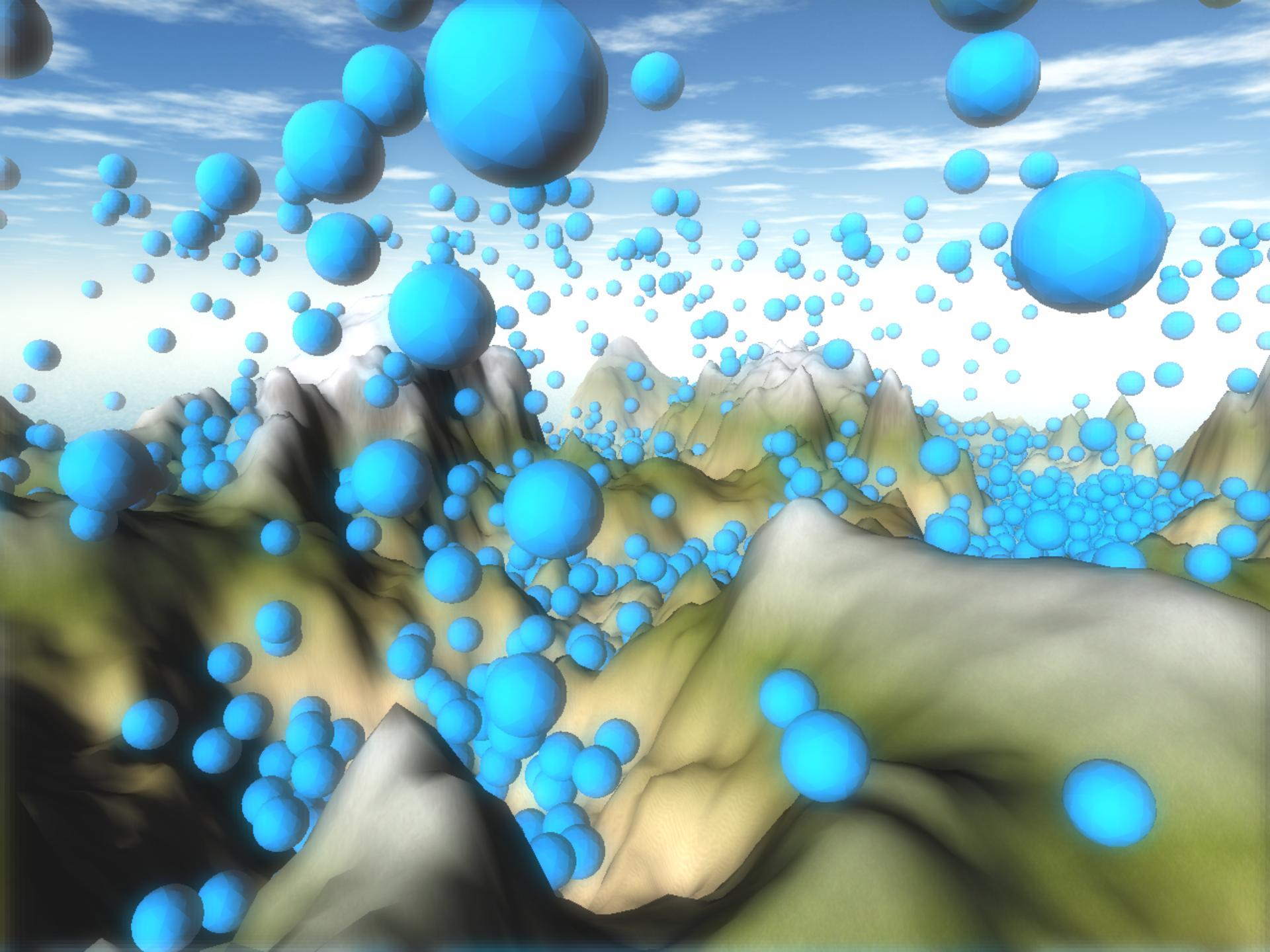
- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - How to determine the impulse direction?
    - Interpolate heightmap normals of the triangle intersected by the ray



[https://en.wikipedia.org/wiki/Normal\\_\(geometry\)](https://en.wikipedia.org/wiki/Normal_(geometry))

# Geology-based terrain degradation

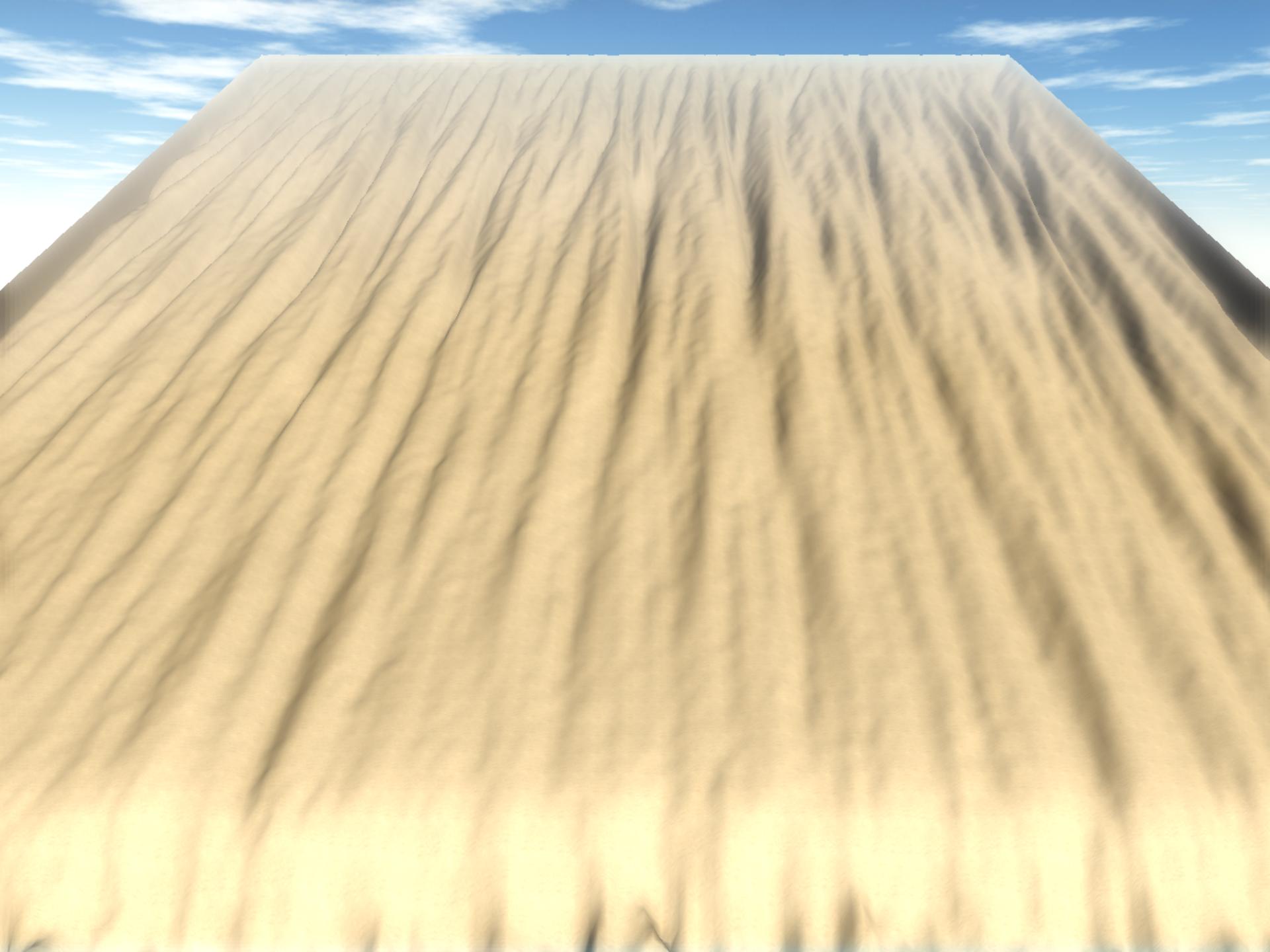


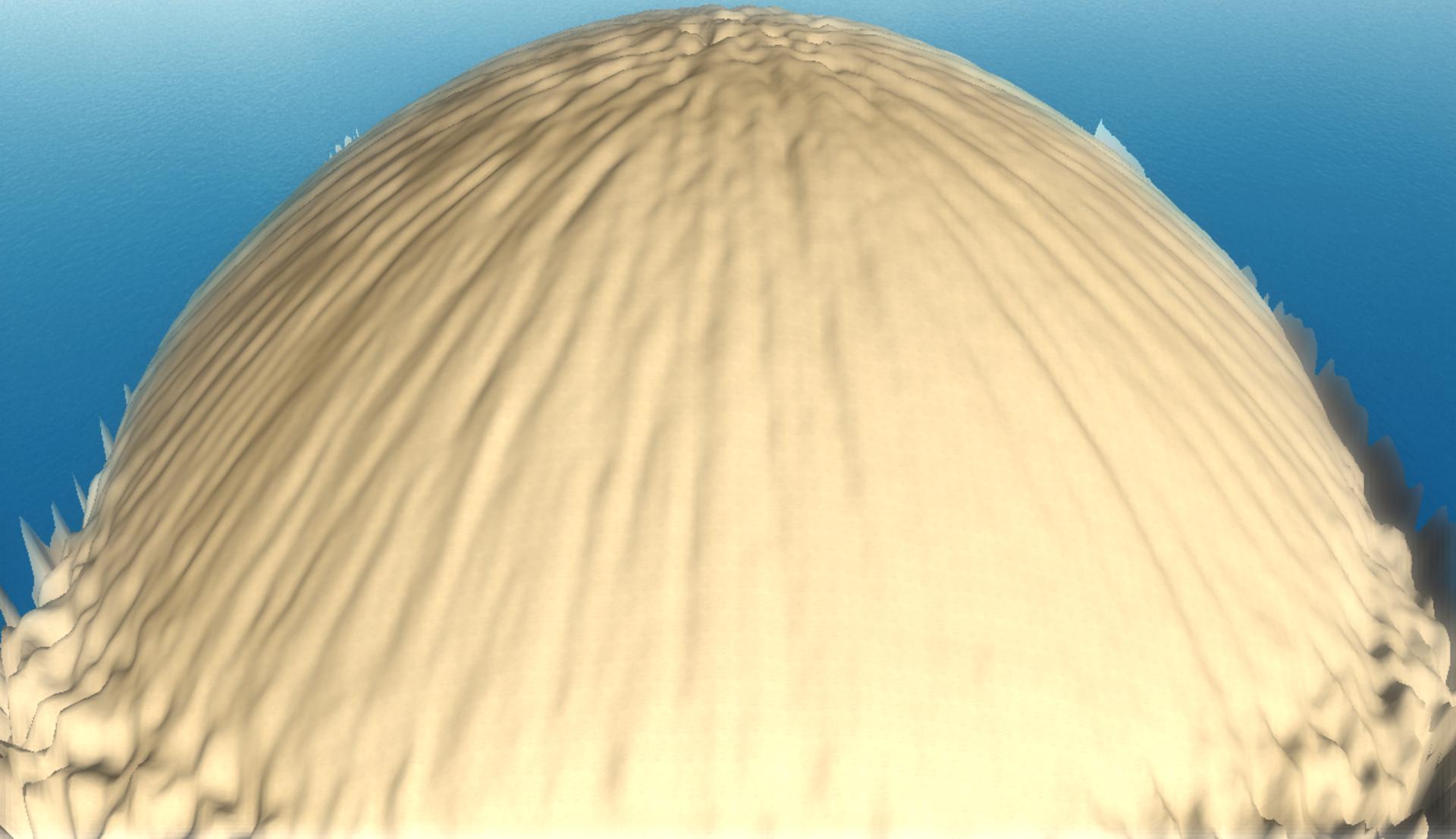


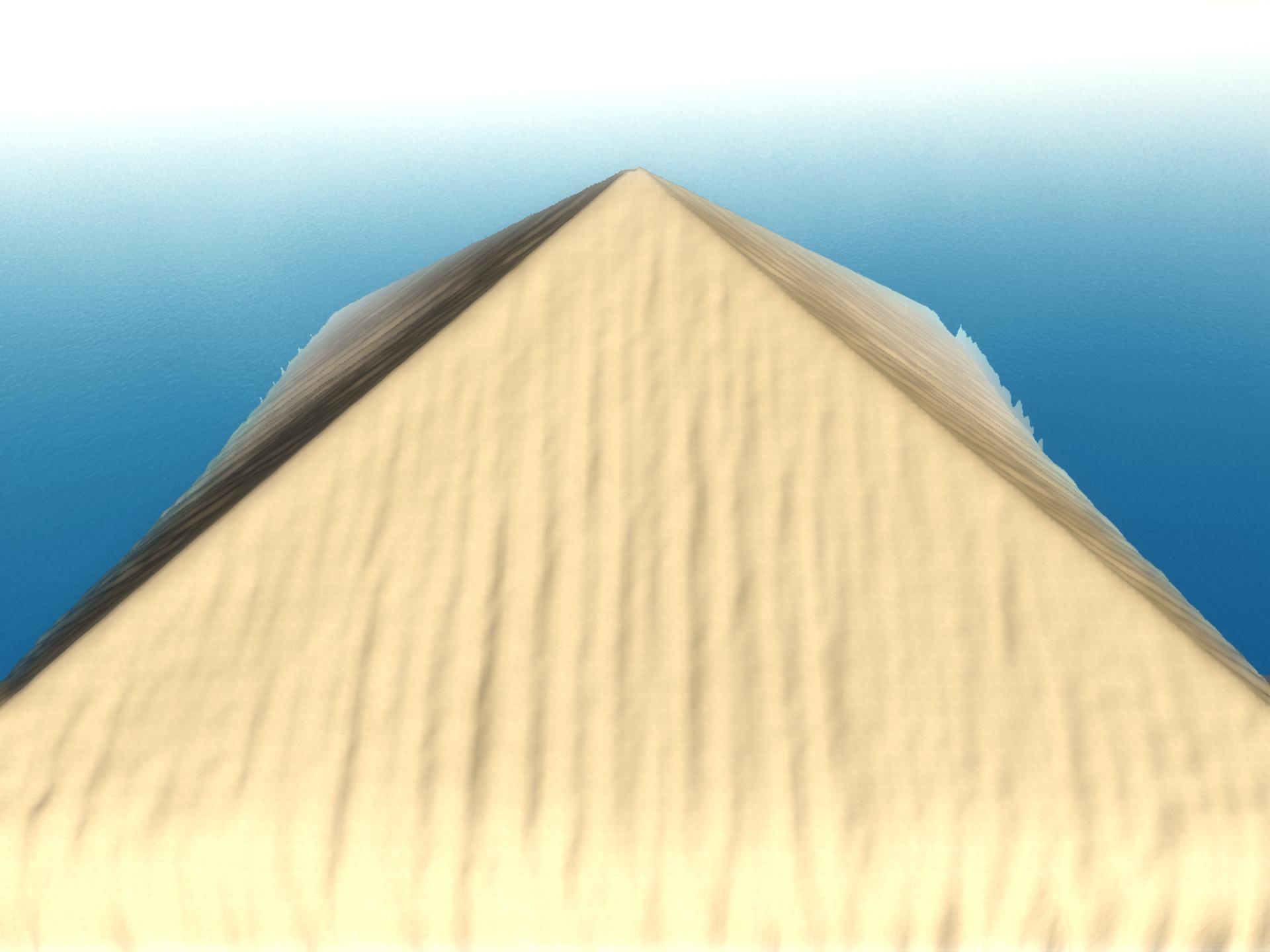
# Geology-based terrain degradation

---

- Smoothed particle hydrodynamics-based hydraulic erosion simulation
  - Particles can absorb sediment
  - Particles have a sediment capacity
  - Particles have a predefined lifetime
  - Particles deposit sediment once they evaporate







# Visualization techniques

---

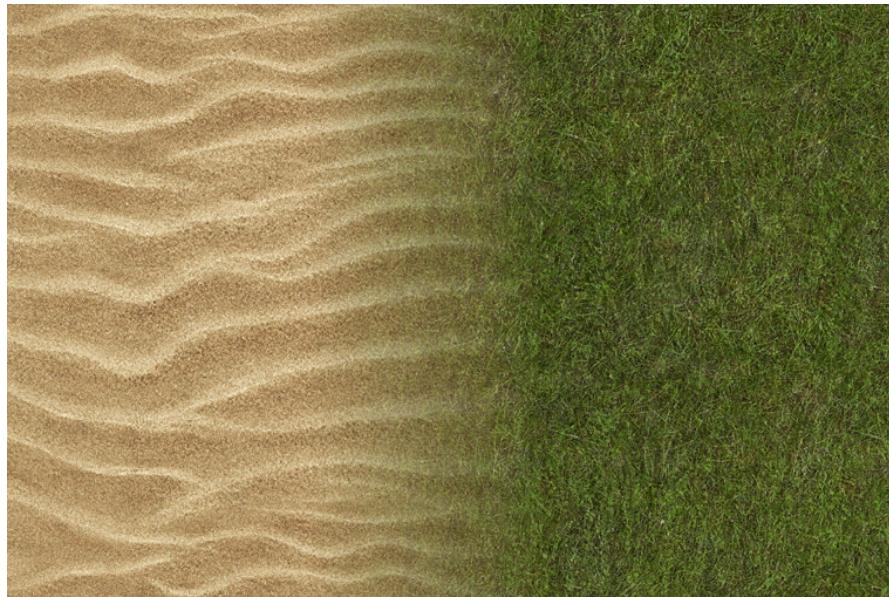
- Texture splatting
- Bloom shader

# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - Is a technique to smoothly transition from one texture to another



# Visualization techniques

---

- Texture splatting
  - Bloom shader
    - How does it work?

# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



100% Sand / 0% Grass

- Linear interpolation:

$$c(t) = a + (b - a) * t; r, g, b \in [0, 255]$$

$$a = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}); b = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}}); t = 0.0$$

$$c(0.0) = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}})$$



# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



95% Sand / 5% Grass

- Linear interpolation:

$$c(t) = a + (b - a) * t; r, g, b \in [0, 255]$$

$$a = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}); b = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}}); t = 0.05$$

$$c(0.05) = 0.95 * (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}) + 0.05 * (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}})$$



# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



90% Sand / 10% Grass

- Linear interpolation:

$$c(t) = a + (b - a) * t; r, g, b \in [0, 255]$$

$$a = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}); b = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}}); t = 0.1$$

$$c(0.1) = 0.9 * (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}) + 0.1 * (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}})$$



# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



50% Sand / 50% Grass

- Linear interpolation:

$$c(t) = a + (b - a) * t; r, g, b \in [0, 255]$$

$$a = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}); b = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}}); t = 0.5$$

$$c(0.5) = 0.5 * (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}) + 0.5 * (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}})$$



# Visualization techniques

---

## → Texture splatting

- Bloom shader
  - How does it work?



0% Sand / 100% Grass

- Linear interpolation:

$$c(t) = a + (b - a) * t; r, g, b \in [0, 255]$$

$$a = (r_{\text{sand}}, g_{\text{sand}}, b_{\text{sand}}); b = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}}); t = 1.0$$

$$c(1.0) = (r_{\text{grass}}, g_{\text{grass}}, b_{\text{grass}})$$



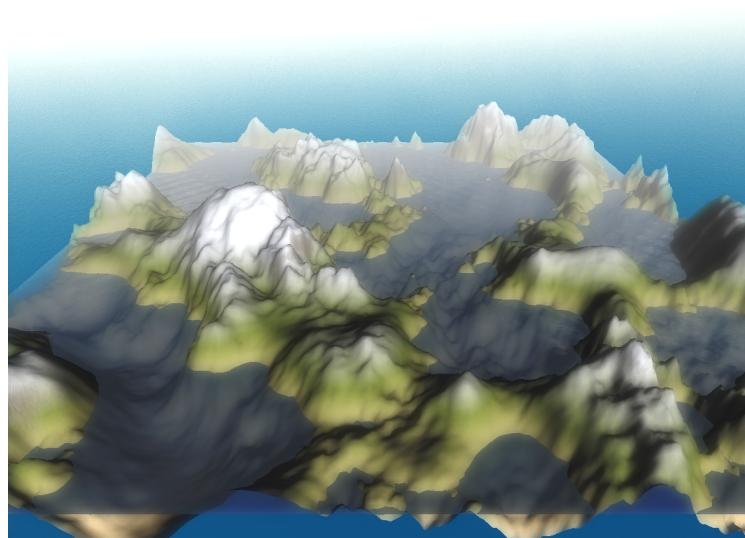
# Visualization techniques

---

## → Texture splatting

- Bloom shader

- What rules to use when generically texturing a mountain?
  - Height thresholds



# Visualization techniques

---

## → Texture splatting

- Bloom shader

- What rules to use when generically texturing a mountain?
  - Slope thresholds



# Visualization techniques

---

- Texture splatting
  - Bloom shader
    - Texture splatting requires seamless textures



# Visualization techniques

---

- Texture splatting
  - Bloom shader
    - Texture splatting requires seamless textures



# Visualization techniques

---

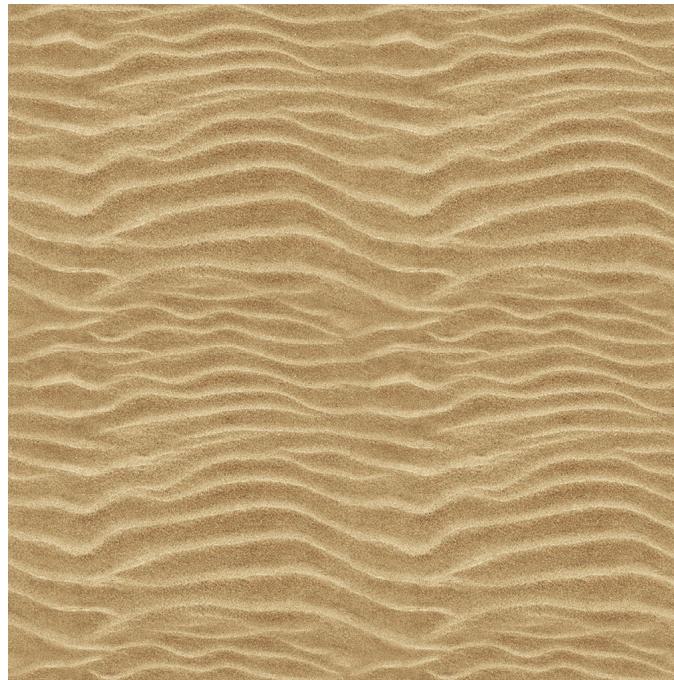
- Texture splatting
  - Bloom shader
    - Texture splatting requires seamless textures



# Visualization techniques

---

- Texture splatting
  - Bloom shader
    - Texture splatting requires seamless textures



# Visualization techniques

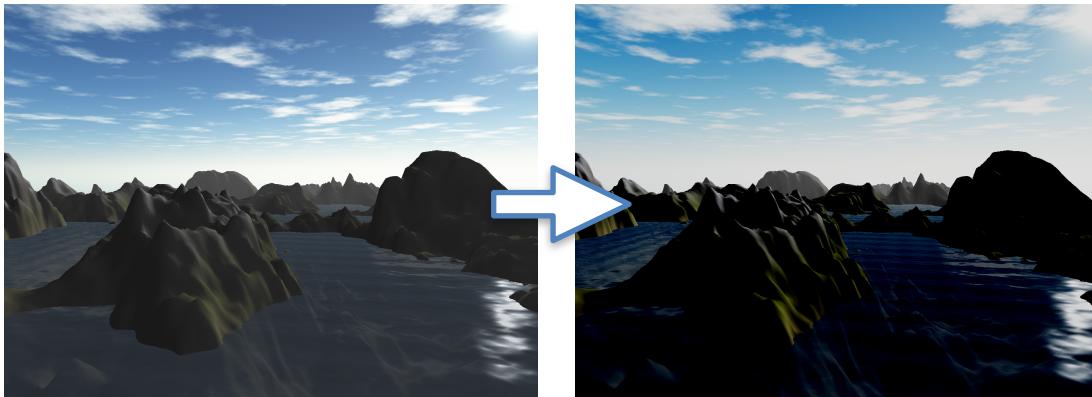
---

- Texture splatting
- Bloom shader
  - Reproduces image imperfections caused by real-world cameras
  - Makes brightly lit spots appear to be glowing
  - Is a combination of two filters:
    - Bright-pass filter
    - Gauss-blur filter

# Visualization techniques

---

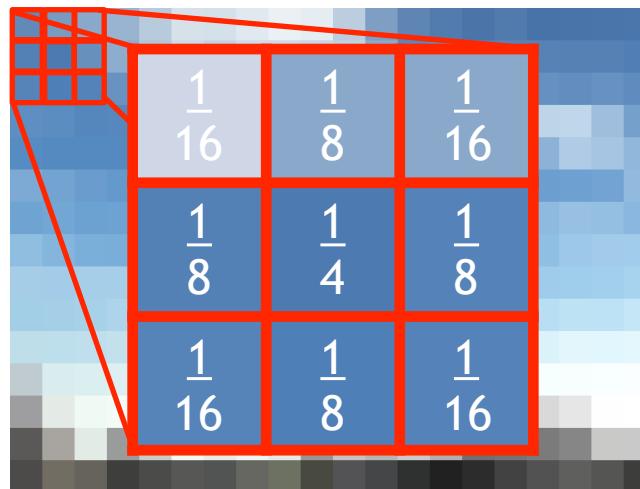
- Texture splatting
- Bloom shader
  - Bright-pass filter
    - Bright colors will pass in an unimpeded manner
    - Dark colors will be filtered



# Visualization techniques

---

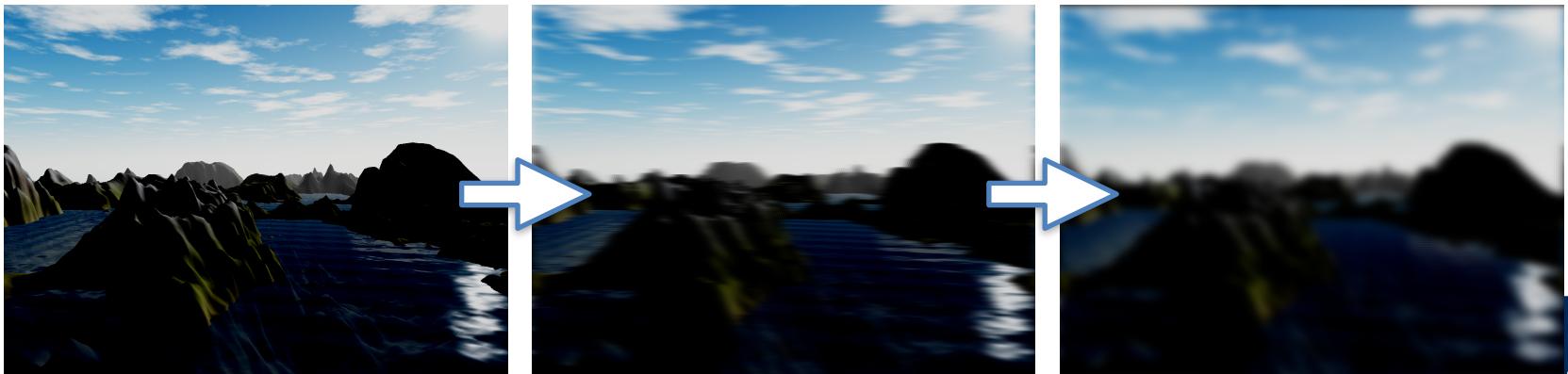
- Texture splatting
- Bloom shader
  - Gauss-blur filter
    - How does gauss-blurring work?
    - Simple example (3x3 gaussian kernel):



# Visualization techniques

---

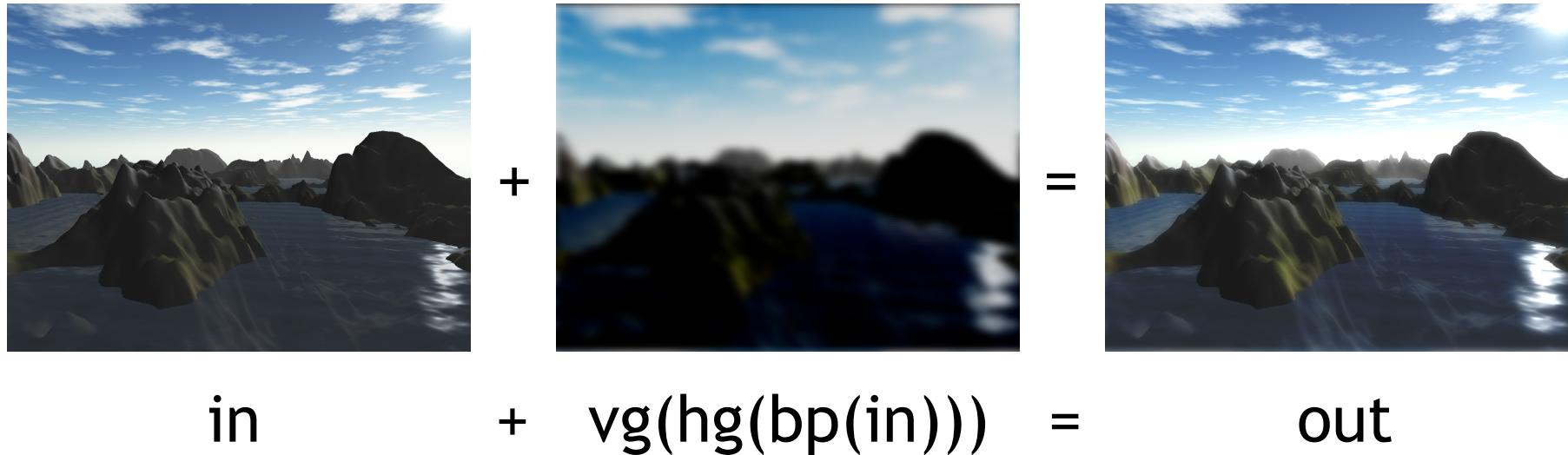
- Texture splatting
- Bloom shader
  - Gauss-blur filter
    - Performance optimization:
      - Horizontal Gauss-blur filter
      - Vertical Gauss-blur filter

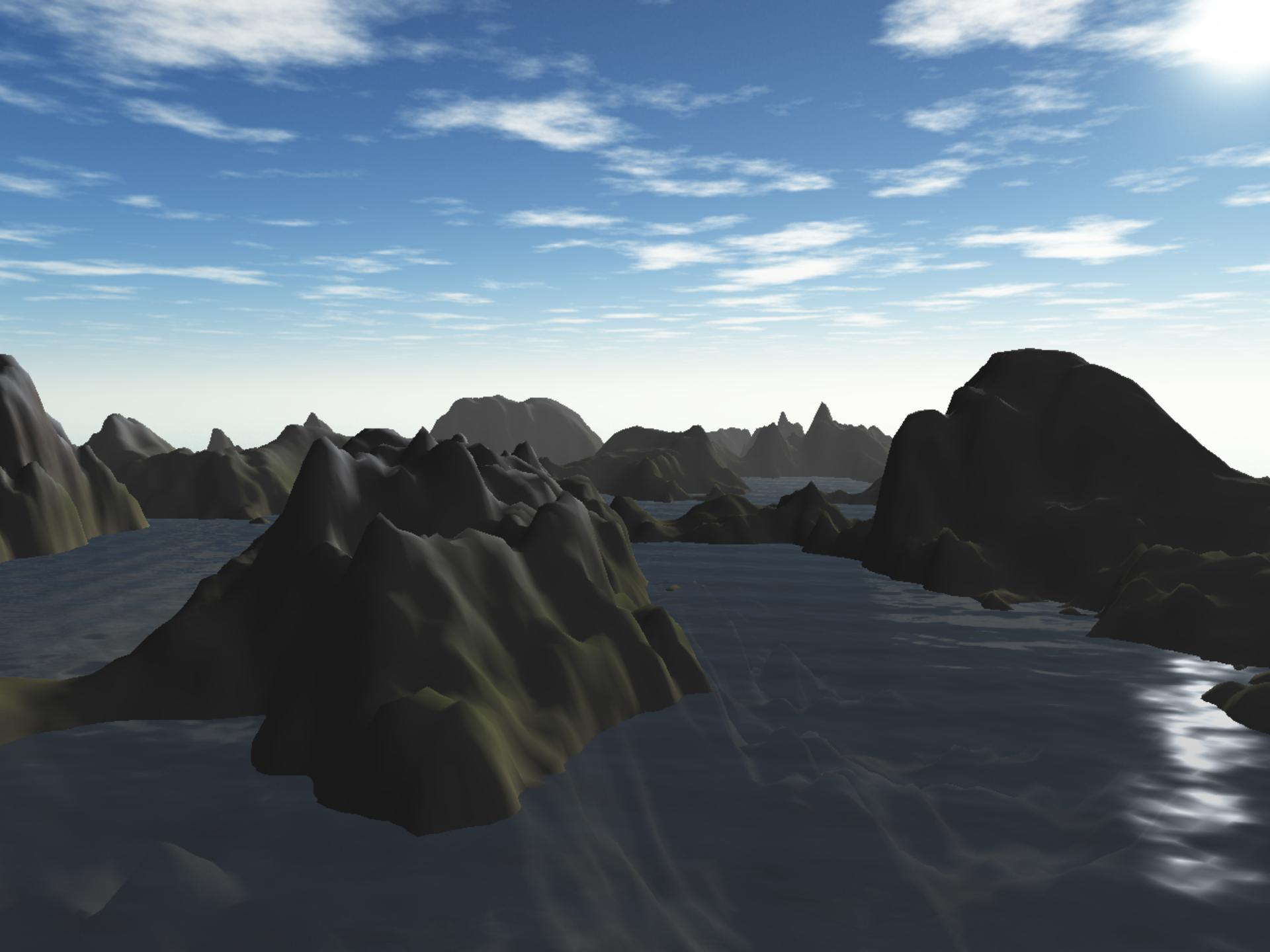


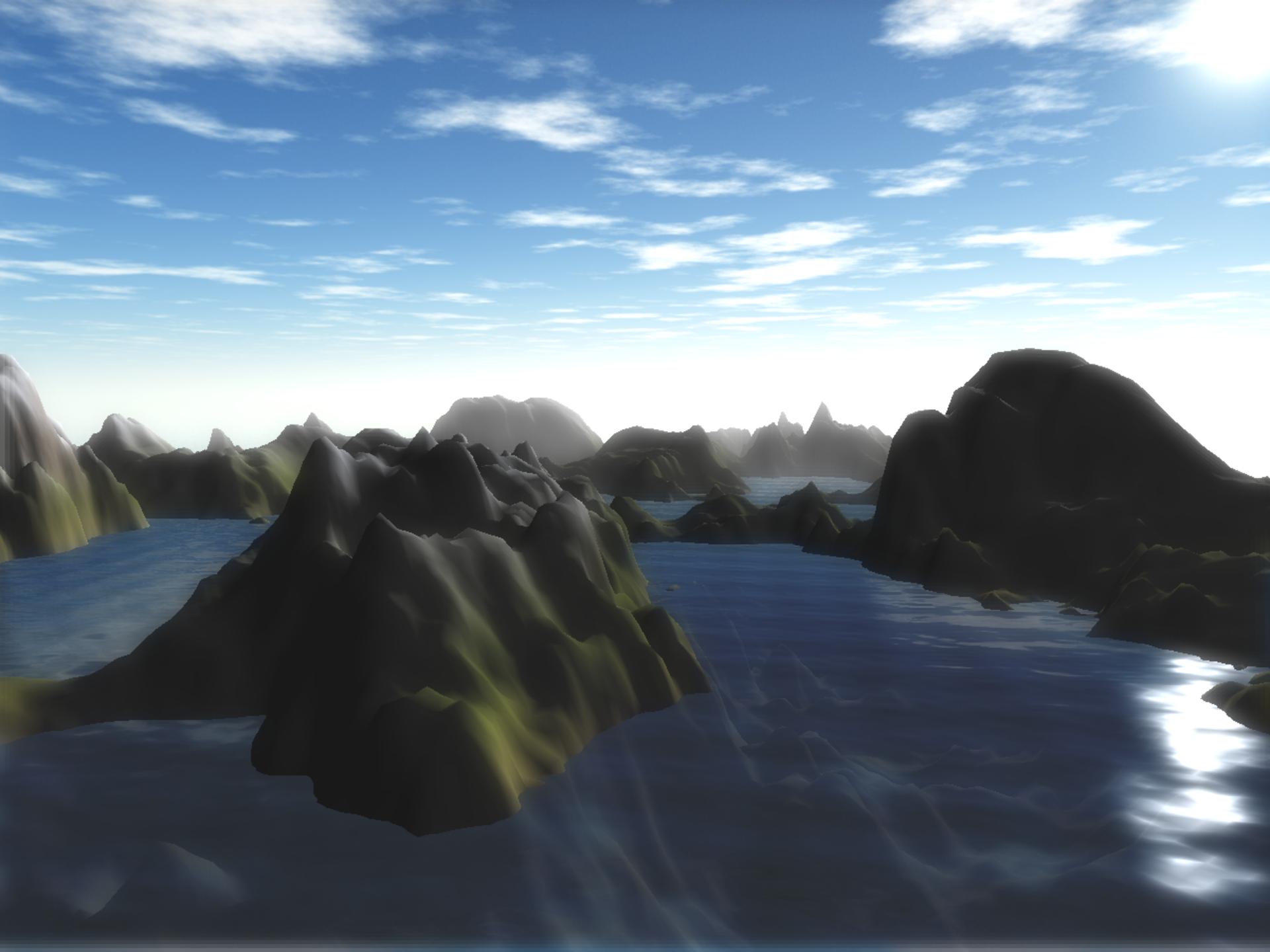
# Visualization techniques

---

- Texture splatting
- Bloom shader

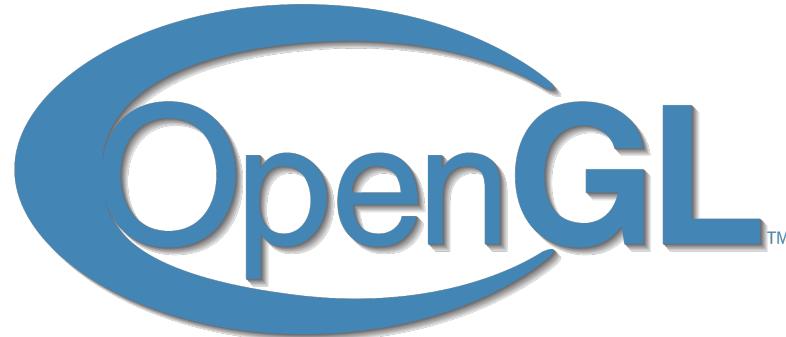




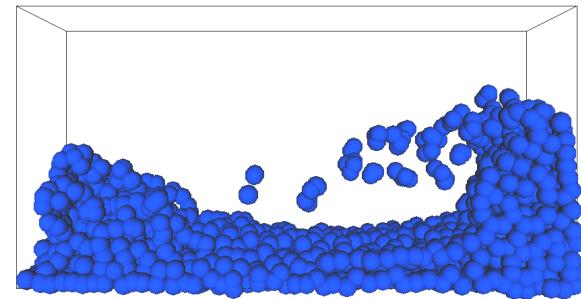


# What technologies have been used?

---



C++



stblib

 FFmpeg

Box2D 

# References

---

- Paul Boxley, Terrain generation with the diamond square algorithm.  
<http://www.paulboxley.com/blog/2011/03/terrain-generation-mark-one>
- Chris Wellons, A GPU Approach to Voronoi Diagrams.  
<http://nullprogram.com/blog/2014/06/01/>
- Saeed Mahani, SPH 3D Real-Time Fluid Simulation.  
<https://github.com/saeedmahani/SPH-3D-Fluid-Simulation>