Procedural Terrain Generation

Generating a beautiful world at runtime

Brothers of Destruction
2005041 Md Imtiaz Kabir
2005070 Md Irtiaz Kabir

July 31, 2025

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

Index

How does an endless scroller become endless?

Why Learn This?

Why Learn This?

3D Voxel terrain in Minecraft

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?
- Create a single terrain by hand and store it somewhere, repeat when it ends.

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?
- Create a single terrain by hand and store it somewhere, repeat when it ends.
- Have a set of terrains and choose one at random at start.

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?
- Create a single terrain by hand and store it somewhere, repeat when it ends.
- Have a set of terrains and choose one at random at start.
- How do we ensure nothing repeats without having infinite memory and infinite time?

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?
- Create a single terrain by hand and store it somewhere, repeat when it ends.
- Have a set of terrains and choose one at random at start.
- How do we ensure nothing repeats without having infinite memory and infinite time?
- Analytic terrains ($y = sin(x)$) are boring and not realistic

Why Learn This?

3D Voxel terrain in Minecraft

2D terrain of Terraria

- How do your favourite games create these **beautiful** endless terrains?
- Create a single terrain by hand and store it somewhere, repeat when it ends.
- Have a set of terrains and choose one at random at start.
- How do we ensure nothing repeats without having infinite memory and infinite time?
- Analytic terrains ($y = sin(x)$) are boring and not realistic
- We need to somehow accommodate the randomness of nature.

Generation Algorithms

Cellullar automata

Steps of algorithm

Step 1: Start with random grid with 1 as wall and 0 as passage.

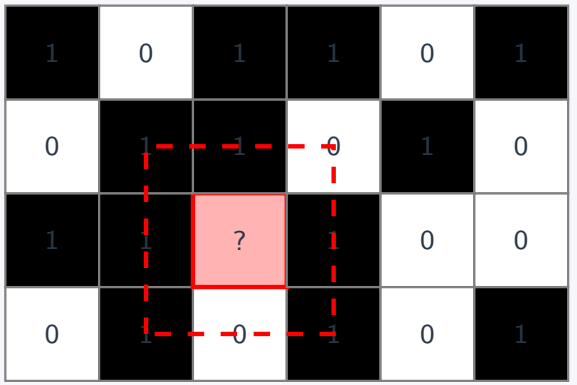
0	0	0	0	0	0
1	0	0	0	0	1
1	1	1	0	1	0
0	1	0	1	1	1
1	1	0	1	0	1
1	1	0	0	0	1

Random Initial Grid

Cellullar automata

Steps of algorithm

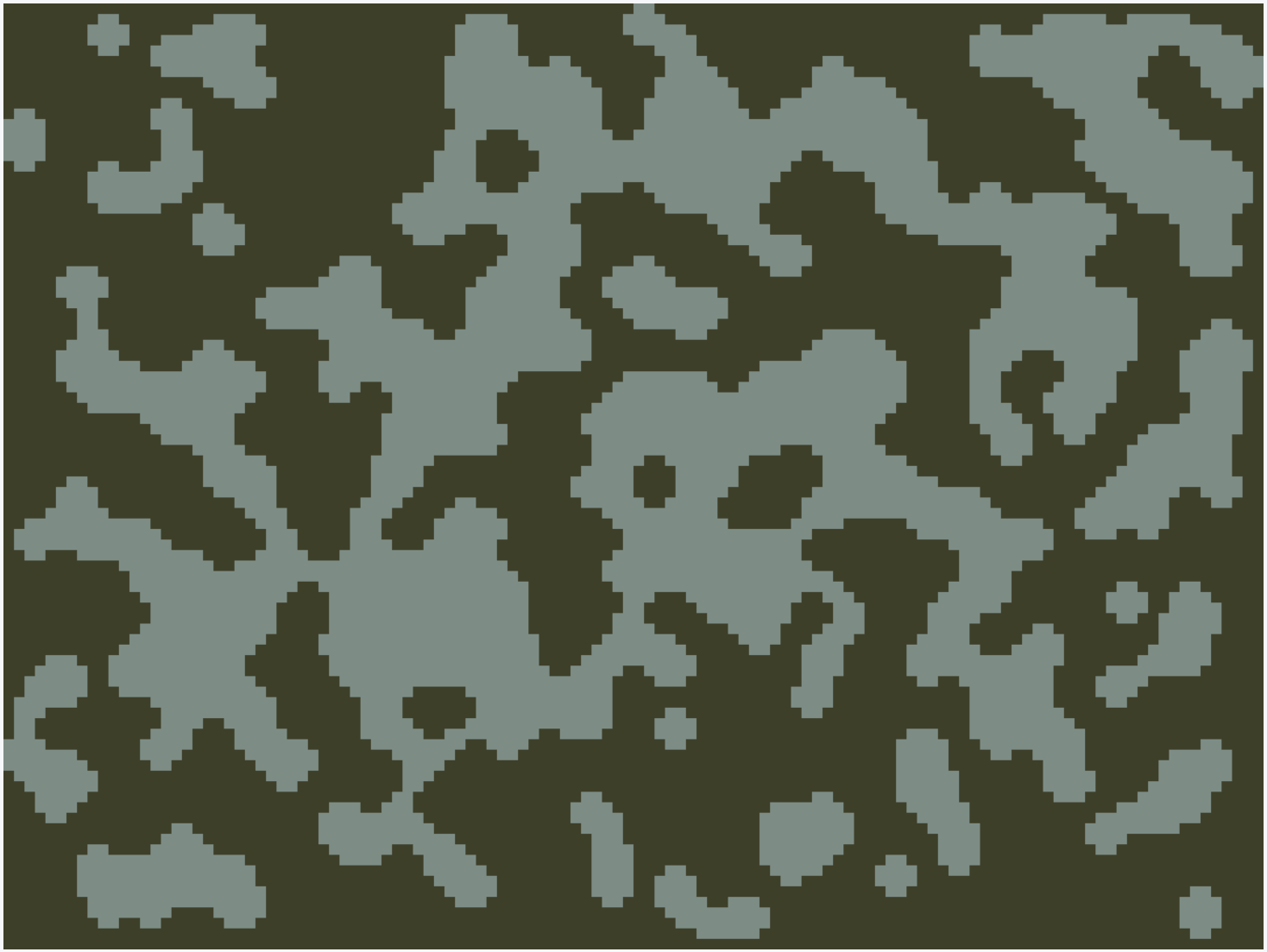
Step 2: Each next iteration, change a cells value according to the following rule.
A cell becomes a wall, if in it's Moore neighborhood (including itself), majority of cell is wall.
Otherwise it becomes a passage.



Moore Neighborhood Rule

Cellullar automata: Demonstration

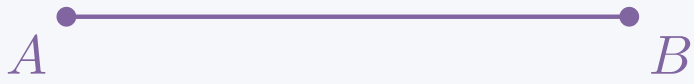
Cellullar automata: Demonstration



Midpoint Displacement Algorithm

Steps of algorithm

Step 1: Begin with a straight line segment AB.



Midpoint Displacement Algorithm

Steps of algorithm

Step 2: Compute the midpoint C of AB

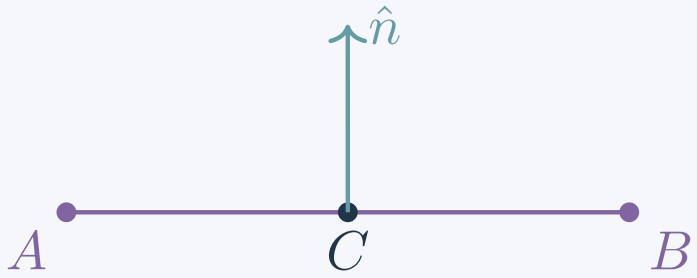
$$C = \frac{A + B}{2}$$

Midpoint Displacement Algorithm

Steps of algorithm

Step 3: Compute unit normal $\hat{n}$ of segment AB

$$\hat{n} = \frac{B - A}{|B - A|} \times \hat{k}$$

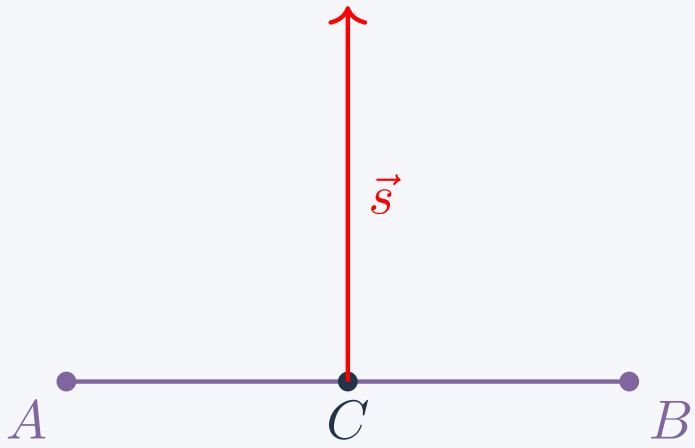


Midpoint Displacement Algorithm

Steps of algorithm

Step 4: Multiply with `random(-h, h)` to find displacement

$$\vec{s} = \hat{n} \times \texttt{random(-h, h)}$$

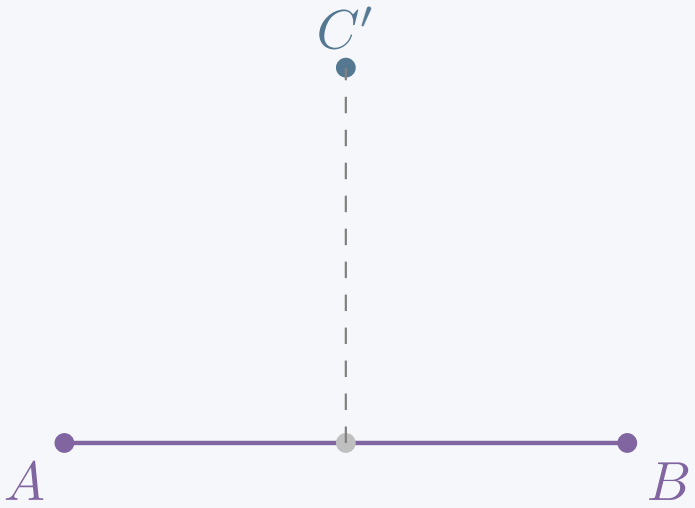


Midpoint Displacement Algorithm

Steps of algorithm

Step 5: Displace midpoint

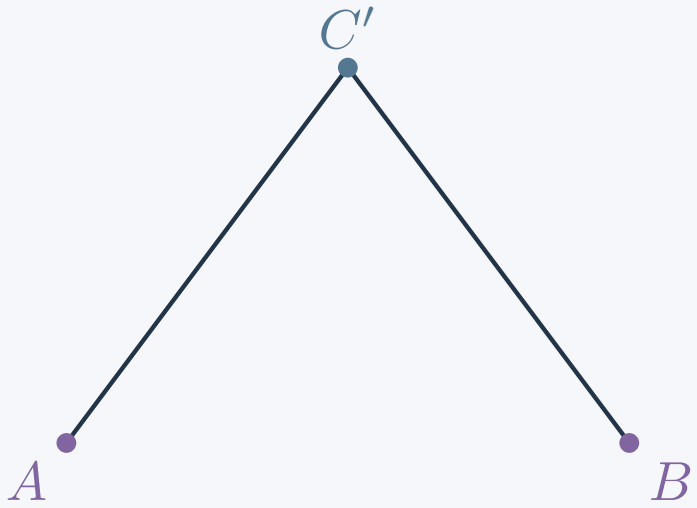
$$C' = C + \vec{s}$$



Midpoint Displacement Algorithm

Steps of algorithm

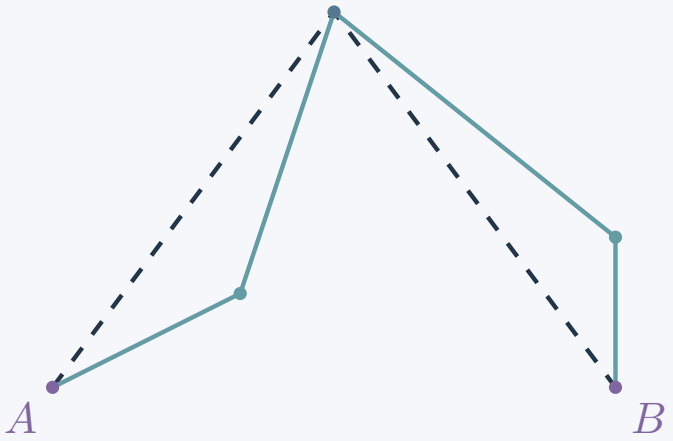
Step 6: Remove old line segment (AB) and add new segments (AC', BC')



Midpoint Displacement Algorithm

Steps of algorithm

Step 7: Repeat the process (with geometrically reduced h) n times to create a terrain with $2^n + 1$ points

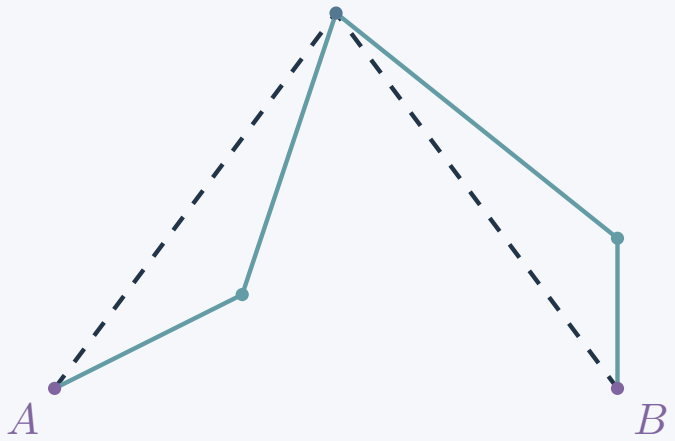


Terrain after 2 iterations

Midpoint Displacement Algorithm

Steps of algorithm

Step 7: Repeat the process (with geometrically reduced h) n times to create a terrain with $2^n + 1$ points



Terrain after 2 iterations

Making it endless

- **Two terrains** Create two overlapping terrains
- **Swap and create** When the first terrain goes out of screen swap the terrains and replace the later terrain with a new one.

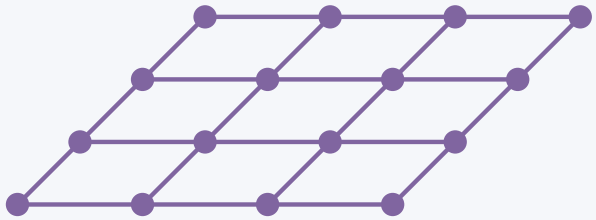
Midpoint Displacement Algorithm: Demonstration

Midpoint Displacement Algorithm: Demonstration

Terrain with Perlin Noise

Steps of algorithm

Step 1: Begin with a tessellation of a region in xy plane.
Store the tessellation as a list of lattice points.



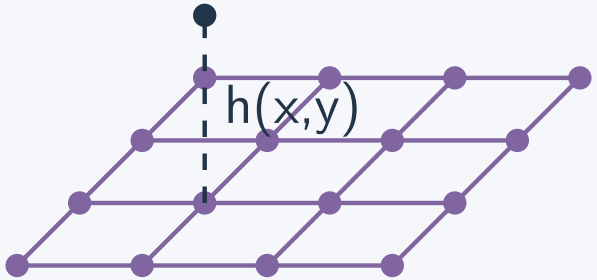
Connected Grid

Terrain with Perlin Noise

Steps of algorithm

Step 2: For each lattice points use Perlin noise to compute the terrain height.

$$h(x, y) = \texttt{height} \times \texttt{perlin(x,y)}$$

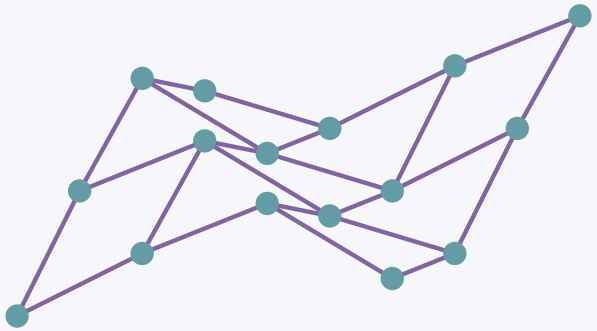


Terrain with Perlin Noise

Steps of algorithm

Step 3: Extrude each lattice point in tessellation

$$C' = C + h(x, y) \times \hat{k}$$



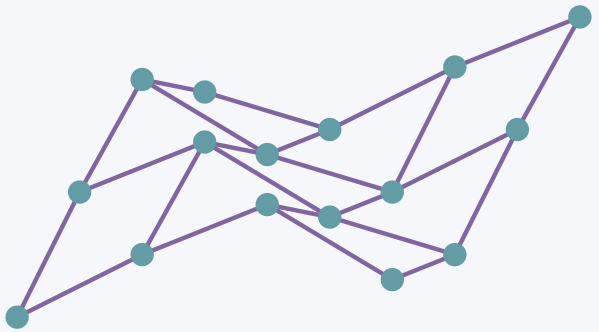
3D Terrain

Terrain with Perlin Noise

Steps of algorithm

Step 3: Extrude each lattice point in tessellation

$$C' = C + h(x, y) \times \hat{k}$$



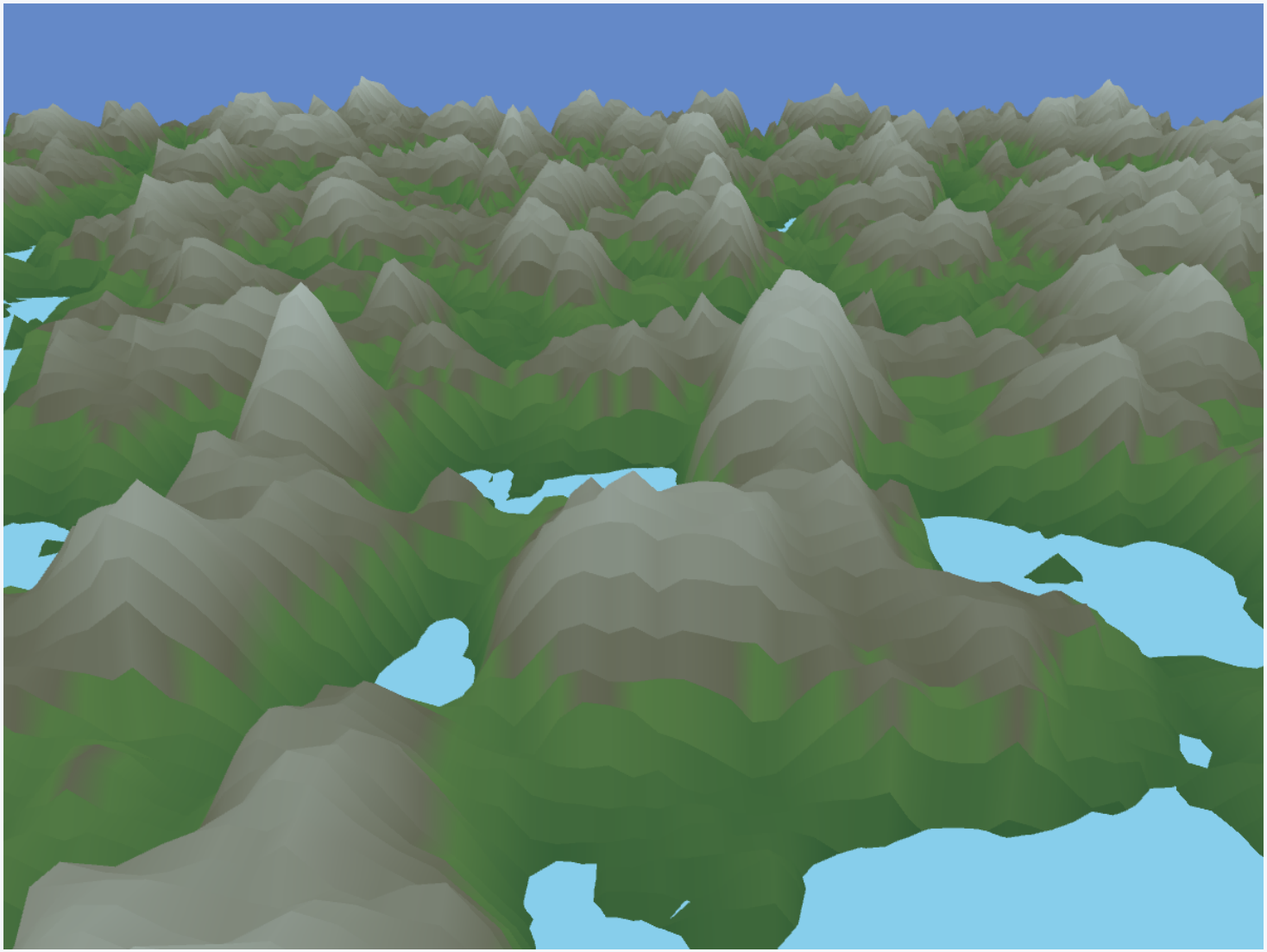
3D Terrain

N.B.:

- **Fixed world** World grows in one side and shrink in another to make it feel endless
- **Noise heightmap** Other noises like **Worley noise** or **Simplex noise** can also be used in a similar way to generate heightmap.

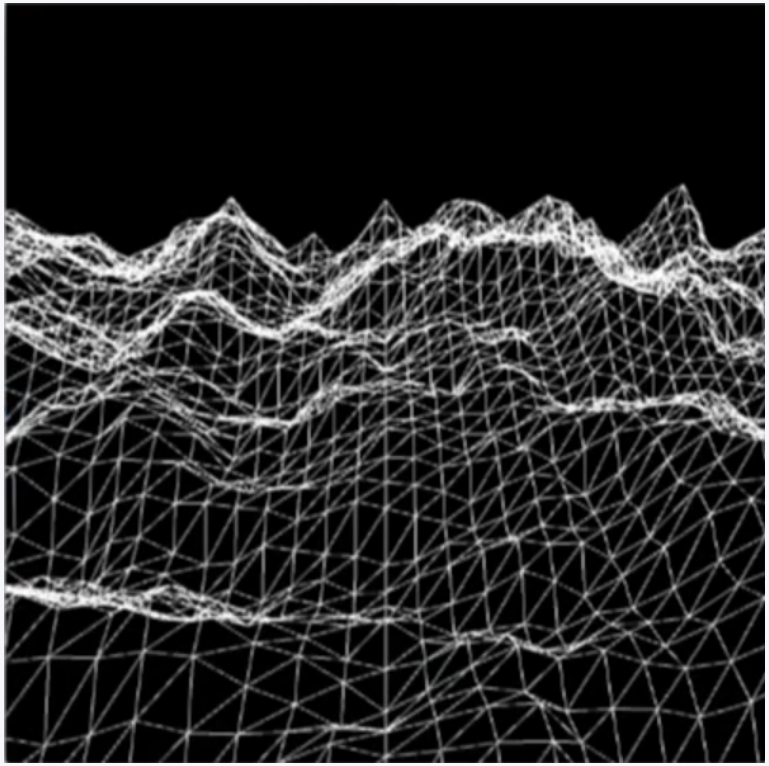
Terrain with Perlin Noise: Demonstration

Terrain with Perlin Noise: Demonstration



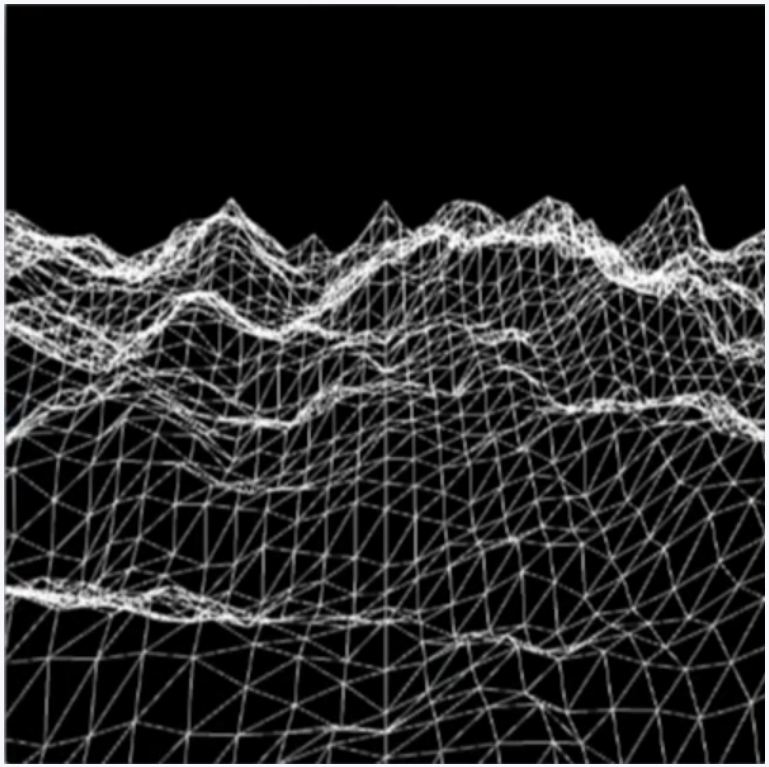
Making it realistic

Realistic terrains

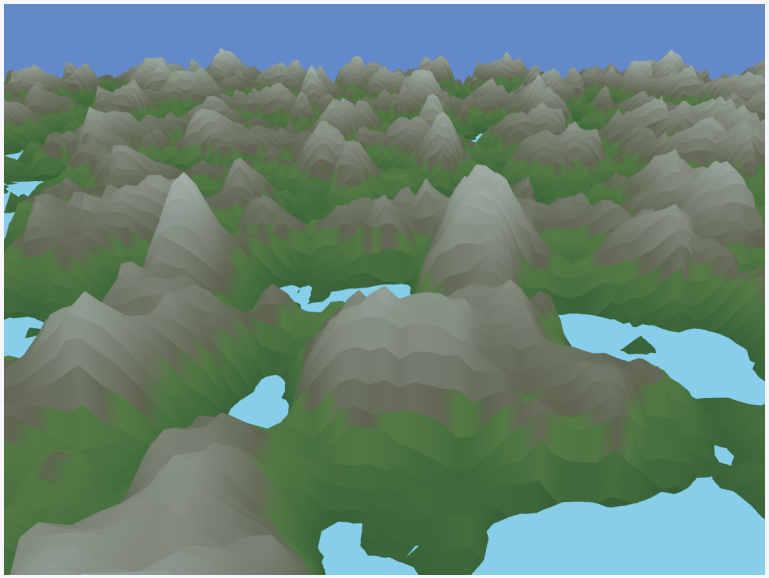


Your ugly wireframe terrain

Realistic terrains

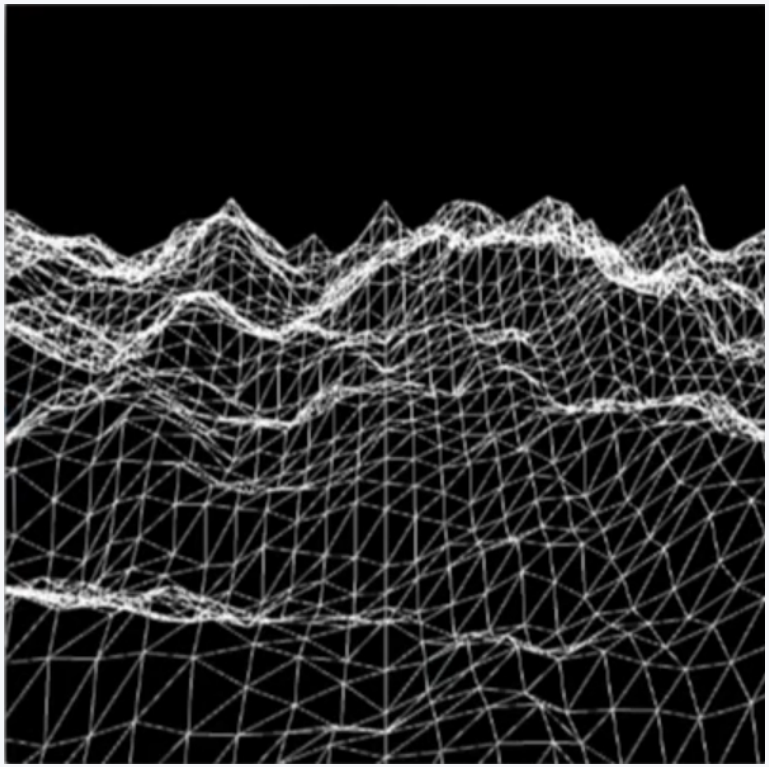


Your ugly wireframe terrain

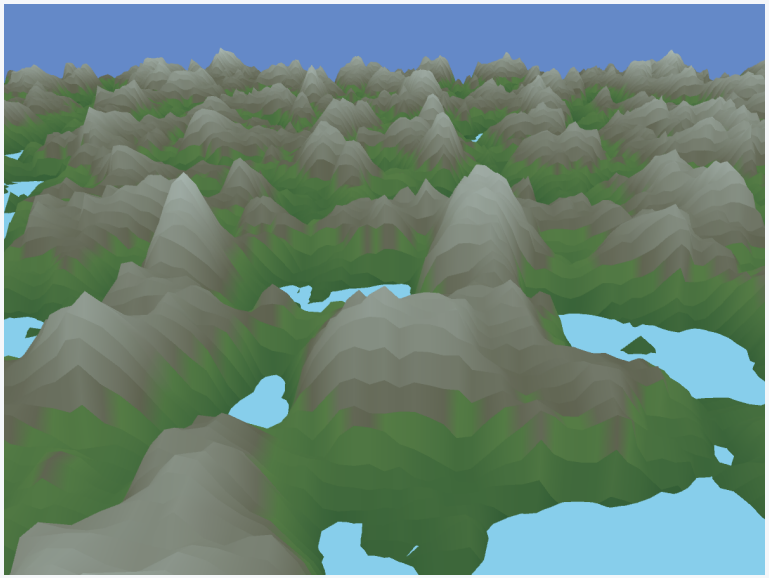


Beautifully rendered terrain

Realistic terrains



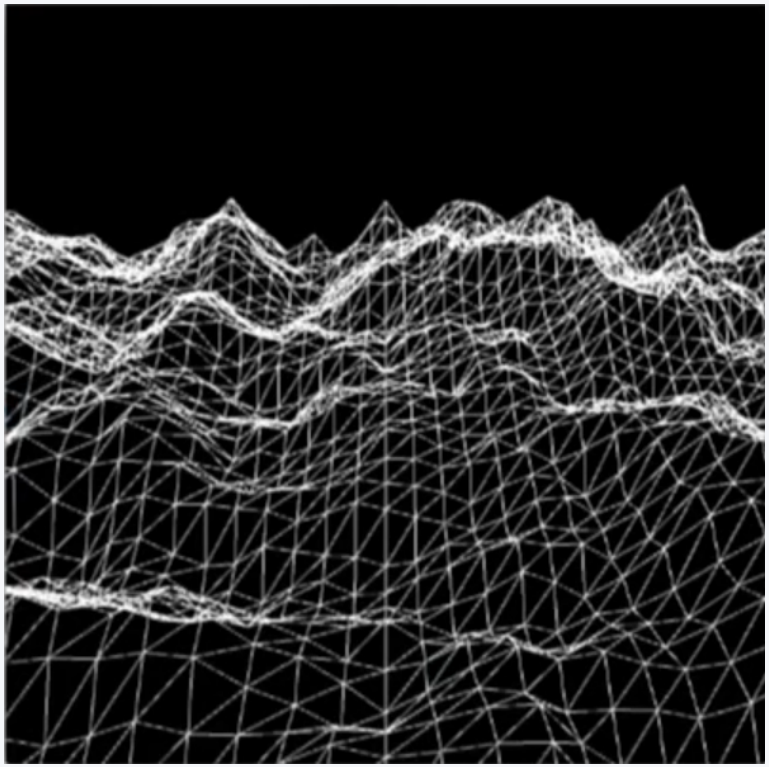
Your ugly wireframe terrain



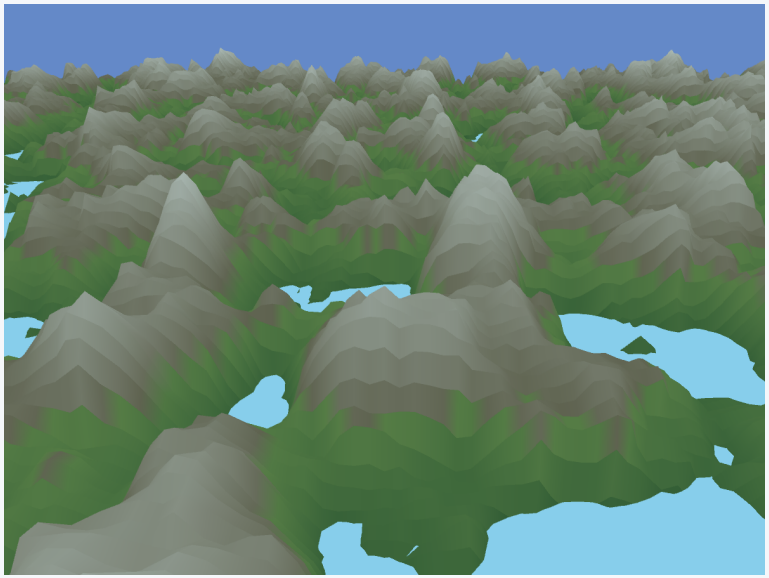
Beautifully rendered terrain

- Choose color based on height.

Realistic terrains



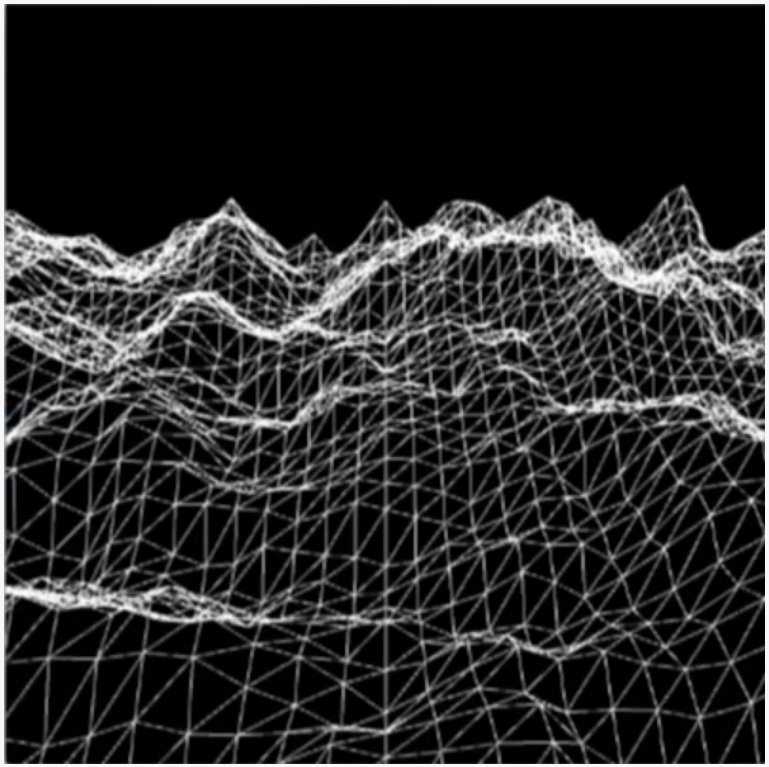
Your ugly wireframe terrain



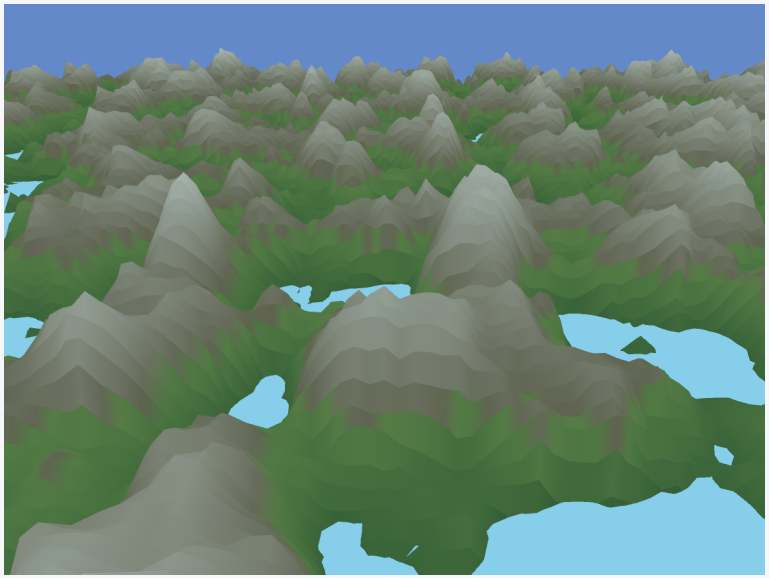
Beautifully rendered terrain

- Choose color based on height.
- $z > z_{thresh}$ can be colored with the color of mountains

Realistic terrains



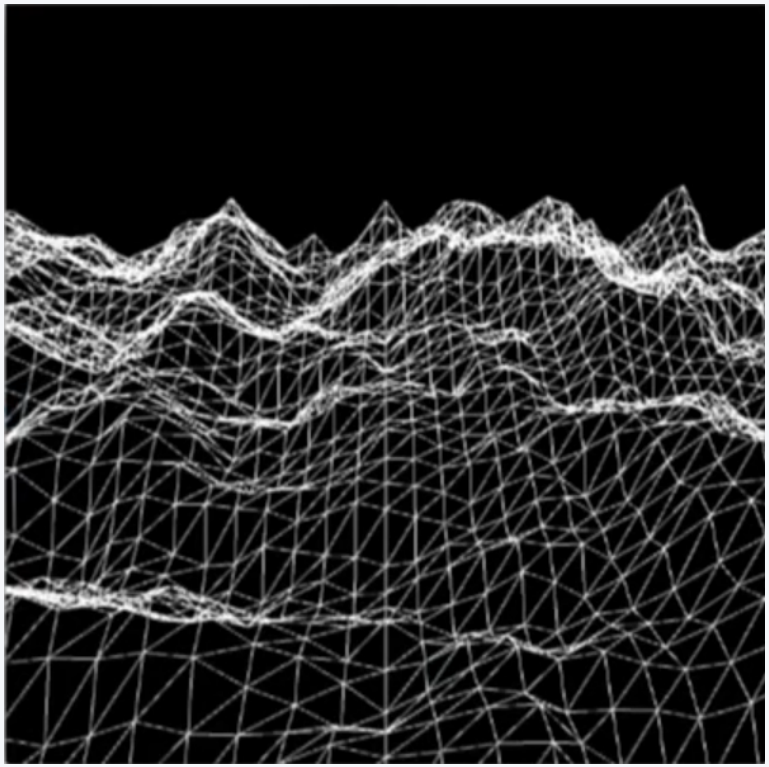
Your ugly wireframe terrain



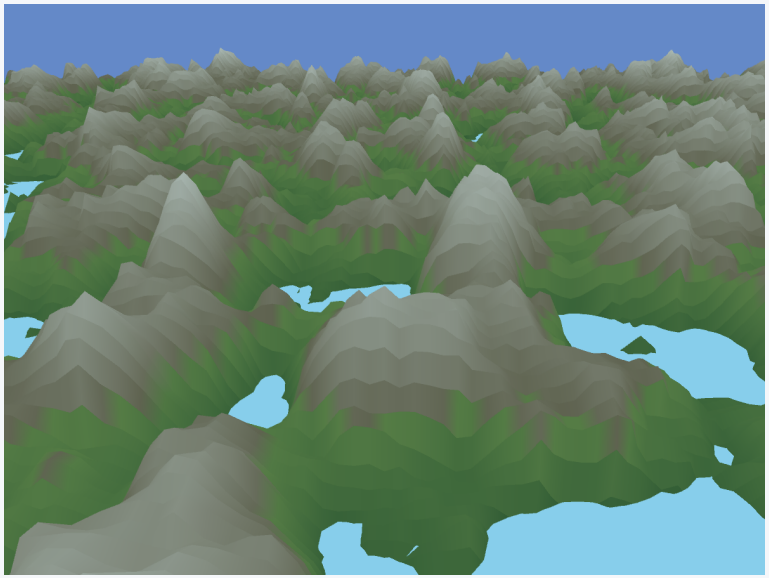
Beautifully rendered terrain

- Choose color based on height.
- $z > z_{thresh}$ can be colored with the color of mountains
- $z \leq z_{thresh}$ can be colored with the color of ground.

Realistic terrains



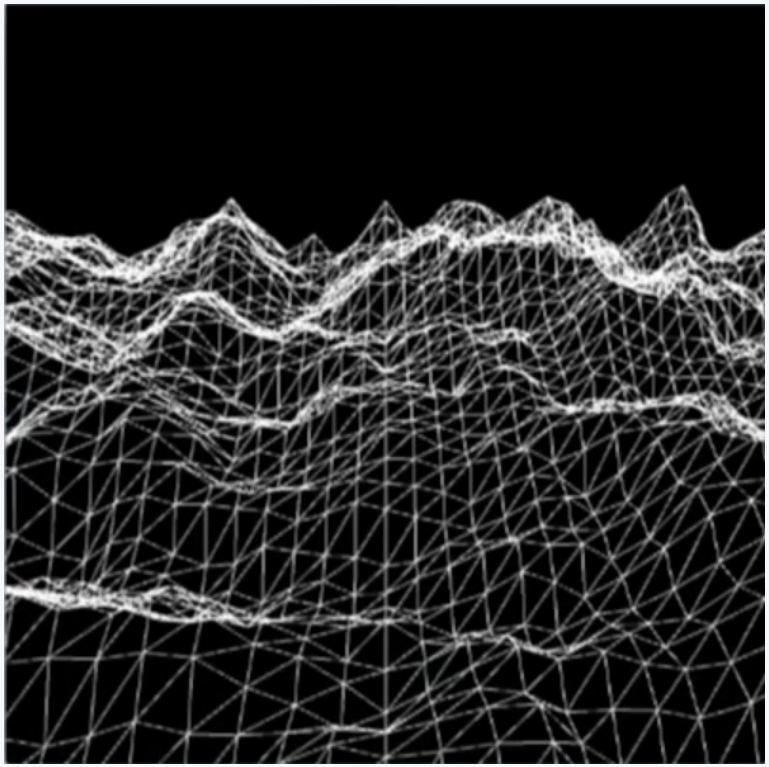
Your ugly wireframe terrain



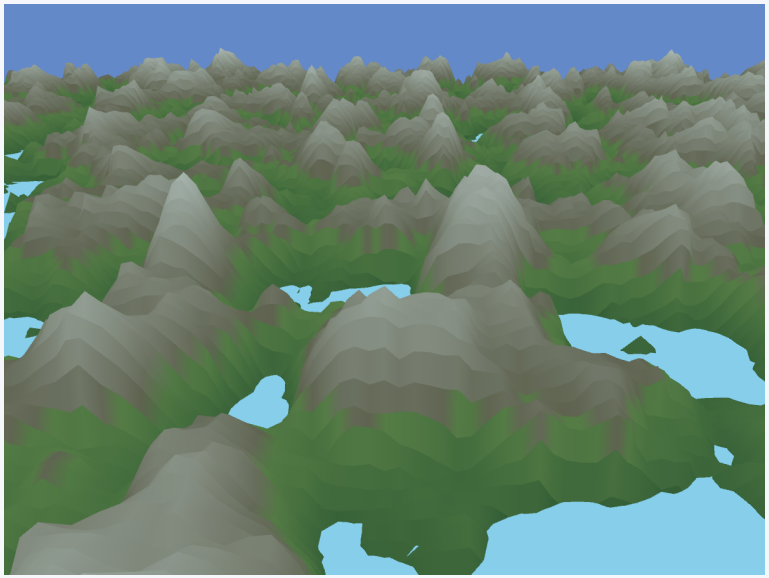
Beautifully rendered terrain

- Choose color based on height.
- $z > z_{thresh}$ can be colored with the color of mountains
- $z \leq z_{thresh}$ can be colored with the color of ground.
- Create more levels of z_{thresh} to create more interesting scenes

Realistic terrains



Your ugly wireframe terrain



Beautifully rendered terrain

- Choose color based on height.
- $z > z_{thresh}$ can be colored with the color of mountains
- $z \leq z_{thresh}$ can be colored with the color of ground.
- Create more levels of z_{thresh} to create more interesting scenes
- Apply texture for extra pizazz

References & Further Reading

- *Kabir Brothers guide to terrains for dummy dum dums* Slide code
- *Fundamentals of Terrain Generation*. Available as PDF
- *Landscape generation with midpoint displacement algorithm*. Availabe online
- Daniel Shiffman *3D Terrain Generation with Perlin Noise in Processing*. Available as video