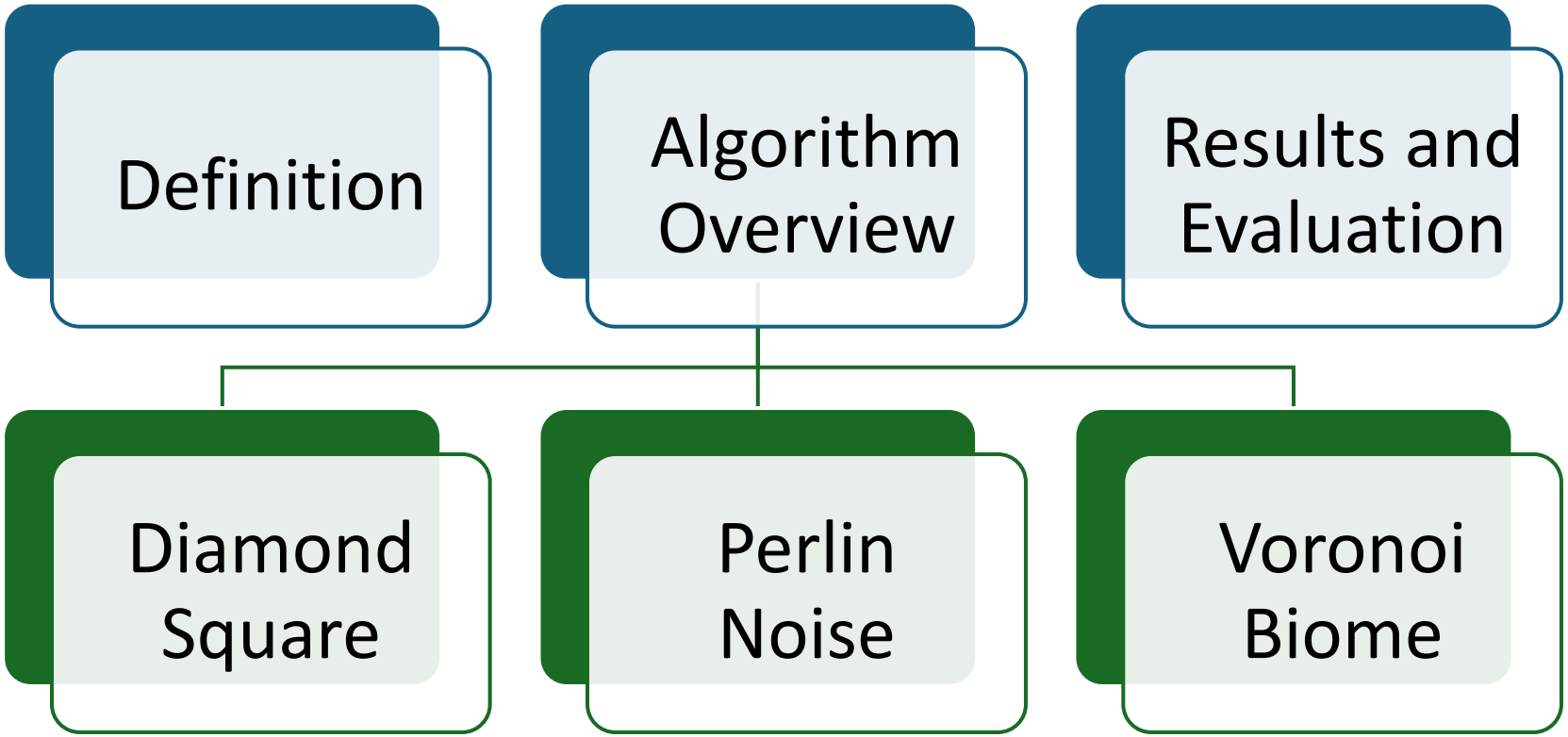
Procedural Terrain Generation

Qitong Liu and Ryan Vera

Outline


Definition
Algorithm Overview
Results and Evaluation
Diamond Square
Perlin Noise
Voronoi Biome

What is Procedural Terrain Generation

- Method used to create landscapes algorithmically rather than manually
  - Uses mathematical models, algorithms, and randomness to simulate realistic terrains

Diamond Square

Diamond Square Overview

Introduced by Fournier, Fussell, and Carpenter in 1982

Generates fractal-based heightmaps using midpoint displacement

- Ideal for Natural-looking terrain like mountains and valleys

Two alternating steps

- Diamond – fills center of square
- Square – fills midpoints of edges

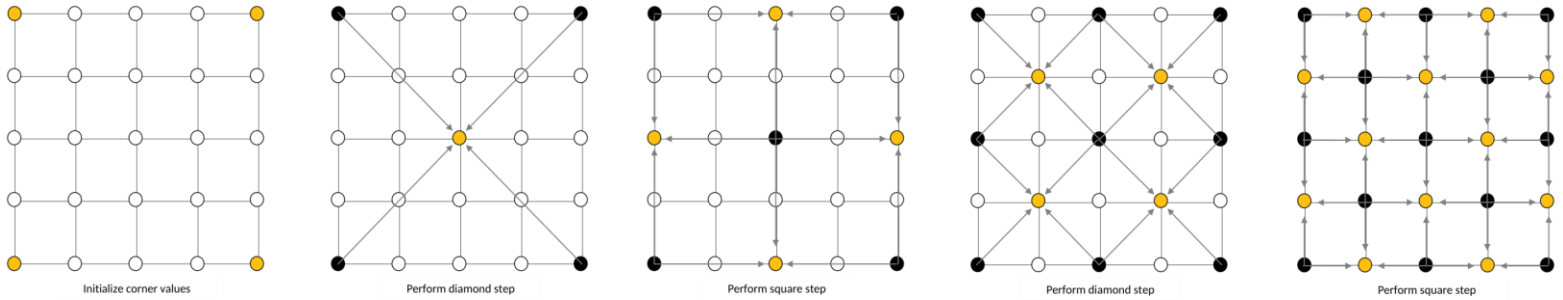
Diamond Square Pseudocode

Grid Setup

- Create a grid size of 2^n + 1
- Set corner values to initial heights

Repeat until Step size is 1

- Diamond – set center of each square to average of corners + random offset
- Square – set height of edge midpoints to average of neighbors + offset
- Reduce offset scale by multiplying by some factor



Diamond Square Variants

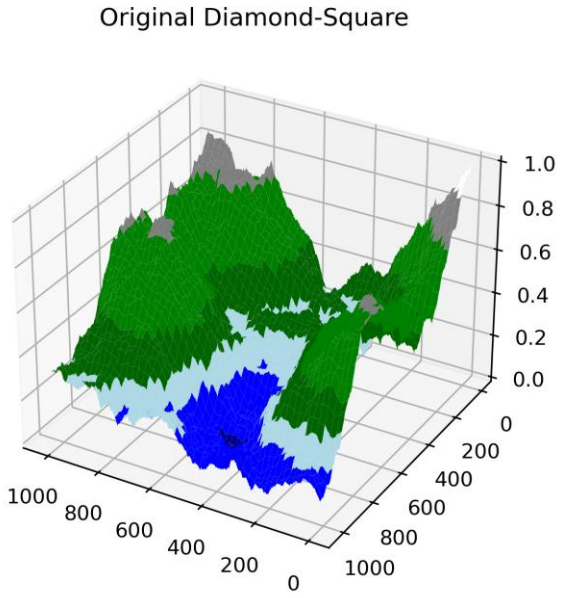
Wrapped Diamond-Square

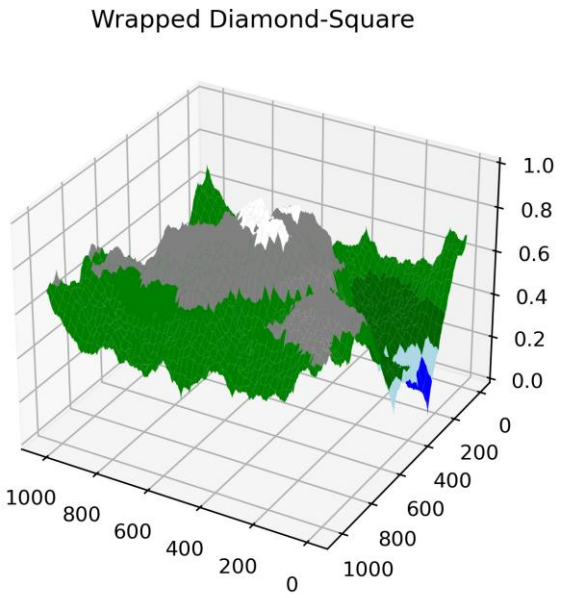
- Sets all 4 corners to the same value
- Matches all edge values via wrapping
  - Done with modulo indexing
- Ideal for infinite or looping maps

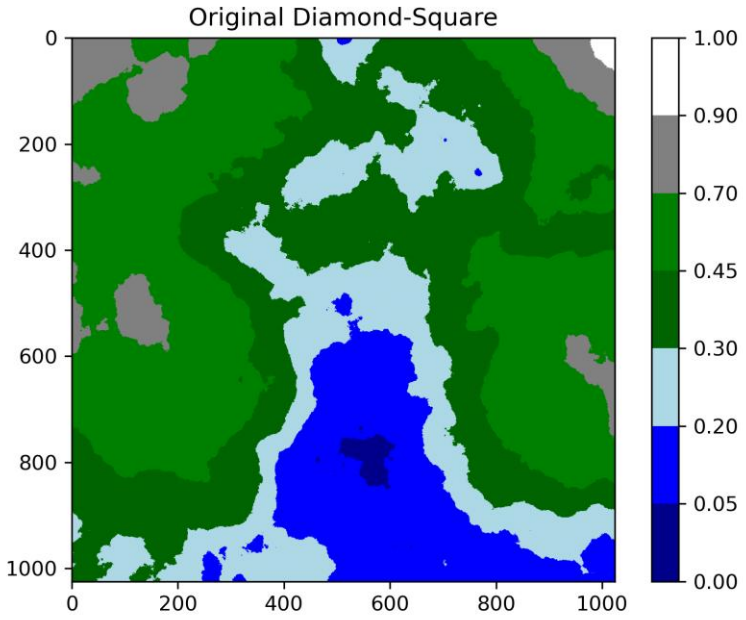
Radial falloff

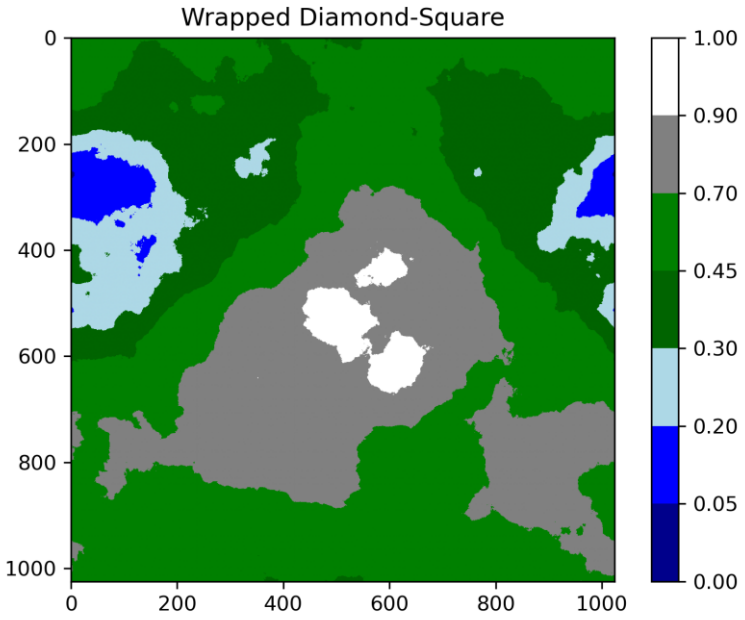
- Multiplies terrain height with a value decreasing as it gets further away from the center of the grid
- Emphasizes island-like elevation

Diamond-Square Examples

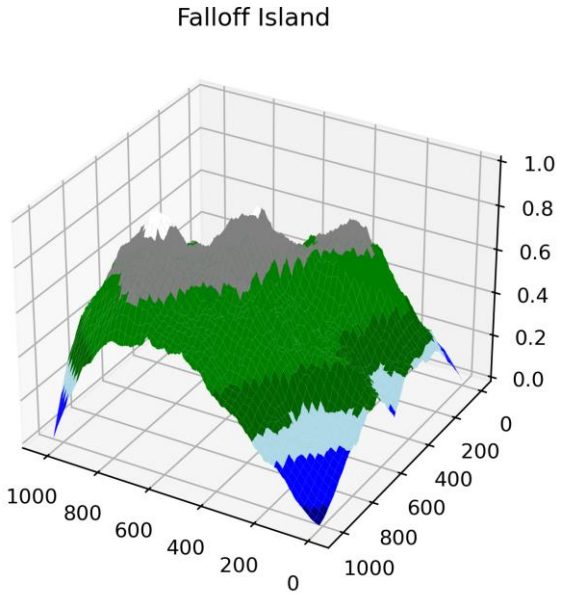


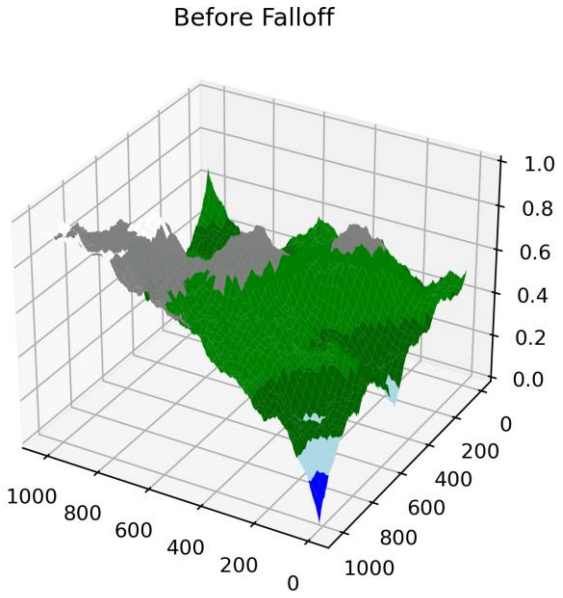


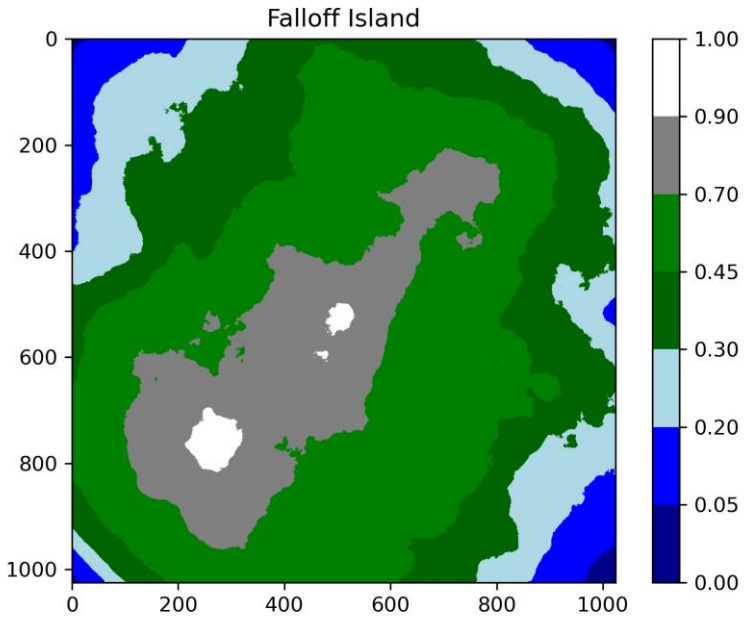


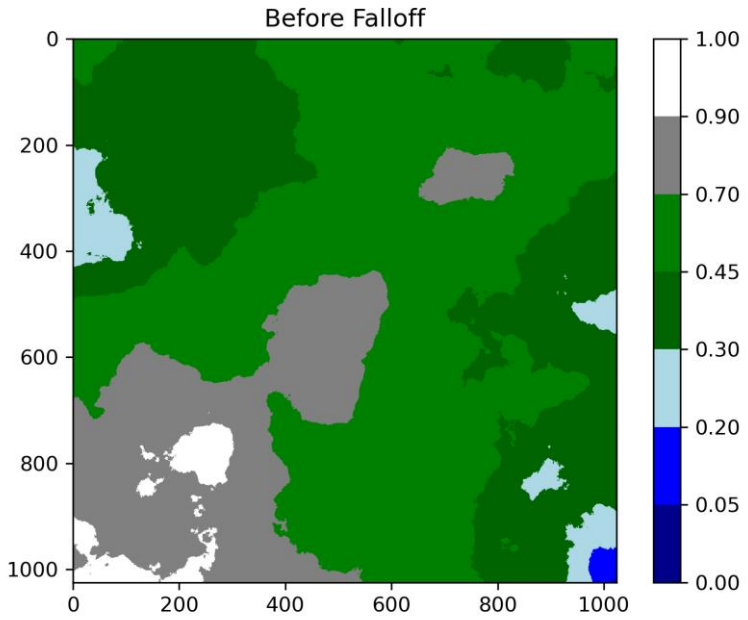


Radial Falloff Examples



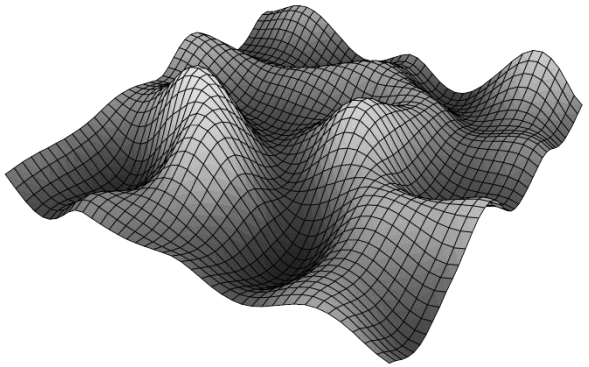




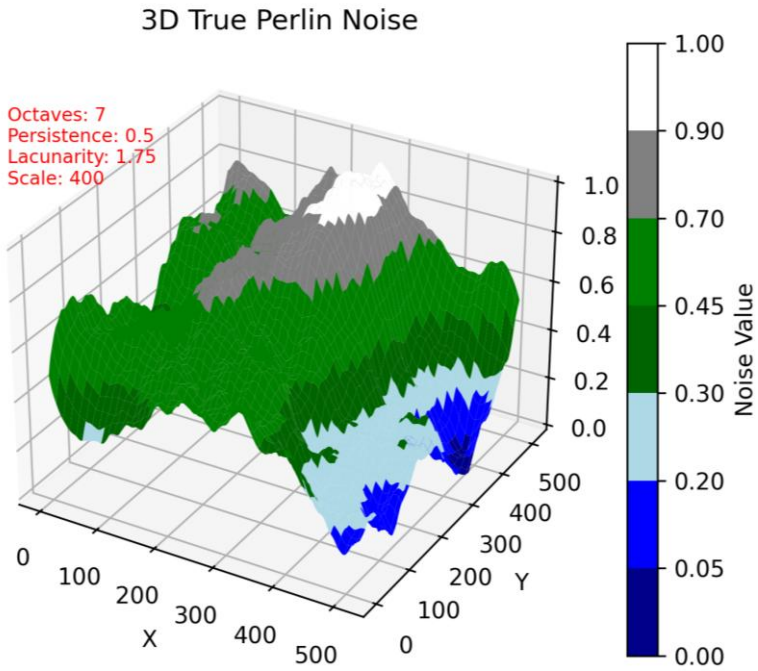


Perlin Noise

Perlin Noise Overview



- Created by Ken Perlin in 1983
- Produces smooth transition between values
  - Ideal for organic patterns
- Two main smoothing algorithms
  - Fractal and Gaussian



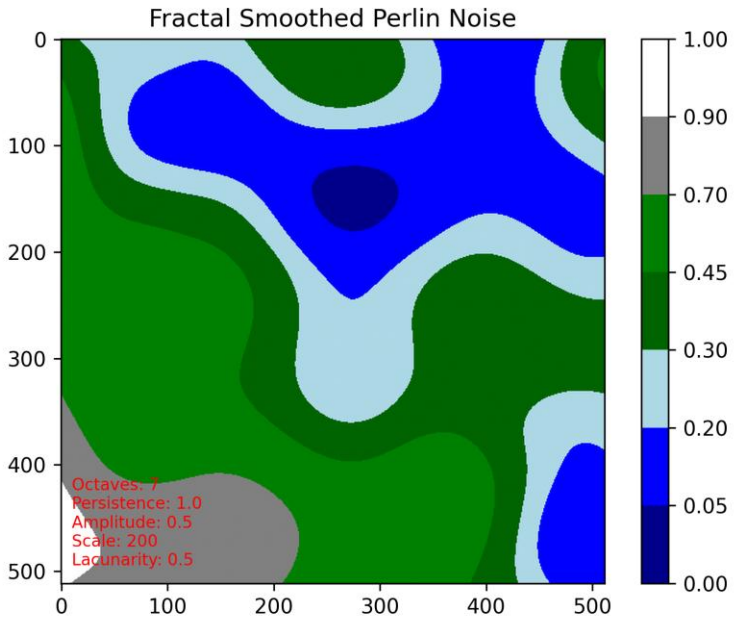
Perlin Noise Pseudocode

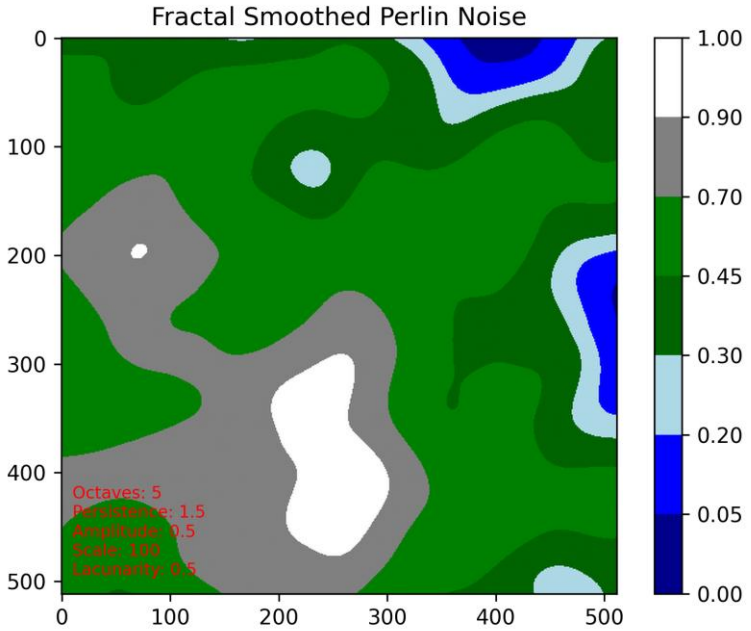
Grid Setup	• Creates grid where each cell is assigned a random gradient vector • Gradients will influence final value
Input Point and Grid Cell	• Identify surrounding cells
Dot Product Calculation	• Compute dot product between gradient vector and each grid point
Interpolation	• Smoothly blends grid points together
Final Noise Value	• Combines all interpolated values

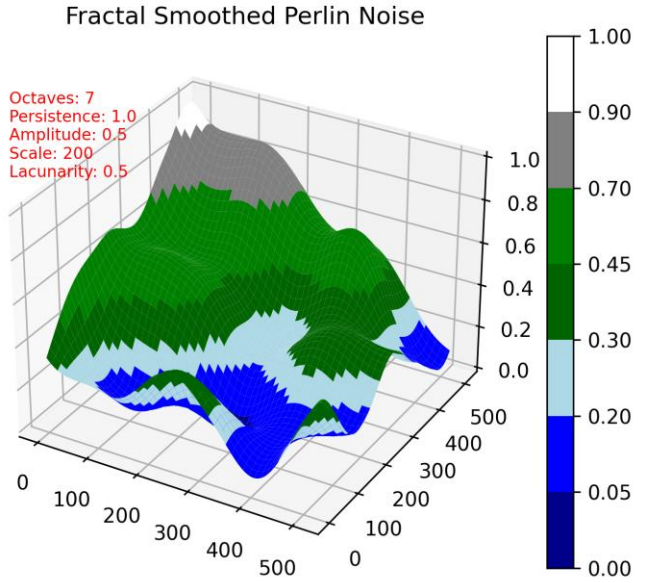
Fractal Smoothing

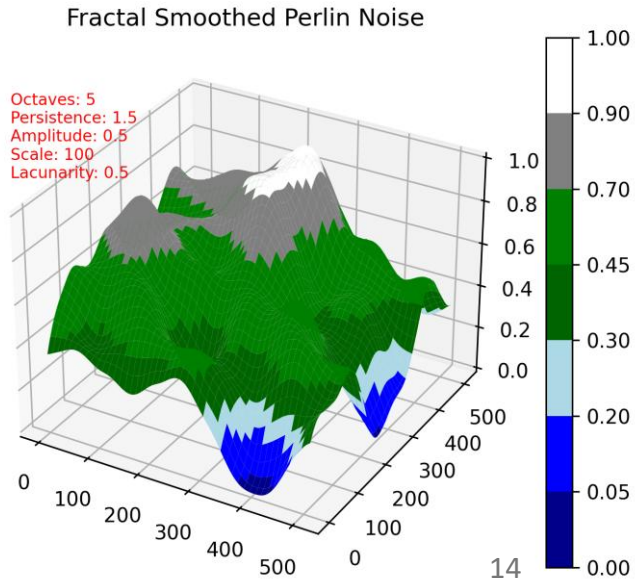
- A technique that combines multiple layers (or "octaves") of Perlin noise to create more detailed and realistic patterns
- How it works
  - Layering Noise (Octaves)
    - Multiple layers of Perlin noise are generated, each with increasing frequency and decreasing amplitude
    - Higher frequencies capture finer details, while lower frequencies define broader features
  - Combining Layers
    - The layers are summed together to produce the final noise value
    - The contribution of each layer is weighted by its amplitude
  - Smoothing Transitions
    - A smoothing function ensures seamless blending between layers

Fractal Smoothing Examples

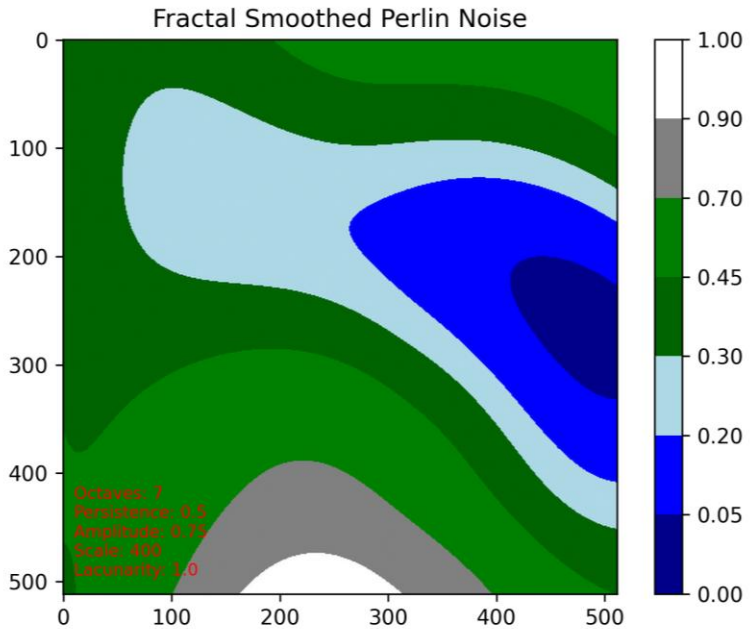


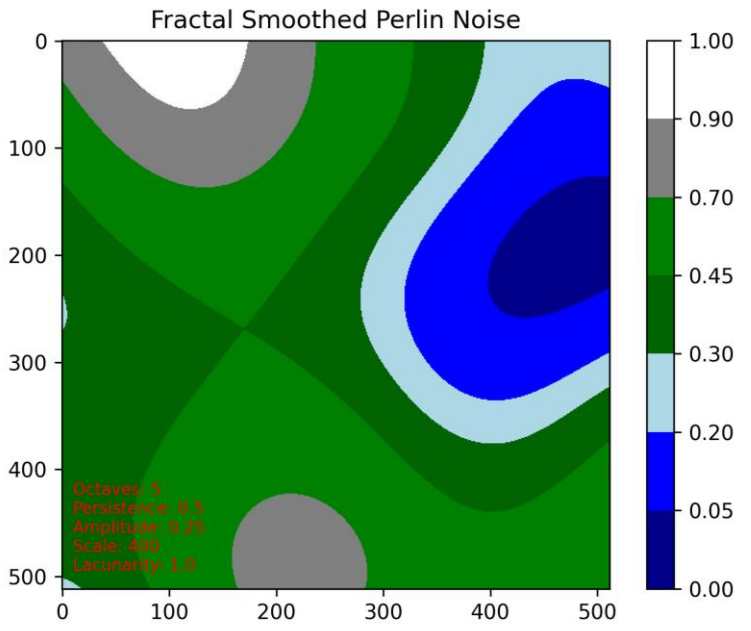


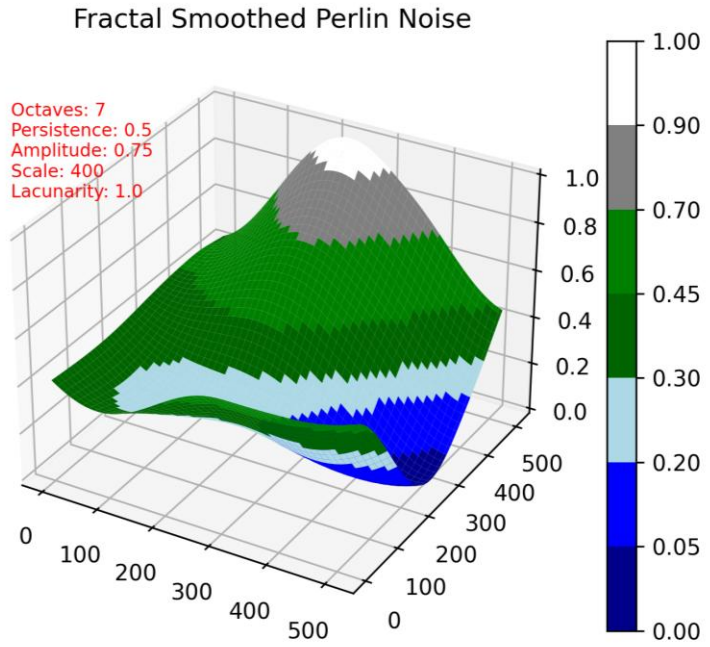


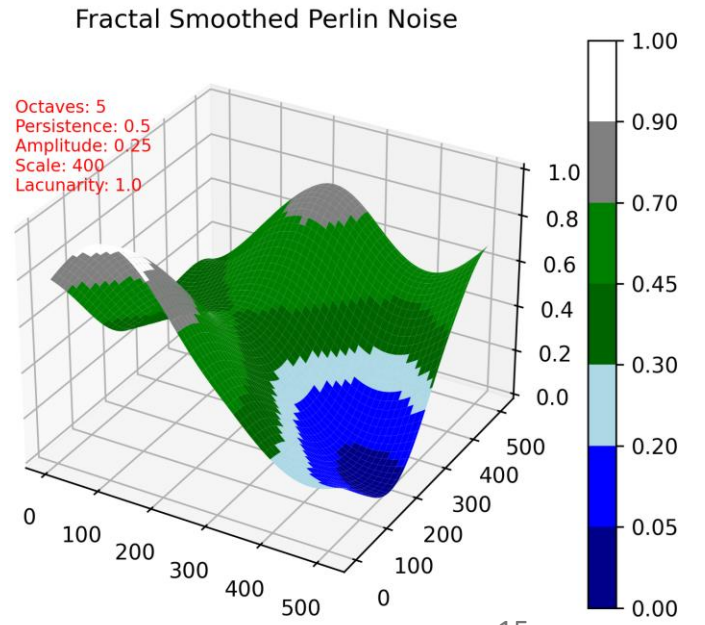


Fractal Smoothing Examples







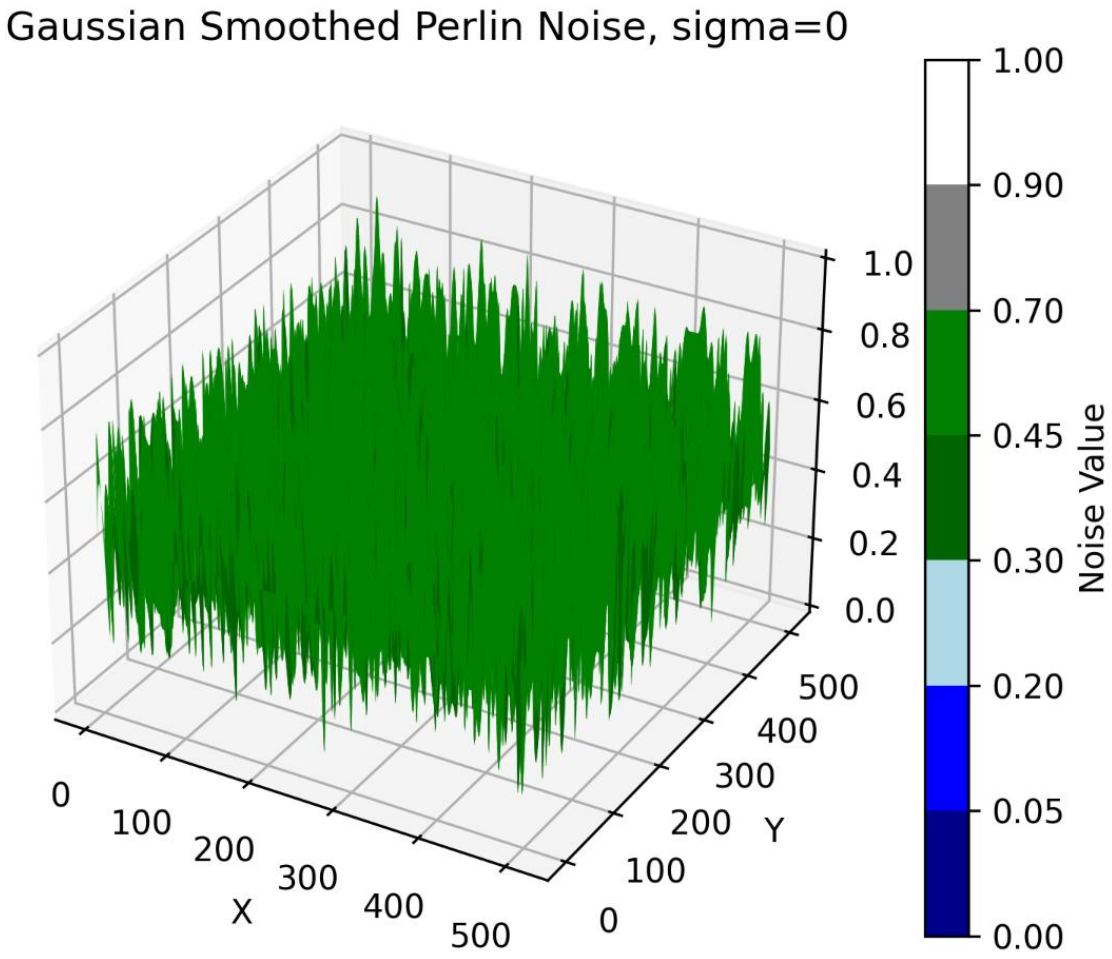


Gaussian Smoothing

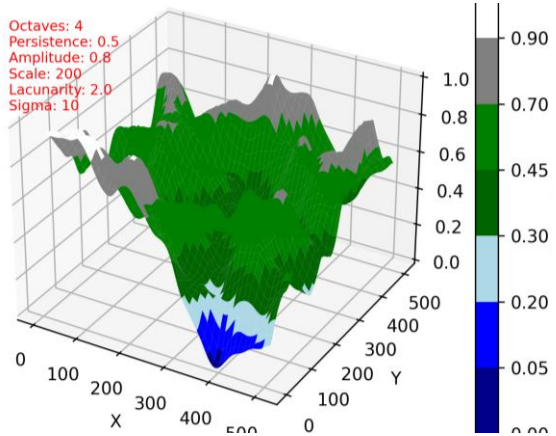
A technique used in image processing and data analysis to reduce noise and smooth data

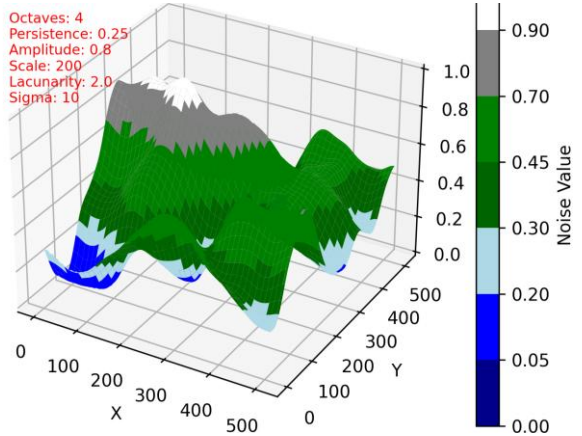
Applies a Gaussian function as a filter to weight nearby values more heavily than distant ones

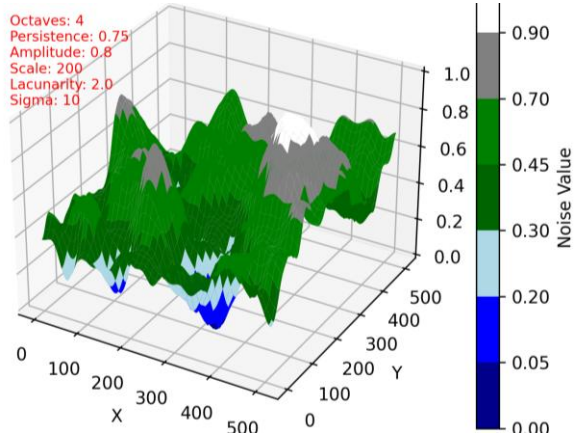
Gaussian Smoothing Example

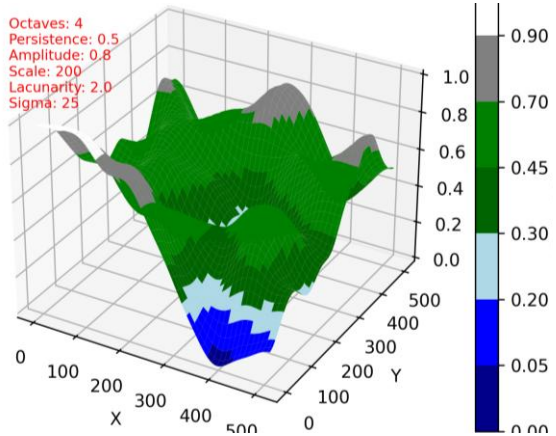


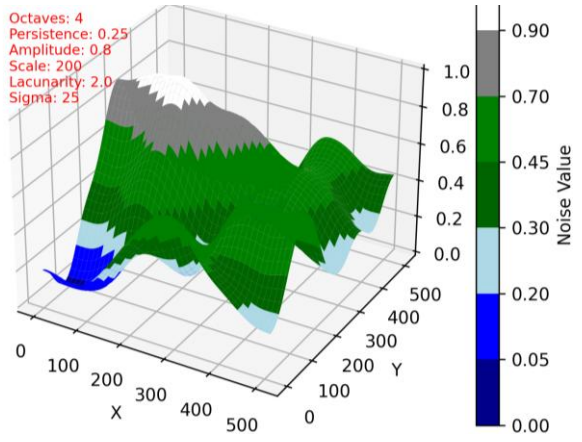
Combined Smoothing Examples

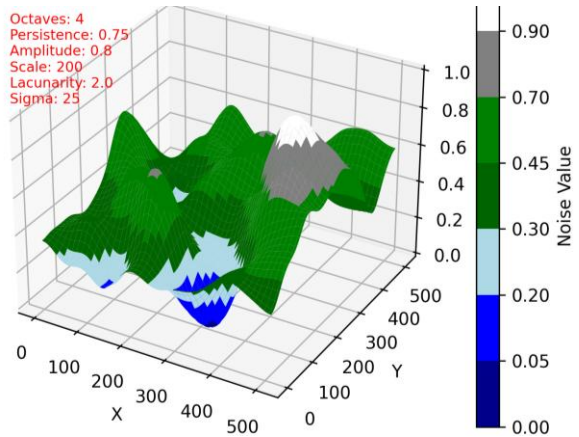












Voronoi Biome

Voronoi Biome Overview

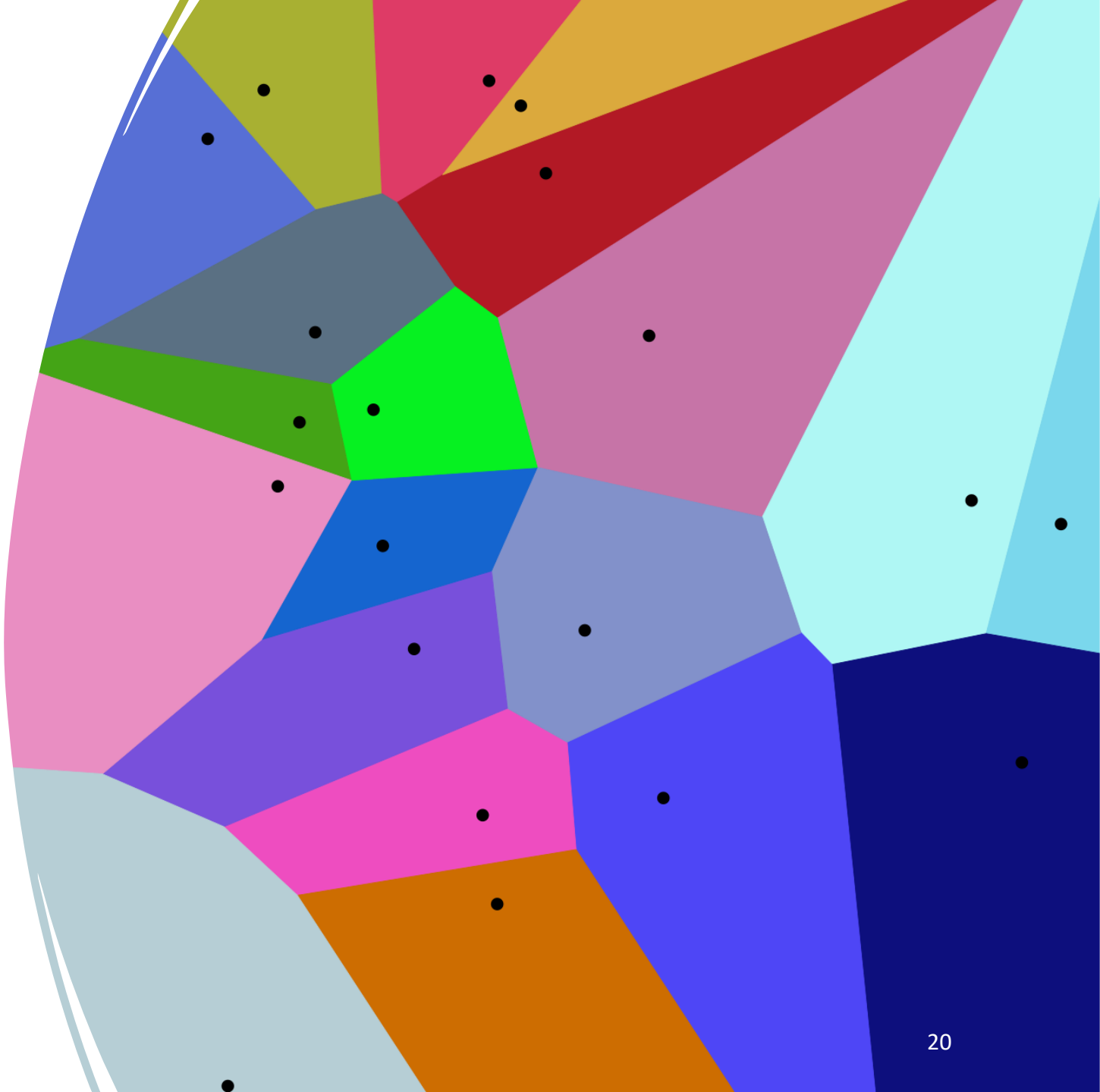
Based on Voronoi Diagrams, which can be traced back to 1644, but named after Georgy Voronoi who defined it in 1908

- Creates regions based on distance to randomly selected seed points
- Importantly, does not create any height

Ideal for region-based world generation with smooth transitions

Two-tiered system

- Major seeds – base biomes
- Minor seeds – blend between biomes for transitions



Voronoi Biome Pseudocode

Grid Setup

- Define a list of biomes and assign colors

Major Seeds

- Select random positions for the desired number of major seeds
- Assign each major seed a biome, optionally ensure each biome is selected at least once

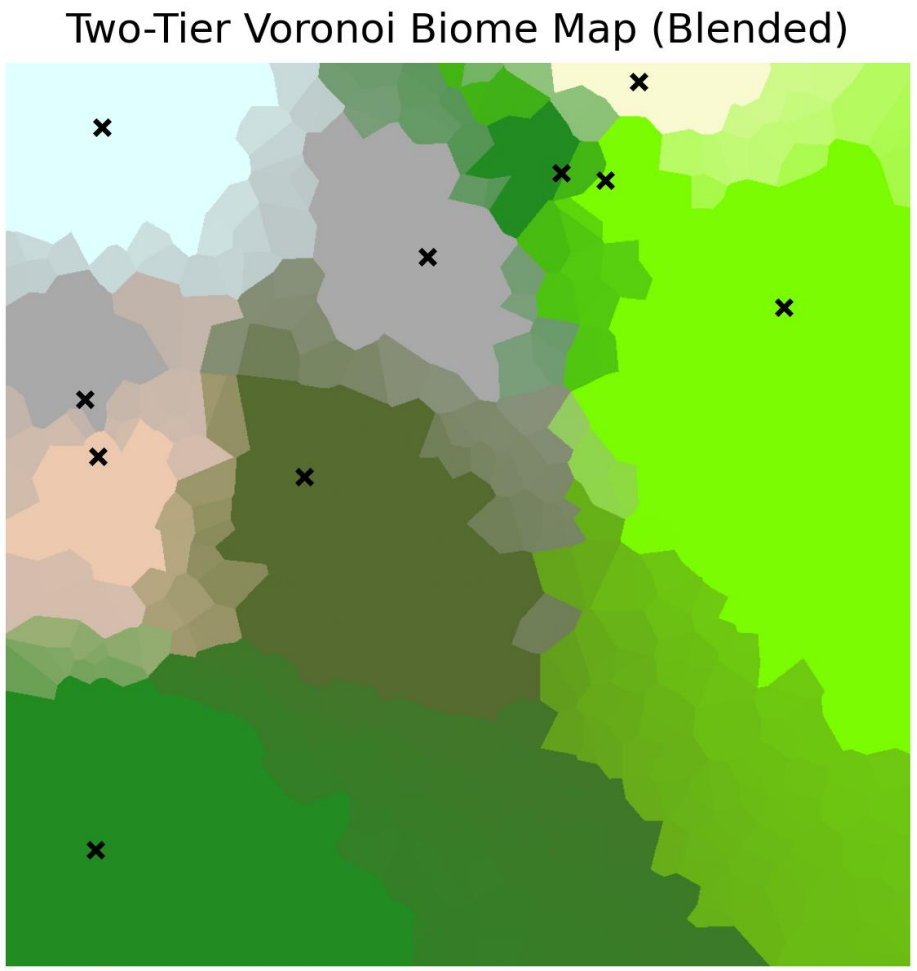
Minor Seeds

- Generate random positions for the desired number of minor seeds
- For each minor seed, find the two nearest major seeds
  - If distance is about the same, blend color proportionally
  - Otherwise, match closest major seed color

Fill-in

- Assign colors to every pixel based on closest minor seed

Two-Tier Voronoi Biome Map



Comparative Analysis

Comparative Analysis

- Perlin Noise
  - Generates smooth, continuous terrain variations, ideal for gradual slopes and natural landscapes.
- Diamond Square
  - Uses recursive subdivision to create rugged, mountainous terrain with dramatic elevation changes.
- Voronoi Biome
  - Divides terrain into distinct regions, simulating biome separations or geological formations.
- Comparison
  - Each algorithm has unique strengths but can be integrated to enhance terrain complexity and realism.
- Combined Approach
  - Perlin Noise for base elevation
  - Diamond Square for added ruggedness
  - Voronoi Biome for biome distribution enables diverse and adaptable procedural environments.

Questions and Demo