Lecture 24
Shader Topics
A - Noise

The Model In Review

Each vertex is independent

Each fragment is independent

We compute each thing separately

- it's OK, parallel so it's fast!

What about randomness?

Regular patterns look too boring

We don't want the patterns to be obvious...

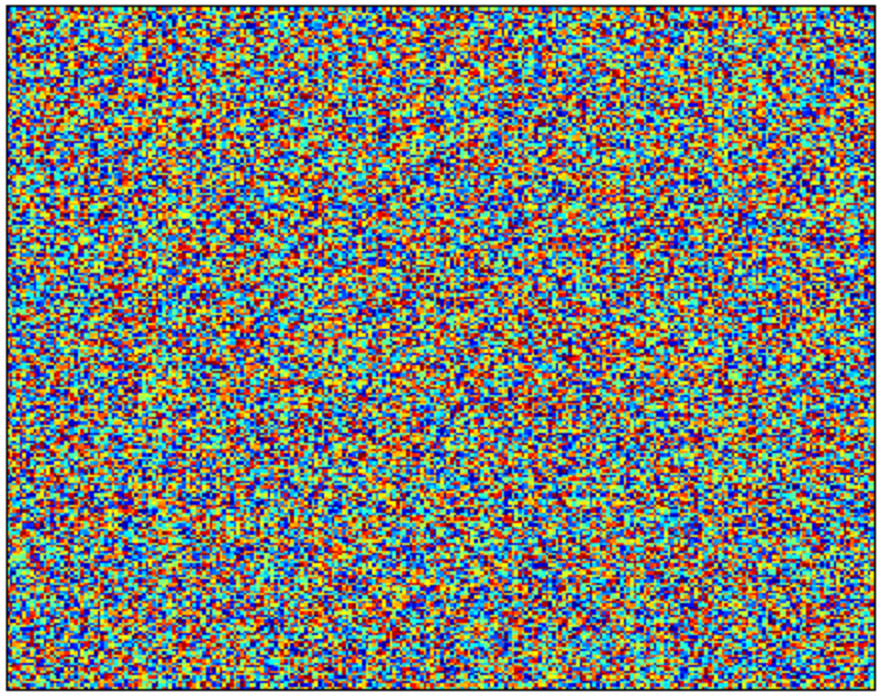
"Controlled Randomness"

- no randomness? boring
- totally random? boring

- structure + controlled randomness? Good!

Purely Random (each pixel)

White Noise 256x256 (1-channel)

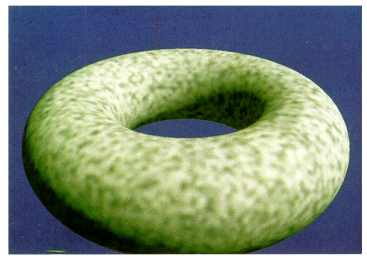


Blank Slide

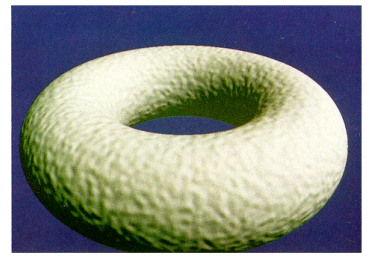
Historic Examples

Ken Perlin, 1985

He worked on "Tron"

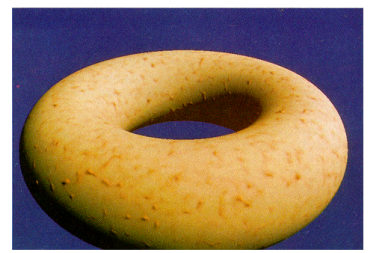


Spotted Donut



Bumpy Donut

Stucco Donut



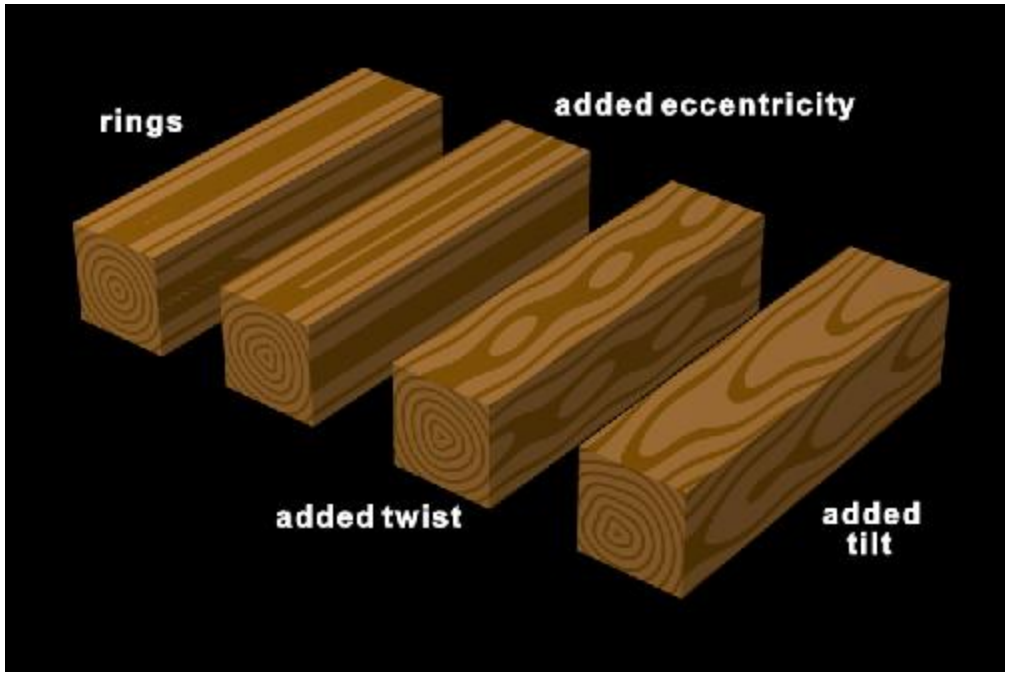
Disgusting Donut

Bozo's Donut

Wrinkled Donut

Wood

Wood



Simple Stuff Becomes Fancy

We can't really use randomness

- Each frame independent
  - random would cause it to change each time
- Each fragment independent
  - no way to have structure between fragments

Noise

1. Psuedo-random
    - pattern too complex to see
    - but still is controlled / deterministic
2. Structured
    - control properties that we care about

Noise - A Simple Method

Start with a 1D pattern (since it's simpler than 2D)

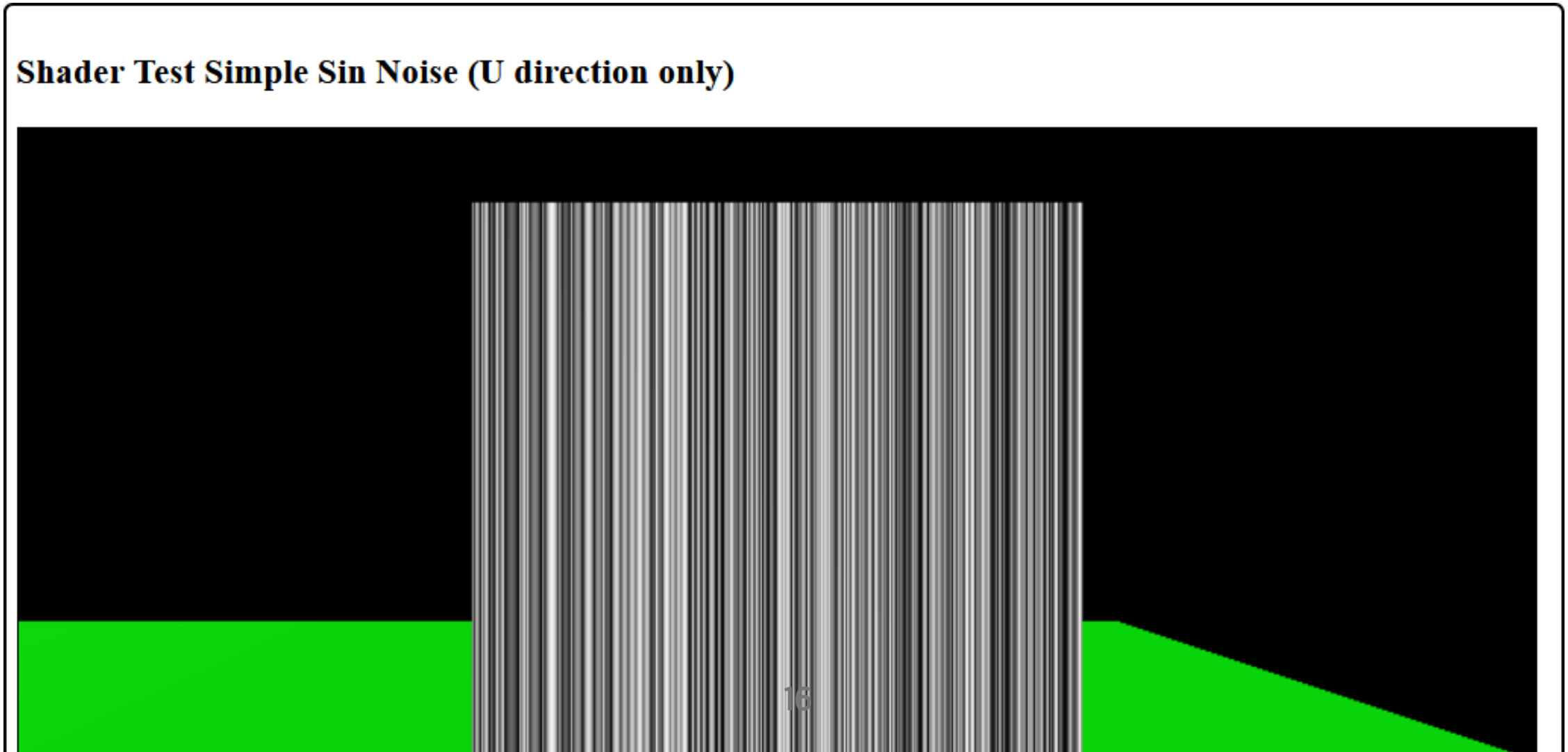
color = f(u) (like stripes)

Simple Psuedo-Random

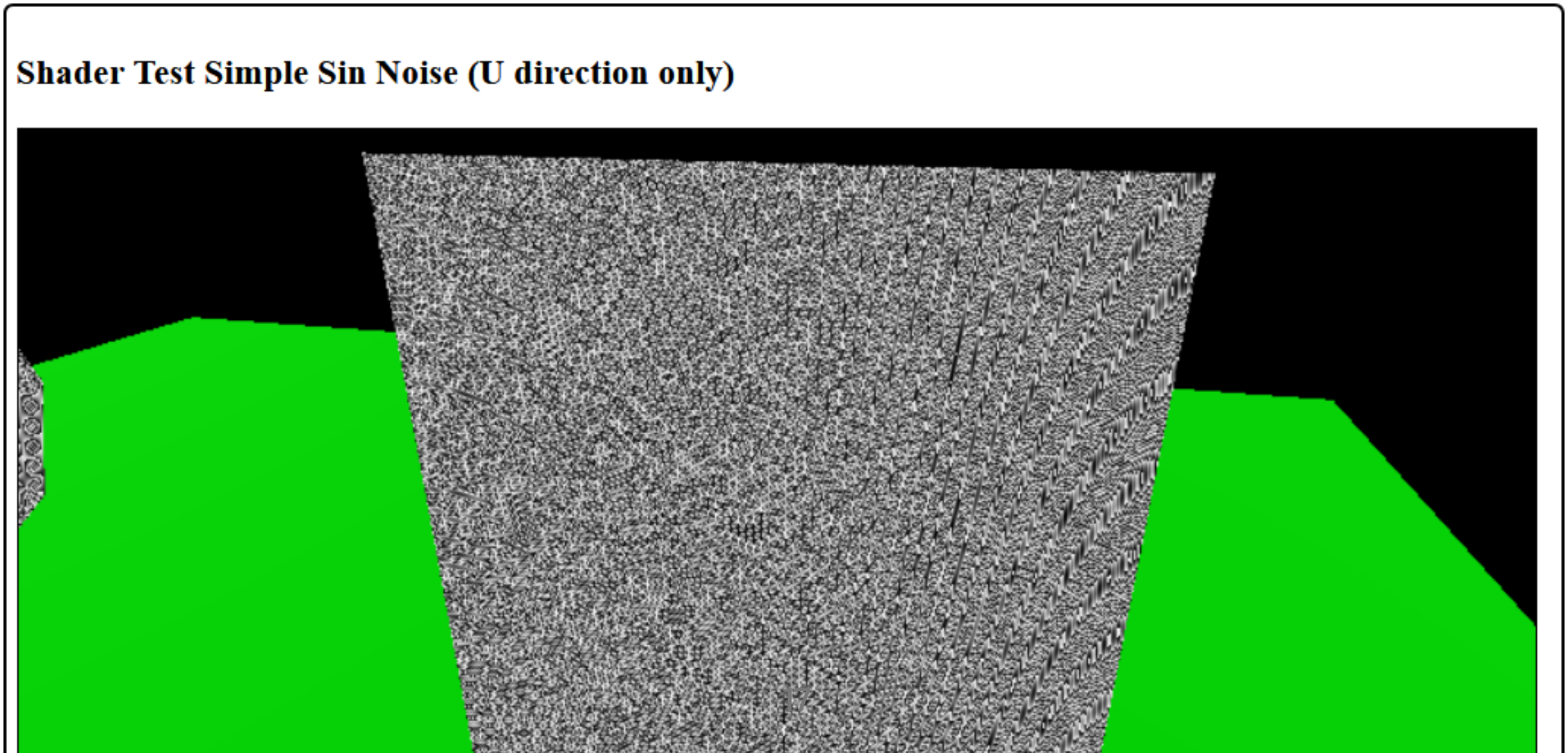
(many better choices)

```
r = fract( sin(u) * 100000. )
```

What does this look like?



A problem



What's happening?

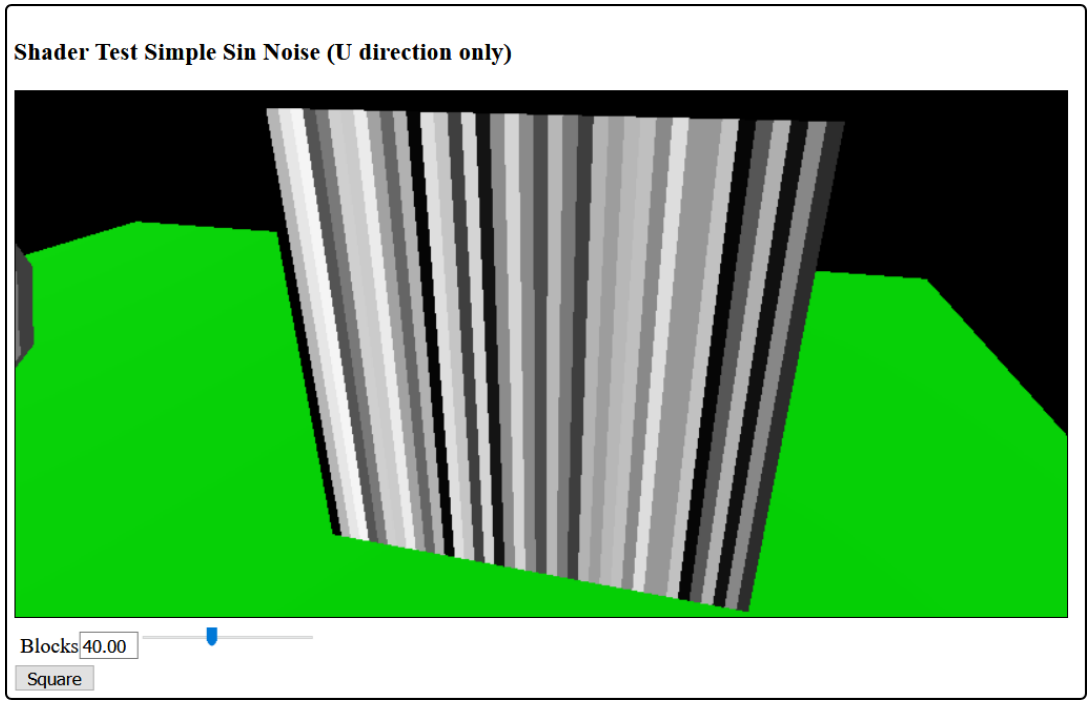
Structure: change slowly

Sample (points along line)

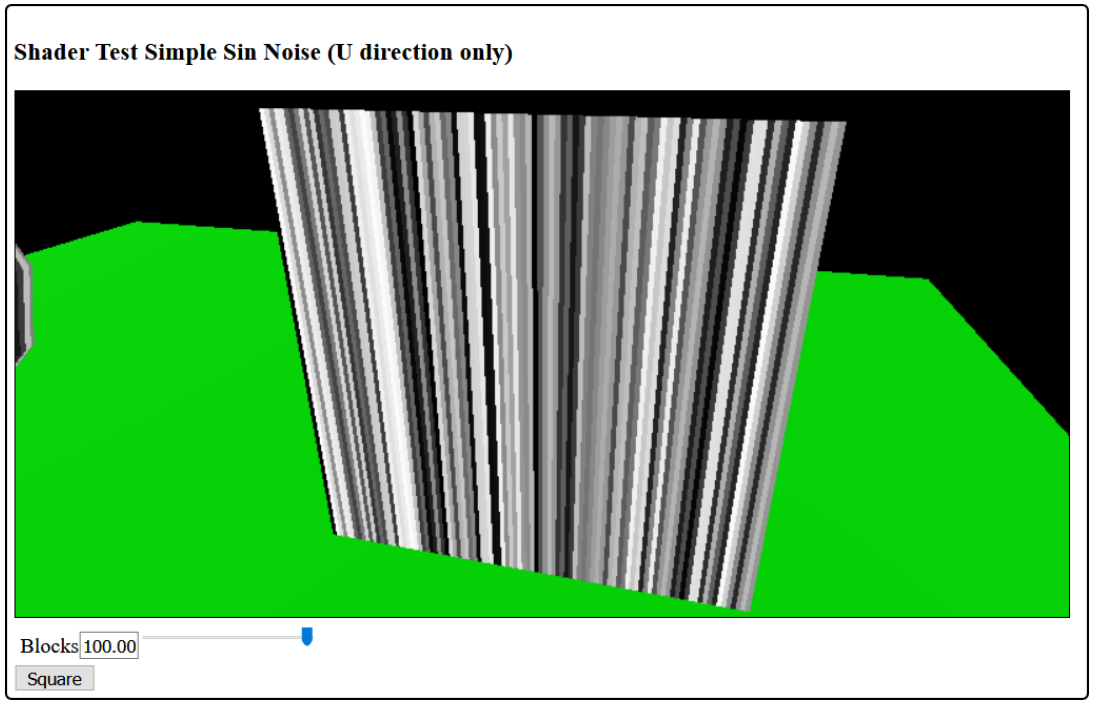
Aliasing - but adds to randomness

Using Simple Randomness

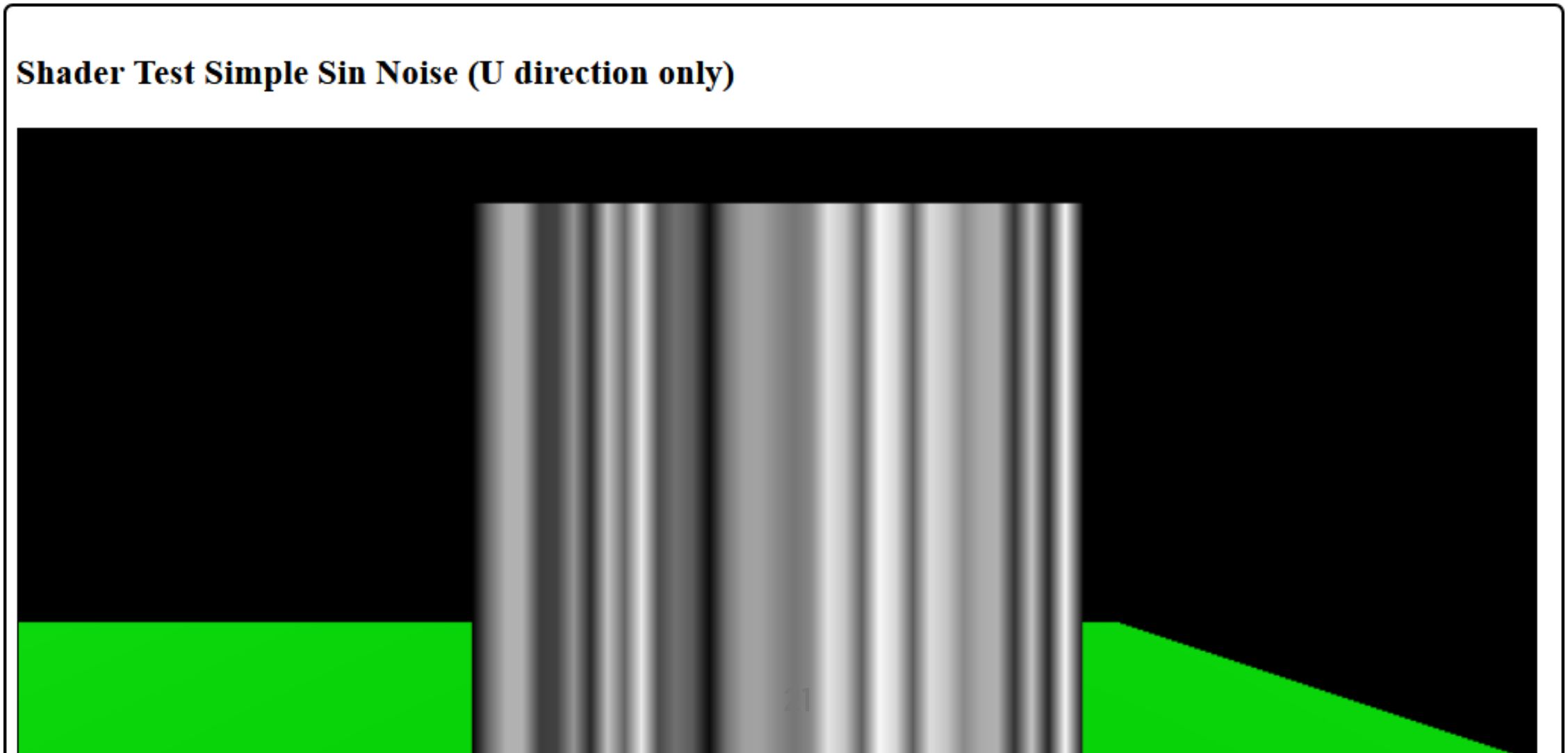
40 points



100 points

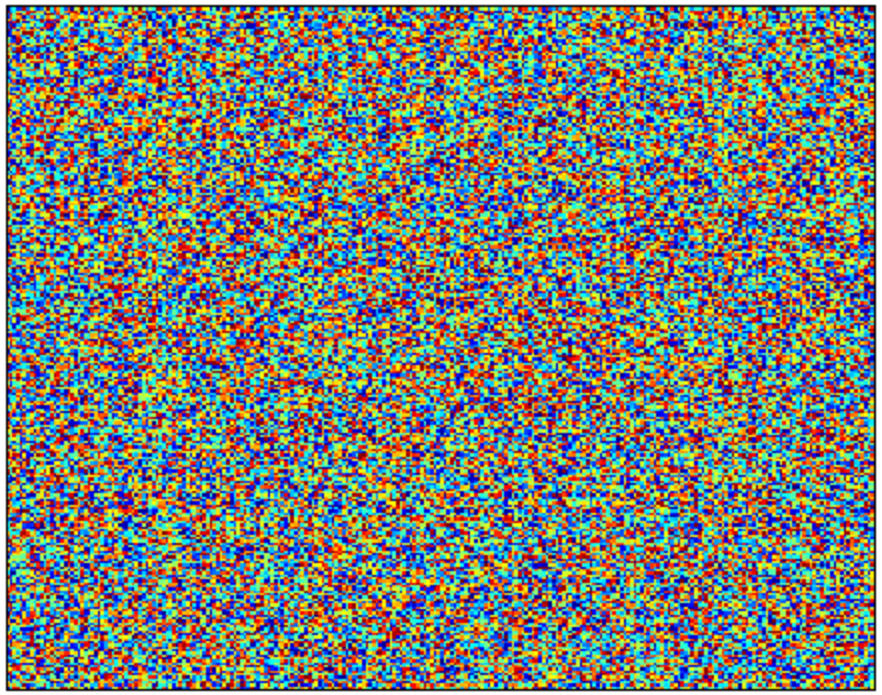


Interpolate to make smooth



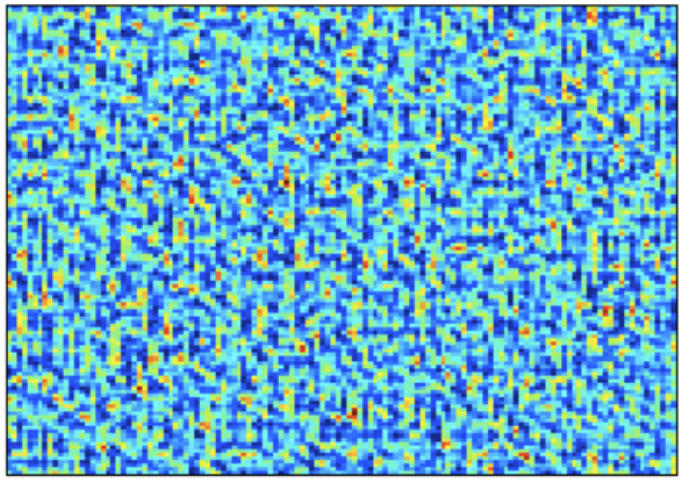
In 2D

White Noise 256x256 (1-channel)

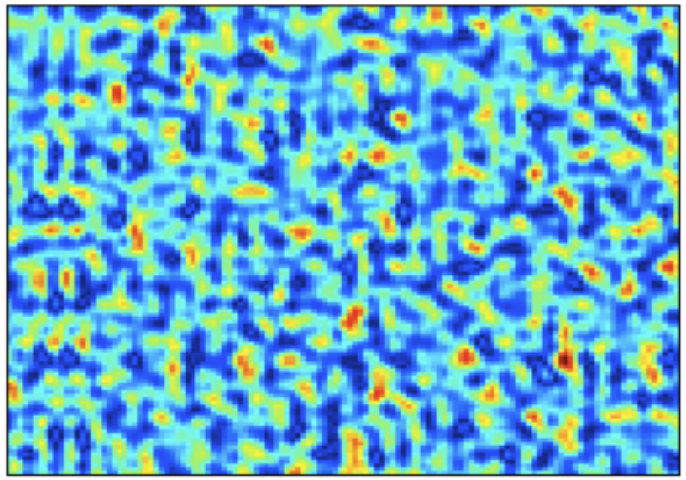


Note: this is a better "random function"

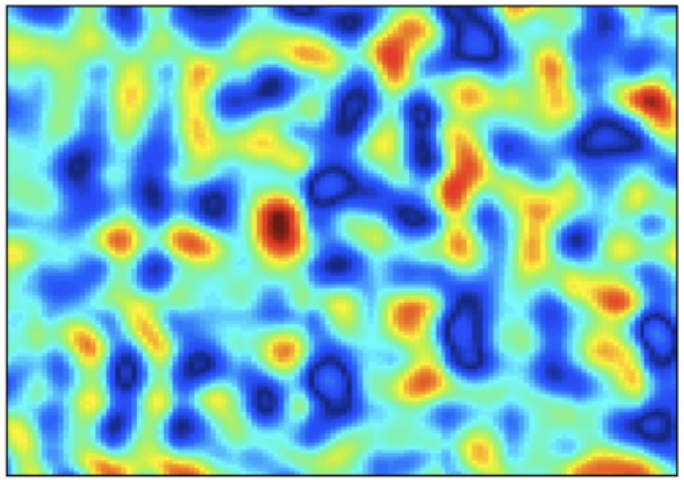
50x50 grid



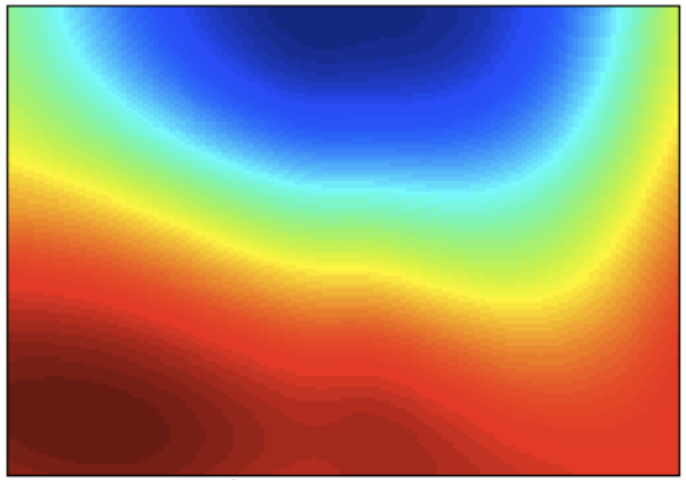
25x25 grid



10x10 grid



1x1 grid



You can do a lot better...

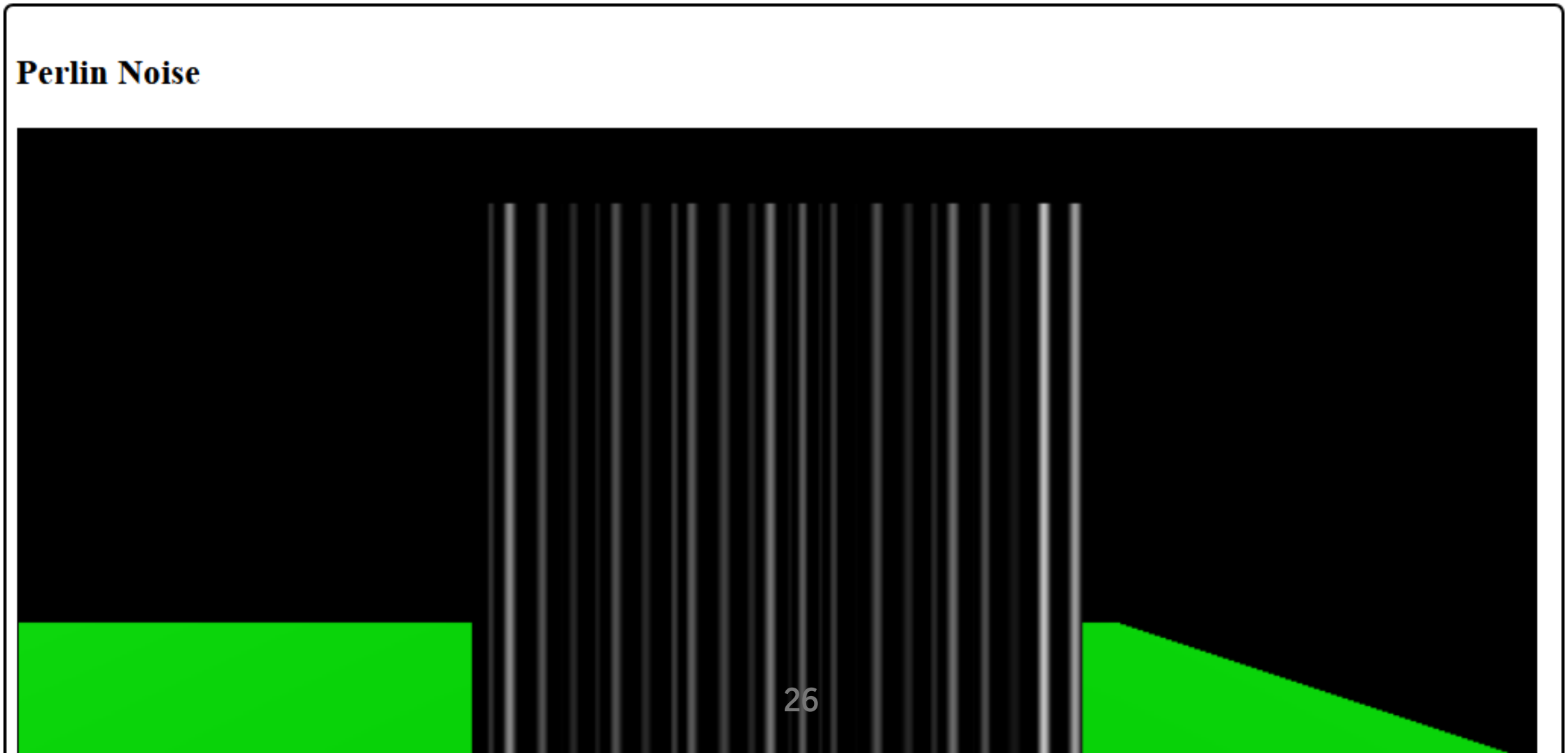
- Better Psuedo-random functions
- Efficient in multi-dimensions (2D, 3D)
- Tileable
- Better interpolation
- Multiple frequencies

Perlin Noise

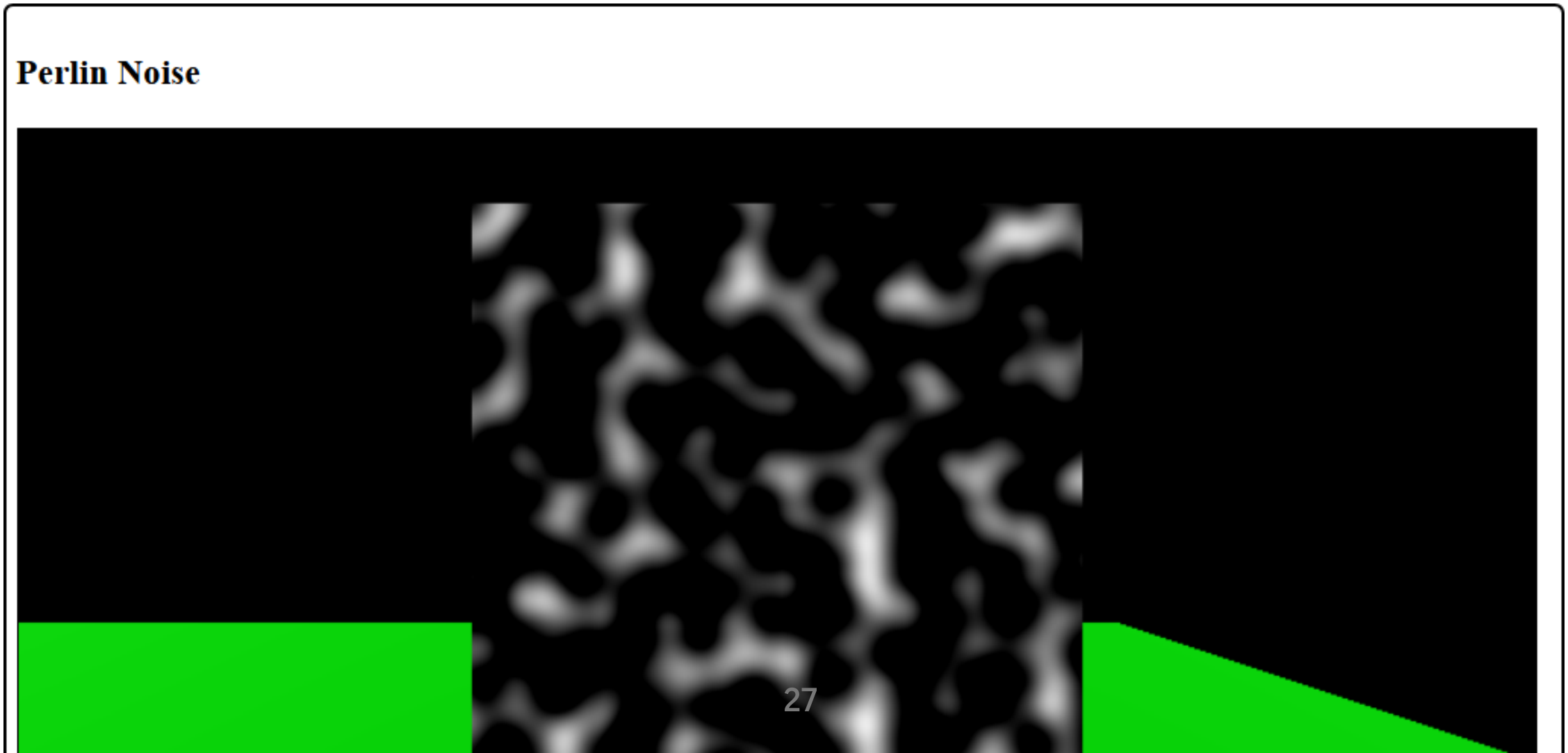
The classic noise function

- newer variants are more efficient on GPUs
- even better psuedo-randomness
    - noise - controlled psuedo-randomness
        - Perlin noise
            - coherence at different frequencies
            - demo (1D)
            - demo (2D)

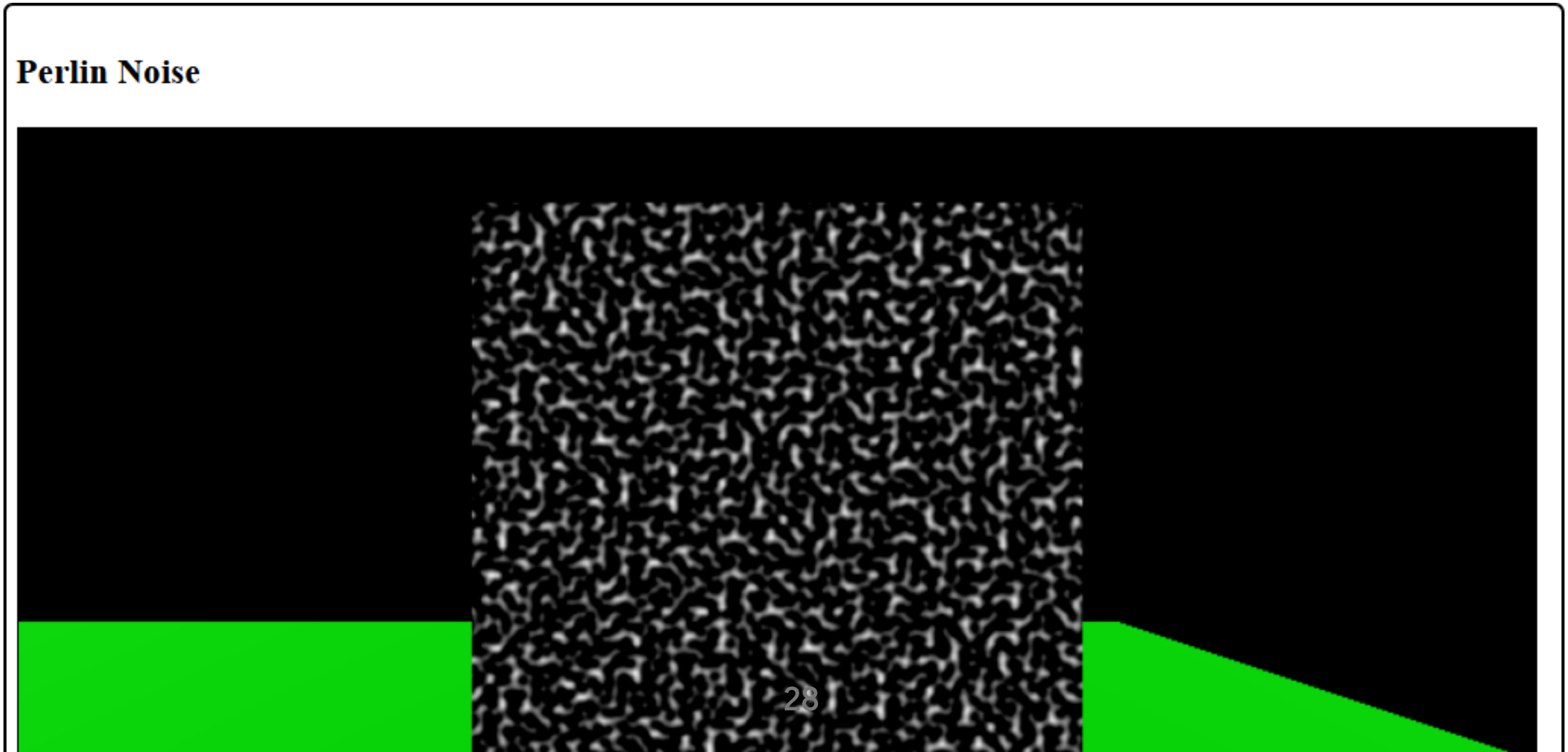
Perlin Noise in 1D



Perlin Noise in 2D

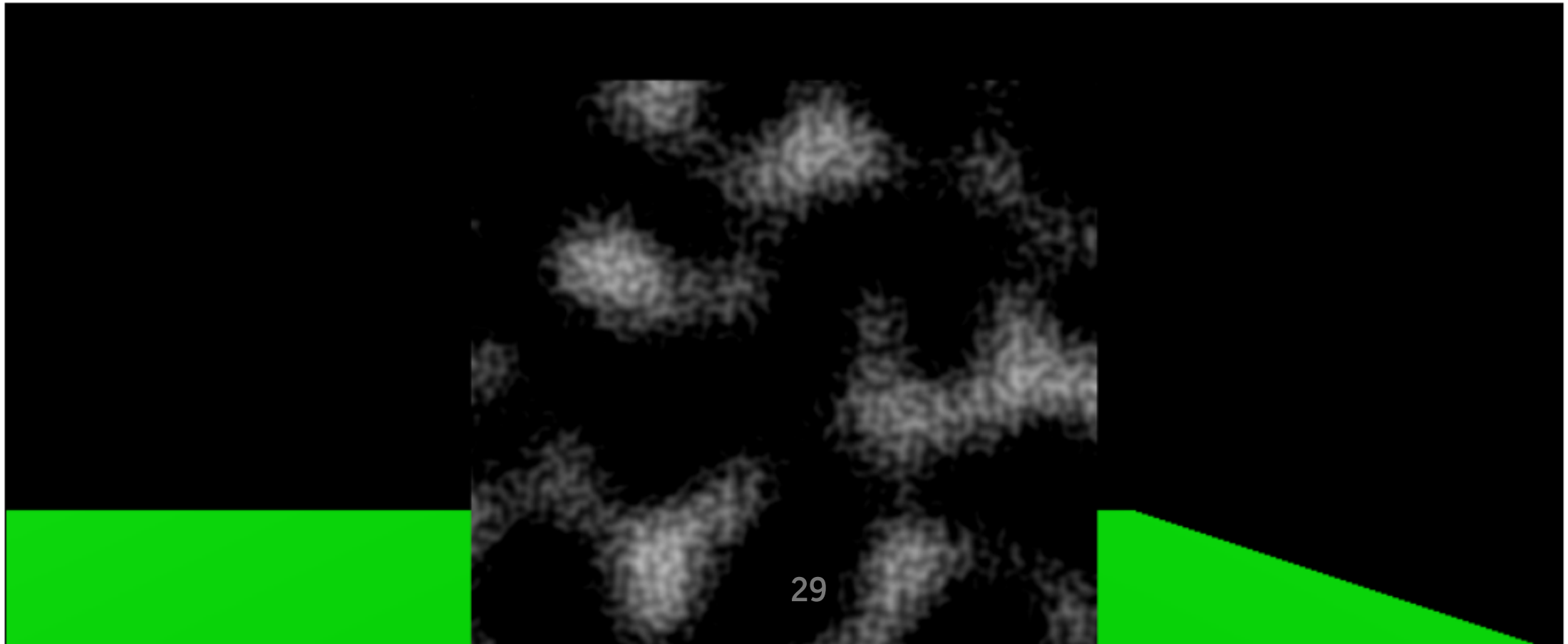


Perlin Noise in 2D - High Frequency



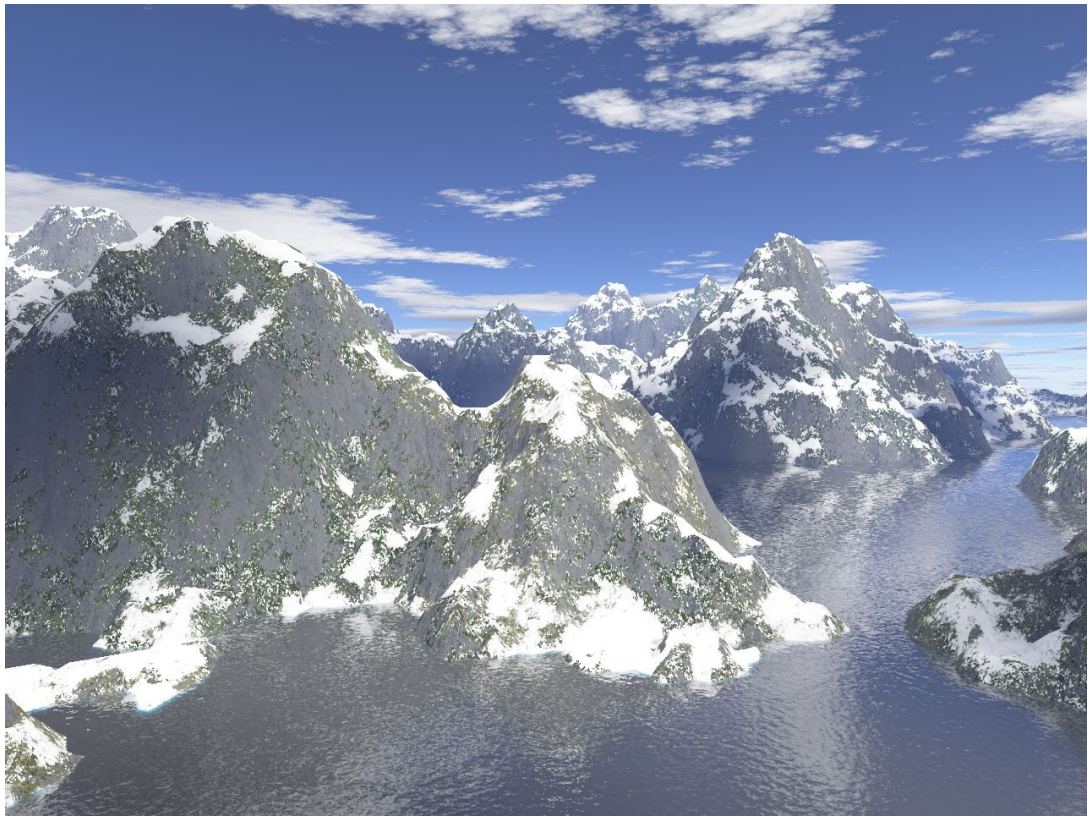
Multi-Frequency: Low + High

Perlin Noise

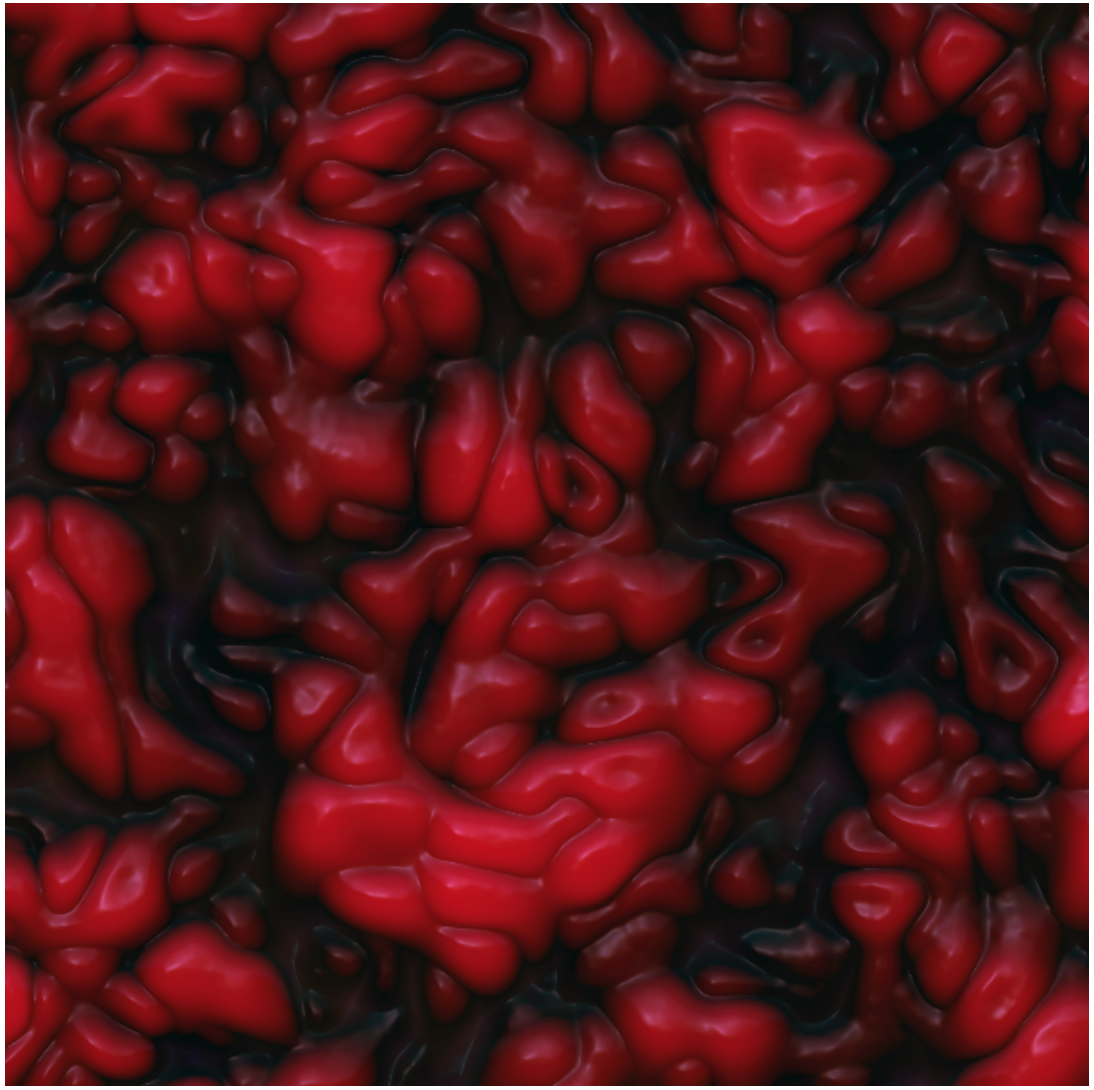


How do you use this?

- Find an implementation on the web
- Mix different frequencies to get desired effects
- Add noise to make things less "perfect"
- It's an art



By Stevo-88 - self-made, used Adobe Photoshop for Perlin noise creation and Terragen for rendering., Public Domain, https://commons.wikimedia.org/w/index.php?curid=2208011



By Simon Strandgaard from Kastrup, Danmark - pink/red liquid using perlin noise + bump + coloring, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=76348609