

Procedural generation of textures and geometry

An introduction to procedural methods. Preliminary contents:

- 1) Basic procedural methods for images and textures in CPU and GPU
- 2) Random numbers in shaders
- 3) Noise generation.
- 4) Texture generation with noise
- 5) Fractals, L-systems
- 6) Fractal Brownian motion for terrain generation
- 7) Tools for procedural generation of geometry

I can't cover all this in depth so I will choose what I hope is the most interesting.

I hope that you are familiar with

- pixels
- RGB color
- recursion

but how familiar are with these?

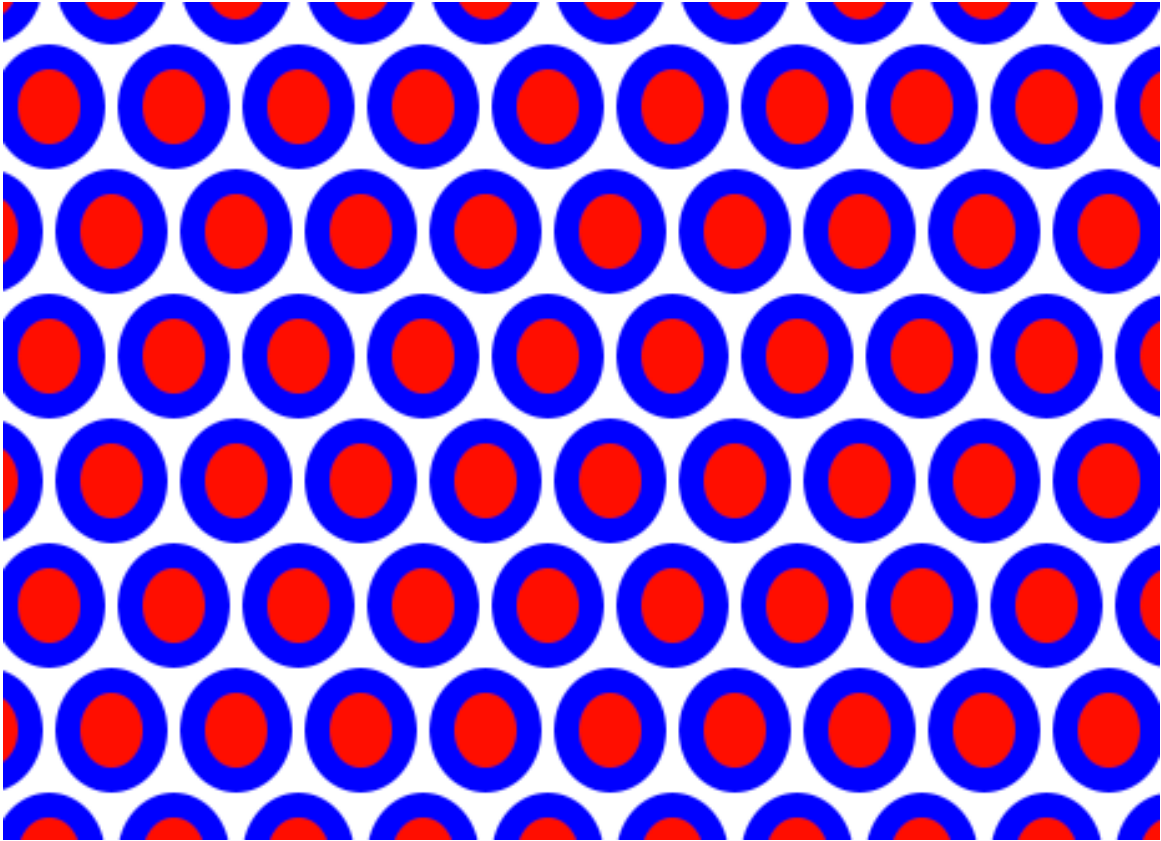
- shaders
- polygon rendering
- OpenGL

I need your input during the session to help me have the right level.

1. Basic procedural methods.

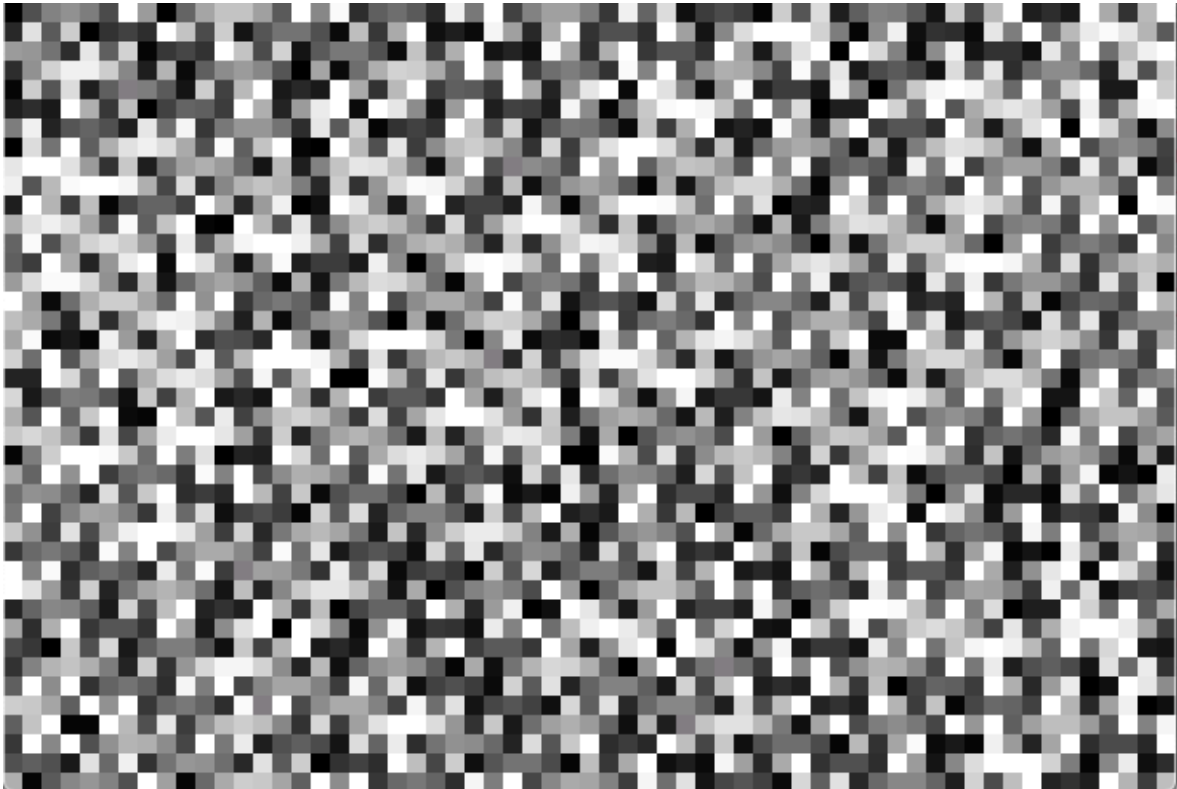
Brief introduction to shader programming, in case you are new to it.

We can easily create images like the one below, but if you do that in a shader?



2. Random number in shaders

Randomness turns out to be harder than you might expect. We can not use the sequential methods any more.



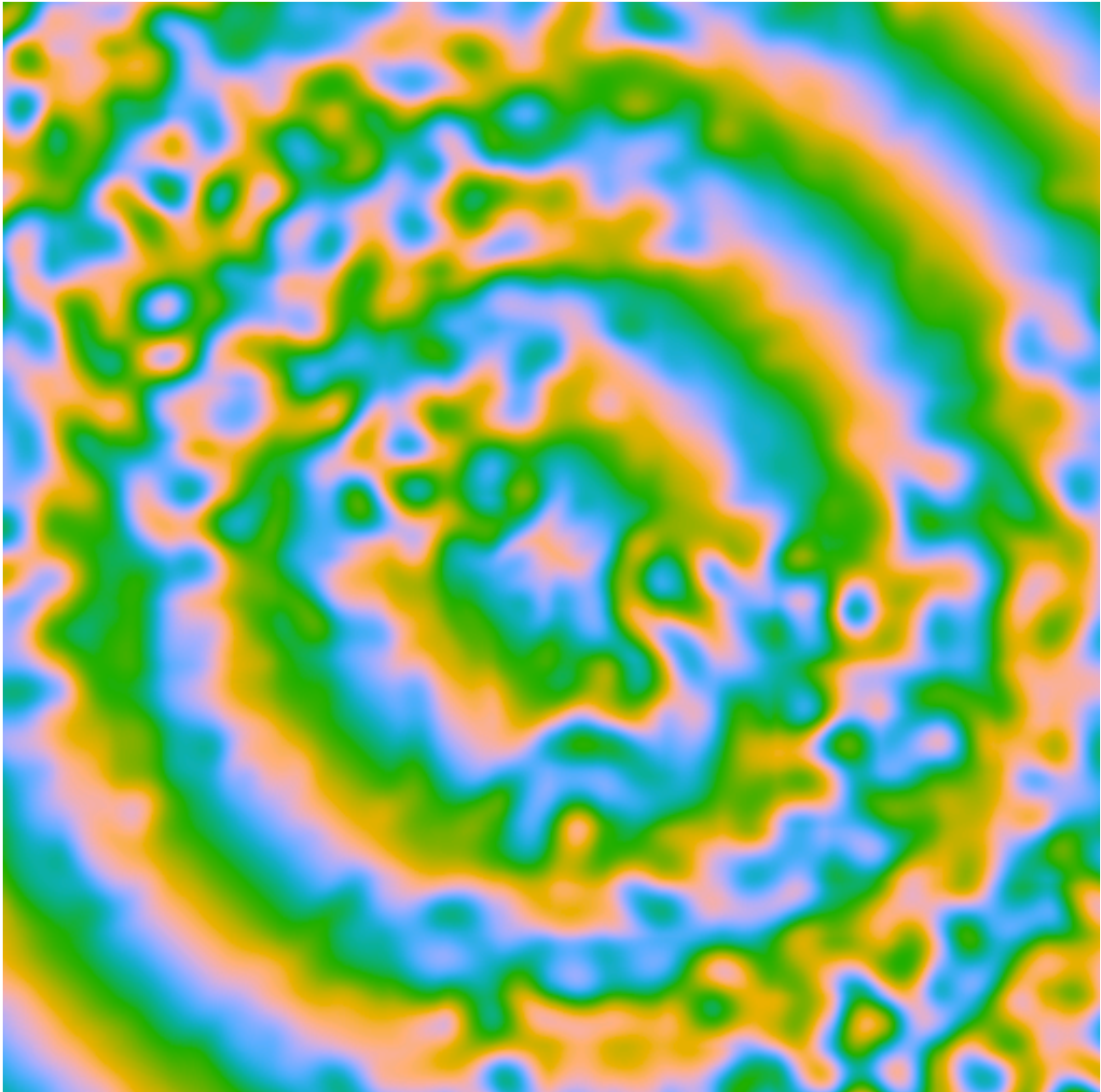
3. Noise generation

Randomness gives noise... Yes, but you can do that much smarter.



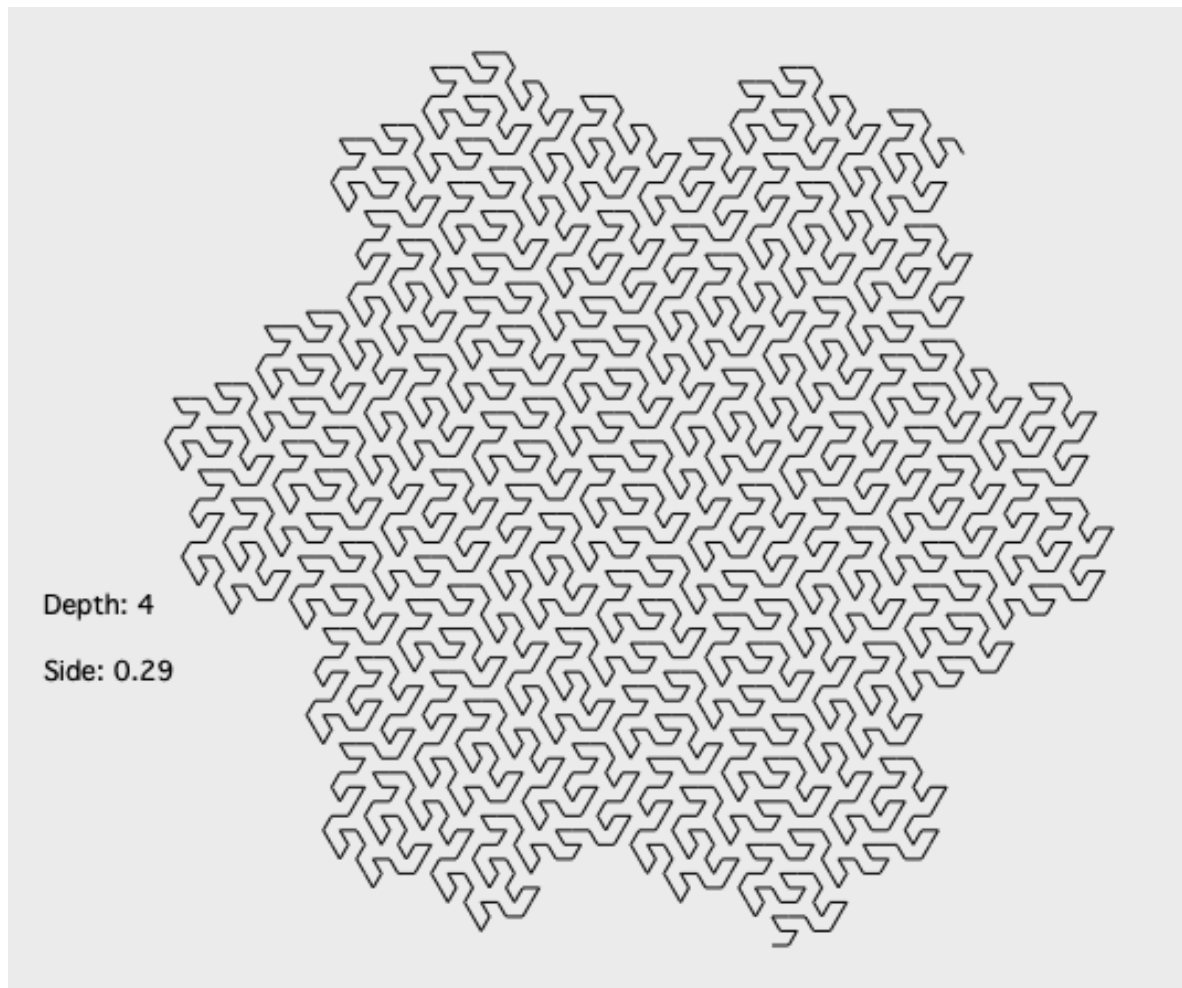
4. Texture generation with noise

Now we have better noise. Let's be create with it! This is a mix of noise and a sinus wave:



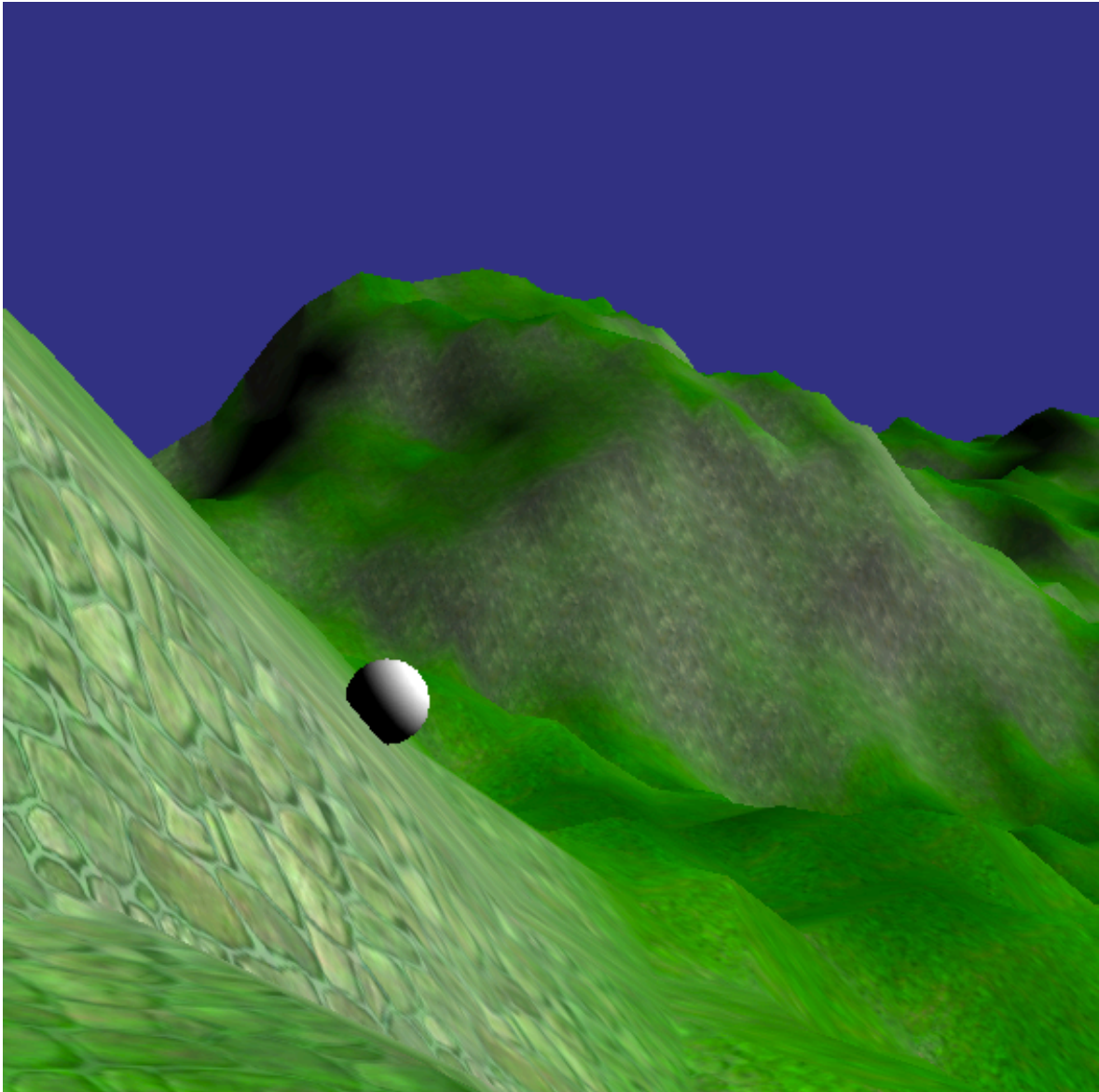
5. Fractals, L-systems

How to create interesting images by a rule system.



6. Fractal Brownian motion for terrain generation

Fractal Brownian Motion has nothing to do with motion but a special kind of fractals that is useful for things like creating terrains.



7. Tools for procedural generation of geometry

We have some in-house tools for making it easier to produce geometry from code.

