Progress Report

Bruce Alcock

A Procedural, Minimal Input, Natural Terrain Plug-in for Blender

Supervisors: Kevin Glass and Shaun Bangay and Hannah Slay

Date: 16 August 2007

1 Previous Objectives

- Do a comprehensive visual analysis of all the factors influencing the Fractional Brownian Motion (fBm) generator with the intention of being able to control not only the maximum height at a position, but create smoothly varying river banks too for instance.
- Experiment with creating Squig curves to generate the river paths [1].

2 Progress

2.1 Poster

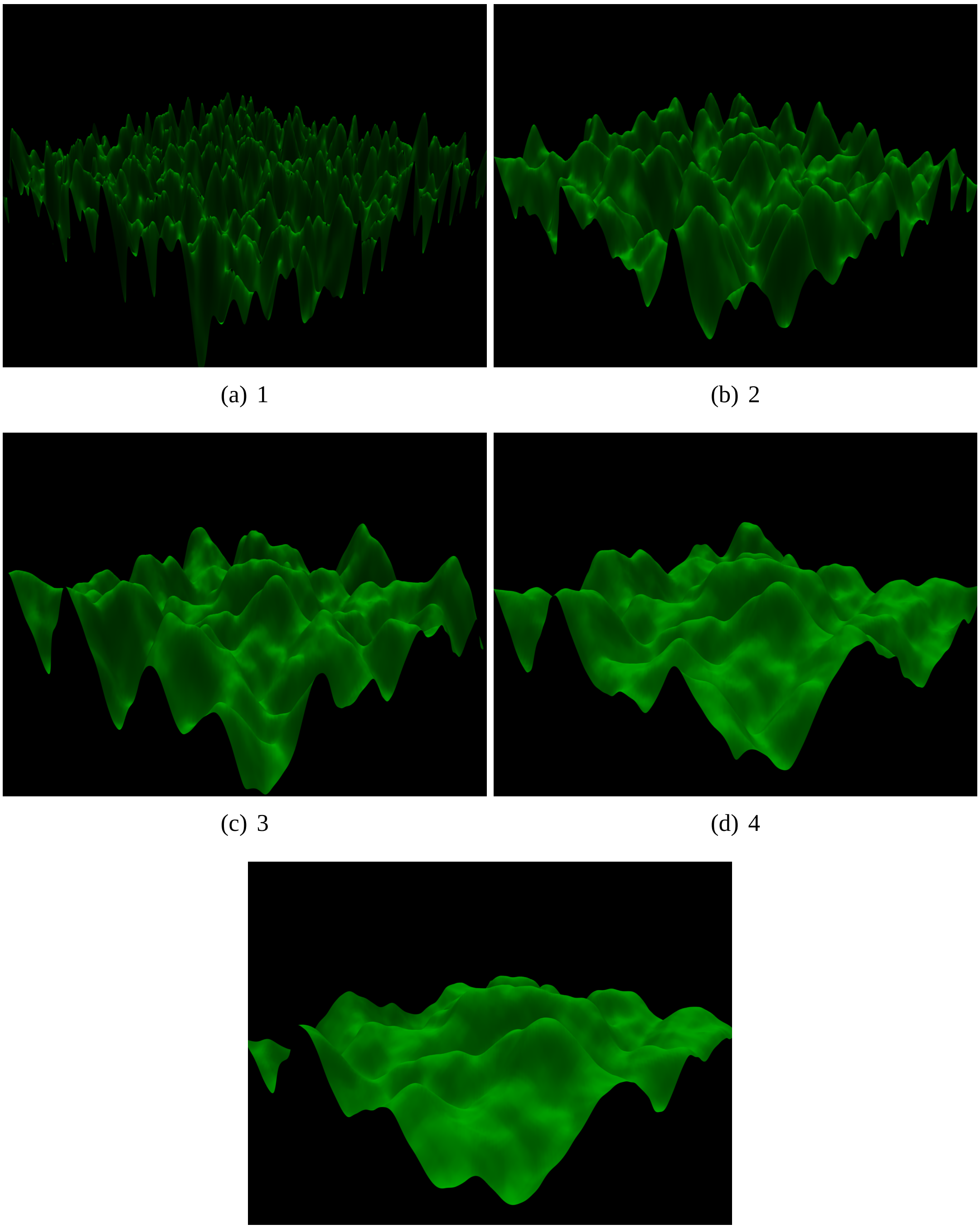
Drafted poster ready.

2.2 Visual Analysis of fBm Parameters

The effects of changing the fBm generator needed to be visually analyzed in order to determine how to proceed with feature generation and interpolation with the remainder of the terrain. The results of this are documented below and visible in various figures.

2.2.1 Division

This is denoted by "div" in Algorithm 1, and refers to changing to input X and Z parameters since the grid tested with ranges over 20 intervals and needs to be scaled. Higher divisions create more realistic and smoothly varying terrain, refer to Figure 1.



(a) 1

(b) 2

(c) 3

(d) 4

(e) 5

Figure 1: fBm Parameter: Division

2.2.2 Fractal Increment

This is denoted by “H” in Algorithm 1. Setting this parameter to a low value produces very coarse, quickly varying terrain, but still maintains the overall shape. This would be useful for rocky or inaccesible areas in terrain, but generally the higher values produce a more realistic approximation, refer to Figure 2.

2.2.3 Lacunarity

This is denoted by “L” in Algorithm 3 and refers to the gap between successive frequencies. Smaller values for this parameter produce extreme highs and lows and would be suitable for something like grass perhaps, but for terrain higher values are better, refer to Figure 3.

2.2.4 Octaves

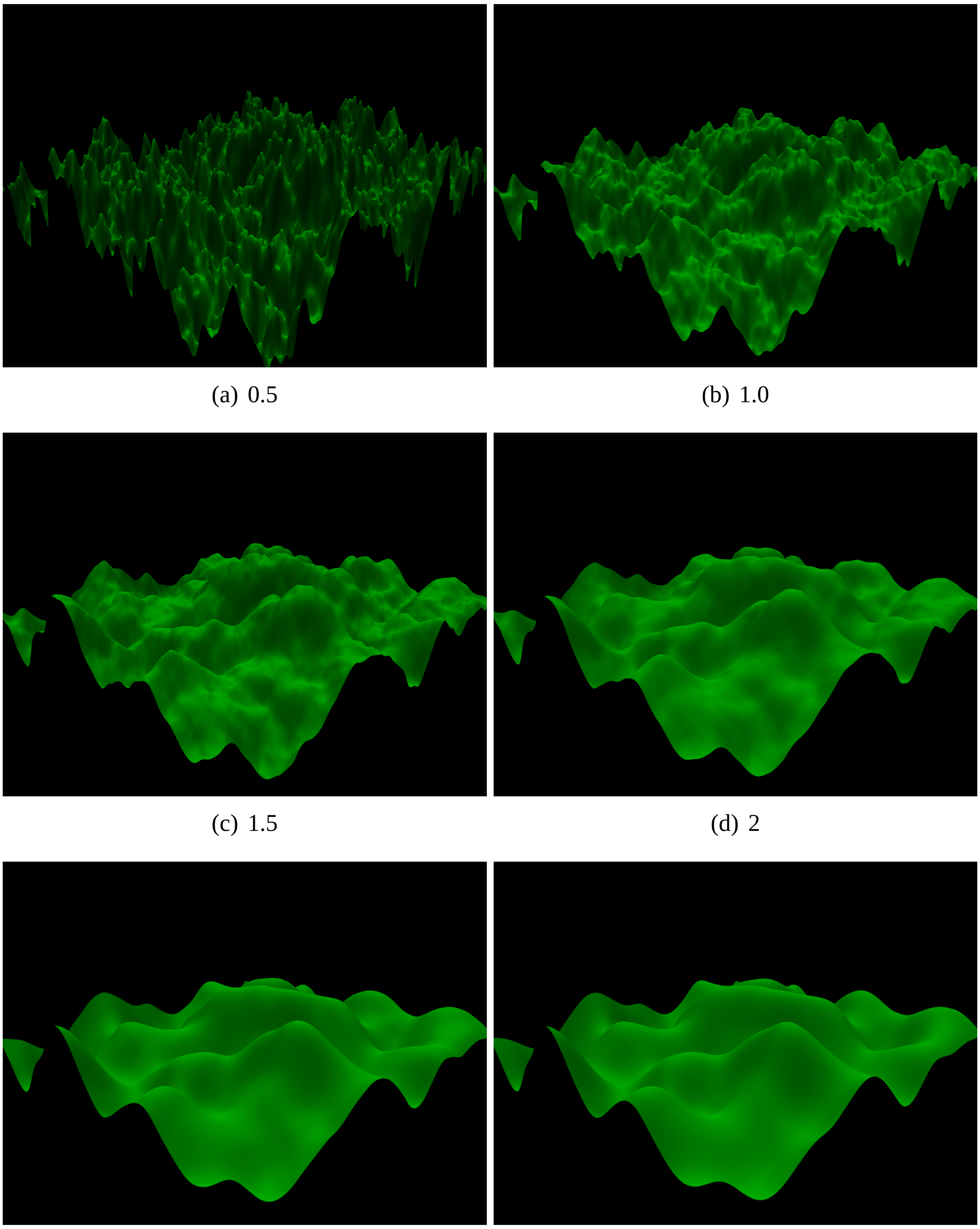
This is denoted by “O” in Algorithm 1 and refers to the number of frequencies used. This parameter is easily visible, in that higher values produce more defined terrain. Refer to Figure 4.

2.2.5 Height

This is denoted by “height” in Algorithm 1 and refers to the scaling of the end value, because fBm ranges between -1.0 and 1.0. Low values for this parameter look like waves, and could be good for smoothing from a beach or river region into a flat terrain. Refer to Figure 5.

2.2.6 Noise Type (NoiseType)

This is denoted by “NoiseType” in Algorithm 1 and refers to the base noise function used to generate the frequencies of the fBm. All three tested look suitable for creating terrain, and the New Perlin method is the one commonly used in this project. Refer to Figure 6.



(a) 0.5

(b) 1.0

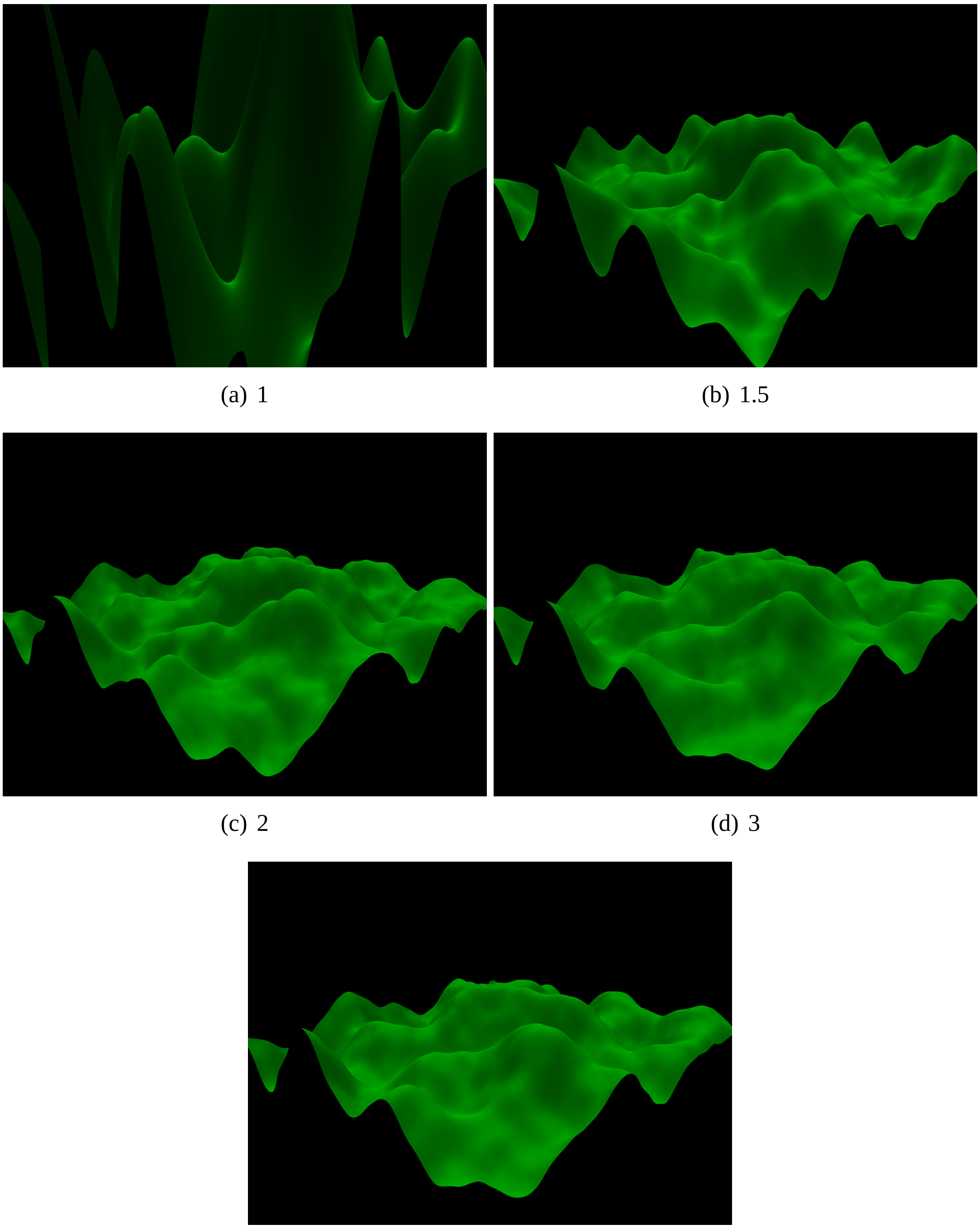
(c) 1.5

(d) 2

(e) 3

(f) 4

Figure 2: fBm Parameter: Fractal Increment



(a) 1

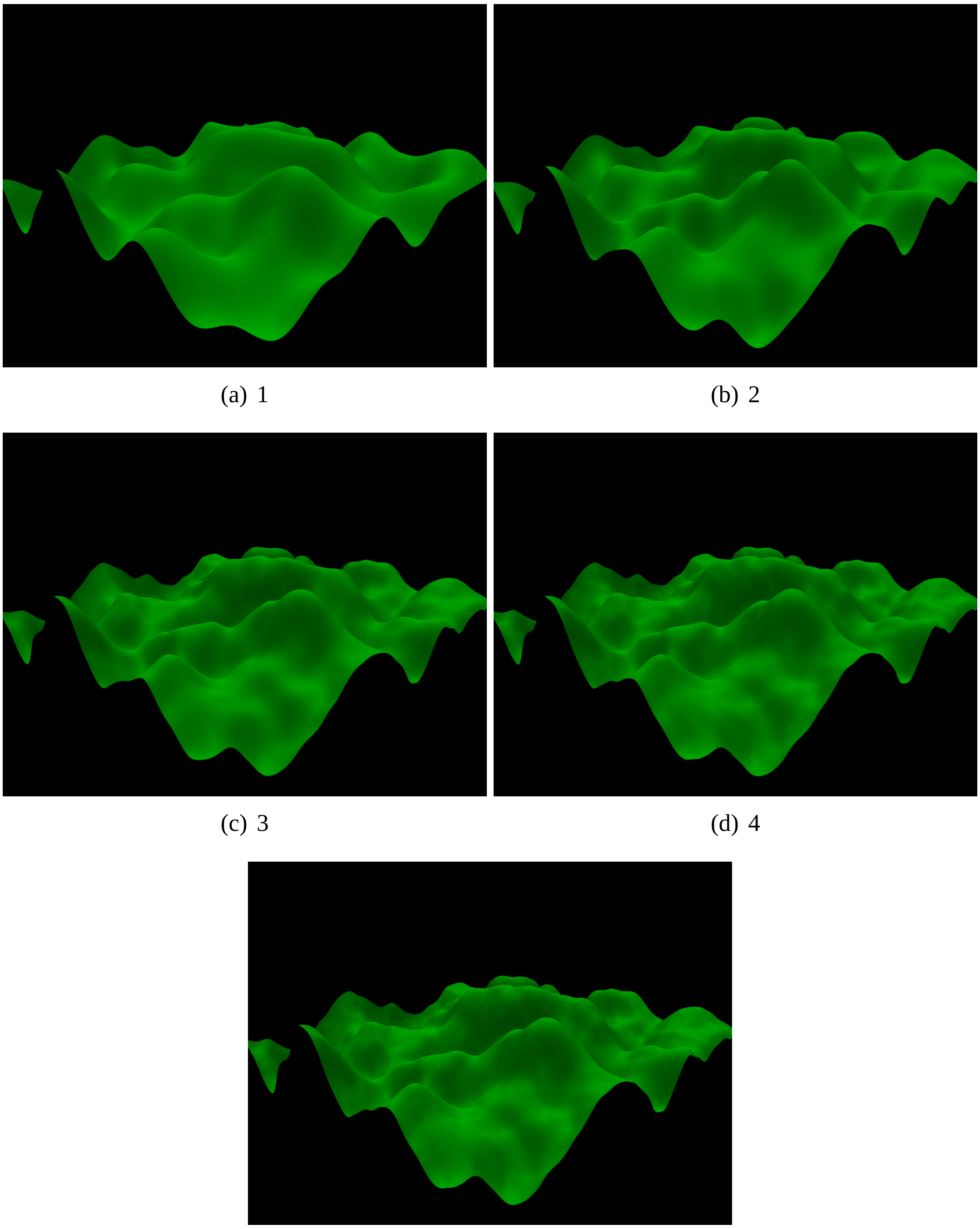
(b) 1.5

(c) 2

(d) 3

(e) 4

Figure 3: fBm Parameter: Lacunarity



(a) 1

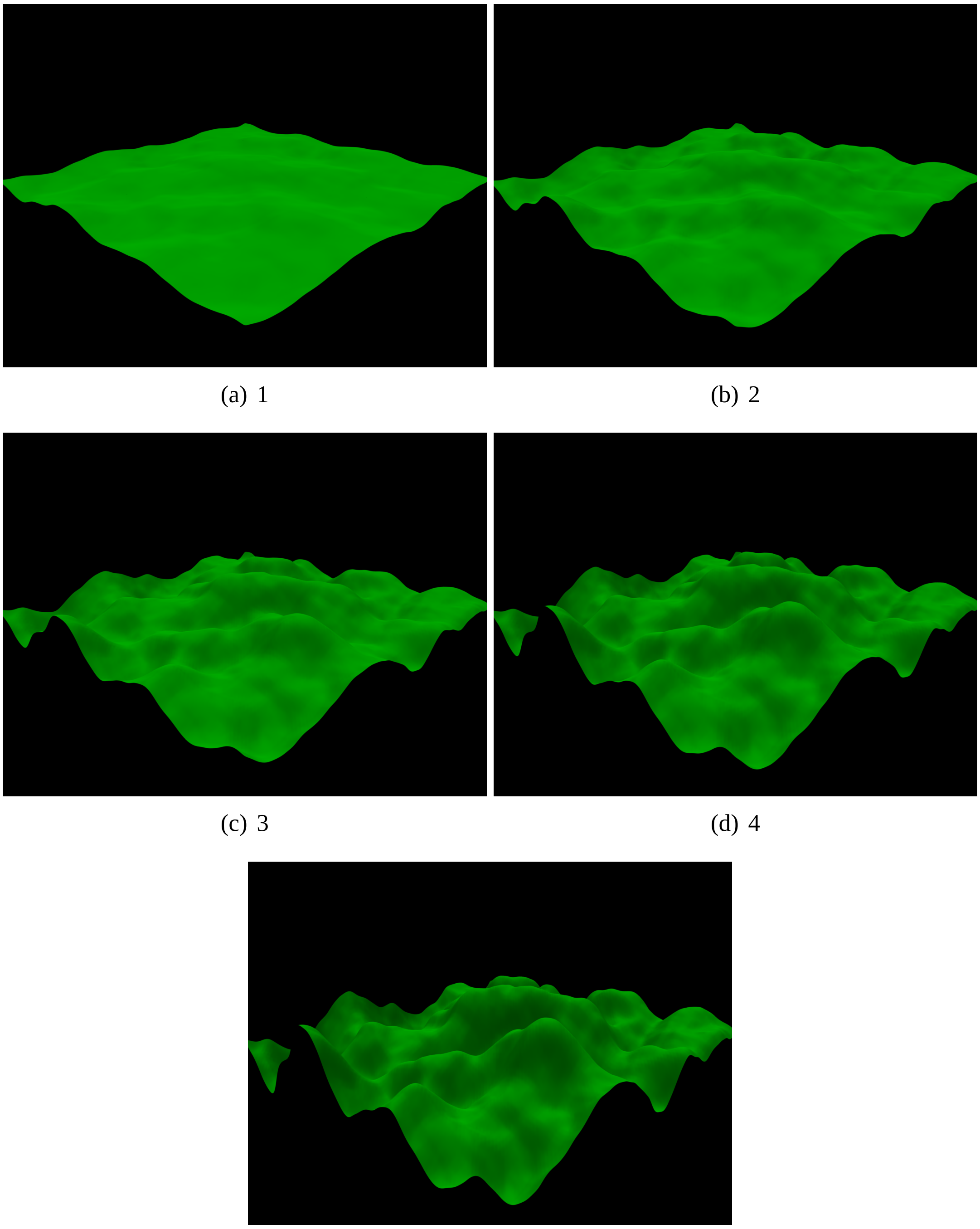
(b) 2

(c) 3

(d) 4

(e) 5

Figure 4: fBm Parameter: Octaves



(a) 1

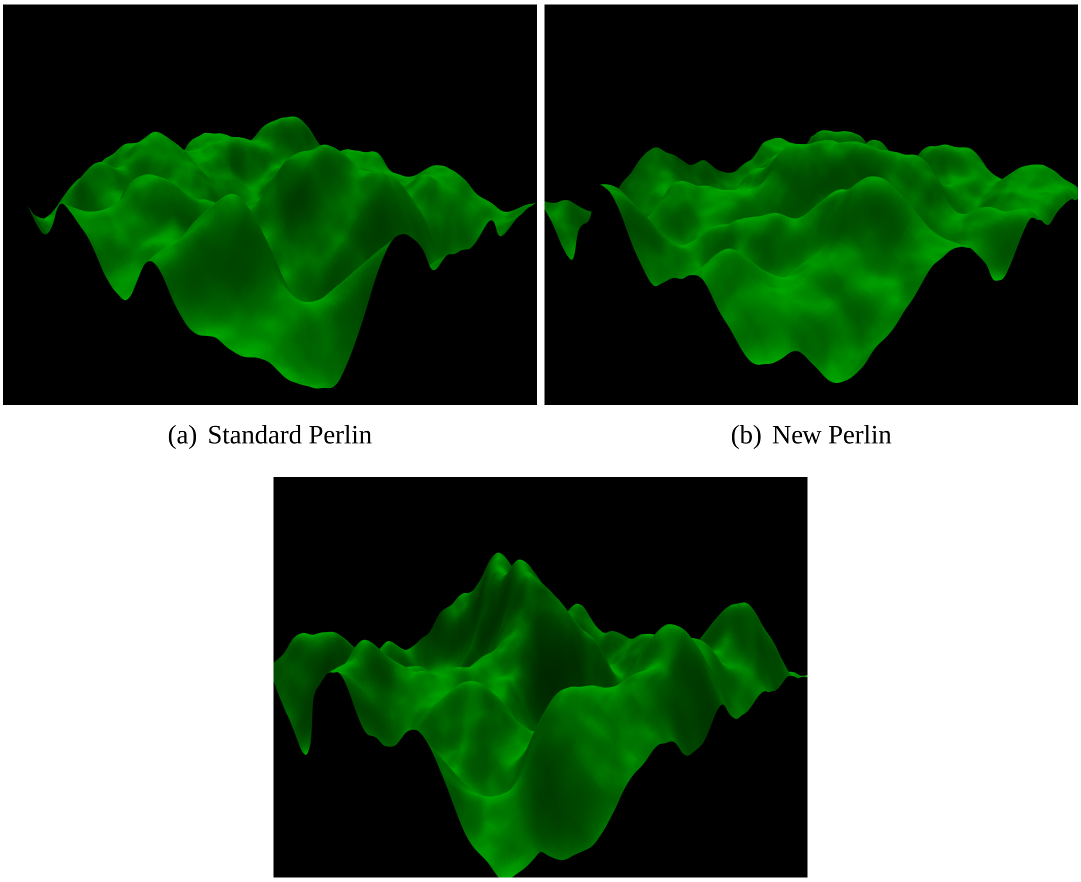
(b) 2

(c) 3

(d) 4

(e) 5

Figure 5: fBm Parameter: Height



(a) Standard Perlin

(b) New Perlin

(c) Blender

Figure 6: fBm Parameter: Noise Type

Algorithm 1 fBm Terrain Creation

```
xyz = (x*distX-distX*xIter/2)/div, 0.0, (z*distZ-distZ*xIter/2)/div
arr [x][z] = (Blender.Noise.fBm(xyz, H, L, O, NoiseType)*height, arr[x][z][1], arr[x][z][2], arr[x][z][3])
```

3 Problems

4 Objectives for Next Week

Squig curve implementation to generate river paths.

References

[1] Przemyslaw Prusinkiewicz and Mark Hammel. A fractal model of mountains with rivers. In *Proceeding of Graphics Interface '93*, pages 174–180, Toronto, Ontario, May 1993.