

Perlin Noise

Creating Procedural Solid Textures

Student: Alexandre Chapiro, Advisor: Luiz Velho

Topics

Topics

Introduction

Theory

Some examples

Software

More examples!

- Solid textures.
- Perlin Noise.
- What can we do with all this?

Introduction

Topics

Introduction

Theory

Some examples

Software

More examples!

→ What is Perlin Noise?

- Perlin Noise is a method used to develop several interesting procedural textures:
- Clouds / Water / Fire / some materials (marble, wood) ...
- ... and much more!

Theory

Topics

Introduction

Theory

Some examples

Software

More examples!

→ What is Perlin Noise?

- Perlin Noise is a mapping from \mathbb{R}^n to \mathbb{R} .
- First, create a grid of points with integer coordinates,
- For each point, find it's closest neighbours on this grid,
- Use a hash table to assign each of the grid points a gradient vector, which defines a linear function,
- Weight these functions using an S shaped polynomial (currently $6x^5 - 15x^4 + 10x^3$),
- Sum them all, and the result is Perlin Noise.

Theory

Topics

Introduction

Theory

Some examples

Software

More examples!

→ What does it look like?

- Perlin Noise alone isn't very useful.
- We can, however, combine it with several other functions!
- For instance, $\sum(1/f) * (\text{noise}(f * p))$ is called turbulence, and can be used to generate lots of interesting effects.



↑ This

Some examples

Topics

Introduction

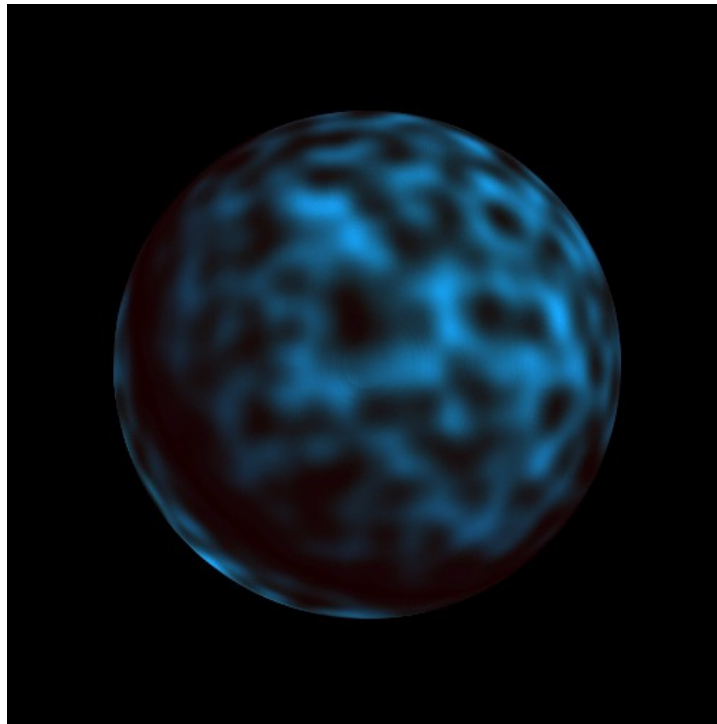
Theory

Some examples

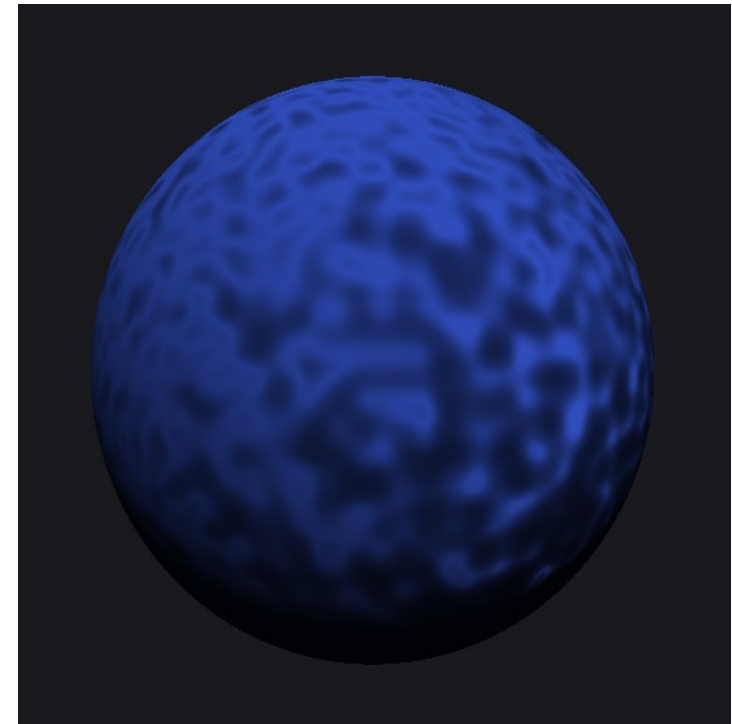
Software

More examples!

→ Some examples from Perlin's "Making Noise" talk (and my attempts at replicating them):



→Original Image



→My example

A water ball with ripples

Topics

Introduction

Theory

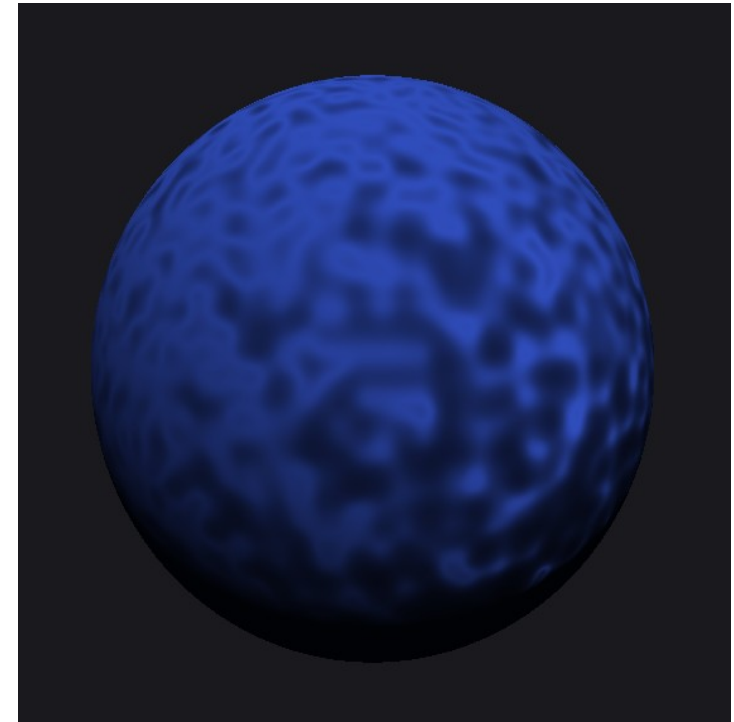
Some examples

Software

More examples!

→ How was it obtained?

- Only bump mapping, no color variation;
- The normal vector at each point 'p' is obtained through:
$$n += \text{pnoise}(p * \text{parameter});$$
- 'n' is the original normal;
- Parameter is chosen big enough for a pleasing effect. (more on this later)



Some examples

Topics

Introduction

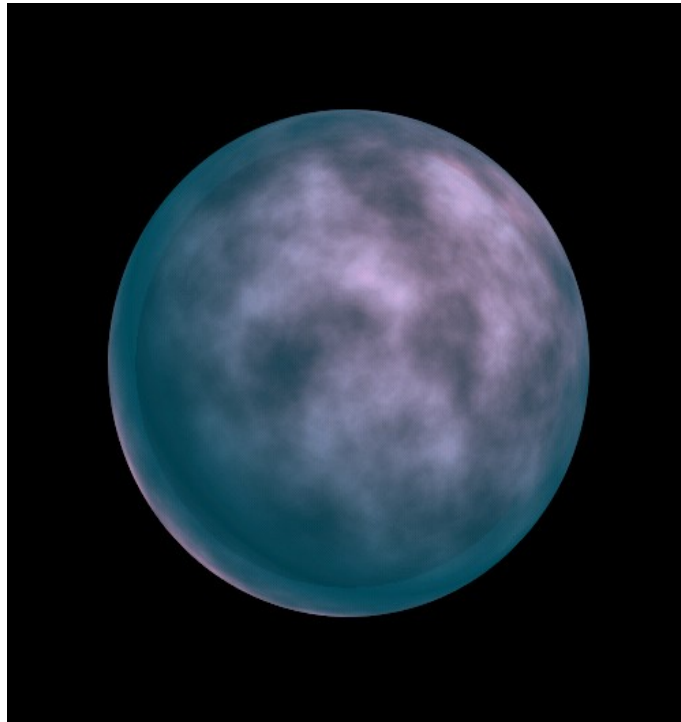
Theory

Some examples

Software

More examples!

→ Another example from Perlin's "Making Noise" talk:



→Original Image



→My example

A green-ish planet with clouds

Topics

Introduction

Theory

Some examples

Software

More examples!

→ How was it obtained?

- Outside of the ball brightened, to make it look like a clearer atmosphere around a darker landmass;
- Planet's color is static, with clouds added to the original color;
- Cloud's color obtained with:
`c += turbulence (8, p);`
- 8 is the amount of octaves;



Some examples

Topics

Introduction

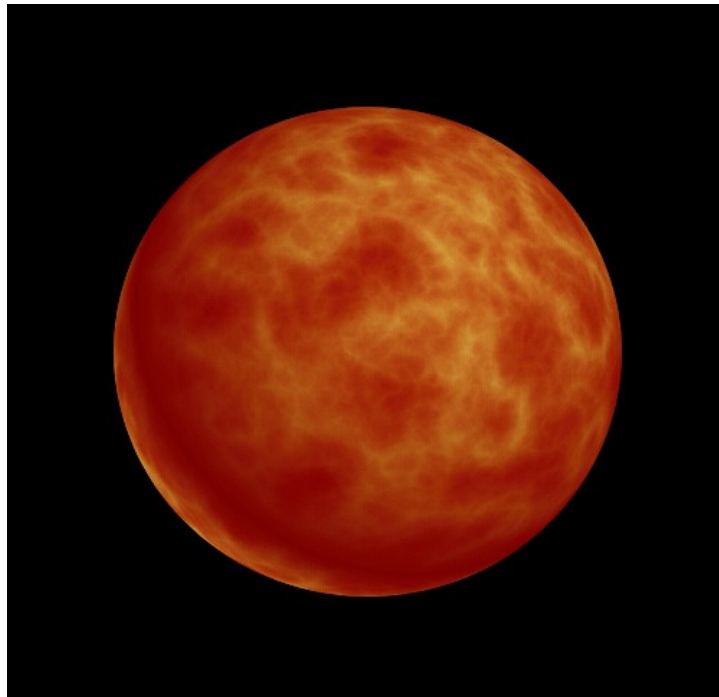
Theory

Some examples

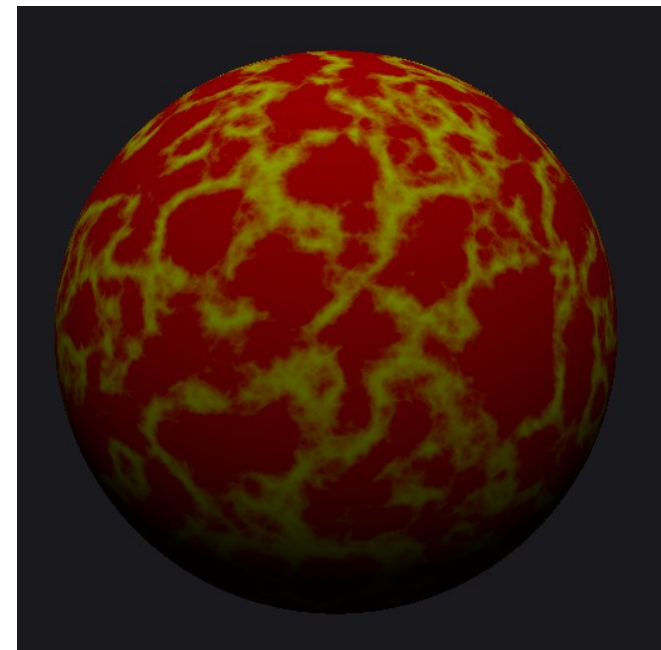
Software

More examples!

→ Yet another example from Perlin's "Making Noise" talk:



→Original Image



→My example

Flame ball

Topics

Introduction

Theory

Some examples

Software

More examples!

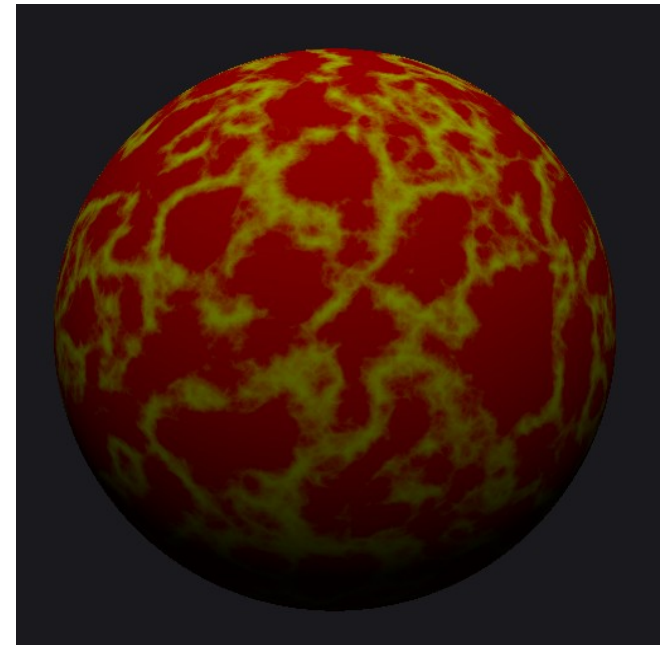
→ How was it obtained?

- Initial colour set as bright orange (lots of R and G);

- The color is then obtained by keeping R the same and varying G:

```
G -= 4* [turbulence (15,  
parameter*p,  
WITH_ABSOLUTE_VALUE)]^2;
```

- 15 octaves, turbulence with absolute value, ^2 for thinner lines.



Some examples

Topics

Introduction

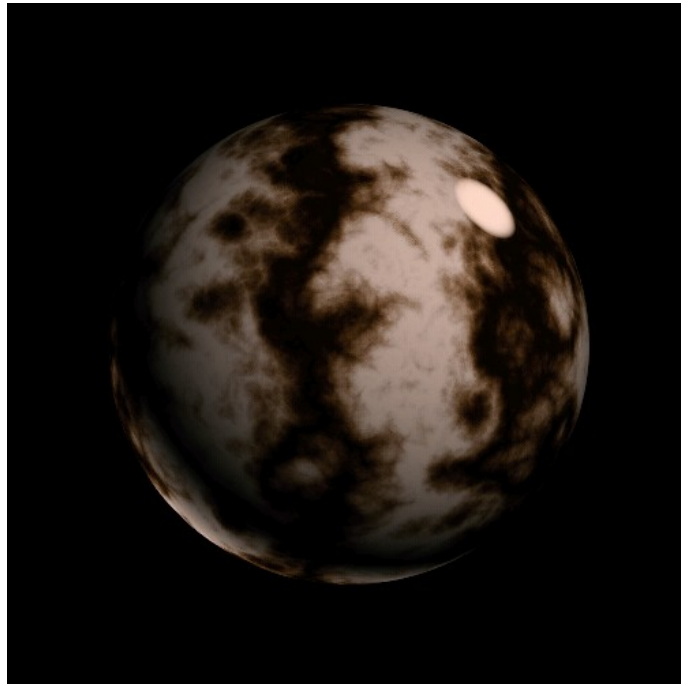
Theory

Some examples

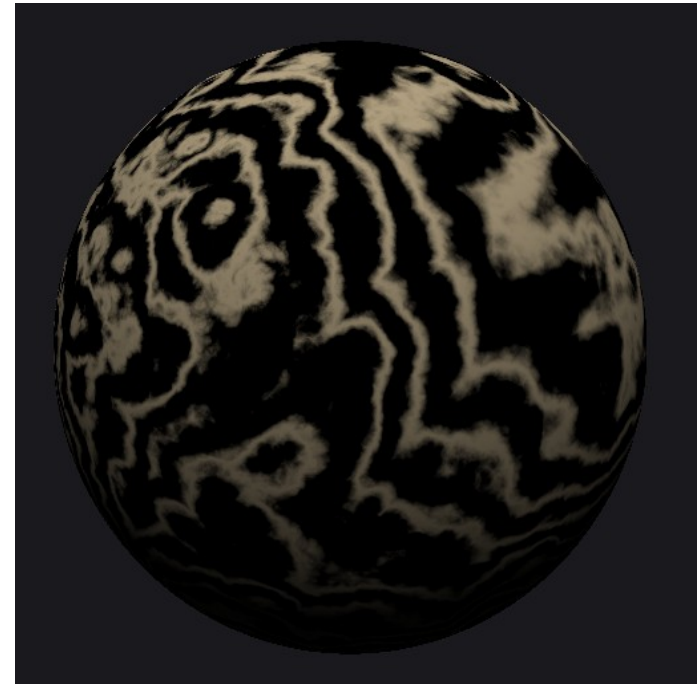
Software

More examples!

→ One last example from Perlin's "Making Noise" talk:



→Original Image



→My example

Marble

Topics

Introduction

Theory

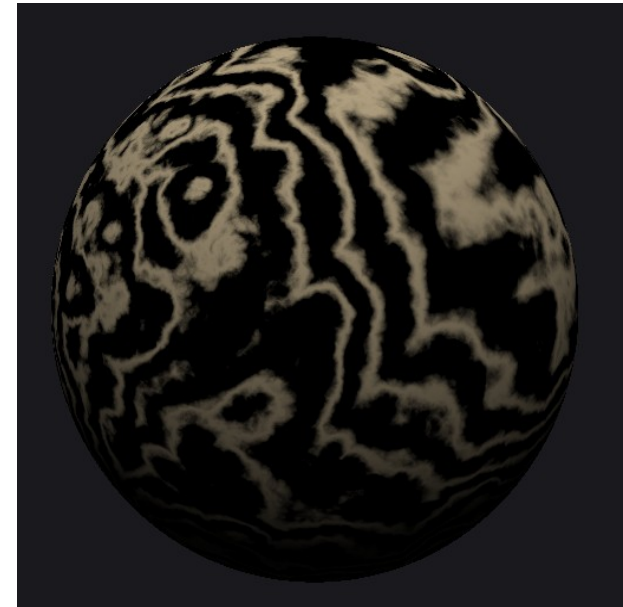
Some examples

Software

More examples!

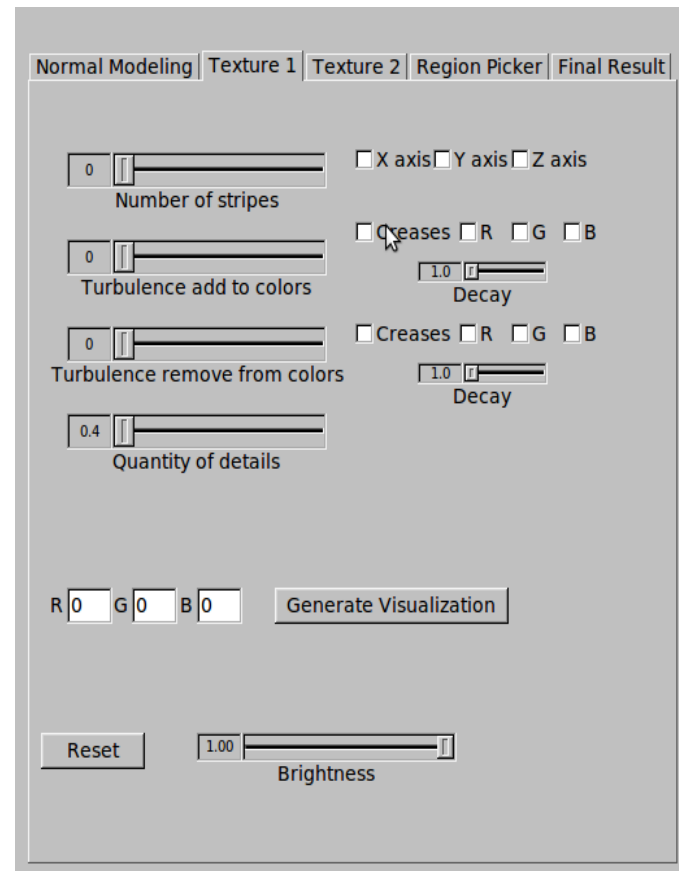
→ How was it obtained?

- Initial colour set as “hay”, then multiplied by 'f';
- `K = turbulence(15, p, WITH_ABSOLUTE_VALUE);`
`C = 'hay'* sin(4*PI*(p.x + 2*k^2));`
- 15 octaves, turbulence with absolute value, $\wedge 2$ for thinner lines, $4*PI$ sets the ammount of oscillation of the 'sin' function (black x colored lines).



The program.

→ I've created an interactive program that allows the user to create his own textures using Perlin Noise / Turbulence.



Normal editing.

Topics

Introduction

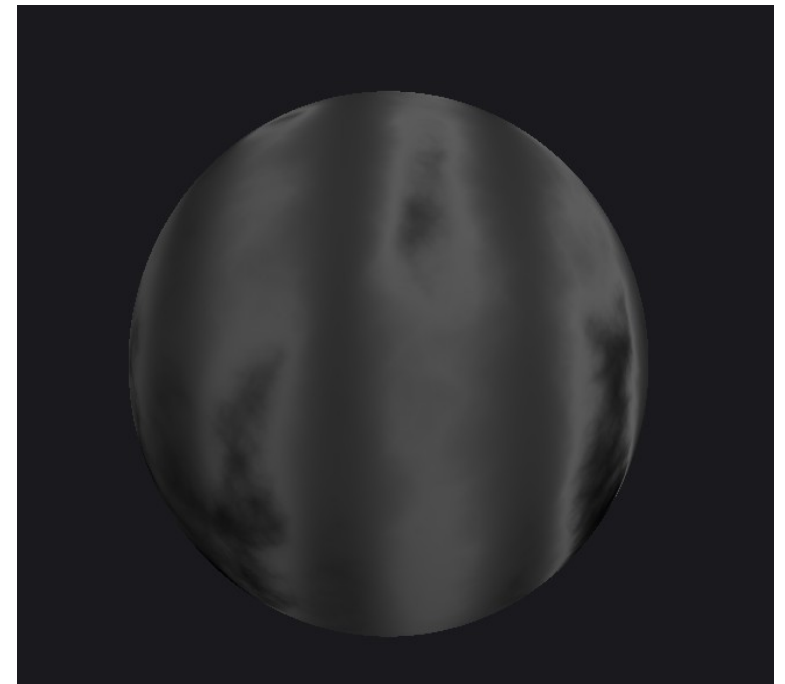
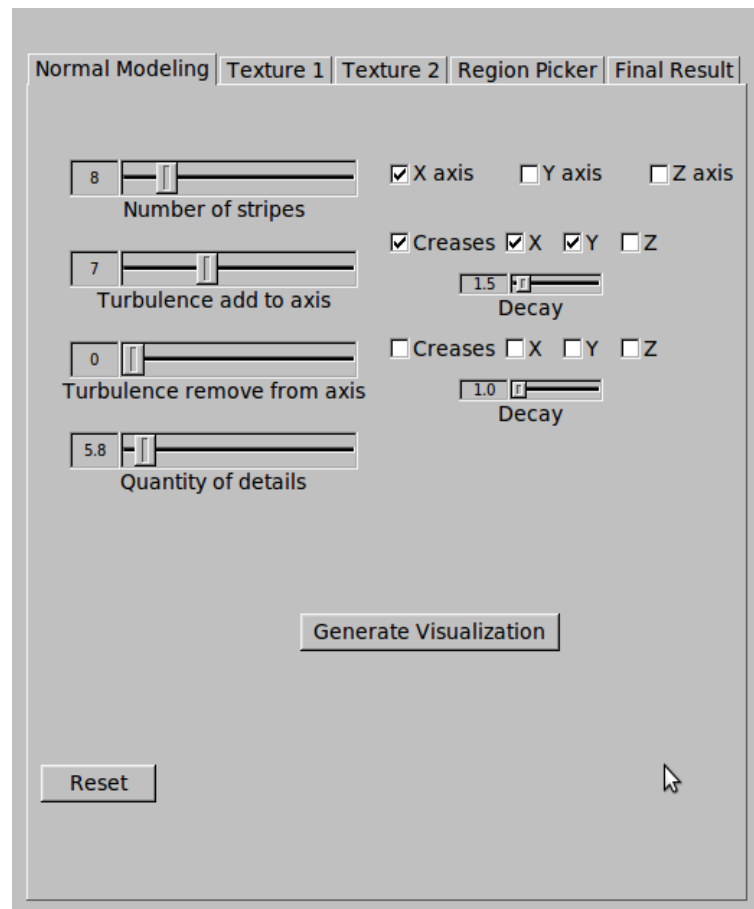
Theory

Some examples

Software

More examples!

→ The first tab allows the user to create bump maps using different kinds of Turbulence, stripes, and several possibilities.



Texture creation.

Topics

Introduction

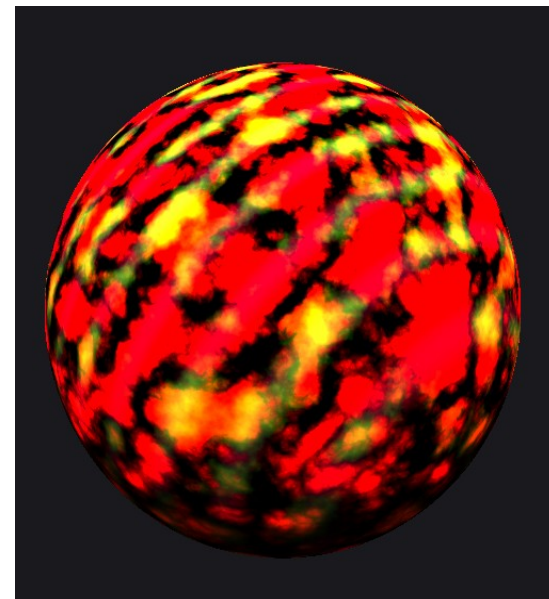
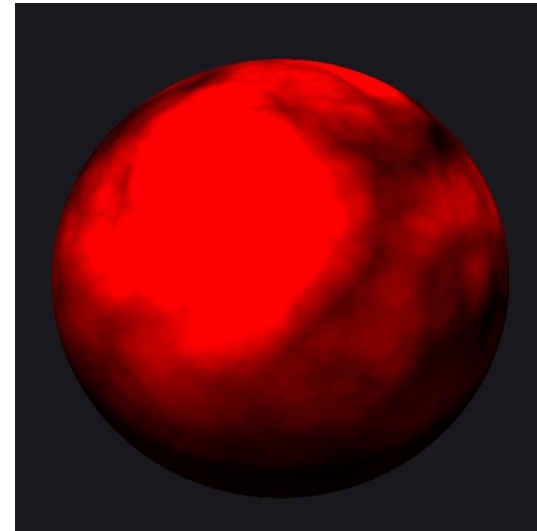
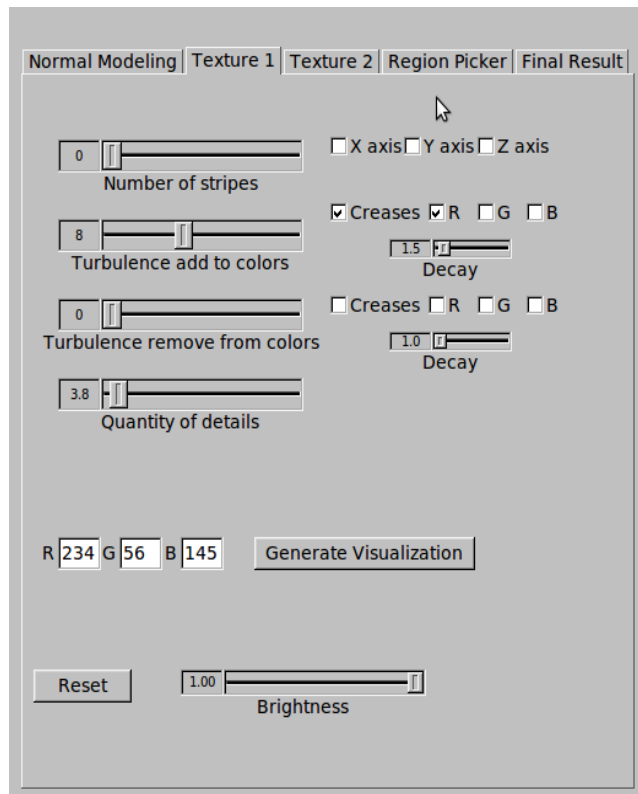
Theory

Some examples

Software

More examples!

→Users can create their own textures.
Some examples:



Defines Regions.

→The program also allows the user to create a division on the surface using turbulence or stripes...

Topics

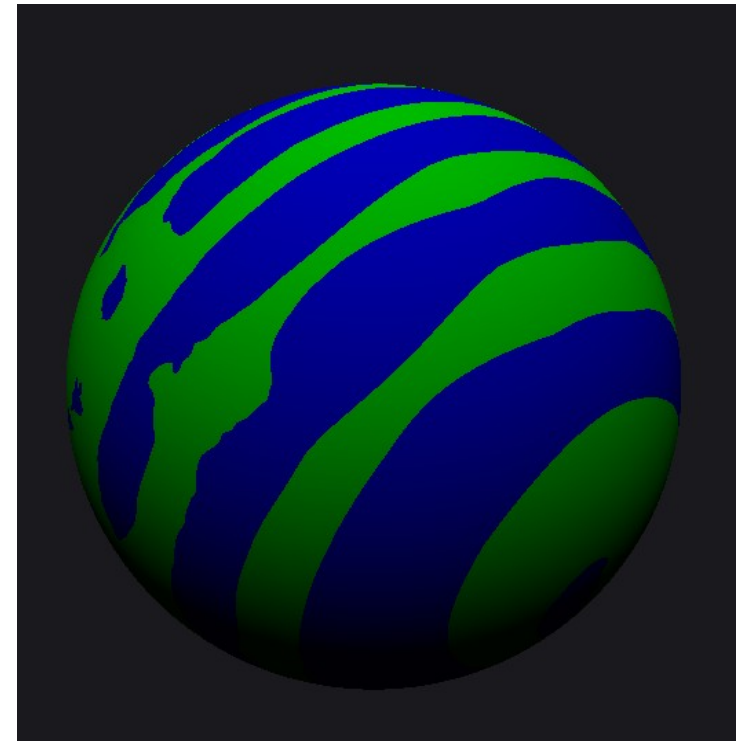
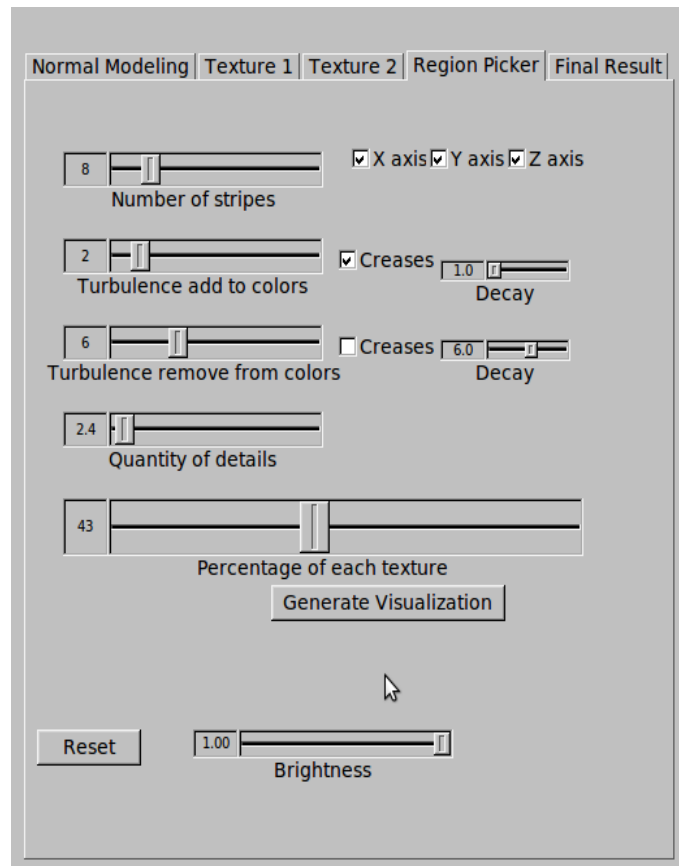
Introduction

Theory

Some examples

Software

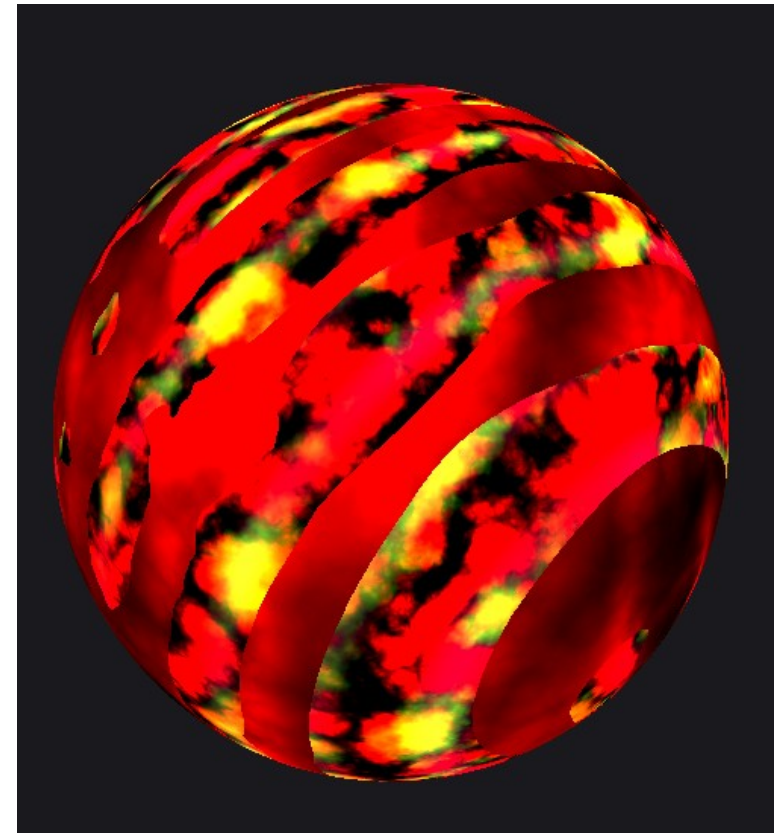
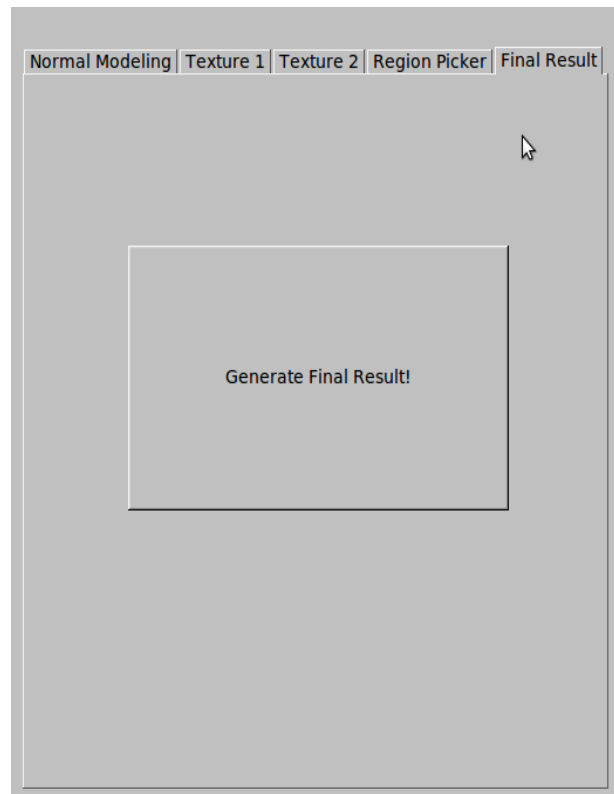
More examples!

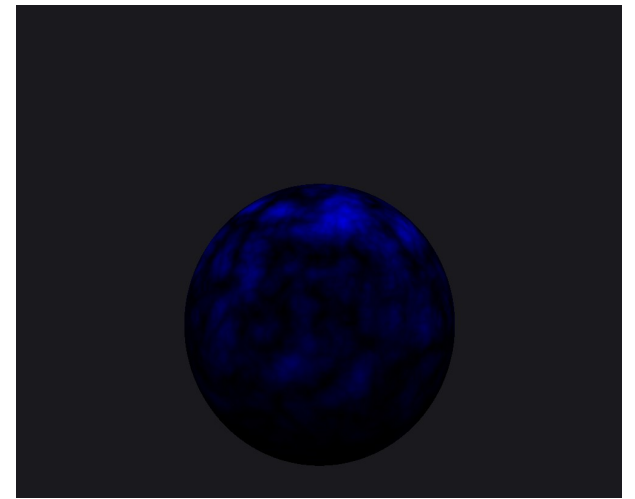
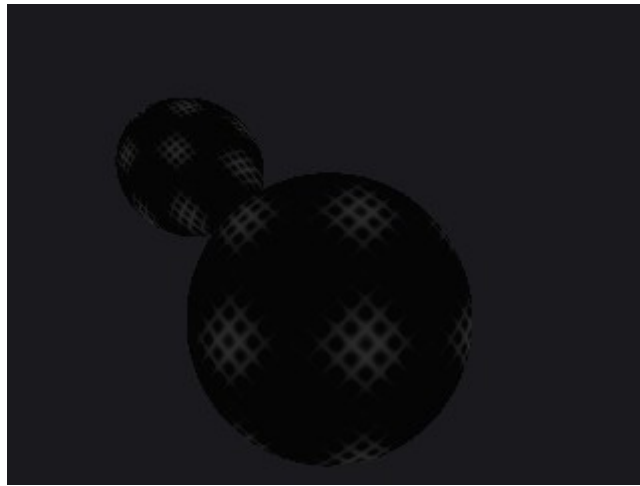


The final result!

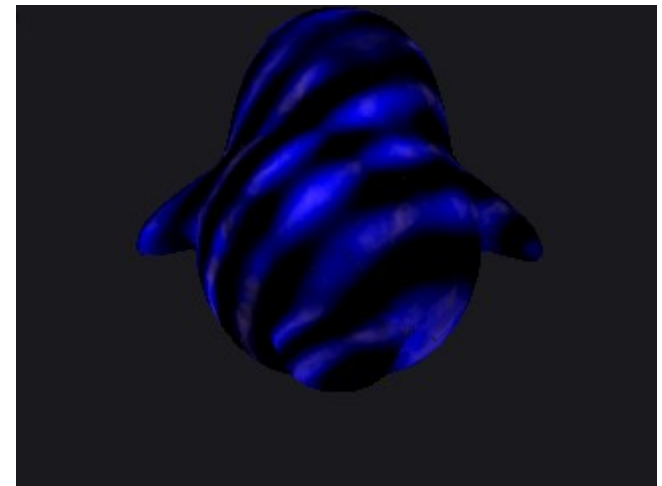
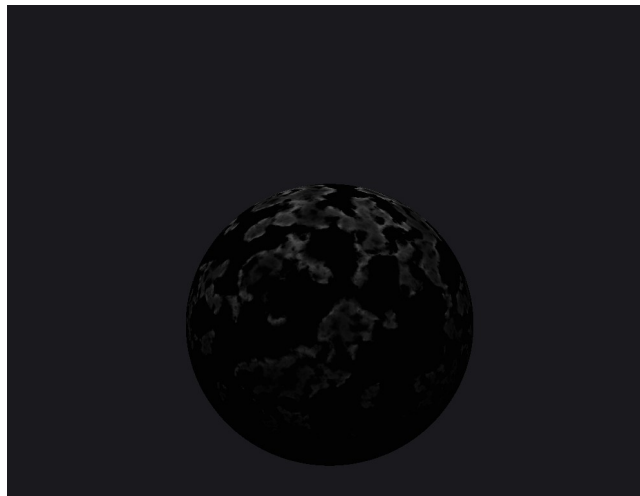
→ Finally, you can create an image that contains the respective textures in the created regions, on a sphere with the bump map you used before!

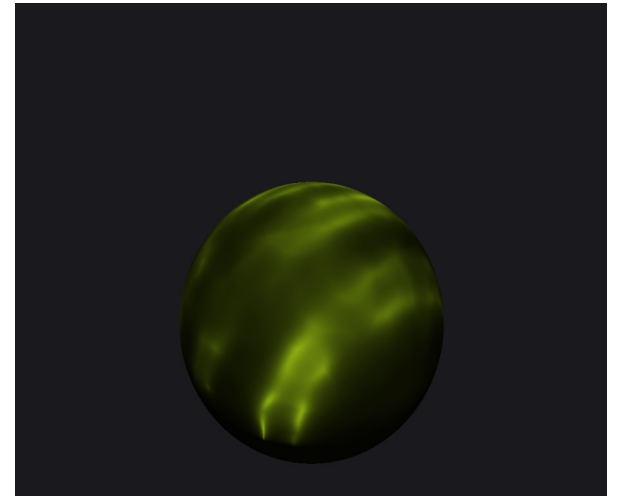
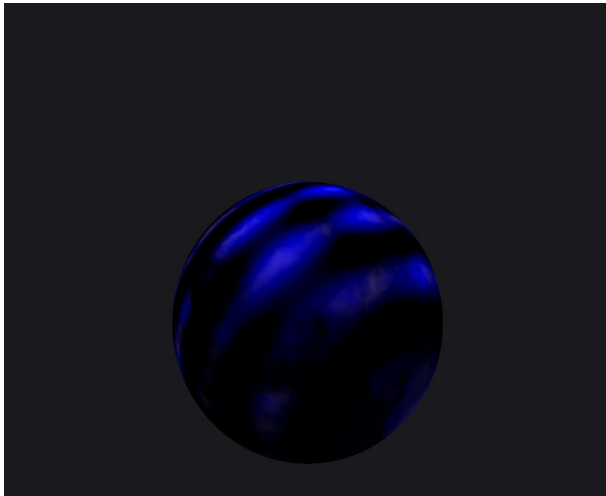
Topics
Introduction
Theory
Some examples
Software
More examples!



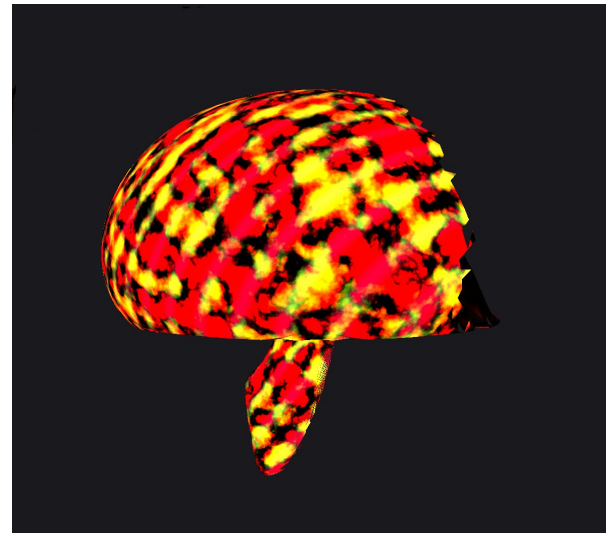
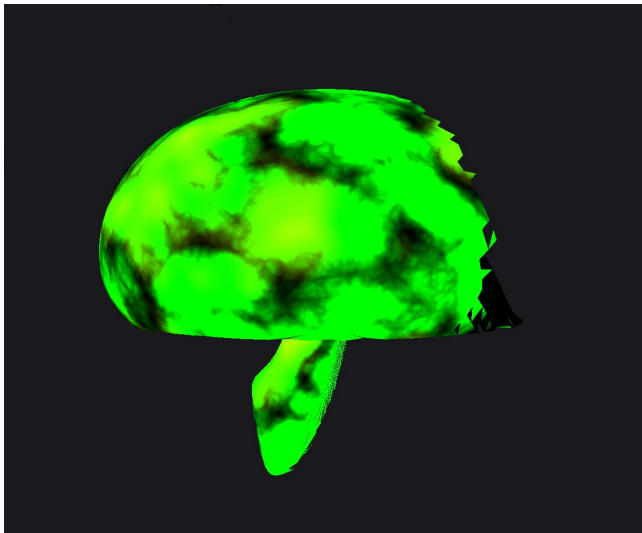


More results!





Few more results!



References:

Papers:

- 1) Ken Perlin, “An Image Synthesizer”, SIGGRAPH Volume 19, number 3, July 1985.
- 2) Ken Perlin, “Improving Noise”, Proceedings of ACM SIGGRAPH 2002.
- 3) A. Lagae et al., “State of the Art in Procedural Noise Functions”, EUROGRAPHICS, STAR – State of The Art Report, 2010.

Talks:

- Ken Perlin, “Making Noise”, <http://www.noisemachine.com/talk1>

Thank You!