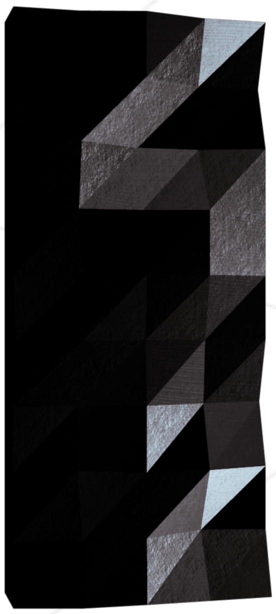

Algorithmic project

mod1

Staff pedago pedago@42.fr

Summary: This project is a water flow simulator on a surface.

Version: 1

Contents

Chapter I

Foreword

A little bit of hydrostatic
For once, it isn't the Mythbusters who have created this experiment but the Frenchies from "We are not only guinea pigs". Pascal's Principle (Basically an XVIIth century French scientist that you have never heard of but used to be so famous that, way back then, had his face on France's biggest bank note) so Pascal's Principle is a great example of hydrostatic pressure. It is defined by the height of a given water container, of its surface on the ground, and of the earth's gravity. The famous formula $p = S.h.g$ allows this calculation. It is important to note that it is incomplete: the volume mass of the fluid considered is usually to be taken into account and to be added to the equation, here simplified and approximated since the density of water is in fact 0,99997 and not 1. But go have a look at this fun experience. Since the video is in french and you don't speak French here is a scientific explanation of Pascal's principle.

Archimedes was not murdered in his tub
"Eurêka" is not the scream of a dying man, it is in fact in victory that Archimedes screamed when he realized one of the founding principles of the dynamic of fluids: its famous push! It works like that: any body plunged into a liquid (so it gets wet, ok...) will resend a vertical push equal to the weight of the liquid moved around. A body that floats is therefore in a state of balance. Kinda like the second law of Newton (the one from the Da Vinci Code or the apple: same dude) We have $\sum f = m.a$ and that an equilibrium means that there is no acceleration, therefore the sum of the forces is nil. Archimedes' push, which corresponds therefore to the water displaced is equal to $m.g$. Please note also that it is due to the gradient of hydrostatic pressure according to depth, still according to the laws described in the previous chapter.

Cherry it's the Bomb Baby!
The must is the moving water, dancing, swirling waters that change and twirl in the air. Take any confined space, for example the bathroom water pipes of an elementary school (In Springfield for ex), ask Bart to throw a cherry bomb in there or (and it's really safer) ask Jamie to inject 20 psi or 1,38 bars (which can also be called 137 895 Pascal, yup same Pascal). Then you will see water all over the place. Adams made a very sweet video about that.

Reminder: "Don't try this at home".

Chapter II

Introduction

Why *mod1*?

Actually no ones knows why this topic is called like that.

Chapter III

Objectives

This project is about creating a terrestrial simulation followed by a hydrodynamic one.

At first, you will extrapolate from very restricted provided data a surface (more or less realistic) representing a scene. Once that scene is defined, it will be covered with water, in multiple shapes.

Your program will get at least one external file (extensions *.mod1*) as parameter that contains the minimum information you have at your disposal. It is a series of coordinates in 3D. They define a few sparse points through which your scene will imperatively go across. It can be either 5, 10 or 20 points for example. A file containing more than 50 points is off topic. This is not about creating a fdf.

Chapter IV

General Instructions

- This project will be corrected by humans only. You're allowed to organise and name your files as you see fit, but you must follow the following rules.
- The executable file must be named `mod1`.
- The langage for this project is up to you but if you but if you decide to do it in C usual rules apply.
- A `Makefile` or something similar must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.
- If you are clever, you will use your library for your `mod1`. Submit also your folder `libft` including its own `Makefile` at the root of your repository. Your `Makefile` will have to compile the library, and then compile your project.
- Your project must be written in accordance with the Norm. Only norminette is authoritative.
- There are no particular constraints on the tools or the libraries you can use for the drawings.
- Naturally, no library that can do the job for you is allowed: nothing to extrapolate the surface for you, nothing to manage the water falling for you.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- You'll have to submit a file called `author` containing your usernames followed by a '\n' at the root of your repository.

```
$>cat -e author
xlogin$
ylogin$
$>
```

- You can ask your questions on the forum, on slack...

Chapter V

Mandatory part

Your first job is to represent a flat surface that goes through those points. Segment the whole thing in space, such as relief in sand in a transparent cube. The edges of this surface must have an altitude of nil. This surface cannot contain an edge (again we are not making a fdf) bbe rounded. So, use a bit of algo, to define the scale both to define the scale, the limits of your space and to calculate the surface. Methods exists. You must search for them on the Internet...

Your second job is to put water everywhere. The presence of dripping water on the landscape must seem logical and natural. A minimum of three different possibilities must be presented:

- An even rise in water all over the surface, covering little by little the lower terrain and not the higher ones.
- A wave that arrives crashing on one side and slowly submerging the whole thing.
- Rain that floods your landscape slowly.

You need to create a graphic representation of the surface, as well as of the water and its movements. There are no particular constraints on the tools or the libraries you can use for the drawings. The minilibx should be enough but you can also use SDL or use OpenGL if you want (or QT, GTK etc...) Remember that it is an algorithm project. The graphic representation must show the requested functionalities. Quality is good. A super duper interface will get you bonus points.

Chapter VI

Bonus part

It is of course a good idea to create a few bonuses in your mod1. An interface that is blowing us away in 3D, more water dripping scenarios, a flush etc...

All right then! Grab your umbrellas!

Chapter VII

Pretty pictures

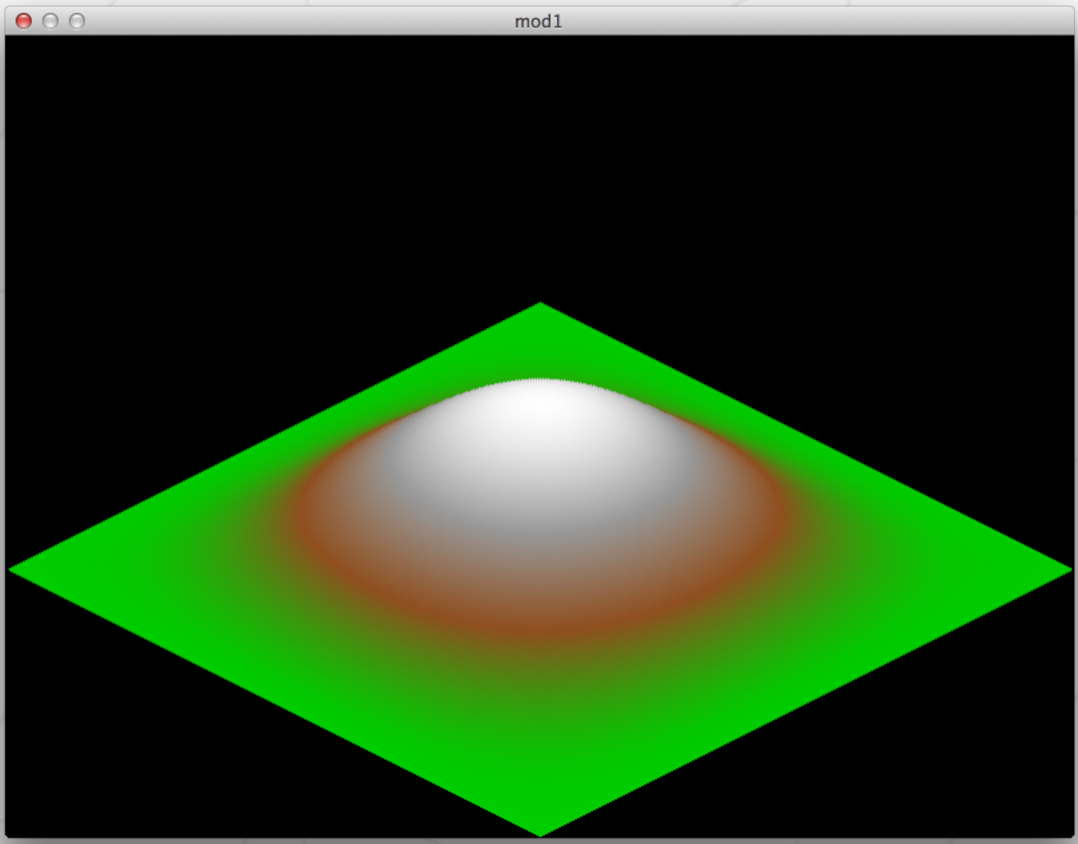

Figure VII.1: A surface

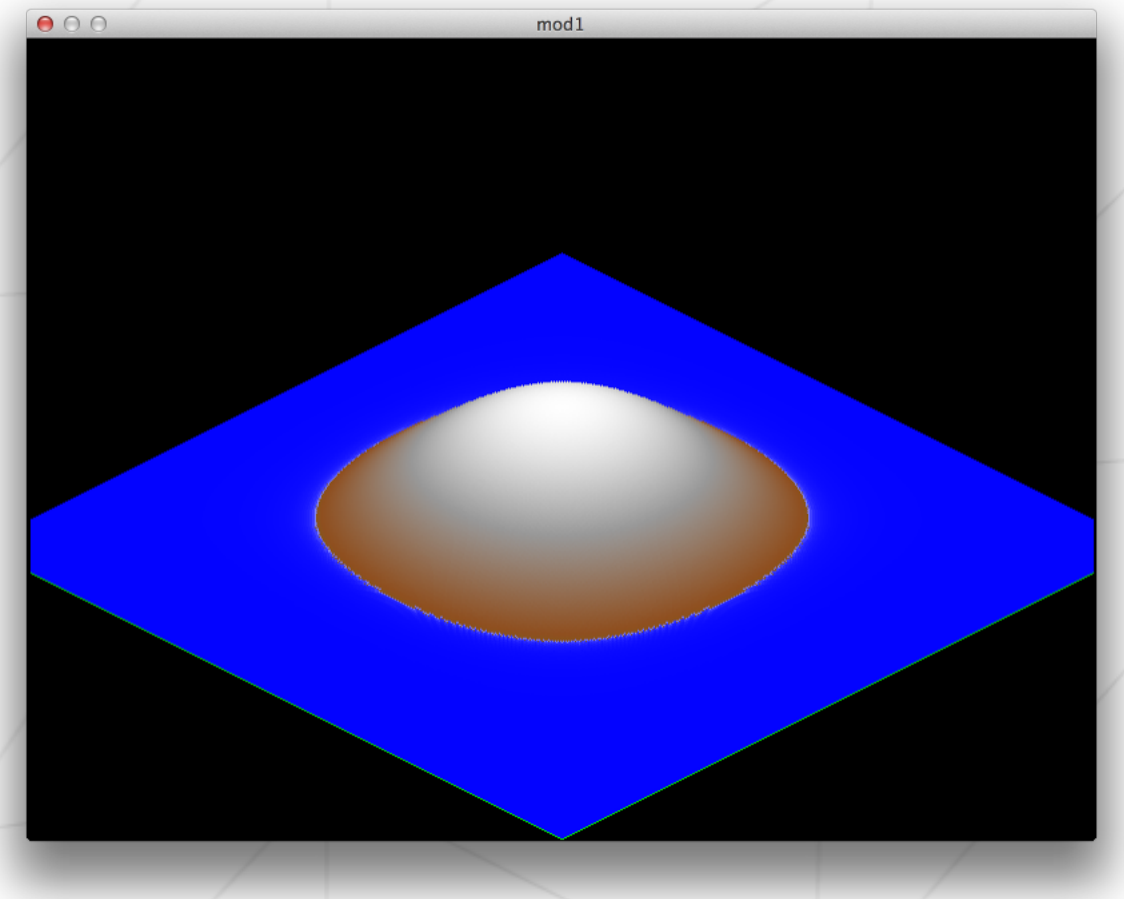

Figure VII.2: A surface with water

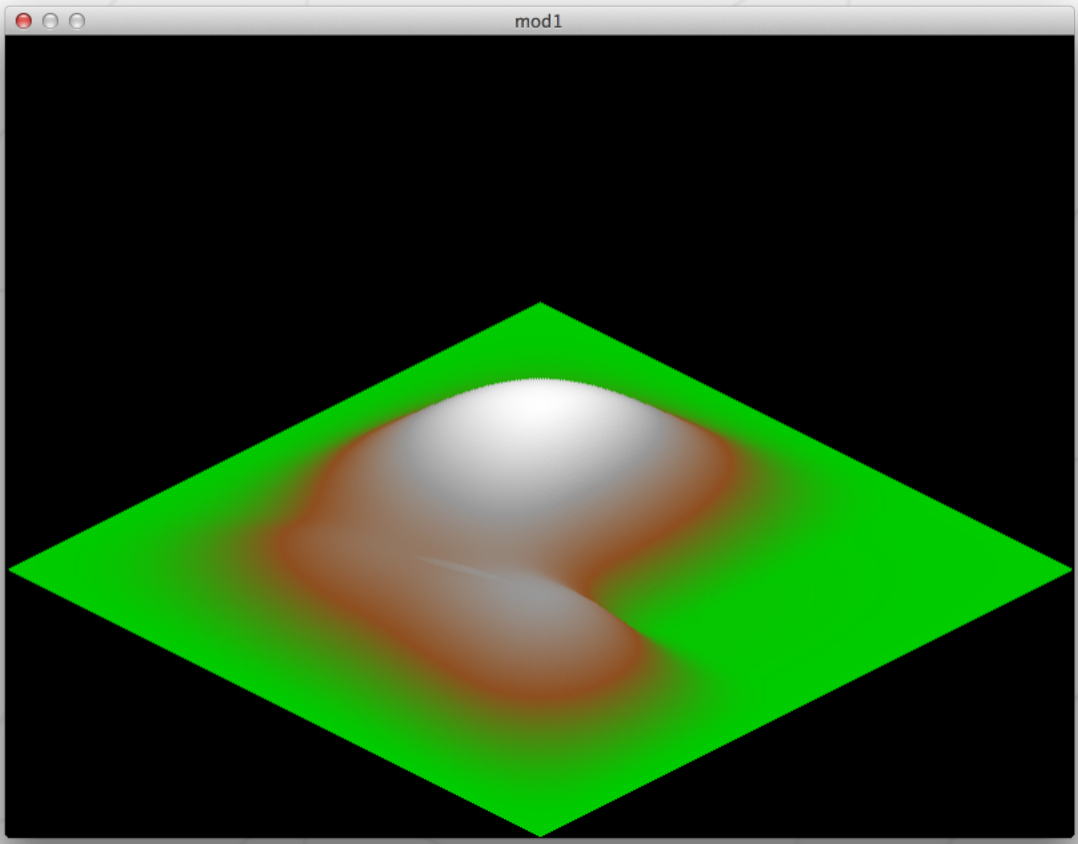

Figure VII.3: A potatoe-like island...

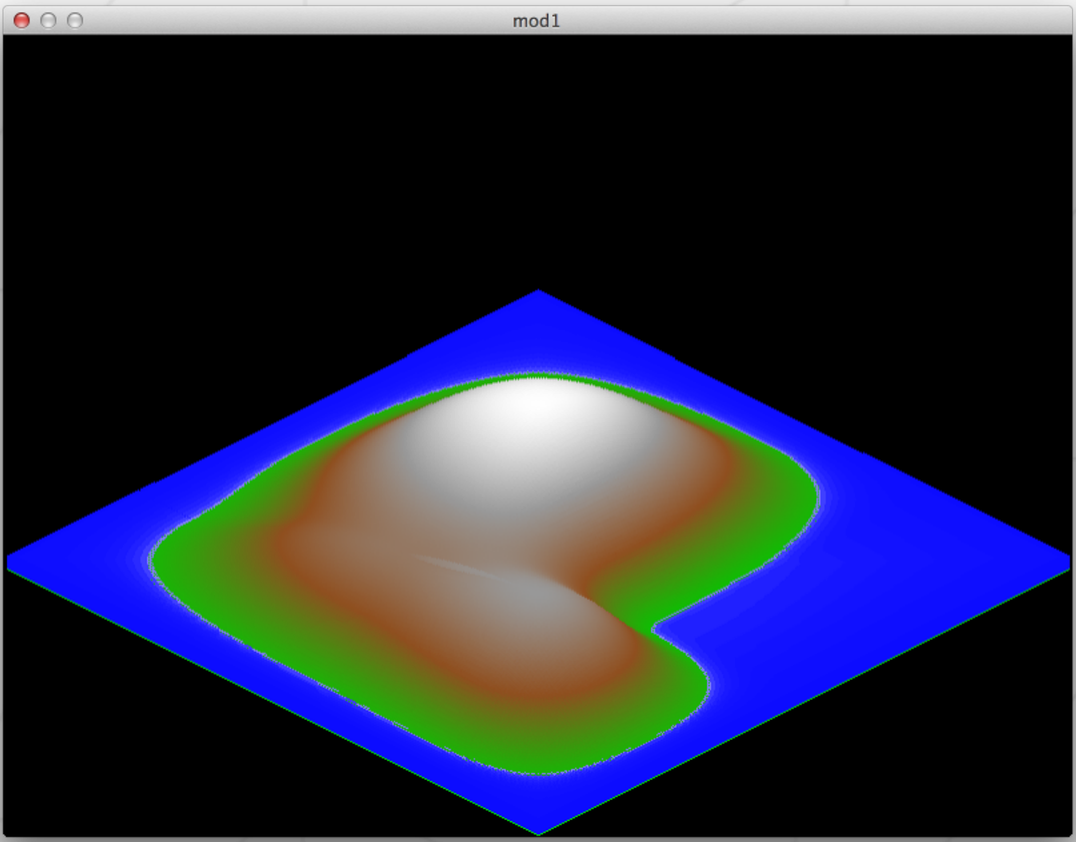

Figure VII.4: ...with a little bit of water...

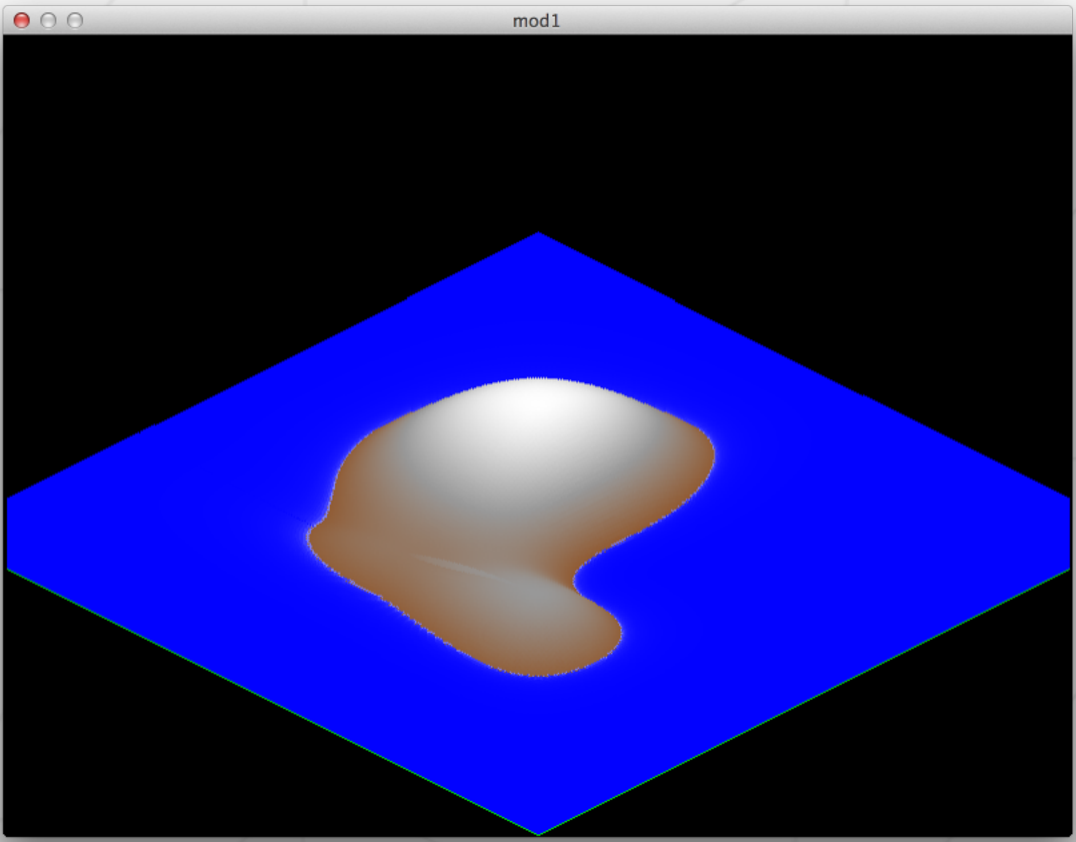

Figure VII.5: ...or a lot of water

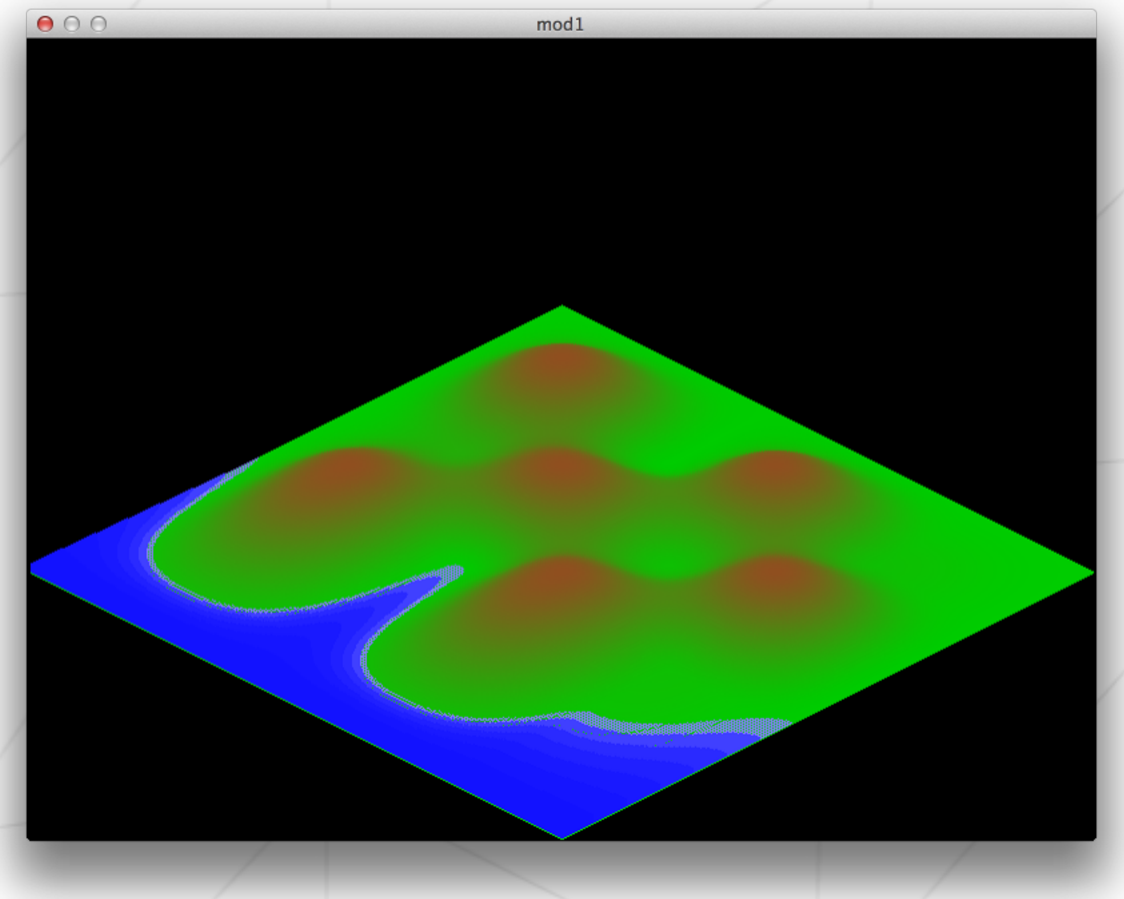

Figure VII.6: A wave over the hills

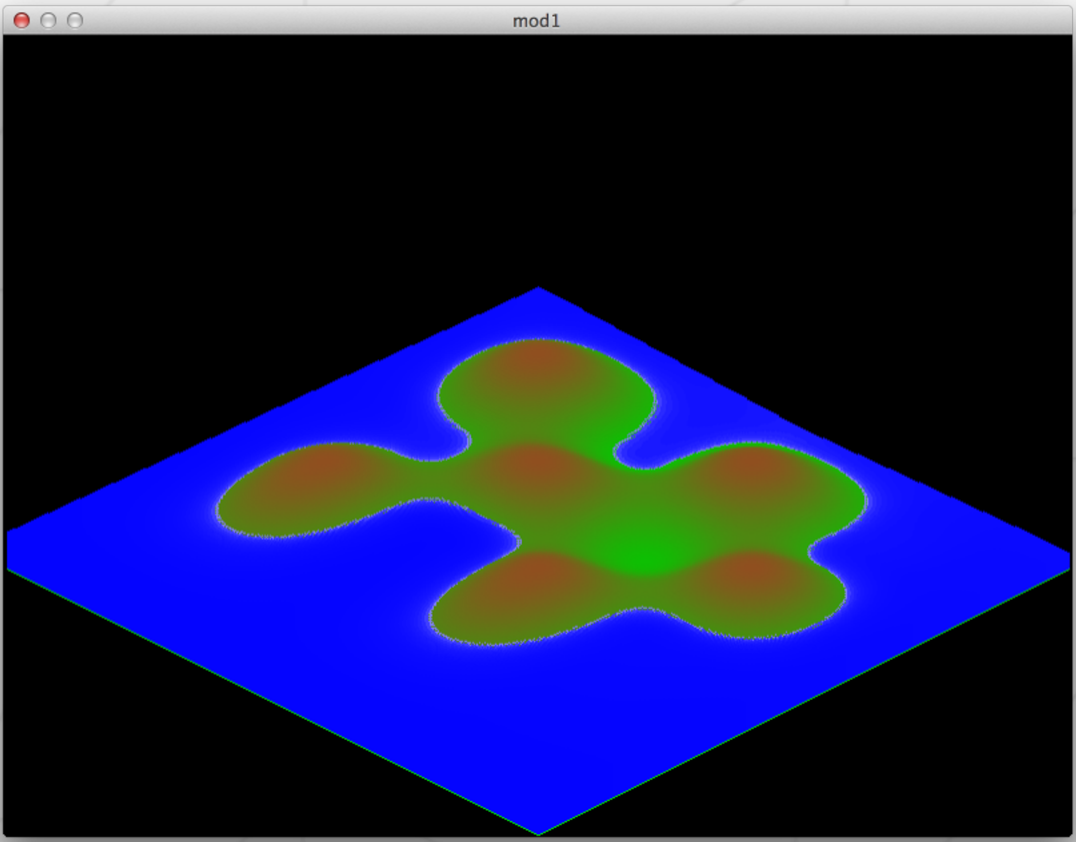

Figure VII.7: and in the end, islands.

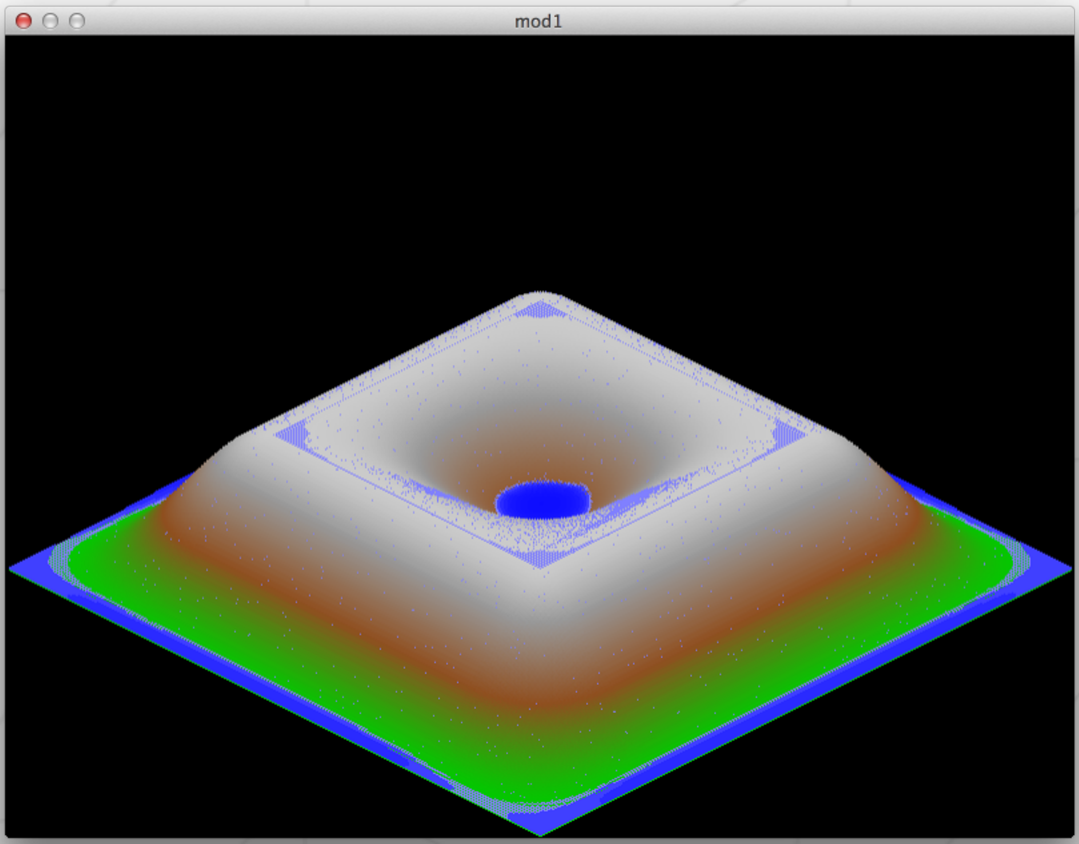

Figure VII.8: It rains over the crater