Undoubtedly GLorYous Terrain

BONIS Thomas - 11409417
CREVITS Francois - 11609835
RAMBURE Hugo - 11808749

1 Introduction

1.1 Project Structure

The project uses four third party libraries :

- *Eigen*, a header only library for vector calculations and mathematical operations.
- *FastNoise* for noise calculations.
- *ImGui* for the interfaced version of the application
- *Catch*, a header only infrastructure to perform unitary tests

The project is managed using CMake, third party libraries are compiled along with the *terrain* library, and linked to three main applications :

- *tests* to launch unitary tests of the base code
- *genTerrain* to launch integration tests
- *genTerrainGraphique* to launch the graphical interface, managing most of the functionalities.

2 Terrain generation techniques

2.1 Basic fractal noise

As seen in class, a terrain can be generated using a weighted sum of Perlin noises. Each Perlin noise used is called an octave. When an octave is added, its frequency is double the one of the previous octave and its amplitude is half. The number of octaves gives the level of details of the terrain.

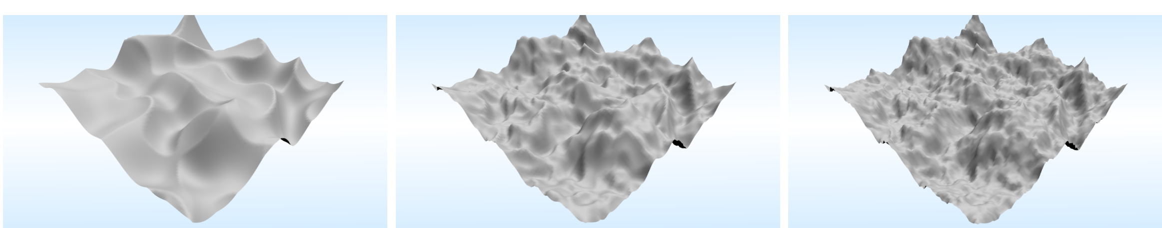


Figure 1: Basic fractal noise. From left to right : 2, 4, and 8 octaves. 1 by 1 kilometer, 200 meters high, 100 by 100 subdivision grid

This noise does not replicate the differences in roughness that are observable in nature. The goal is to obtain smooth surfaces on valleys (low altitude), and rough terrains in mountains (high altitude).

2.2 Modulated fractal noise

In order to do this, a solution is to lower the impact of iterations when applied to a low altitude, weighting the noise with the already present terrain. The result is an all or nothing system, with either flat surfaces or mountains, as we can see on 2.

When using fractal noise, the first octave might represent the overall mountain areas, the second medium hills and variations, the third irregularities in the terrain, and so on. With this analogy, the previously mentioned roughness of mountains would be generated by the higher octaves. So it is possible to lower the impact of height on an octave according to which is calculated. The higher the octave, the more it is affected by height. The result can be seen on Fig.3.

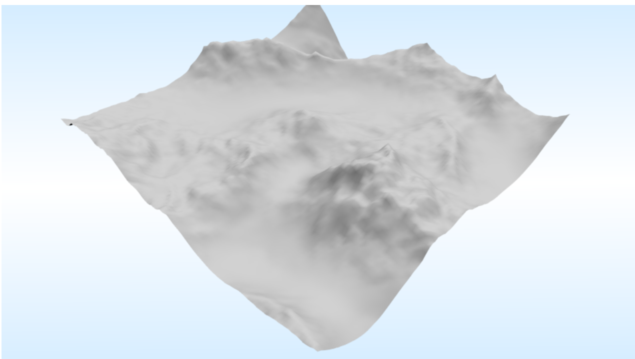


Figure 2: Height based fractal noise

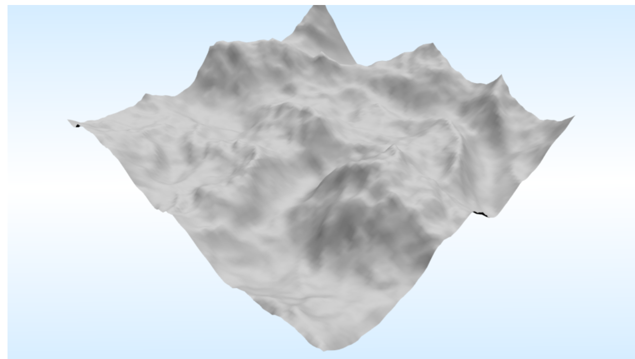


Figure 3: Height and octave based noise

Other interesting results can be obtained. This noise value can then be scaled using a different noise, effectively reproducing the result in Fig.2 but directly on the terrain and not the details.

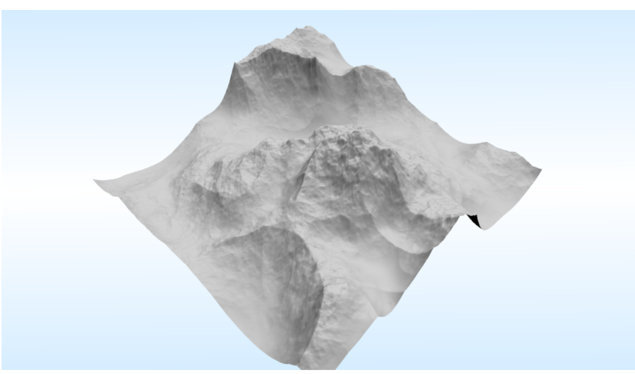


Figure 4: Terrain modulated with half the base frequency

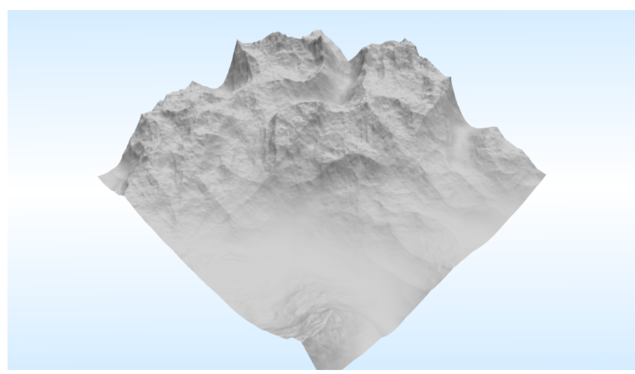


Figure 5: Terrain modulated with a fourth of the base frequency

3 Terrain data

It is possible to calculate various data from the terrain. Those information are then used for different simulations such as erosion and ecosystems.

The data fields can be saved as images. All the values are normalized and displayed as a greyscale.

Fig.6 shows the height map observed in 5. There, no calculation is done besides normalization. For this image and the following, the upper right corner corresponds to the closet point of the camera in Fig.5.

Fig.8 represents the exposure field. A white pixel corresponds to a very exposed location on the terrain. In order to do the exposure calculation, a radius and a number of directions are provided. The number of directions only determines the precision of the calculation, but the radius holds semantic information. A small radius will give information on very local data, showing more details, whilst a big radius will give global information of an are.

It is then possible to use the two type of information together.

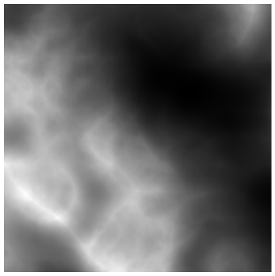


Figure 6: Heightmap of the terrain in 5

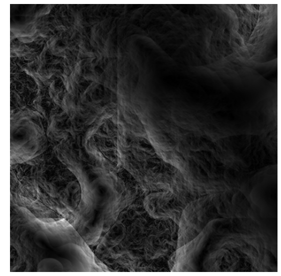


Figure 7: slopes

Fig.9 is the cumulative area. All the points of the terrain are put into an array, sorted by height and given an area value of one. They are then unstacked and their area spread like a fluid to the lower neighbors. It can be calculated in two ways :

- Steepest : All the accumulated *fluid* is given to the neighbor where the slope is the highest
- Distribution : The accumulated *fluid* is spread to all the neighbors proportionally to their slope.

This information can be combined with the slope to calculate an approximation of the humidity at each point. The formula used for this calculation is the following :

$$humidity = \sqrt{area}/(1 + k * slope)$$

The factor k represents the impact of the slope on the humidity level. A high value of k means the humidity only stays on flat areas. The result with k = 4 can be observed on Fig.10

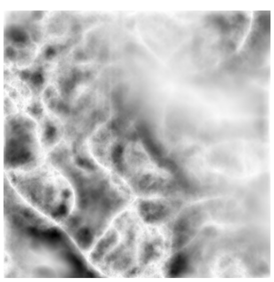


Figure 8: light exposure

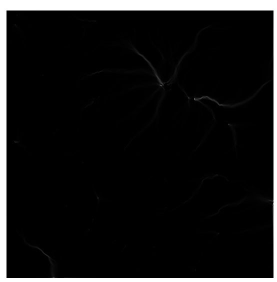


Figure 9: Cumulative area

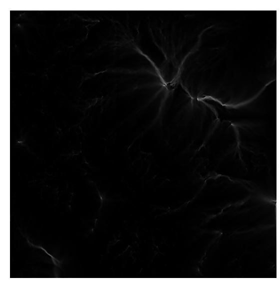


Figure 10: Humidity

Here all the data are normalized and used as such. This is done to avoid searching for physical values when using the information. The exposure is expressed can take values between 0 and 1 with 0 meaning that a point receives the smallest amount of light on the terrain and 1 that a point receives the highest amount of light. The simulations then are effectively biased and non scientific, but it is still possible to produce meaningful results.

4 Erosion

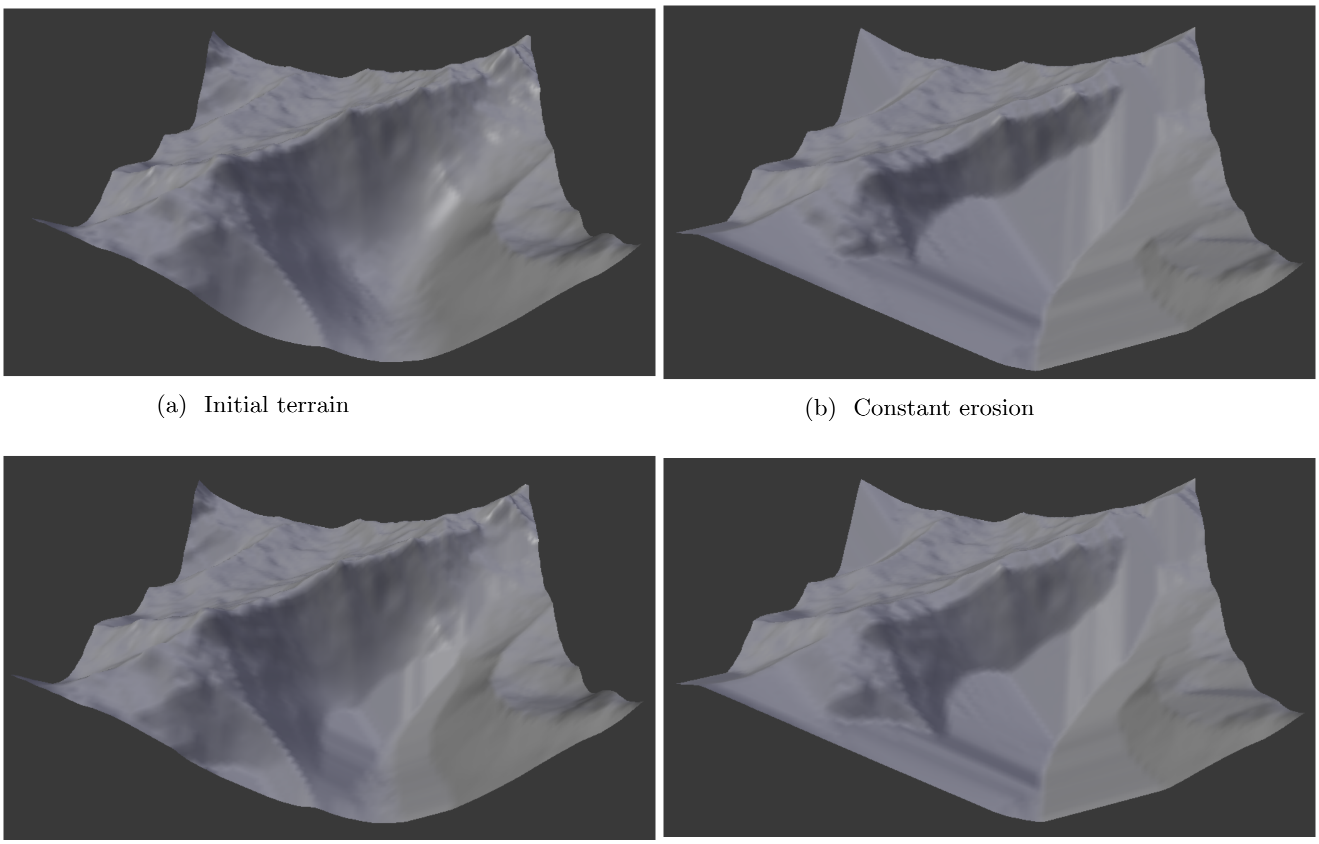
Two kinds of erosion can be calculated: thermal and hydraulic. Both have specific ways to compute erosion and transportation.

4.1 Thermal Erosion

Thermal erosion occurs when rocks on the surface are fragmented into smaller pieces, the sediments, and transported until the whole sediment pile is stable. The fragmentation of bedrock into sediments can be modeled using various approximations of the real phenomenon. The terrain is considered to be bedrock only at the initialization of every simulation. A constant amount of bedrock all over the terrain can be converted into sediments like in Fig.11b.

Another solution is to weight the converted amount based on the terrain slope. The slope can be computed either between the cell and each of its neighbors or between the opposite neighbors with regard to the cell. The weight is equal to the mean or the median of these values, both generates similar results with minimal differences on the edges of the terrain. The result of slope based erosion can be seen in Fig.11c. The fragmentation is less intense because of the weighting but we can see that high frequency details of the terrain are more visible than when using constant fragmentation.

The terrain exposure can replace slope as the weight for conversion and this gives the best results among all solutions as seen in Fig.11d. The exposure based method erodes the sharp edge in the middle of the terrain a lot more than slope based fragmentation but keeps more high frequency details on the terrain.



(a) Initial terrain

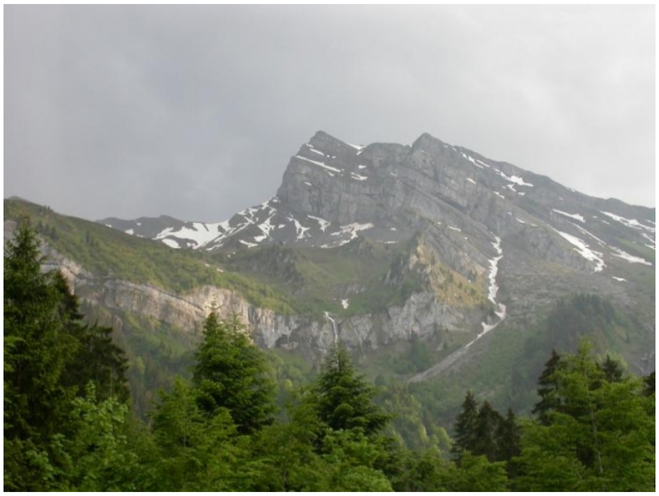
(b) Constant erosion

(c) Mean slope based erosion

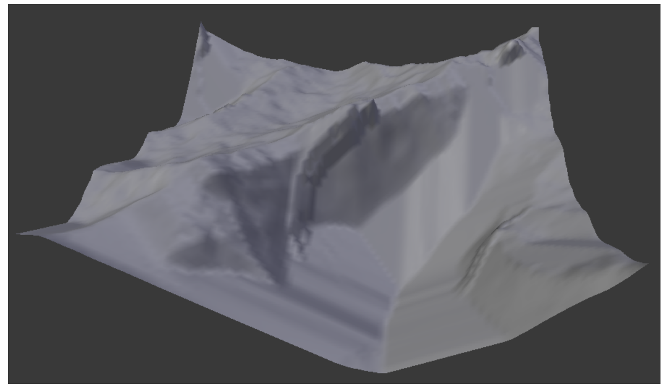
(d) Exposure based erosion

Figure 11: Result of 30 iterations of erosion with a fragmentation factor of 0.1 and transport with a stability angle of 30 degrees, using different fragmentation models. The terrain is a 100x100 grid generated using the method from section 2.

These models consider a terrain with constant material hardness which is unrealistic. A better approximation would be to consider a tilted stack of layers of various hardness. The implementation does so by using a list of user-specified layer altitudes and fragmentation factors as well as a global layer tilt. The results in Fig.12b and Fig.12d are similar to the real terrains from Fig.12a and Fig.12c. These simulations are a great tool to show that the geological structure of the underground has a much stronger effect on the shapes of the terrain than the initial topology.

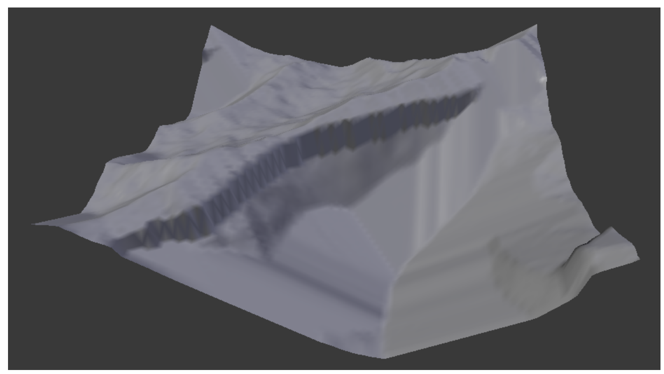


(a) Picture taken in the Swiss Apls, [Géosciences Montpellier, J. Malavieille, insu.cnrs.fr]



(b) Layer based erosion after 30 iterations of exposure based fragmentation and transport with a 30 degree stability angle. Layers are tilted by 10 degrees and the two hard layers visible are 1000 times harder than the soft layers.

(c) Picture taken in the Jura Mountains, [Laboratoire de Géologie de Lyon / ENS Lyon, Pierre Thomas, http://planet-terre.ens-lyon.fr]

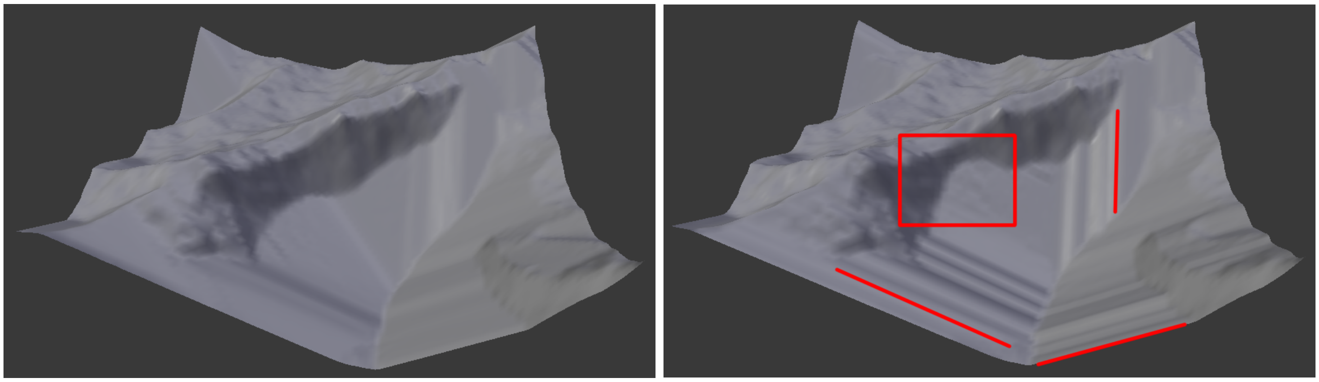


(d) Layer based erosion after 70 iterations of exposure based fragmentation and transport with a 30 degree stability angle. Layers are not tilted and the hard layers visible is 1000 times harder than the soft layers.

Figure 12: Comparisons between real terrains and the result of layer based erosion. Both simulations are initialized with the same terrain.

After fragmentation sediments are transported along the slope until stable. The stability of sediments is determined using a stability angle for the terrain material given by the user. A cell is unstable when its neighbors have a lower altitude, such that it creates an angle bigger than the stability angle. At each step, all neighbors making the cell unstable are selected and an equal quantity of sediment is transported on all of them in order to stabilize the one with the closest altitude. This process is repeated until the cell is stable.

The terrain can be considered as 8-connex or 4-connex. When using 4-connexity (Fig.13b) the terrain appears less realistic: straight lines created by the transport of sediments are a lot more visible than in 8-connexity (Fig.13a) and sediments accumulate a lot more in some places which looks less realistic.

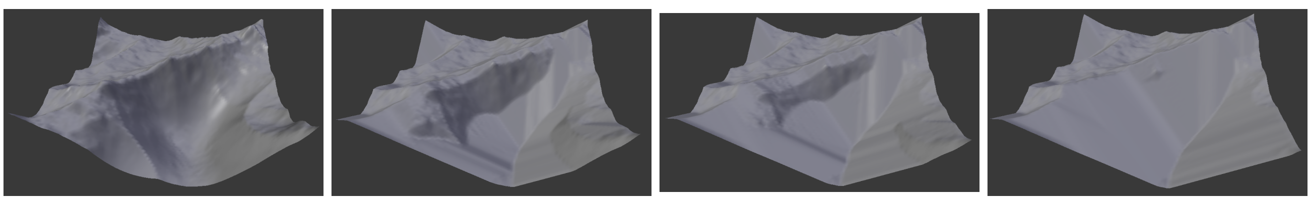


(a) Transportation using 8-connexity (b) Transportation using 4-connexity

Figure 13: Comparison of sediment transportation using 4-connexity and 8-connexity and a stability angle of 30 degrees

The stability angle should be modulated with the water drainage area to simulate a more realistic behavior for sediments. However, the water drainage area is too costly to be computed at each transportation step and the transportation process does not converge when using the same map during an entire iteration.

The evolution of a terrain during exposure based fragmentation can be seen in Fig.14. It shows that when the terrain has only one material, the system takes the shape of a sand pile.



(a) Initial terrain (b) Iteration 30 (c) Iteration 60 (d) Iteration 100

Figure 14: Evolution of a terrain during 100 iterations of exposure based erosion with a 0.1 fragmentation factor and a transport using a 30 degree stability angle.

4.2 Hydraulic Erosion

One way of computing hydraulic erosion is by using the cumulative area. Either the area obtained with the *steepest* or the *distribution* strategy can be used. However the *steepest* strategy produces less natural results, with sharp angles, and should be smoothed before use. This can be done through convolution with a user provided filter, like a mean filter.

The terrain slope is also taken into account for the erosion computation. As less erosion is desired on plains, the slope value weights the effect of the cumulative area. The following formula provide the eroded quantity at each point :

$$eroded\_quantity = \sqrt{slope * \sqrt{area}}$$

The erosion intensity can be weighted by a constant value *k* provided by the user. The output as is is a good approximation of hydraulic erosion to dig riverbeds or other water streams on the terrain. Although a transportation step can be added to preserve the initial quantity of materials, it does not provide consistency between erosion and deposit areas. The corresponding formula is given below :

$$deposit\_quantity = min(\sqrt{area}, kd)/\sqrt{1 + slope^2}$$

Where *kd* is a limiter preventing apparition of hills instead of plains where the area is high.

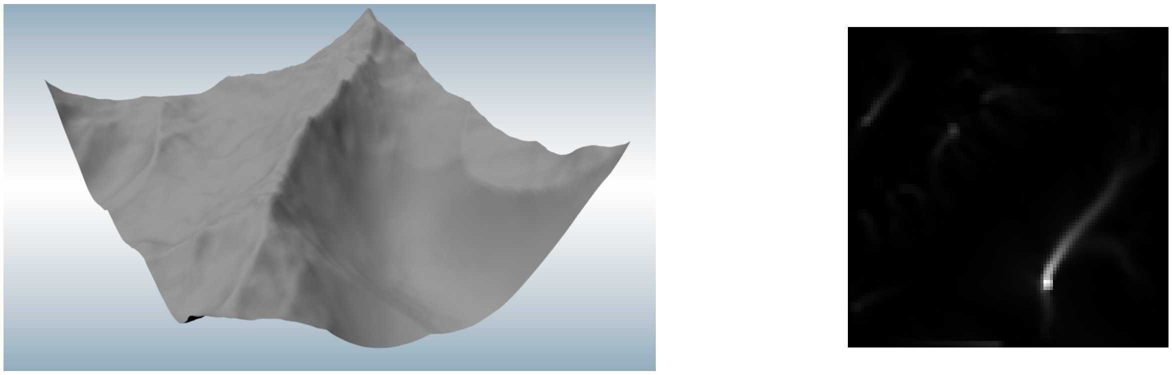


Figure 15: initial terrain for erosion in Fig.16, and the associated distributed cumulative area

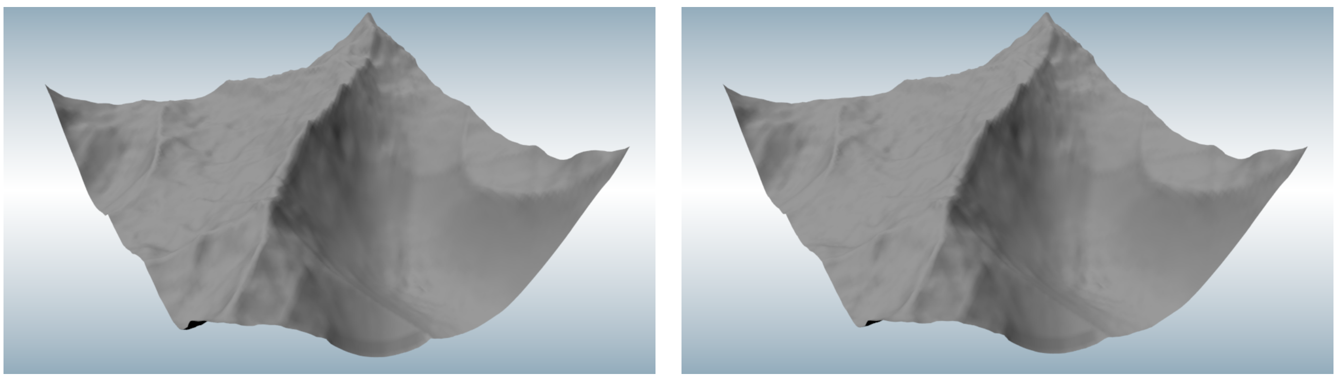


Figure 16: Hydraulic erosion with distributed cumulative area, from left to right : only erosion and erosion and transport. k = 0.2, kd = 0.05

In Fig.16, erosion where cumulative area values are high is visible even after transportation, especially on the right side of the terrain. The cumulative area map has been smoothed by a 3x3 mean filter before erosion computation.

The other way of computing hydraulic erosion is slower, but provides more realistic results. This method is water droplet erosion. Here thousands of water droplets are placed at random and flow along the terrain, one after another. They have very few influence individually, their strength lies in their number. They perform erosion and transportation at the same time, thus guaranteeing consistency between the two phenomena. The slope directly determines a droplet's speed. Each droplet also has a sediment and a water quantity which evaporates a bit after each move. They follow a set of rules:

- choose the next position among the neighbors randomly, with lower ones having more chances of being picked.
- if no lower neighbor is to be found, try deposit sediments until the pit is filled.
- if a pit can't be filled, evaporate immediately to avoid digging spikes.
- in the same spirit, don't erode more than the height difference between the current position and the next one.
- don't erode more than a constant to avoid being too powerful.
- don't deposit more than the current sediment quantity.

Then erosion and deposit are respectively handled by the formulas:

$$eroded\_quantity = min(slope, max\_erode)$$

$$deposit\_quantity = 1 - min(\sqrt{kd + slope}, 1)$$

Where *max_erode* is the maximum erosion allowed by the rules, and *kd* a parameter influencing apparition of sediment plains. As for erosion with cumulative area, a global weight *k* affect the strength of each droplet. A *water_loss* parameter determines the evaporation rate of the droplets. A brush (filter) pattern can also be given in order to spread the effect of droplets a bit, and smooth the results.

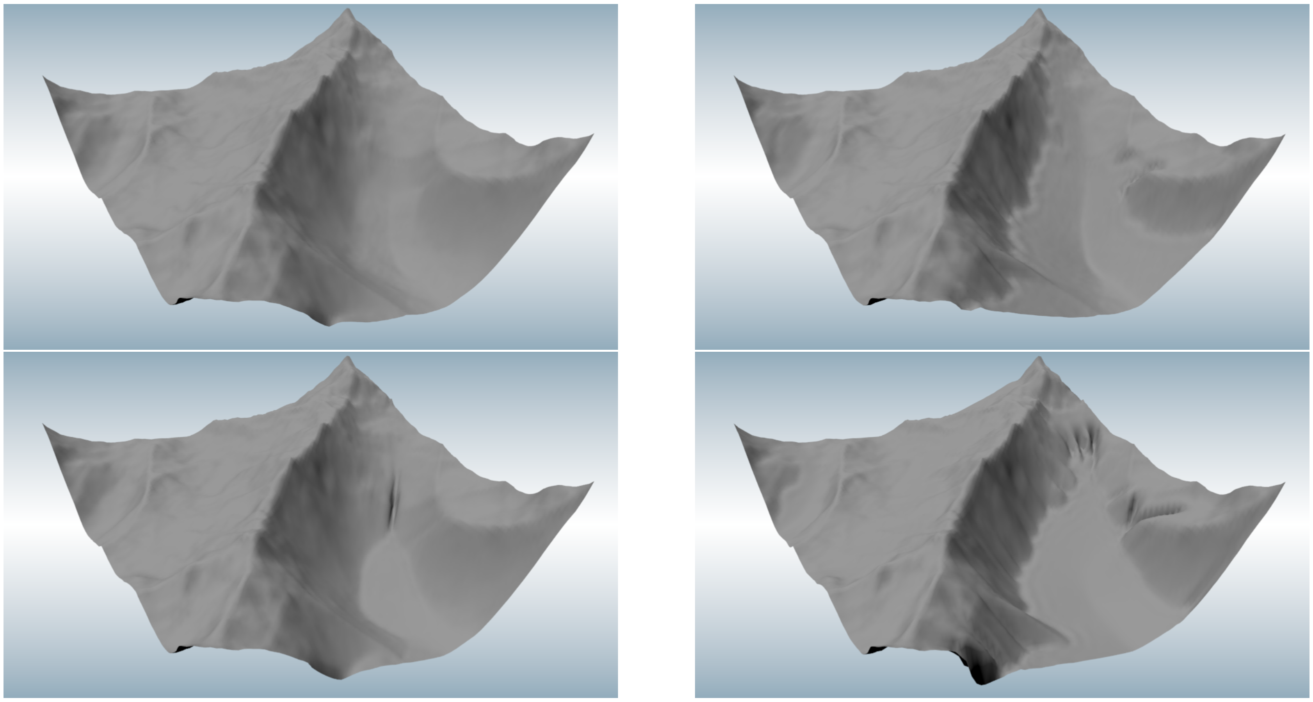


Figure 17: Hydraulic erosion with water droplets, k = 0.01, iterations per droplet = 100. first column: 10k droplets, second column: 100k droplets, first row: kd = 0.1, second row: kd = 0.5

The initial terrain used for erosion in Fig.17 is the same as the one presented in Fig.15. All the presented parameters can be set depending on atmospheric conditions and terrain nature, as rain intensity and soil absorption. A 3x3 brush pattern has been specified, spreading 40% of each droplet's effect on its neighborhood. A high *kd* value will lead to very few deposit along droplets path, which will occur only at the bottom of valleys and forming very flat plains.

5 Vegetation

In order to simulate the vegetation and ecosystem, the previously mentioned terrain data of section 3 are used to determine the suitability of various species. It is then possible to place individuals on the terrain to generate vegetation. This section will be done using a thermally eroded version of the terrain in Fig.5

5.1 Suitability

At the beginning of every simulation, the height, slope, exposure and humidity of the terrain are computed and the quantity of sediments is used as well.

The suitability is a combination of the different parameter and takes values from 0, when the terrain is not suitable, and 1, when it is perfectly suitable.

In this example, four species are defined as followed:

- *Low grass* : It likes humidity, but not heights very much. Its suitability is presented in Fig.18
- *High grass* : It likes humidity, or sediments. Its suitability is presented in Fig.19
- *Bush* : It likes high slopes, light, and a certain height. Its suitability is presented in Fig.20
- *Trees* : It likes light, but not high slopes. Its suitability is presented in Fig.21

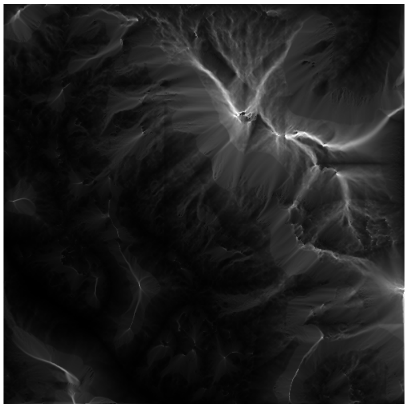


Figure 18: Low grass suitability map :

$$suit = min(1, 1.5 * \sqrt{humidity} * (1 - height))$$

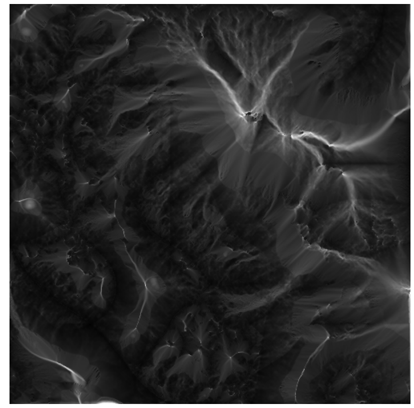


Figure 19: High grass suitability map :

$$suit = min(1, 1.3 * max(\sqrt{humidity}, sediments^2))$$

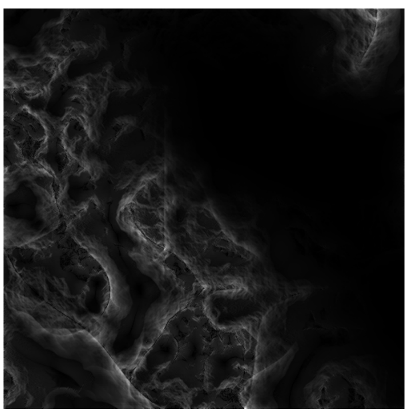


Figure 20: Bush suitability map :
$height\_factor = -7.5 * height^3 + 7.5 * height^2$
$suit = min(1, slope * exposure * height\_factor)$

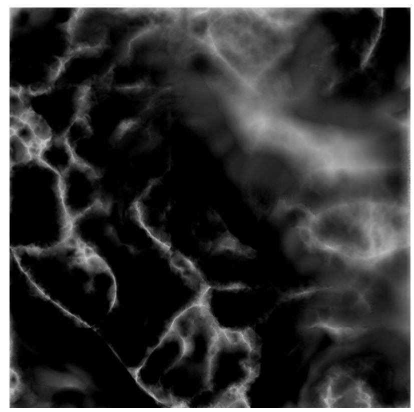


Figure 21: Tree suitability map :

$$suit = max(0, exposure^2 * (1 - slope) - 0.2)$$

The formulas for the density are arbitrary and do not correspond to the environmental requirement of real species. They are here to observe certain behavior and serves as a proof of concept

5.2 Random distribution

A simple approach to generate a map of the vegetation is randomly place a given number of individual. The suitability is used as a probability density function to decide if an individual stays where it was initially positioned

An obvious problem with this solution is the lack of global coherence as it can be observed in Fig.22. As no simulation is computed, individuals do not hold information other than their position.

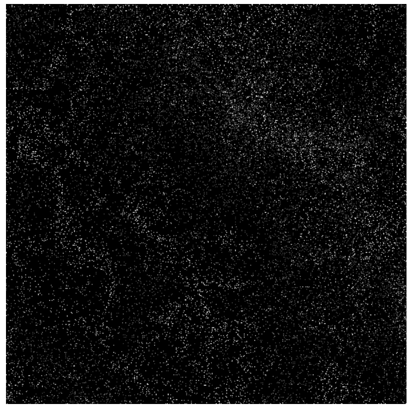


Figure 22: Position of individuals on the terrain, different shades represents different species

5.3 Simulation

To resolve the previously mentioned problem, a simulation infrastructure was put in place. To avoid neighborhood issues, it works like a cellular automaton.

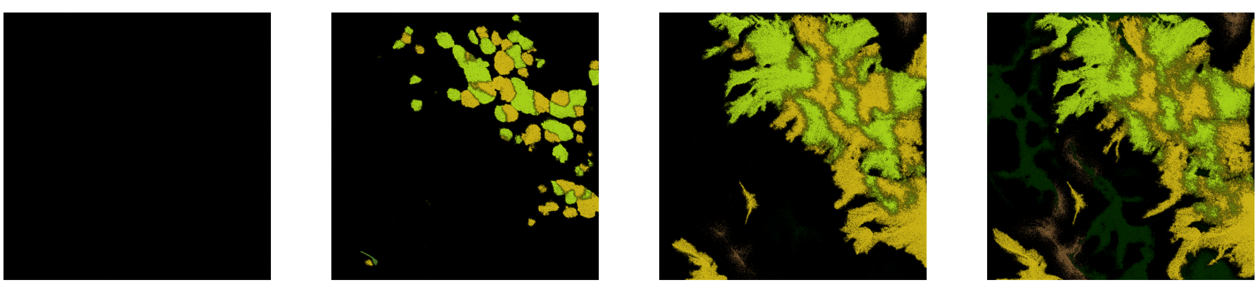
A grid is set up above the terrain, each cell contains a certain number of individual. Each individual is of a certain species, has an identifier, an age of death, an age of reproduction and a health indicator. The grid is initialised with a random distribution

At each step of the simulation, every cell of the terrain is check and every individual inside is updated. The age is incremented. If the age of the individual goes above the age of reproduction, the individual might randomly place a new born. Where the newborn is placed and how likely it is depends on the species.

Before placing a child, the number of individual in the target cell is checked in order to prevent overpopulation. Some species might be more tolerant to crowds. The new individual inherits all the attributes of its parent, and its health is determined by the suitability of the cell it is in.

Death is checked outside the update of an individual. If its current age is above its death age, scaled with its health, then it is considered dead and is removed from the grid.

The stimulation ends up in a stable state.



(a) 0 iterations (b) 100 iterations (c) 500 iterations (d) 1000 iterations

Figure 24: Simulation of the vegetation

The result of such a simulation can be seen on Fig.24. Here there were five species. Two low grass (in lime and yellow), a high grass (in green), bushes (in beige), and trees (in dark green).

We can see on the last iteration that the Strong grass did not survive in the end. The trees and bushes reproduce at a much older age and live longer so it took them time to become visible.

The displayed result is, for each cell, a color corresponding to the present species. But the result of the simulation is a huge map containing individuals, their age, and health. From this it is possible to interpret the result differently.