Using a plateau function for island generation

H.N.

June 22, 2023

1 Introduction

It is often desirable when randomly generating an island group that at the edges the land always descends into the sea. It is often recommended to add a "hill" to your noise to make more land appear in the middle and less or none at the edges. And although this works well when generating a single island, when generating a group of islands this often gives undesirable results (see for example figure **??**) (a large mountain in the middle surrounded by "debris", most often). This repository is meant to give a solution to this exact problem.

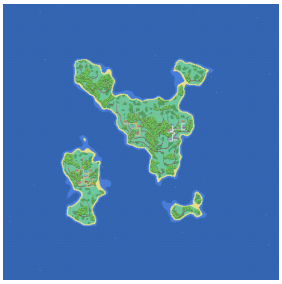

Figure 1: an example using randemon (see References)

2 How to use this repository

2.1 Importing this function

Download the file generator.py and add

```
from generator.py import plateau
```

to the file where you want to use this function.

2.2 Usage

This function takes 5 arguments:

```
plateau(x, y, tau, height, a)
```

and returns the height the plateau has at (x,y).

2.2.1 Parameters

1. x: the x position at which you want to generate the height
2. y: the y position at which you want to generate the height
3. tau: the characteristic drop length of the plateau, a lower tau will give a sharper edge, a higher tau will give a rounder edge (this value should NEVER be zero or smaller). (see figure 2-4 for an ilustration)
4. height: the maximum height of the plateau (this value should NEVER be zero or smaller) (see section 4 if a height of 0 or lower is desirable)
5. a: the distance at which the plateau significantly drops (this value should NEVER be zero or smaller)

2.2.2 Returns

This function returns a float that represents the height of the plateau at that point. The plateau is centered around (0,0). (see section 4 if this is not desirable)

2.2.3 Generating islands using this plateau

As with when using a "hill", when generating islands, the plateau of a location should be added to the noise on that location. if the plateau is too visible, it or the noise can be multiplied by a factor to make it stronger/weaker.

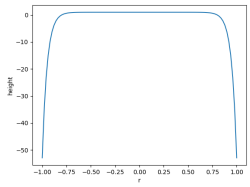

Figure 2: a plateau with tau = 0.04 and a = 0.8

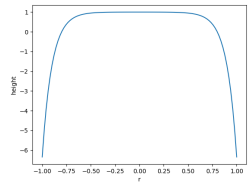

Figure 3: a plateau with tau = 0.1 and a = 0.8

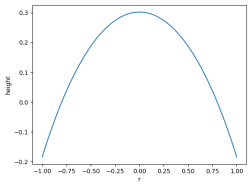

Figure 4: a plateau with tau = 1 and a = 0.8

3 Theory

3.1 Mathematical formula

This function is based on the mathematical formula:

$$f(r) = h(1 - e^{-\frac{r+a}{\tau}})(1 - e^{\frac{r-a}{\tau}}) \tag{1}$$

For the right value of τ, this function is almost constant around r = 0 due to the asymptotical nature of the function $(1 - e^{\pm\frac{r\mp a}{\tau}})$. However, at the edges this function drops exponentially. This behaviour creates the deisered plateau.

4 Addendum

When generating a height below 0 is desired, a height of 1 should be entered, and every value should be subtracted by 1.

If a center, other than (0,0) is desired, it is possible to shift the function by changing x to $(x - x_{desired})$ and y to $(y - y_{desired})$, where $(x_{desired}, y_{desired})$ is de desired location of the centrum.

To make the plateau neat for every scale, it is recommended to, instead of using the raw x and y values, scale them to a range [-1,1], this can be done by using $(x - x_{desired})/(x_{max})$ instead of x, and $(y - y_{desired})/(y_{max})$ instead of x. Where x_{max} is the maximum value x can have and y_{max} is the maximum value y can have. When using these parameters,

```
plateau((x-xcenter)/(xmax), (y-ycenter)/(ymax), 0.1, height, 0.8)
```

gives a nice plateau. (The values of tau and a can of course still tweaked to give a more desirable result, however, these values can be a starting point)

References

[klaasne]KlaasNe on GitHub, GitHub page, the Randemon site