Procedurally Generated Perlin Terrain

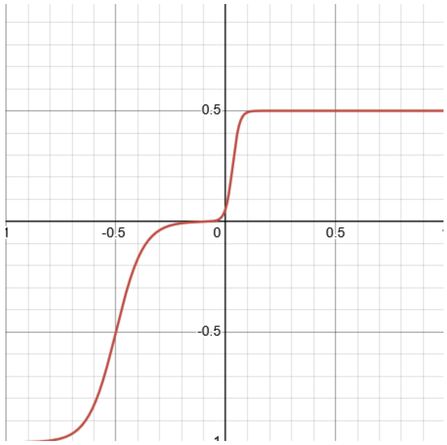
Introduction

This beach, was made using math. .. No blocks were placed by hand. ... This forest too,, to an extent.[1]

Shaping

How are these made? Well, these beaches tend to have a typical shape to them; a cliff,, followed by a portion of sand,, and then, dropping into the ocean. We can ^model this with an equation like so~:

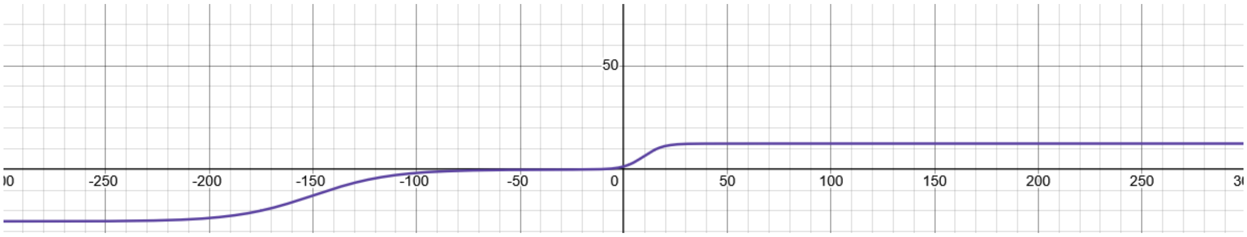
$$y < \frac{1}{1+e^{(-16z-8)}} + \frac{0.5}{1+e^{(-64z+2)}} - 1$$



But this ^isn't our equation.. In reality, it actually looks, alot more,, like this:

$$a = x/300$$
$$y < 25(\frac{1}{1+e^{(-16a-8)}} + \frac{0.5}{1+e^{(-64a+2)}} - 1)$$

[1] The base land was generated by noise, which trees were placed atop; later the groundcover was generated using math.



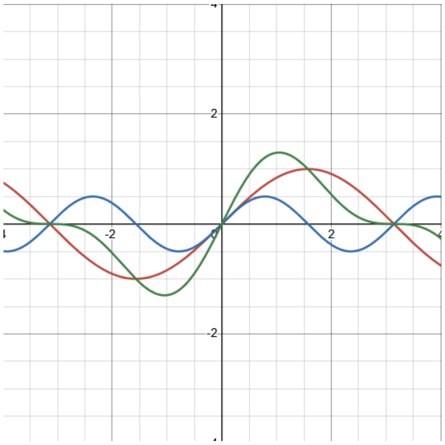
The set of coordinates we use, transforms the function. WorldEdit’s generate uses coordinates from -1 to 1 on all its axes.[2] They’re,, Normalised.

$$0.3(perlin(3209, x, y, z, 0.7, 5, 0.75))$$

^This is` the $perlin()$ function. It’s my; *secret sauce*. It makes noise based on these parameters:

- The Seed,
- The Coordinates,
- The Frequency,
- The Octaves,
- and The Persistence.

You might be` confused about Octaves and Persistence; These can add,, *roughness*. We can, visualise this with ^sine `waves.



[2] Not really. There are flags you can use to change the coordinate system used. For more information, see https://docs.google.com/document/d/1-o4dyb0ahlnn4tDgWfoTvUEQ2yBKDx_9qnUFWMJXgY0/edit?tab=t.0#bookmark=id.61qb0vc8hylt.

$$a = sin(x);\ b = \frac{sin(2x)}{2};\ c = a + b;$$

~Distractions aside``, we can use perlin() to create a, ^marshland.

Glass is good for prototyping as it causes *no* light updates.

When we use multiple perlins, we can use one to` *modify* the other.. A large perlin noise for the general shape~,, and a` smaller one to create ^the details.[3]

Details

After shaping, the decorations are next. These bring it to life. I find that marshes are masters at hiding where the land begins and the water ends.

Emergent plants break the surface of the water using none other than, ^carrots. The only real emergent plants we have in vanilla are dripleaf and mangrove, propagules.

In the low-lying areas you'll find thick reeds and grasses, and` the occasional flower. Higher up you'll find *thick* shrubs and branches`. Up here the grass turns a` different shade of green.

From above, you'll see that the water changes colour, from ice,, to deep blue. It doesn't look great here, but on the water level, it can look like ripples or, ^reflections. ~Do you think that the overhead view more important than the surface one? Is there a way we can~` ^compromise?

Pros & Cons

Now, there's a number of advantages and disadvantages to perlin, terrain over, traditional terraforming. For one, there's,, the scale.. Despite the size, terrain made with equations, have a`,, constant time to build, despite the scale.[4] You might have to be well, versed in the land of

[3] Minecraft uses this technology too, more information from the talk by Hendrik Kniberg: https://www.youtube.com/watch?v=ob3VwY4JyzE

[4] If you do have infinite compute power, it's constant. Unfortunately, I don't have access to those kind of resources.

mathematics and computer science. But~,, it also does` not blend well`,, well, with existing terrain. ^That`, has to be done,, ~manually.

But what do you think? Should *this* be the new way of building? With *math* and *code*?

I’ll leave the source commands for the marsh in the description. I made this with FAWE, but you should be able to translate it to WorldEdit.

See you next time.

Marshland: (x, 0, z)

```
//expand 599 n,e
//expand 49 u
//gen 95 c=(perlin(219,x,0,z,0.5,5,0.75)); d=0.5*perlin(912,x,y/4,z,2.3,5,0.8) y<c*d
//rep "0 <95" 95
        --Repeat until: "(FAWE) 0 blocks have been replaced."
//expand 1 n,w,s,e
//walls barrier
//contract 1 n,w,s,e
//rep "0 =y<=28" water
//rep "95 =y<=28" mud
//rep "95 =y>28" soul_soil
//setbiome stony_peaks
//gmask =y<=27
//setbiome lukewarm_ocean
//gmask
//rep "water >mud =perlin(38,x,y,z,0.3,5,0.5)<0.6*(2^(y-28))" tall_seagrass
//rep "water >mud =perlin(38,x,y,z,0.3,5,0.5)<1*(2^(y-28))" tall_seagrass[half=upper]
//rep "water >mud =perlin(38,x,y,z,0.3,5,0.5)<3*(2^(y-28))" seagrass
//rep "0 >tall_seagrass[half=lower] =perlin(38,x,y,z,0.3,5,0.5)<0.6*(2^(y-28))" beetroots[age=1],beetroots[age=0]
//rep "0 >mud =perlin(38,x,y,z,0.3,5,0.5)<0.2*(2^(y-28))" tall_grass[half=upper]
//rep "0 >mud =perlin(38,x,y,z,0.3,5,0.5)<0.4*(2^(y-28))" wheat[age=5]
//rep "0 >mud =perlin(38,x,y,z,0.3,5,0.5)<0.7*(2^(y-28))" carrots[age=5]
//rep "beetroots =perlin(38,x,y,z,0.3,5,0.5)<0.05*(2^(y-28))" wheat[age=5]
//rep "beetroots =perlin(38,x,y,z,0.3,5,0.5)<0.1*(2^(y-28))" potatoes[age=4]
//rep "mud =perlin(38,x,y,z,0.3,5,0.5)<2*(2^(y-28))" *muddy_mangrove_roots
```

```
//rep "soul_soil =perlin(38,x,y,z,0.3,5,0.5)<0.01*(2^(y-28))" coarse_dirt
//rep "soul_soil =perlin(38,x,y,z,0.3,5,0.5)<0.03*(2^(y-28))" *stripped_spruce_wood
//rep "0 >soul_soil =perlin(38,x,y,z,0.3,5,0.5)<2*(0.5^(y-28))" tall_grass
//rep "0 >soul_soil =perlin(38,x,y,z,0.3,5,0.5)<4*(0.5^(y-28))" tall_grass[half=upper]
//rep "0 >tall_grass[half=lower]" tall_grass[half=upper],0
//rep "0 >soul_soil =perlin(38,x,y,z,0.3,5,0.5)<7*(0.5^(y-28))" short_grass
//rep "tall_grass >tall_grass =perlin(38,x,y,z,0.3,5,0.5)<1*(0.5^(y-28))" lilac[half=upper]
//rep <lilac 10%lilac,90%large_fern
//rep "tall_grass[half=upper] =perlin(38,x,y,z,0.3,5,0.5)<0.03*(2^(y-28))" large_fern[half=upper]
//rep "<large_fern tall_grass" large_fern
//gmask =y>=31
//setbiome plains
//gmask
//rep "coarse_dirt <0" grass_block
//rep "0 >grass_block =perlin(38,x,y,z,0.3,5,0.5)<0.001*(2^(y-28))" oak_leaves
//rep "0 >grass_block =perlin(38,x,y,z,0.3,5,0.5)<0.002*(2^(y-28))" *large_fern
//rep "0 >grass_block,stripped_spruce_wood =perlin(38,x,y,z,0.3,5,0.5)<0.003*(2^(y-28))"
short_grass
//rep "0 >oak_leaves =perlin(38,x,y,z,0.3,5,0.5)<0.00025*(2^(y-28))" short_grass
//rep "soul_soil <soul_soil ~[!#solid]" *dark_oak_wood
//rep "coarse_dirt <soul_soil ~[!#solid]" *stripped_spruce_wood
//rep beetroots[age=0] carrots[age=0]
//rep beetroots[age=1] carrots[age=3]
//rep "tall_grass[half=upper] =y==30 >soul_soil" 70%tall_grass[half=upper],30%wheat[age=5]
//rep "water <0 #offset[0][-1][0][tall_seagrass] =perlin(38,x,y,z,0.3,5,0.5)<0.2*(2^(y-28))"
acacia_leaves[persistent=true,waterlogged=true]
//rep seagrass
90%seagrass,5%*small_dripleaf[half=upper,waterlogged=true],5%mangrove_propagule[waterlogged
=true]
```