

Terrain Shader

03/11/2021

Brayden Tremper

tremperb@oregonstate.edu

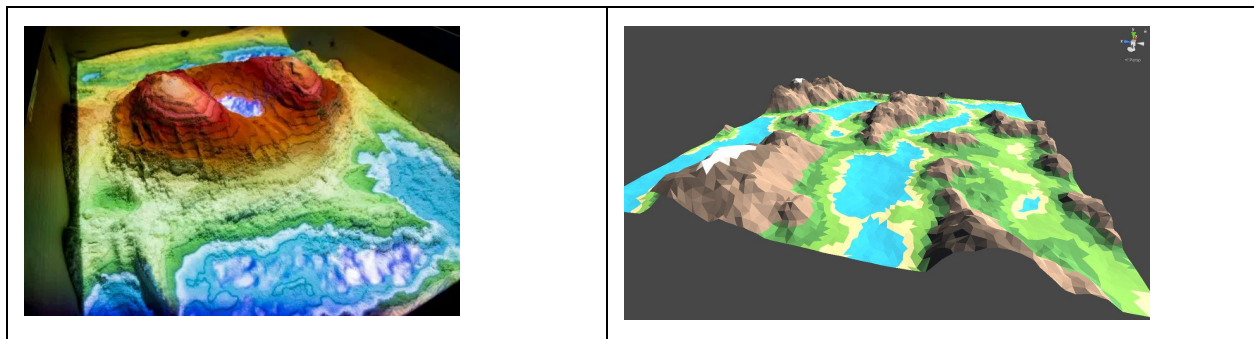
CS457 Winter 2021

Video Link: <https://www.loom.com/share/9487b1ce225146bc894a092027753428>

Proposal

For my project I am seeking to create a 3D terrain shader with multiple levels relative to height. Some example levels could be water, sand, dirt, grass, snow, etc. These general levels could just simply be colored or have a designated texture applied. Specifically, working with some of the concepts we have explored this term such as noise, height maps, frequency, amplitude allowing for both a unique and adjustable terrain. Following this, when considering the lighting and similar aspects I am seeking to target an idea similar to project 3 using per fragment lighting where the user has the ability to play around with some lighting positions and some ambient, specular, and diffuse values allowing the terrain to be seen in an interesting way similar to imitating the sun position.

Breaking this down further, I am seeking to make this work with the idea of a sandbox in mind. Such as, the terrain lives on a given quad where its fields such as amplitude, frequency, and height can be adjusted freely while the terrain maintains the rendering on the given quad using water as the minimum base level. Some rough examples illustrating the general concept I am after can be seen below.



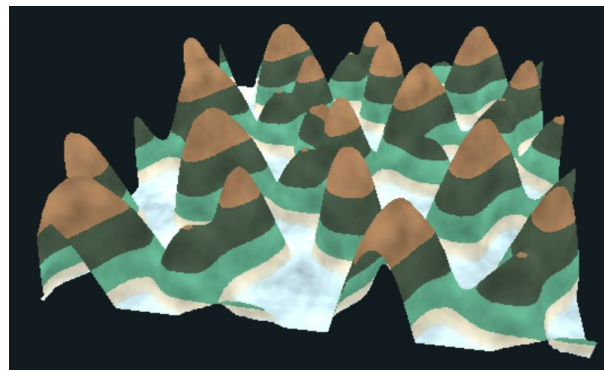
I can't say, without actually jumping to the implementation, how difficult this would be but some optional features which I believe could be applied include adding a reflective/refractive effect to the water, creating an option to view the terrain with more or less octaves to change the roughness, creating terraces, or exploring more noise ideas such as simplex and perlin noise. These are just some general ideas for the project although I am open to any suggestions or advice.

I believe this would be interesting and is actually something I have wanted to pursue with some future implementation being to bring this into Unity where this terrain can be interactable in a VR environment.

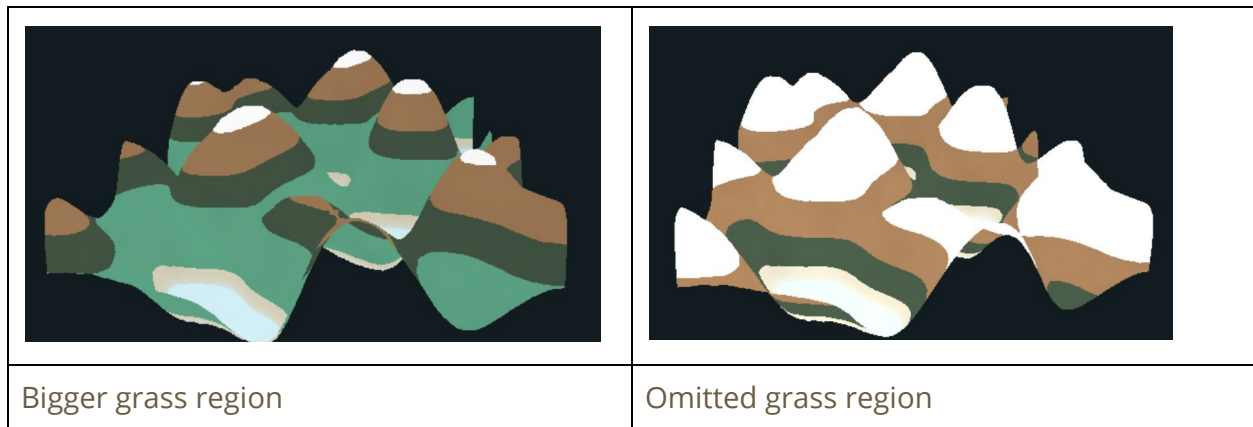
Complete Project



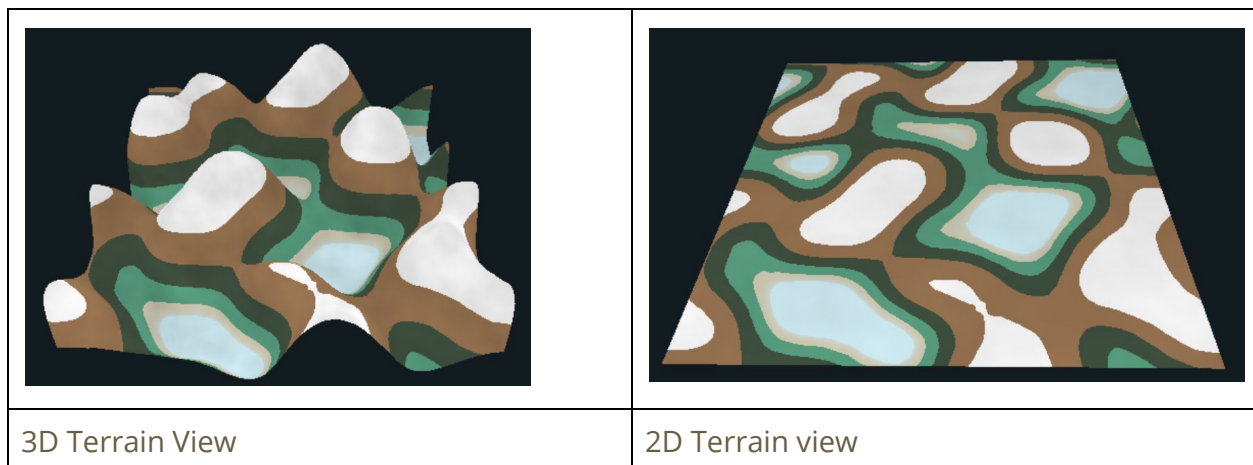
When viewing the completed project all essential items from the project were completed. The noise displacement was completed in the vertex shader by obtaining our height value from our sampler noise and raising the height at a given vertex in correspondence with that value on the condition (plus a few added conditions such as exaggerating the height) that it was greater than our water level. Continuing with this water level it was used as an overall baseline as in the minimum vertex location for our terrain as seen below.



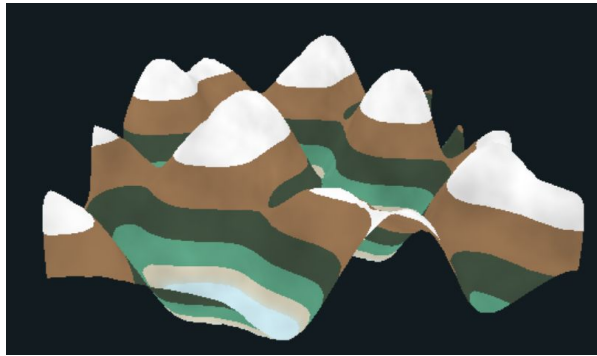
With the method above in place a 3D terrain could be created and live on our water grid but separating the layers of this terrain took place in the fragment shader. Following the idea of utilizing sampler noise the terrain was then colored in corresponding boundaries, each building off the previous, with these boundaries(further referred to as regions) being water, sand, grass, a forest, sirt, and finally snow. These could then be colored and applied to the respective location with an option of mixing between each zone with the utilization of a tolerance value. Building on the previous statement of regions building off of each other, the regions were designed in a way allowing each region to grow in turn pushing the next region(s) up while also giving the user access to complete control over the size of the regions and whether they wish to even omit a region. Example pictures of this functionality can be found below.



Following this the coloring of said regions were set up allowing for a more recognizable terrain to be viewed. Following this, functionality was added to allow the terrain to be transitioned between 2D or 3D space simulating which I felt to be the idea of viewing this data on a map vs in person. This can be seen below.



Upon overall setup of said terrain per fragment lighting was created allowing for adjustable ambient and diffuse lighting while also allowing custom light positions in order to mimic the idea of the sun.



Afternoon Terrain View Idea

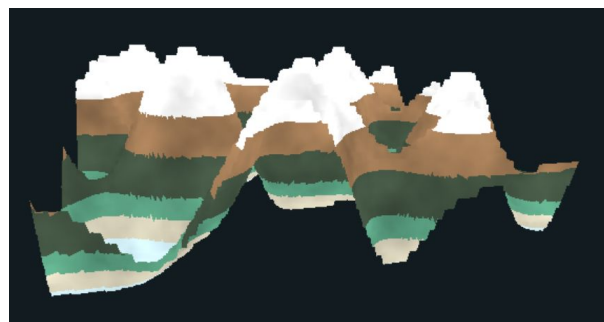


Night Terrain View Idea

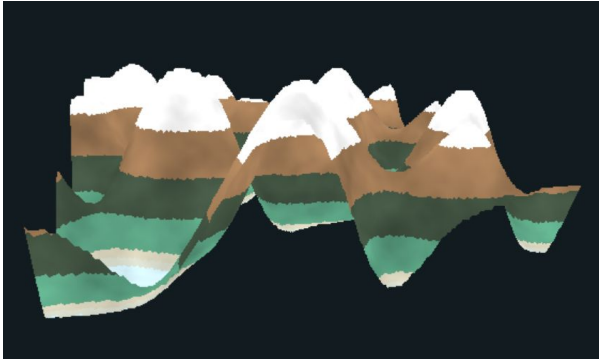
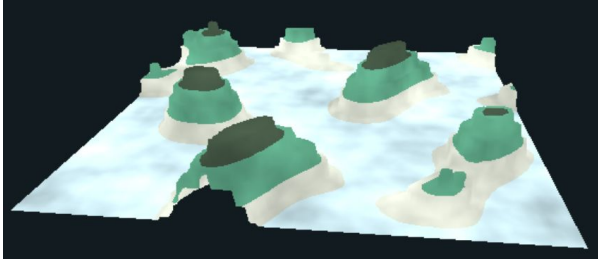
Following the previous modifications all necessary components were completed. However, with continued implementation I was able to create terraces for the terrain. This took place in both the vertex and fragment by working on the idea of rounding a terrain's height value to a desired number of levels, this mainly follows the idea of rounding the value of our height multiplied by the desired level while then dividing this rounded number by the level as well. Examples of this functionality can be seen below.



Approx 16 terrace levels



Approx 40 terrace levels

	
Approx 100 terrace levels	Approx 20 terrace levels with raised water height.

Differences

When viewing differences when comparing the completed project to the proposal few come to mind. All necessary components were completed successfully as I had originally envisioned them when submitting the proposal. However, one item not included was specular lighting, this was simply due to glman not letting me take in the uniform variables to access and adjust it, I could never figure out what was causing this and eventually decided to drift from the idea of specular lighting. However to make up for this, one of the possible extras listed in the proposal, the terraces, were completed as well which I believe turned out well and gave more versatility to the terrain.

Link(Also listed on title page)

Video Link: <https://www.loom.com/share/9487b1ce225146bc894a092027753428>