CIS565 Final Project Proposal
Vimanyu Jain, Mikey Chen

Why Terrain Generation?
There exist various methods to generate terrain. For example, realistic terrain can be created using heightmaps gathered from real world data. This method can be limiting in terms of the output that we can achieve. In contrast, procedural and fractal methods enable us to model terrain with many different looks and features.

Terrain Generation:
We intend to use subdivision and tessellation based on a smooth distance based LOD scheme. Some of the shortcomings of the current approaches to terrain generation are the need to
Limitations of current approaches:

- need to preprocess large data sets
- constant transfers of large datasets from CPU to GPU
- limited viewing areas
- complex algorithms to merge meshes to avoid cracks and seams.

The GPU based algorithm that we intend to implement addresses all these limitations without any significant impact on the CPU. The main algorithms utilized in the system are,

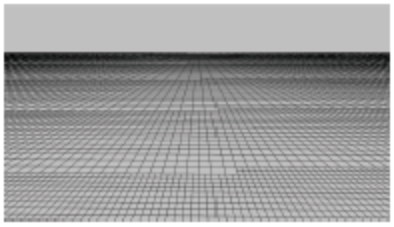
- Subdivision algorithm, to perform region based subdivision
- LOD Transition algorithm, to ensure smooth transitions from one LOD to another
- Procedural Height Generation

Apart from this, we plan to test different models for procedural height generation, and do performance analysis

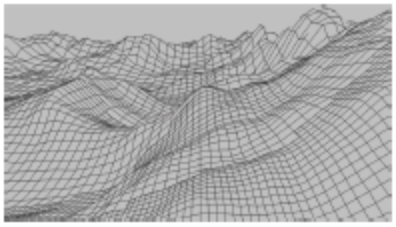
- Ridged multifractal terrain model
- Perlin noise algorithm
- Midpoint displacement
- Diamond-square algorithm

Rendering:
If time permits, we would like to include water as part of our procedural world and have different shading for the water and the mountains.

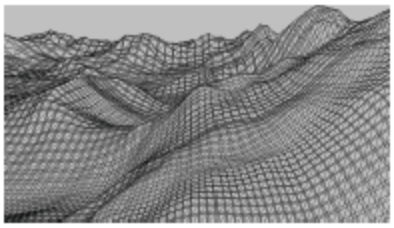
References:
Geometry Shader and Tessellation Shader
Terrain Rendering
GPU Pro Advanced Rendering Techniques - Chapter 1
OpenGL Insights



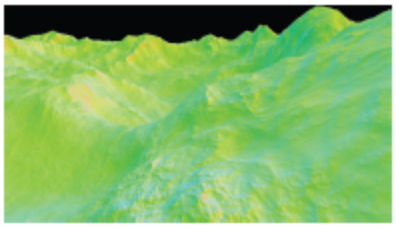
(a)



(b)



(c)



(d)