Terrain Generation

Tiju Thomas

CIS 565 – GPU Programming Final Project
Mid-Point Presentation

3rd December 2012

Outline

- Introduction
- Goal
- Current Progress
- Next Steps

Introduction

- "The wilderness inspires many and fills you with energy and enthusiasm"
- Love the outdoors
- Terragen (http://planetside.co.uk)

Goal

- Create a basic Terrain Generator

- Shading

- Tessellation based on distance from camera

- Different terrains based on noise (and maybe height fields)

Basic Mesh

- Flat Grid

- Height using Perlin Noise

- Displacement in Vertex Shader

- Color assigned using Fragment Shader

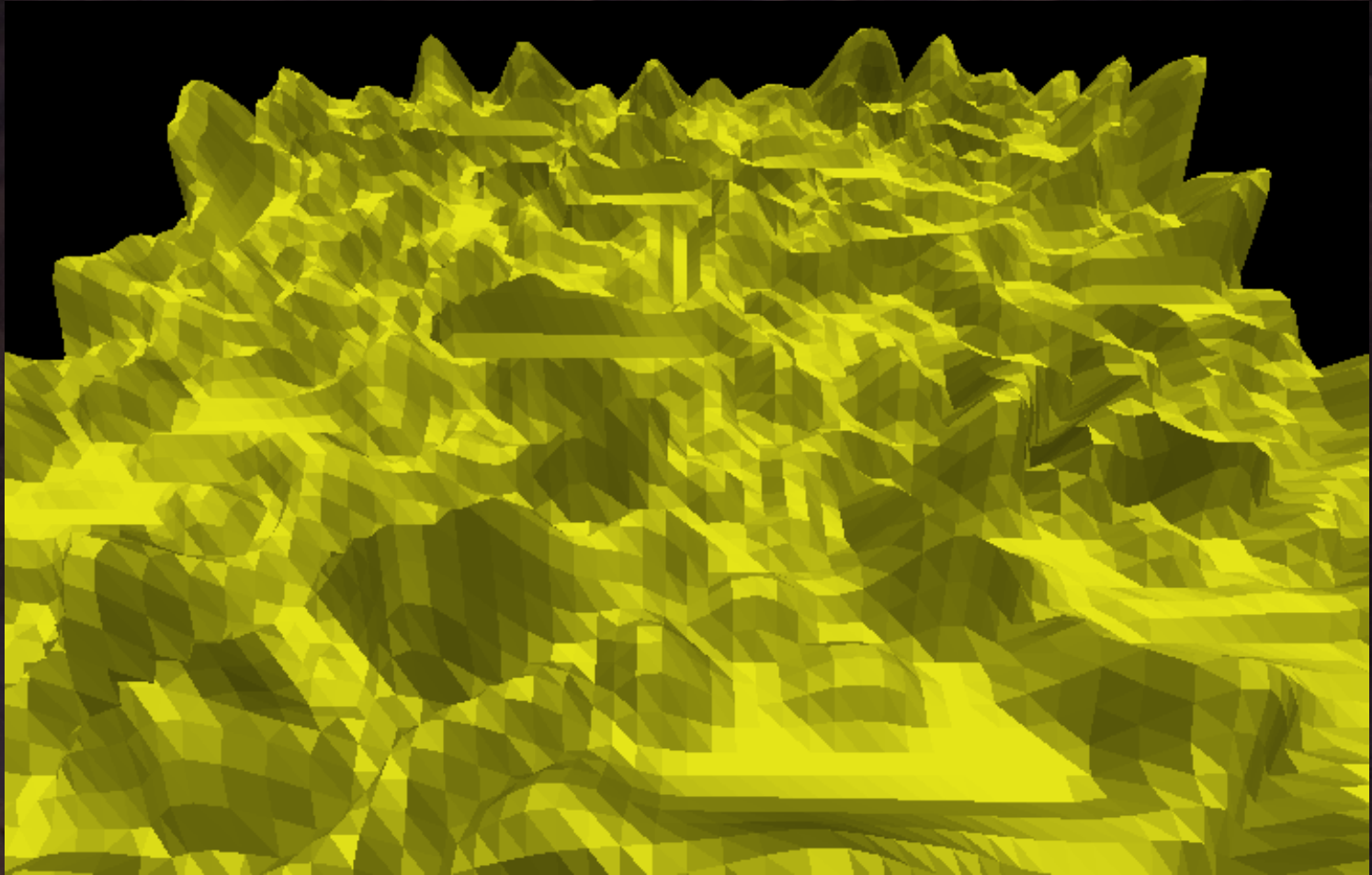- Mesh created using GL_LINE instead of GL_FILL in glPolygonMode

Basic Mesh
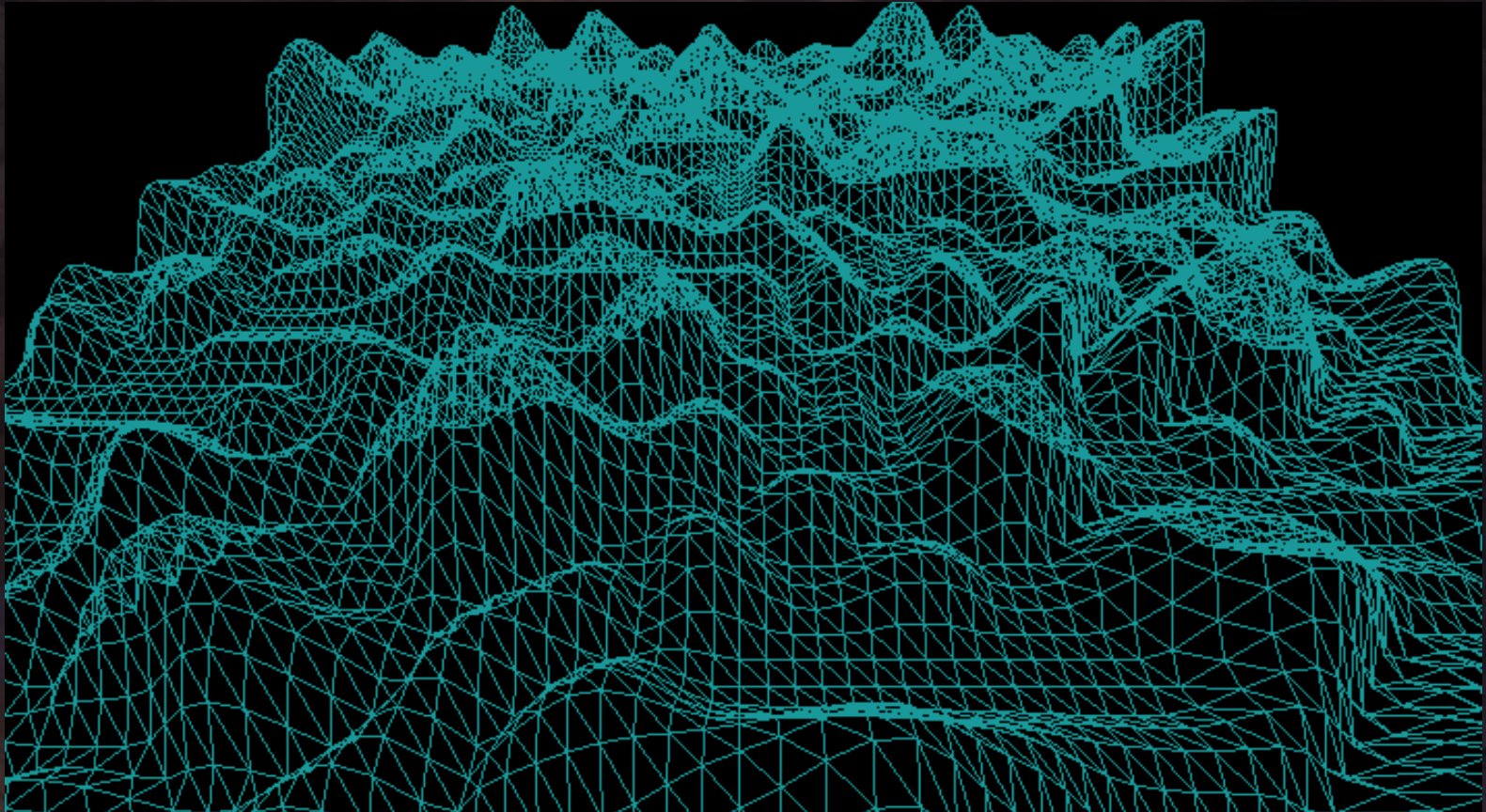
Shading

- Geometry Shader
  - Contains information about the entire primitive
- Generate Normals using vertex data
- Diffuse shading in Fragment Shader based on directional light

Shading

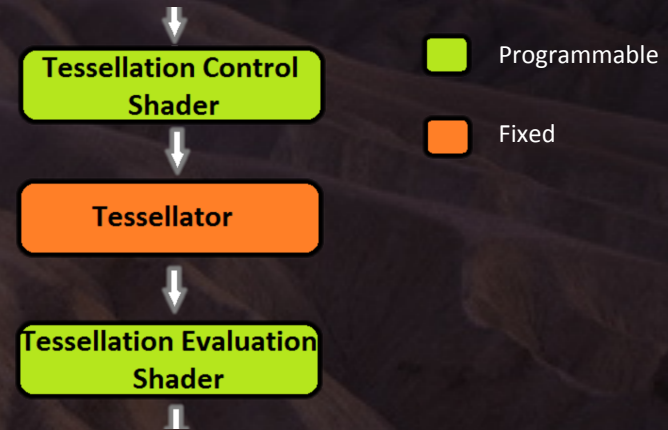Shaded with normals data from geometry shader

Shading

- Geometry Shader
  - Contains information about the entire primitive
- Generate Normals using vertex data
- Diffuse shading in Fragment Shader based on directional light

- **Issue** – Flat shading for each primitive

Tessellation

- Tessellation Shaders
  - Tesselation Control Shader (TCS)
  - Tessellation Evaluation Shader(TES)
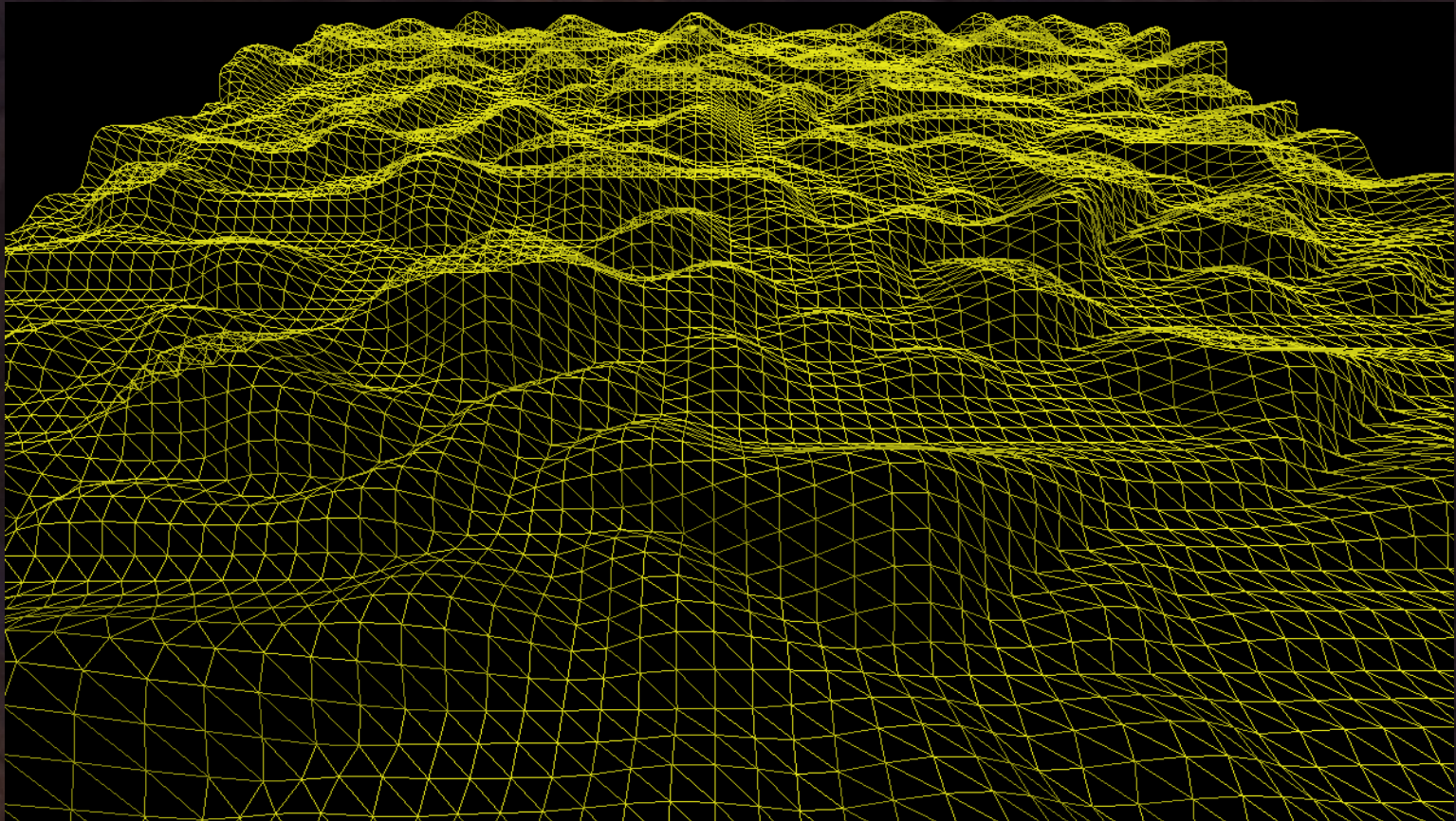

| | Programmable |
| | Fixed |

- Control Parameters (Outer and Inner Tessellation Levels)
- TCS sets values
- Tessellator stage does actual tessellation
- TES computes vertex coordinates

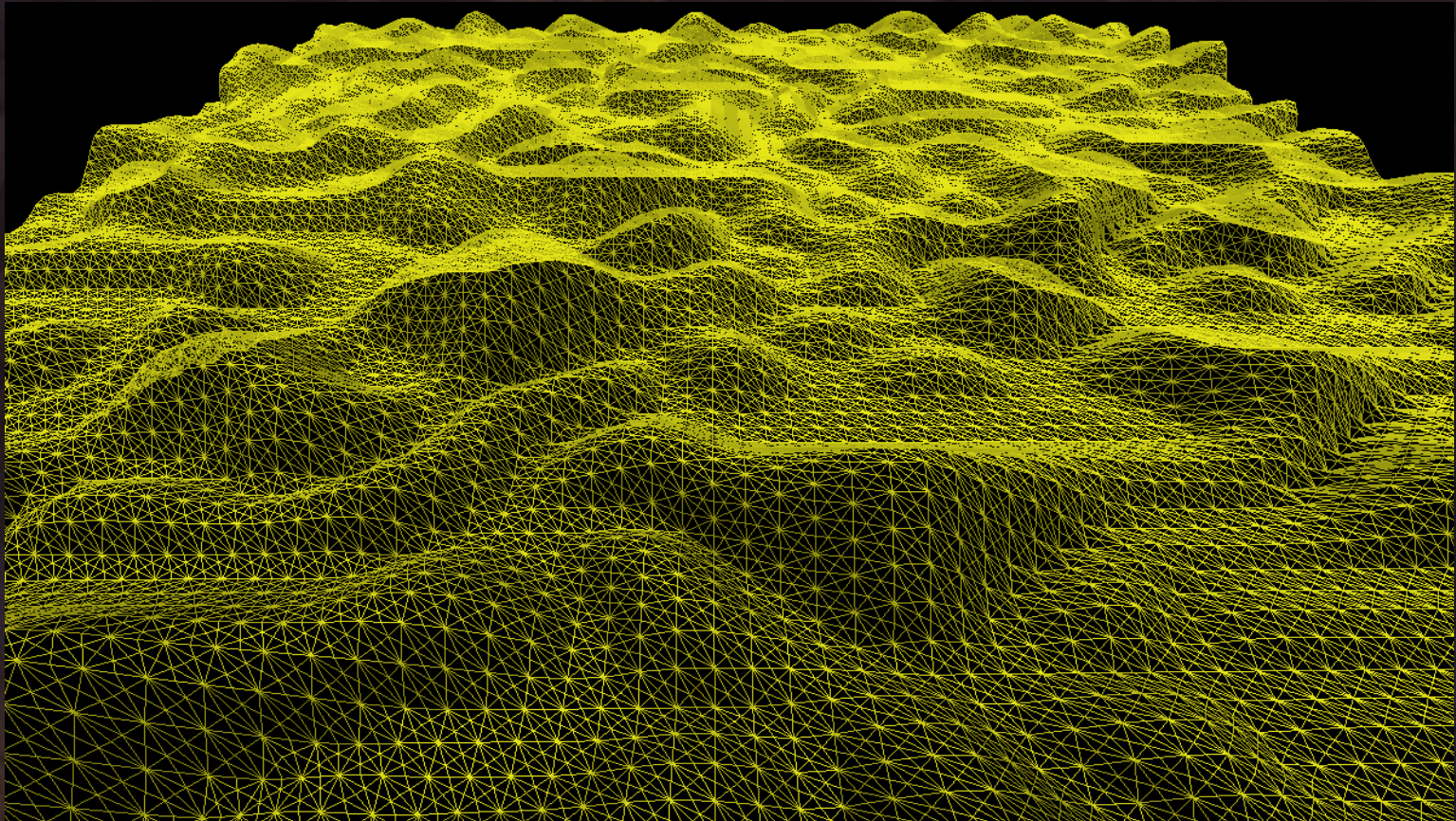# Tessellation

Non-Tessellated Mesh

Running at 30 fps

Tessellation

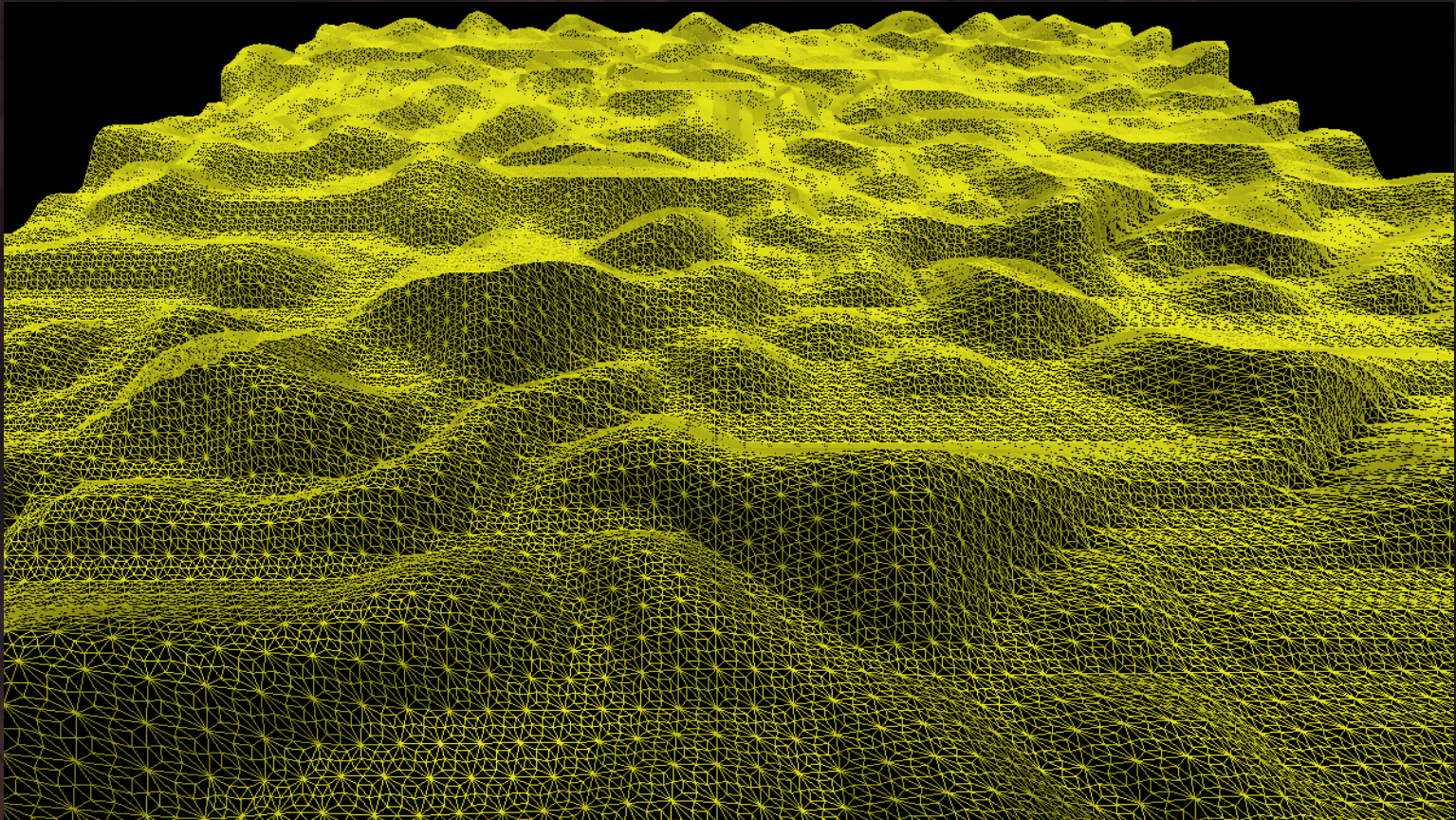Outer Tessellation – 2, Inner Tessellation - 2

Running at 9 fps

Tessellation

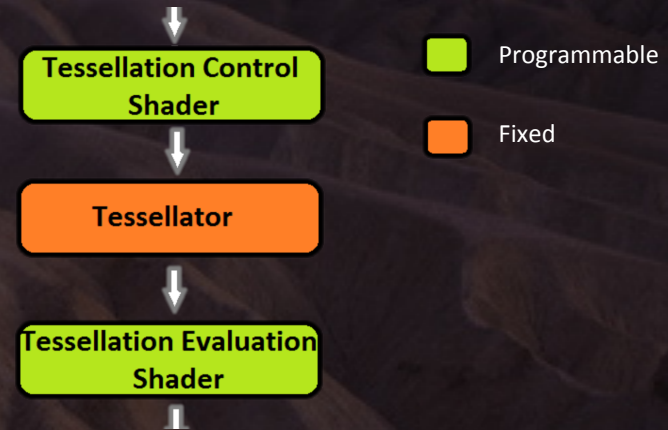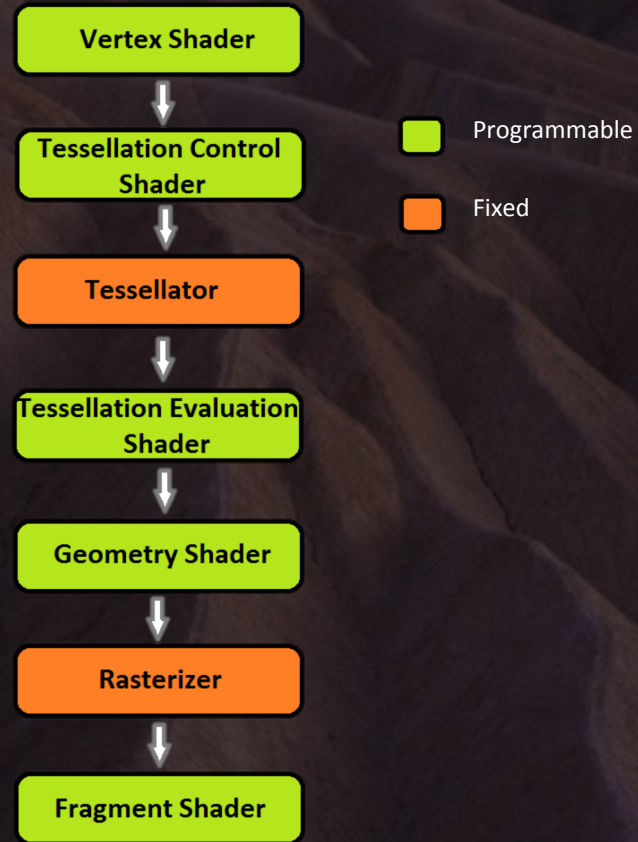Outer Tessellation – 2, Inner Tessellation - 3

Running at 6 fps

Tessellation

- Tessellation Shaders
  - Tesselation Control Shader (TCS)
 - Tessellation Evaluation Shader(TES)

- Control Parameters (Outer and Inner Tessellation Levels)
- TCS sets values
- Tessellator stage does actual tessellation
- TES computes vertex coordinates

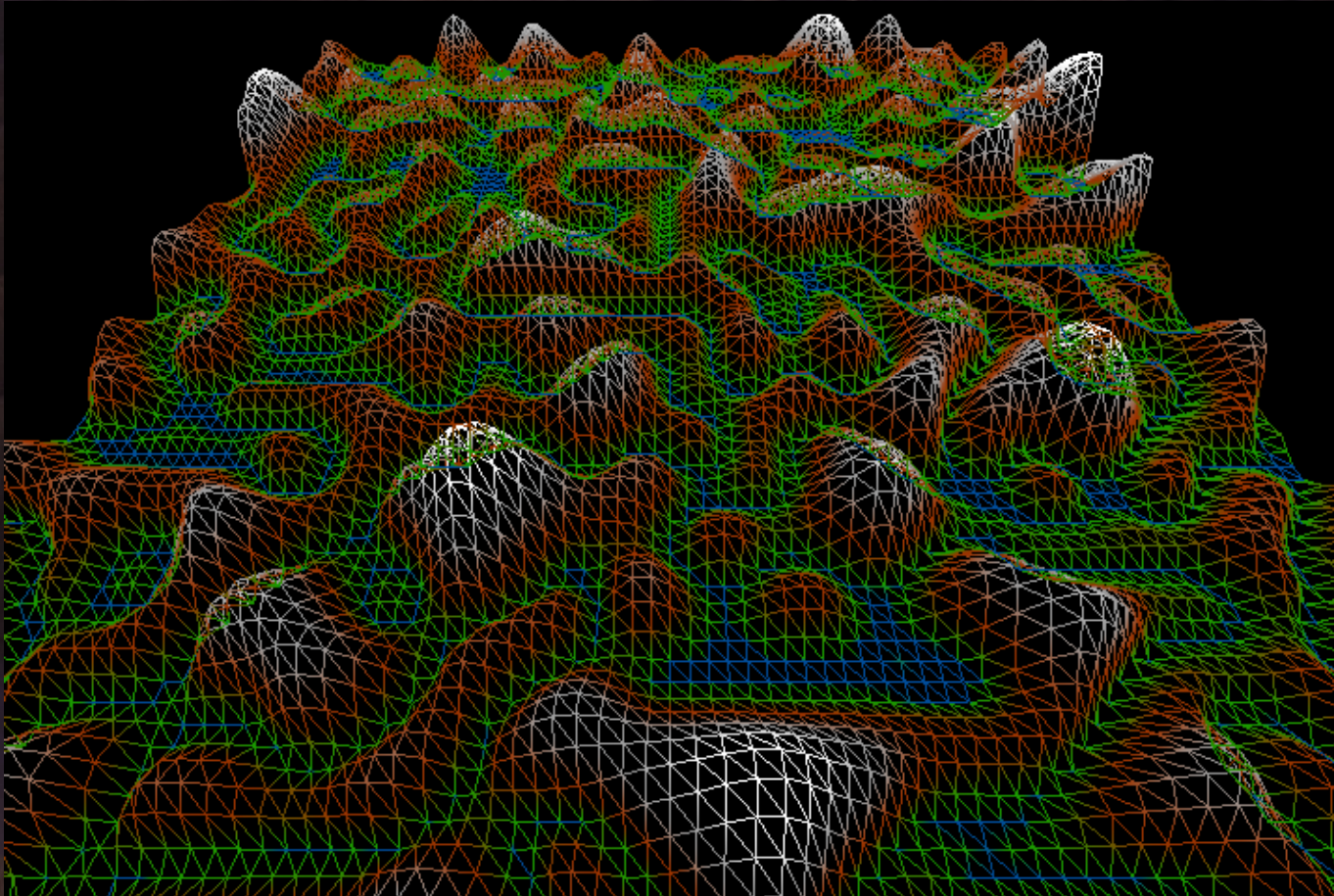- **Current State**: No displacement of vertices after tessellation

Recap

```
        ┌─────────────────────┐
        │    Vertex Shader     │
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐          ■ Programmable
        │  Tessellation Control│
        │       Shader         │          ■ Fixed
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐
        │     Tessellator      │
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐
        │ Tessellation Evaluation│
        │       Shader         │
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐
        │   Geometry Shader    │
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐
        │     Rasterizer       │
        └─────────────────────┘
                  ↓
        ┌─────────────────────┐
        │   Fragment Shader    │
        └─────────────────────┘
```

Depth

Shading based on depth from camera
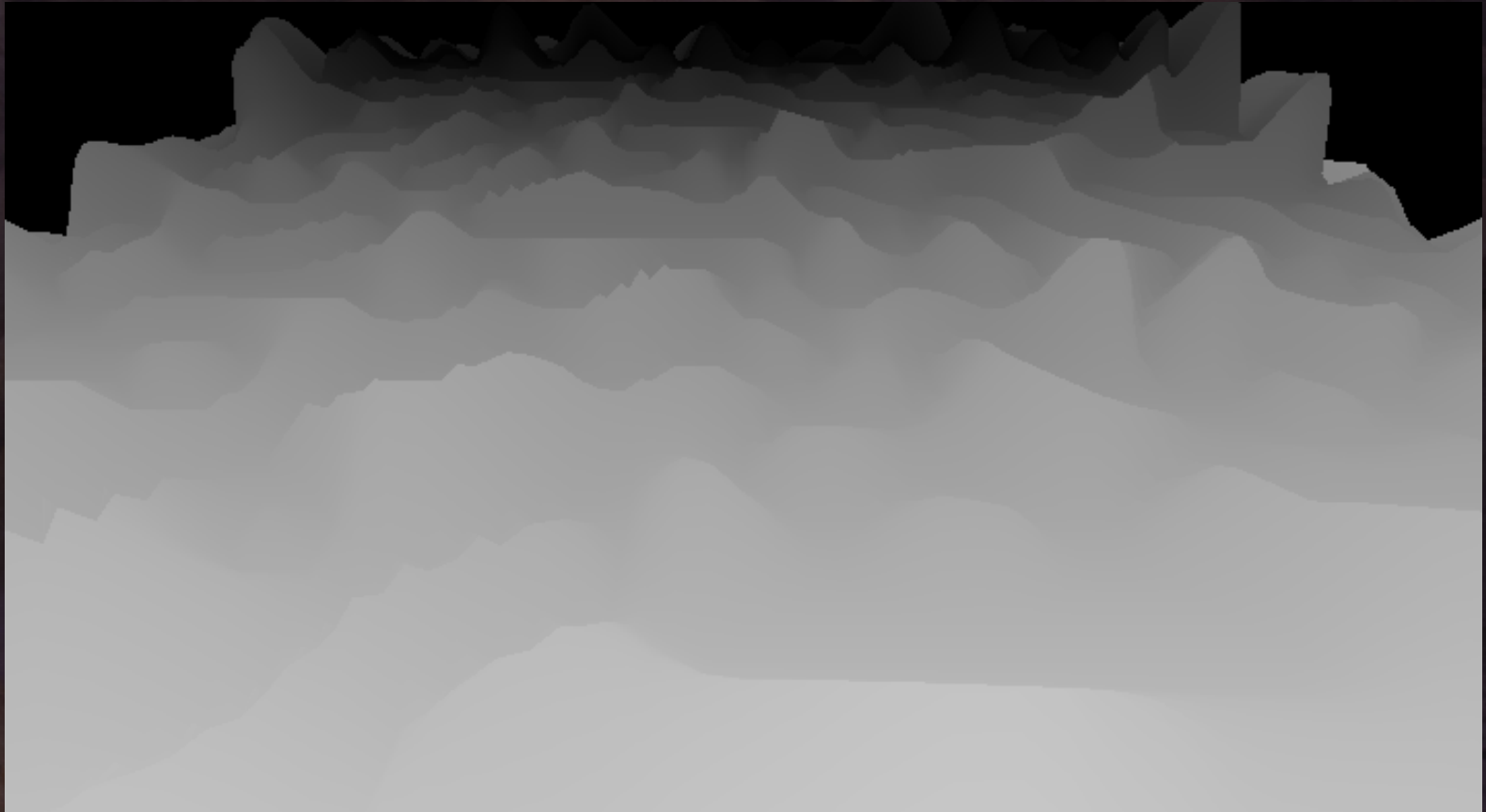
Terrain Height

Color based on height

Next Steps

- Distance based tessellation
- Displacement using tessellated vertices
- Correct shading
- Texturing
- Different Noise functions

References

- OpenGL Insights book

- Texturing And Modeling – A Procedural Approach book

- Perlin Noise (http://www.sci.utah.edu/~leenak/IndStudy_reportfall/Perlin Noise on GPU.html)

- The Little Grasshopper blog (http://prideout.net/blog/?p=48)

Thank you