CS 91T Final Project Art Portfolio Demonstration and Illustration File
Caroline Yao, Yue Zhang

Here are some terrains we generated using the following three algorithms in blender. The skeleton of the terrain is computed using the algorithms, the coloring of the terrain is generally based on real-world terrain, well-known artwork, or fictional literature.

All the coloring are done using the "Shader" editor in Blender with manipulation of the "Color Ramp", "Metallic", and/or "Roughness" value. In particular, the "Color Ramp" maps colors from range 0 to 1, so there is a 1-1 map between the actual height of the position in the terrain and the final color it gets assigned. We had a function for standardizing the height values of different nodes but we manually added a variable that allows to slightly change the height of the entire terrain just for coloring purpose. Both "metallic" and "roughness" value help change the texture of the surface and usually give a more shiny look of the terrain if handled correctly.

Please see at the end for the triangulated mesh in 3D that demonstrates the skeleton of the terrain.

Perlin Noise

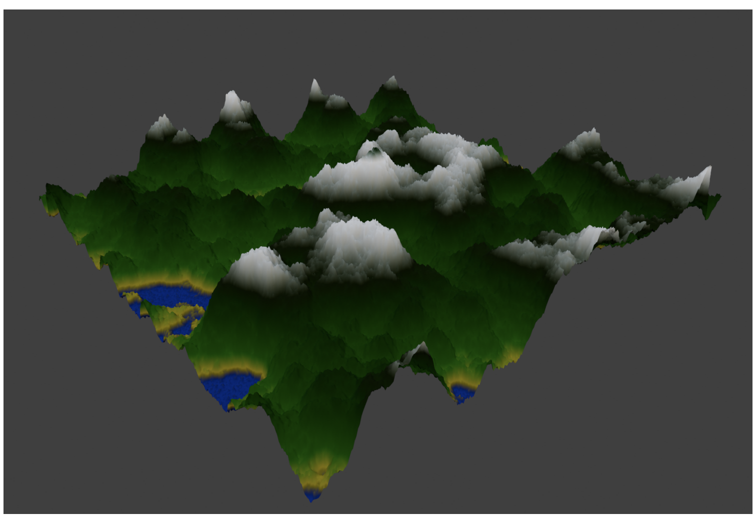

Figure 1: Forest with clouds

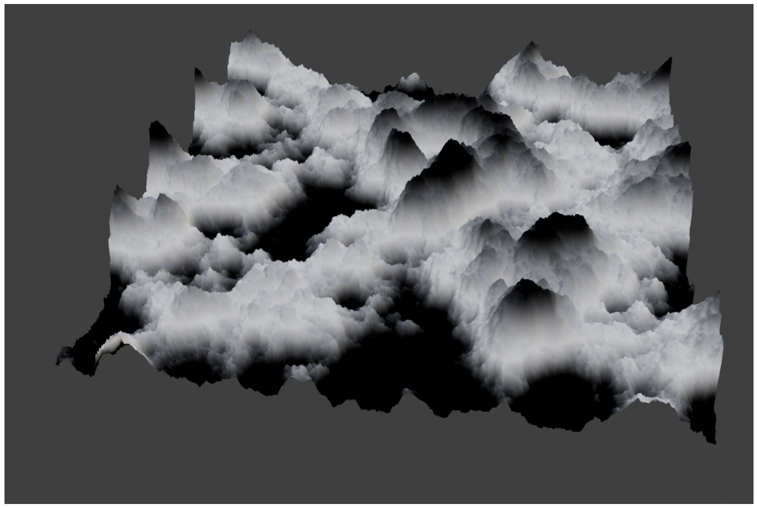

Figure 2: Shan Shui(traditional Chinese painting) style landscape

Diamond-Square

The terrains generated with Diamond-Square Algorithms are less adaptable to imitate different kinds of landscapes. We show two examples following the traditional idea of “mountain with snow above the sea” and one example modeling moon’s surface texture.

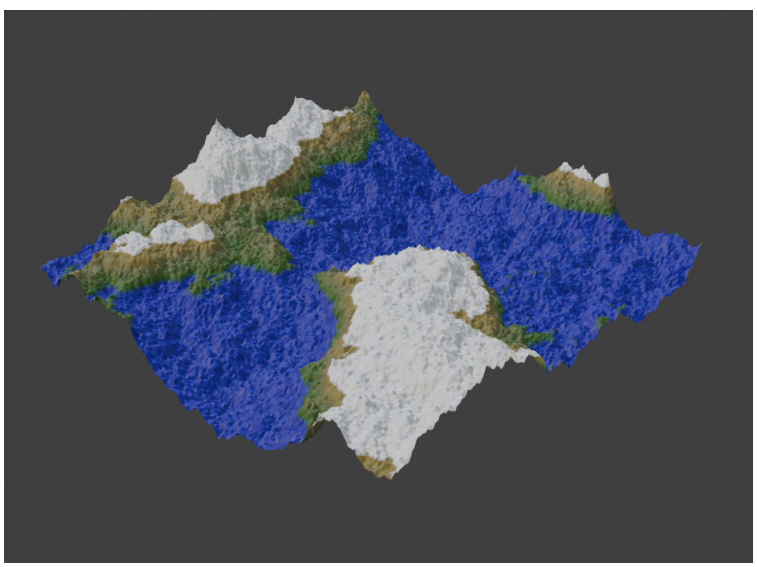

Figure 3: Scattered Mountain with Snow above the Sea 1

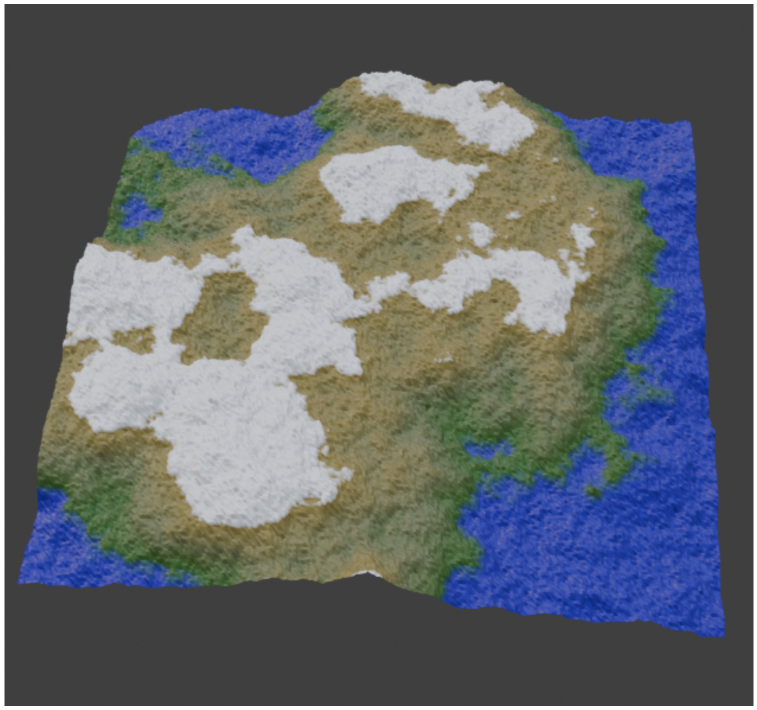

Figure 4: Concentrated land above the Sea 2

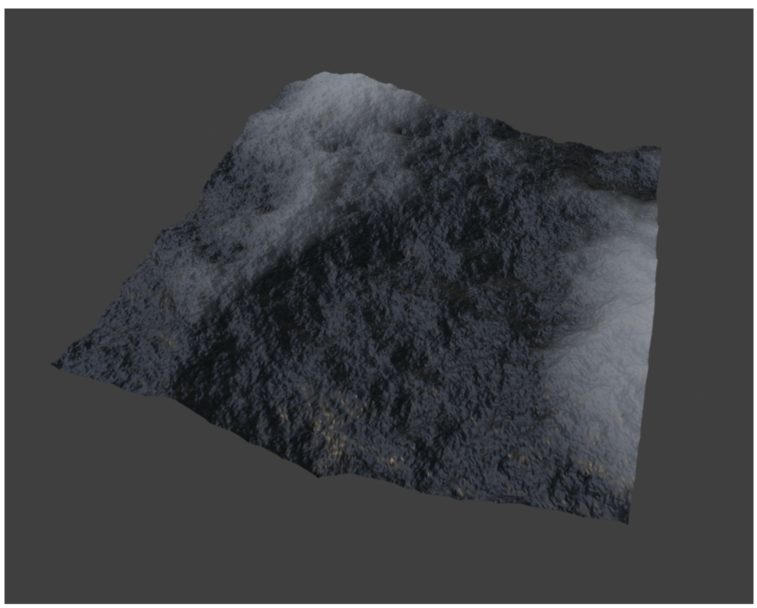

Figure 5: Moon Surface

Worley (Voronoi) Noise

For the Worley noise, we also added the feature that cuts off the surface below a certain level so that we will have "bumps" over a flat land instead of "bumps" and "pits" all over the terrain. We believe that this help beautify the terrain as we color it.

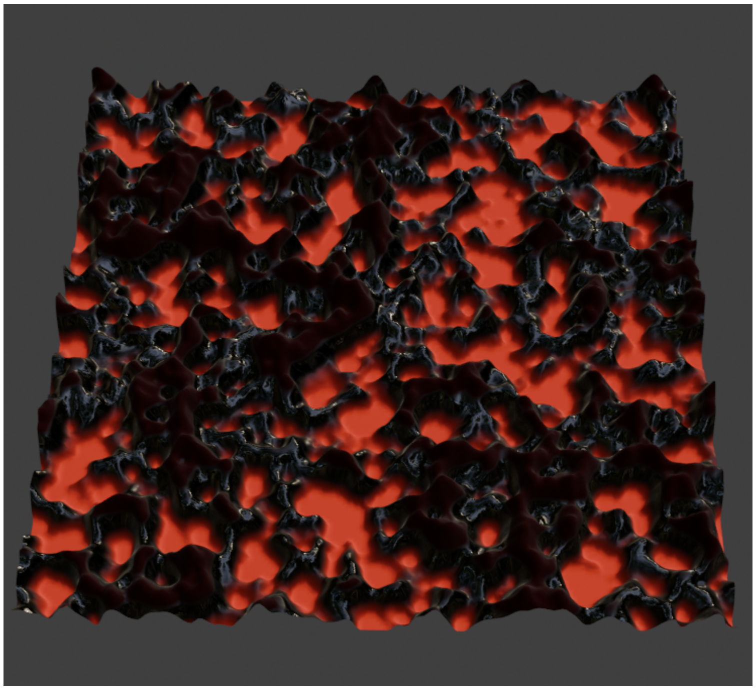

Figure 6: Volcano / Fictional depiction of inferno

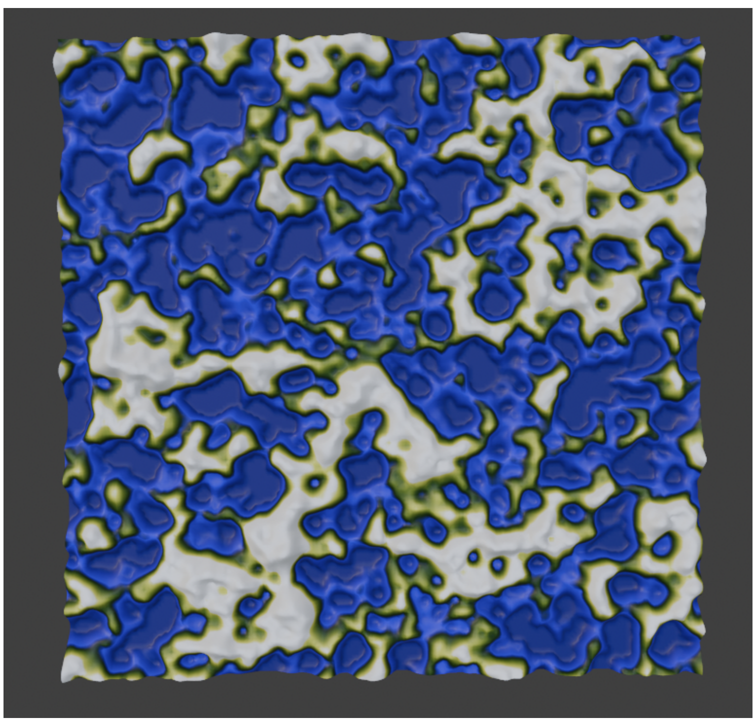

Figure 7: Sea with reefs and many little islands

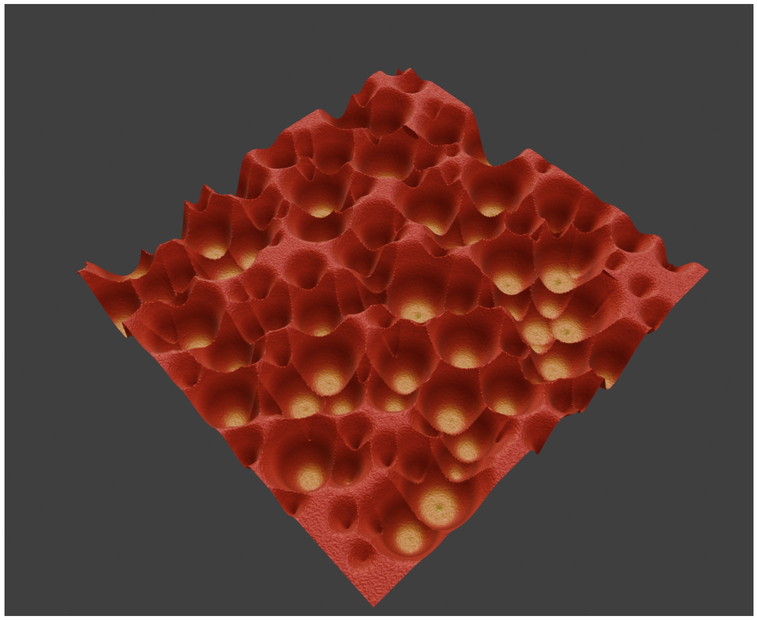

Figure 8: Human Tissue like surface generated using modified (weighted)Worley Noise with lower threshold and extra smoothing

3D Mesh under Triangulation Examples

Since all three terrain generation algorithms rely on square grids that will be later translated to 3D with vertices elevated to different heights, it is relatively time-consuming to perform real Delaunay triangulation. Instead, we apply a pseudo-Delaunay triangulation method. That is, for each square cell of the 2D grid, we draw the shorter diagonal. Since difference in height between any two adjacent vertex is sufficiently small, so this pseudo-Delaunay triangulation should be a good approximation to real Delaunay triangulation in 3D.

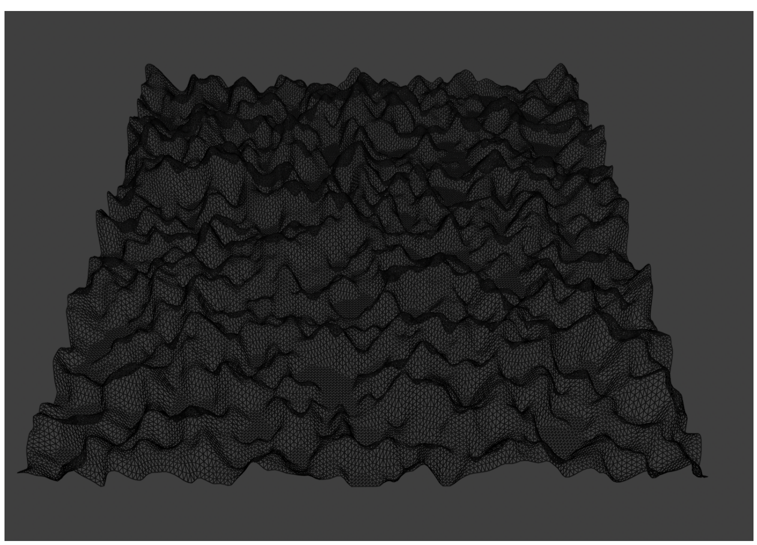

Figure 9: Pseudo-Delaunay Triangulation of Figure 6. (Voronoi Algorithm) realized in Blender with triangulation method set to Shortest Diagonal

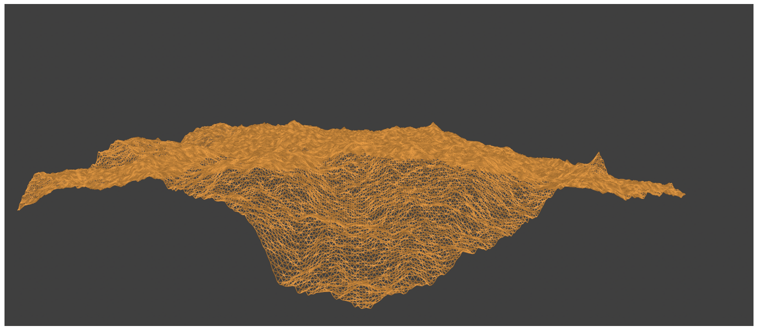

Figure 10: Pseudo-Delaunay Triangulation of Figure 4. (Diamond-Square) positioned at an angle. Highlighted for improved visibility.

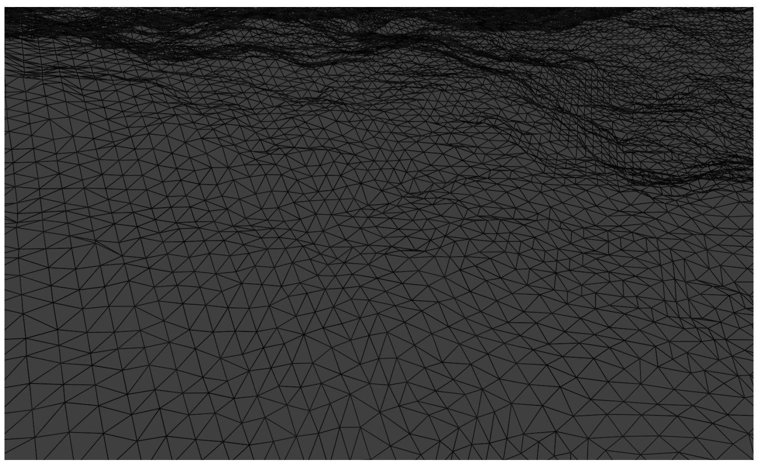

Figure 11: Part of pseudo-Delaunay Triangulation of Figure 5. (Diamond-Square) positioned at an angle. Notice that the direction of diagonals are different among cells.

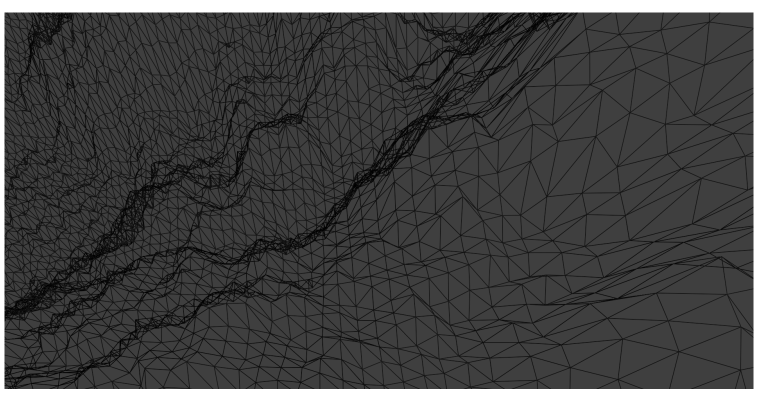

Figure 12: Part of pseudo-Delaunay Triangulation of Figure 2. (Perlin) positioned at an angle. Notice that the direction of diagonals are different among cells.