

Better than Earth

Terrain Generation

Sam Myers, Taylor Sheneman, Willem Thorbecke

What is this?

Dissatisfied with the current state of procedurally generated worlds, we set out to develop a program that generates worlds through the use of not only noise based fractal algorithms, but also evolutionary, macro-scale modeling of physical processes such as tectonic activity and hydraulic erosion. In the future we hope to implement edge wrapping, weather patterns and Holdridge life zones, in addition to scaling everything larger. Our rendering engine creates a voxel-based world, with the eventual intention to move on to a gradient-colored mesh rendering system.

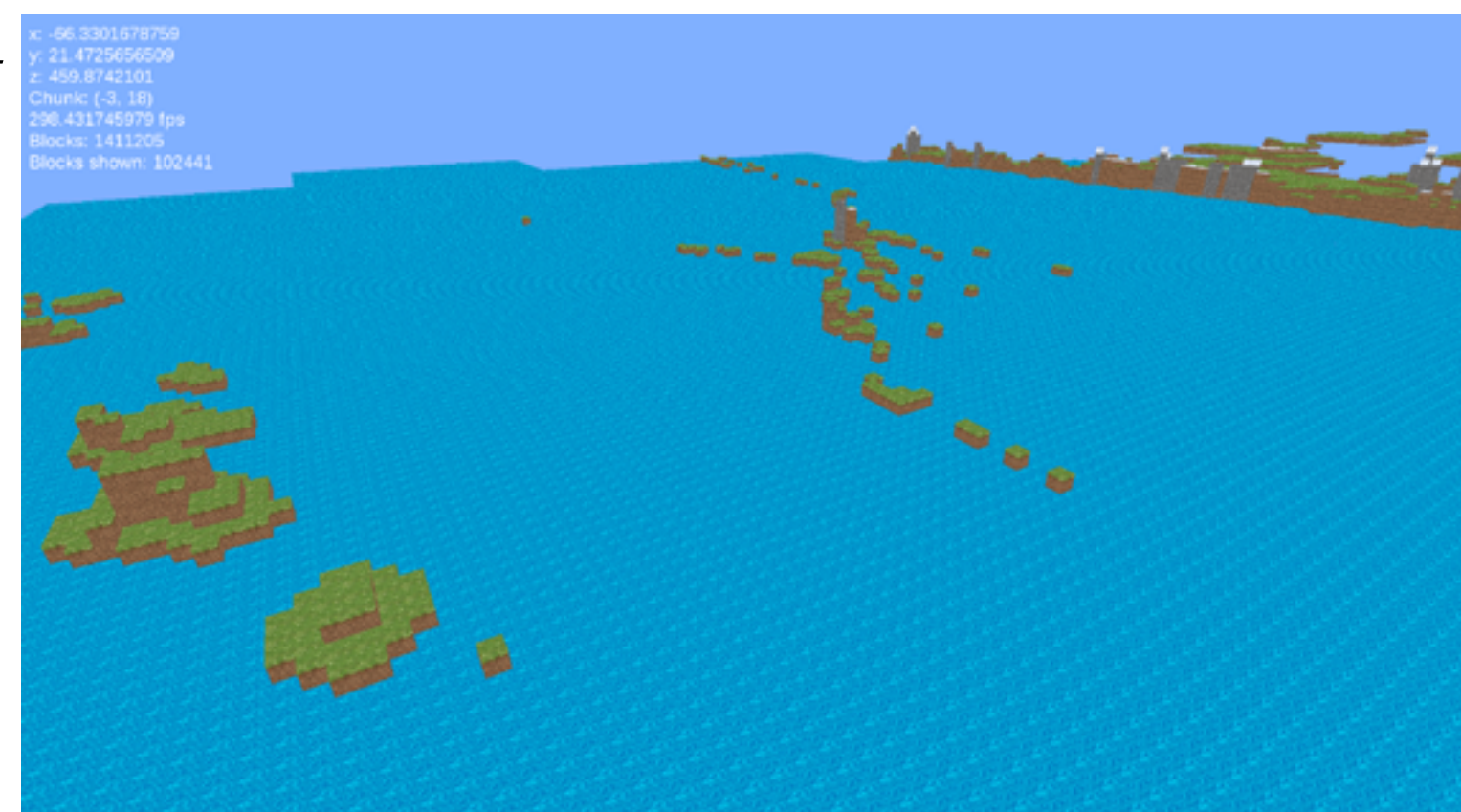
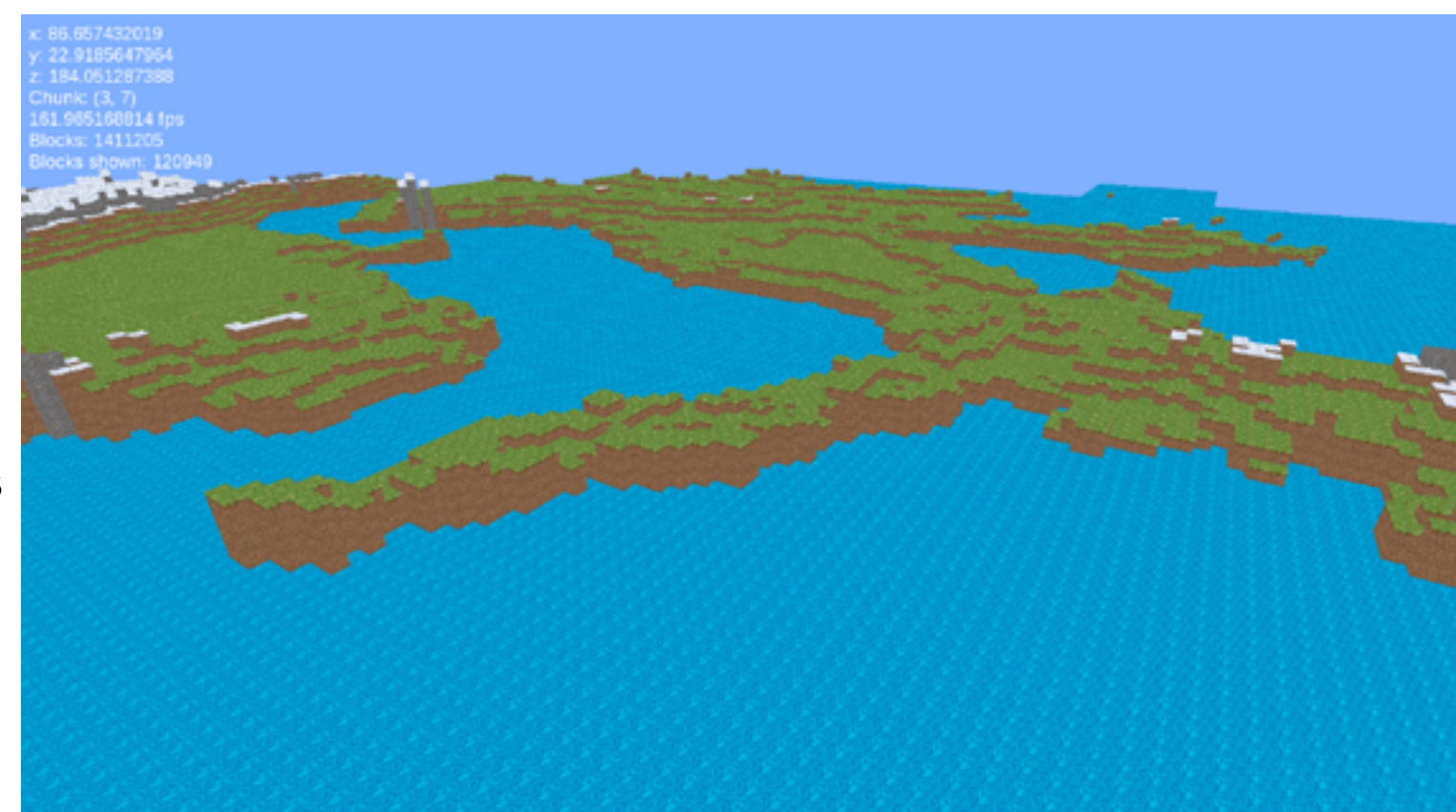
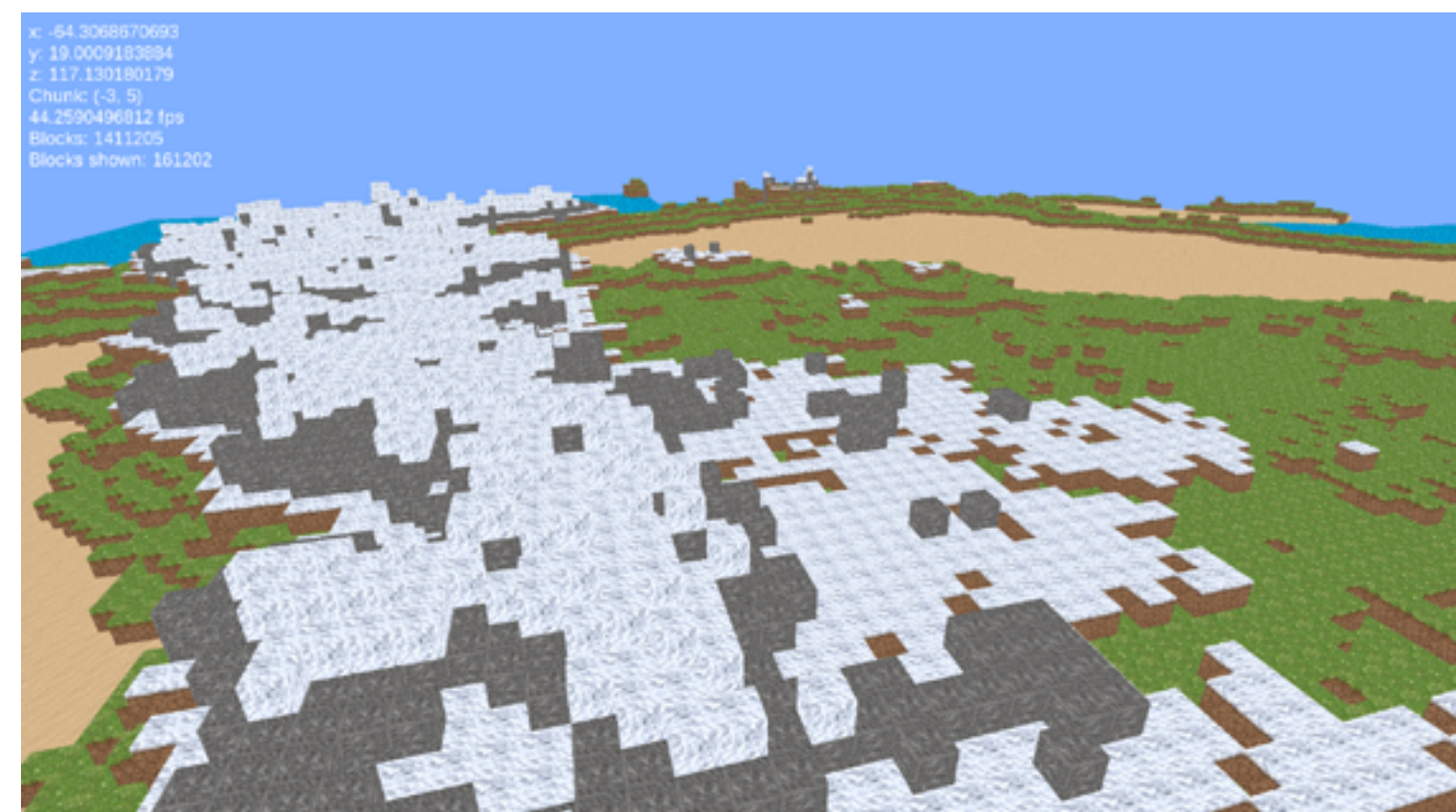
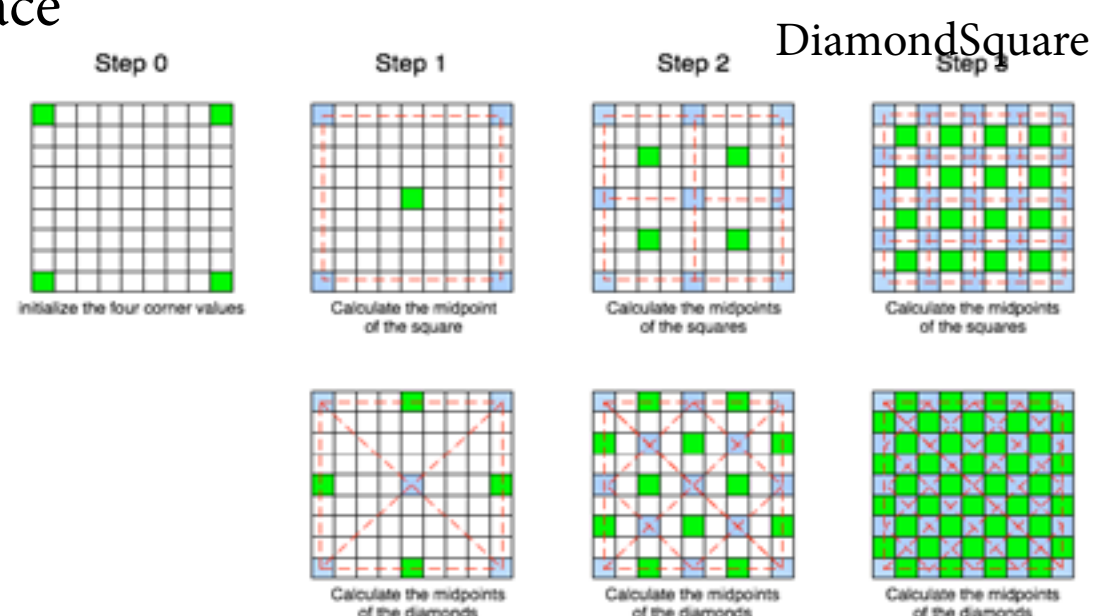
Algorithms Procedural and Evolutionary

PyPlatec: This is a library that uses plate tectonics simulations to accurately model continents and mountain ranges

Diamond Square: Produces pseudo random height maps based off of a fractal algorithm demonstrated on the right

Hydraulic Erosion: The primary mechanism of water erosion. Can be simulated with a simplified pipe model

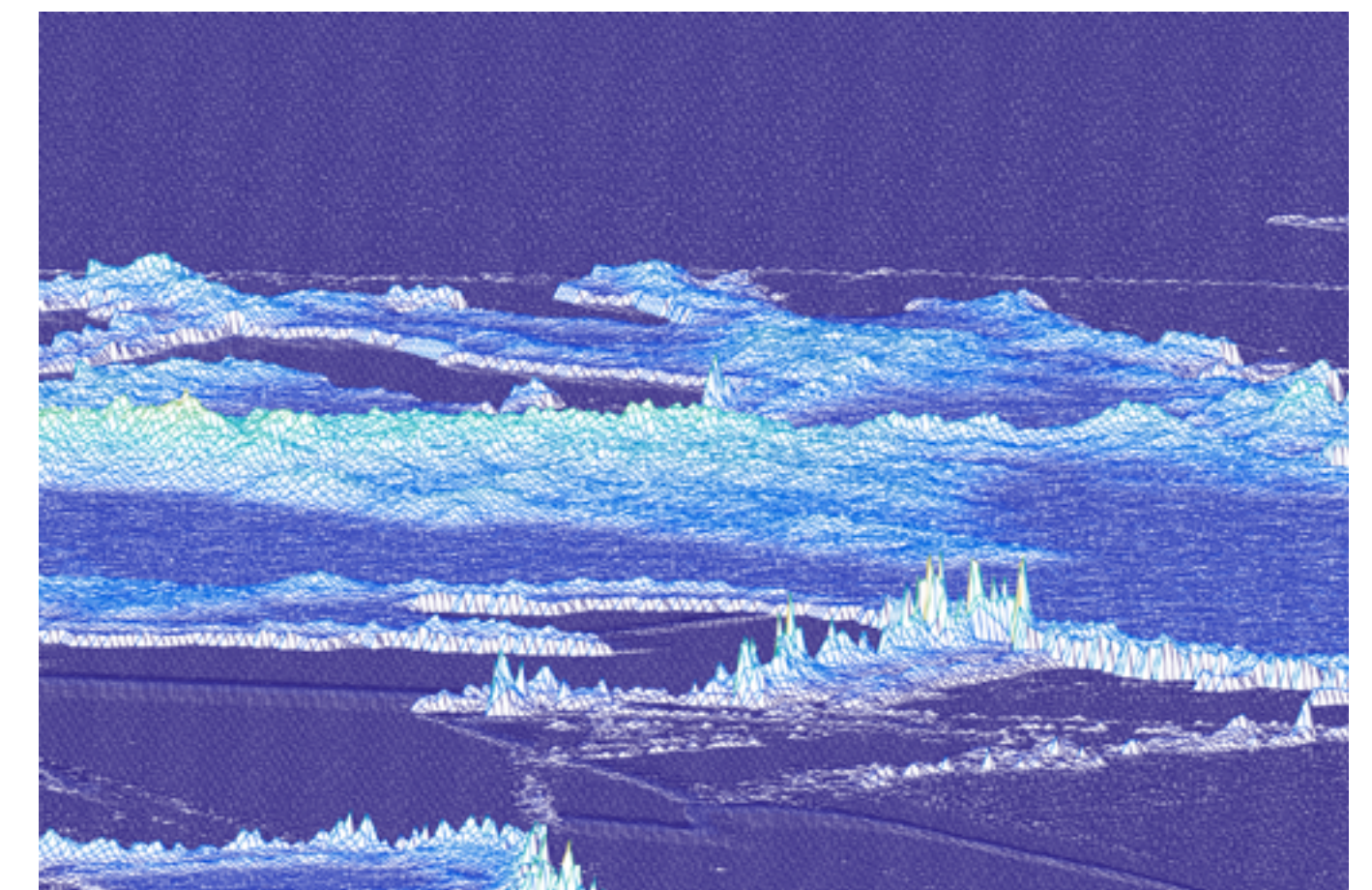
Cave Generation: Generates caves by stitching together Diamond Square derived 2D caves and extruding them into 3D space



Rendering

For the program to render the world, first the processed heightmap is converted to a dictionary of position (3-tuple)-to-block (integer ID) pairs. The world is then divided into 25-by-25-block “chunks”, which are loaded into memory as needed. When a chunk is close enough to the player, the bounding vertices for a cube centered at each position value in that chunk are calculated and stored in a batch. During each frame (at 60 FPS by default) the player’s position is re-checked, chunks are loaded or unloaded if necessary, and current blocks are drawn to the screen.

Refer to sammyers.github.io/pythoncraft for more information on PythonCraft in addition to acknowledgements



PyPlatec Rendered with Matlab