World Generation

Sam, Taylor, Will

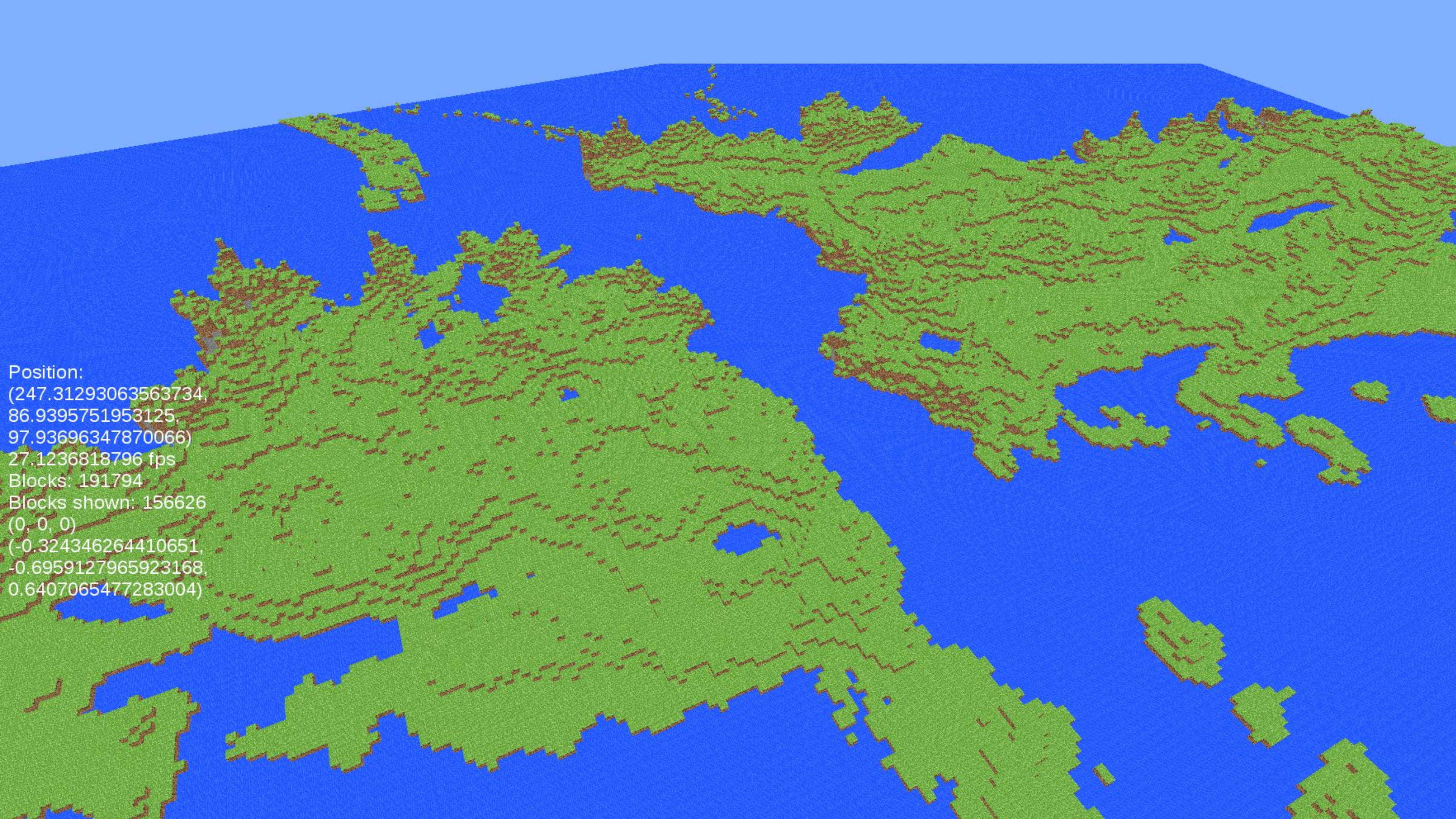
The Point:

(of the thing what is our project)

To design better computer generated terrain, using models of actual physical processes and fractal-generated detail.

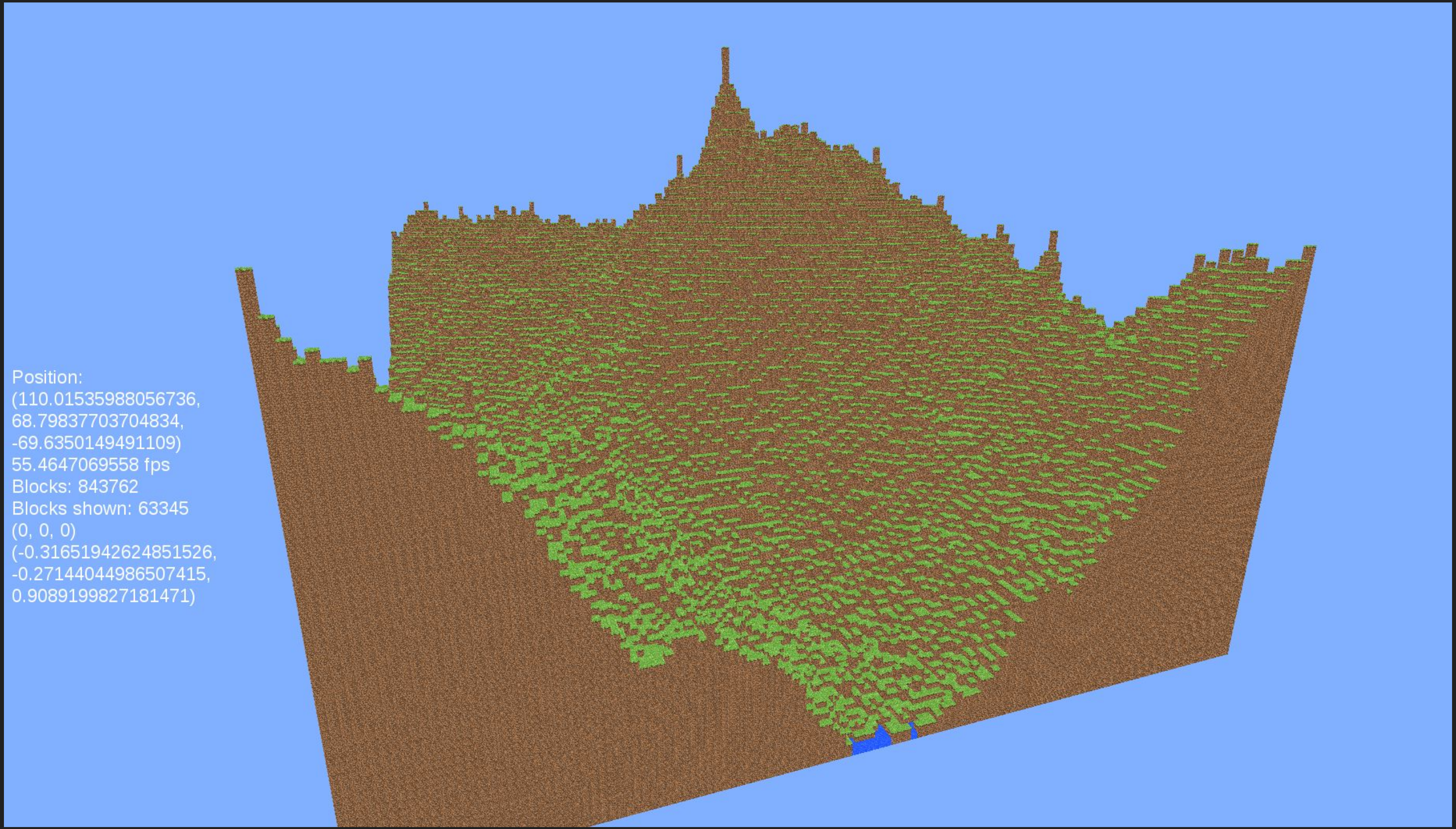
Process

- Tectonic plate simulation
- Fractal surface detailing
- Erosion and precipitation
- Latitude and altitude calculation, biome creation
- Cave generation (?)
- Send Sam block coordinates + IDs
- Render!



```
Position:
(247.31293063563734,
86.9395751953125,
97.93696347870066)
27.1236818796 fps
Blocks: 191794
Blocks shown: 156626
(0, 0, 0)
(-0.324346264410651,
-0.6959127965923168,
0.6407065477283004)
```

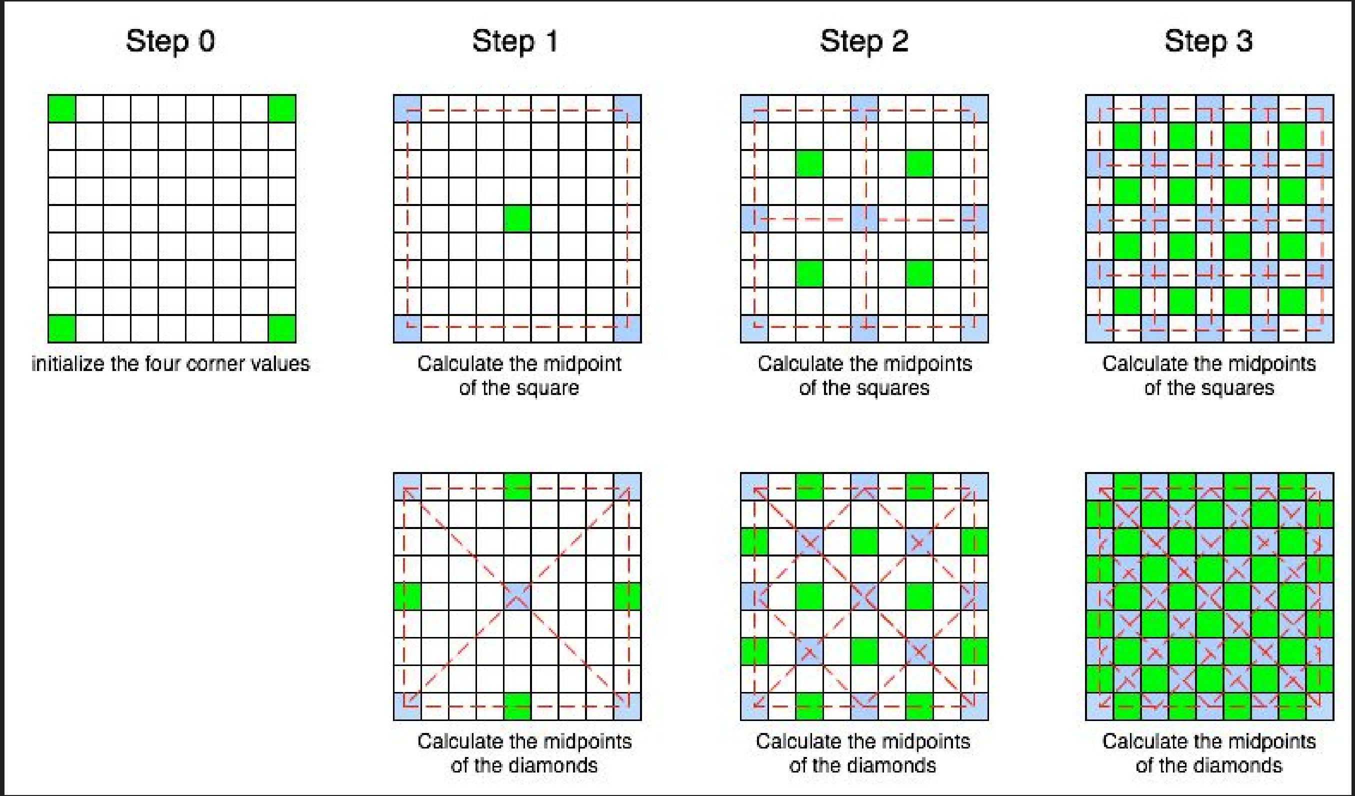
Diamond Square



```
Position:
(110.01535988056736,
68.79837703704834,
-69.6350149491109)
55.4647069558 fps
Blocks: 843762
Blocks shown: 63345
(0, 0, 0)
(-0.31651942624851526,
-0.27144044986507415,
0.9089199827181471)
```

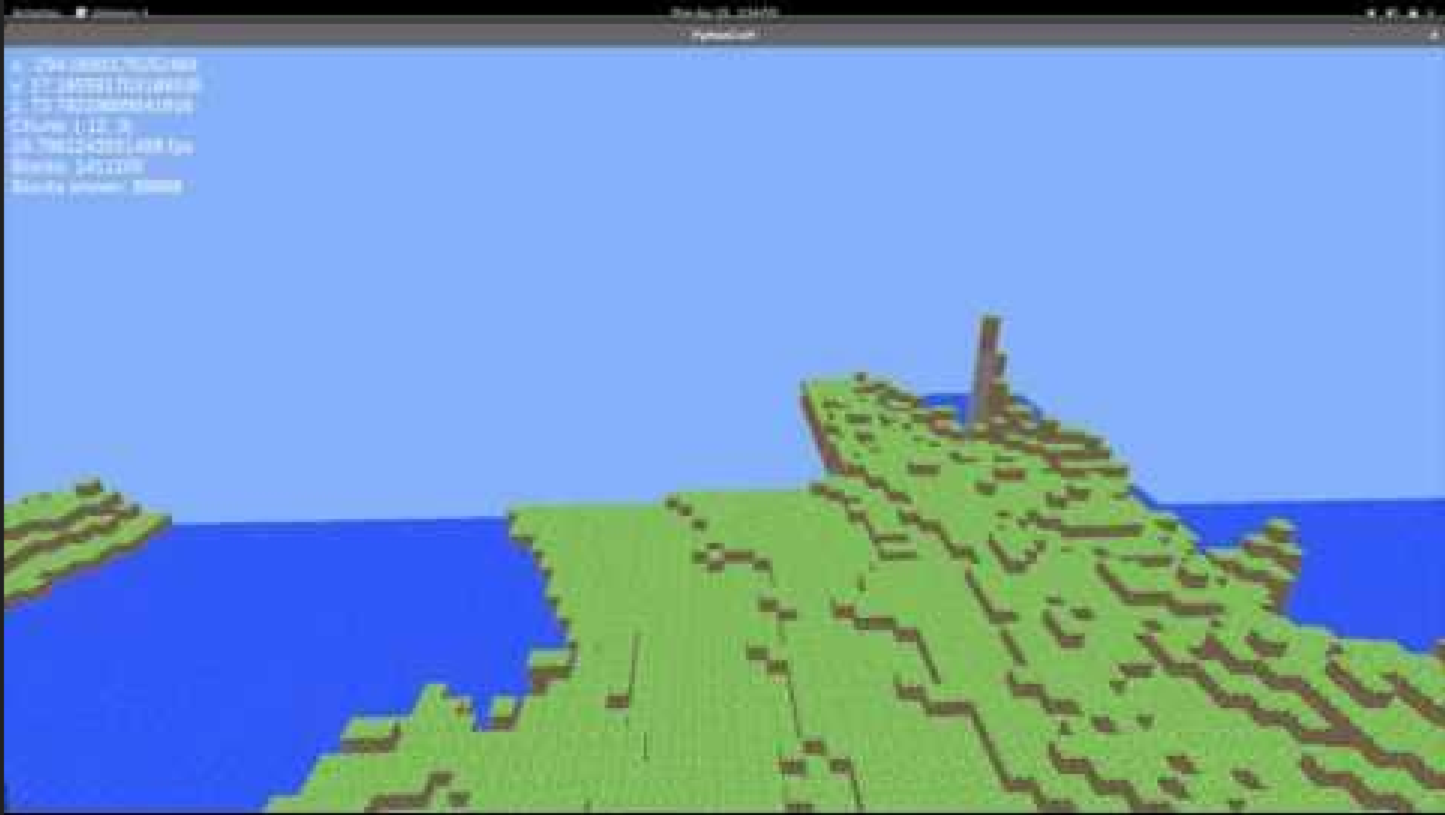
- Relatively high-res iteration of diamond square
- Unwanted ‘chimneys’
- Mostly correct implementation

How it works


Step 0
Step 1
Step 2
Step 3
initialize the four corner values
Calculate the midpoint of the square
Calculate the midpoints of the squares
Calculate the midpoints of the squares
Calculate the midpoints of the diamonds
Calculate the midpoints of the diamonds
Calculate the midpoints of the diamonds

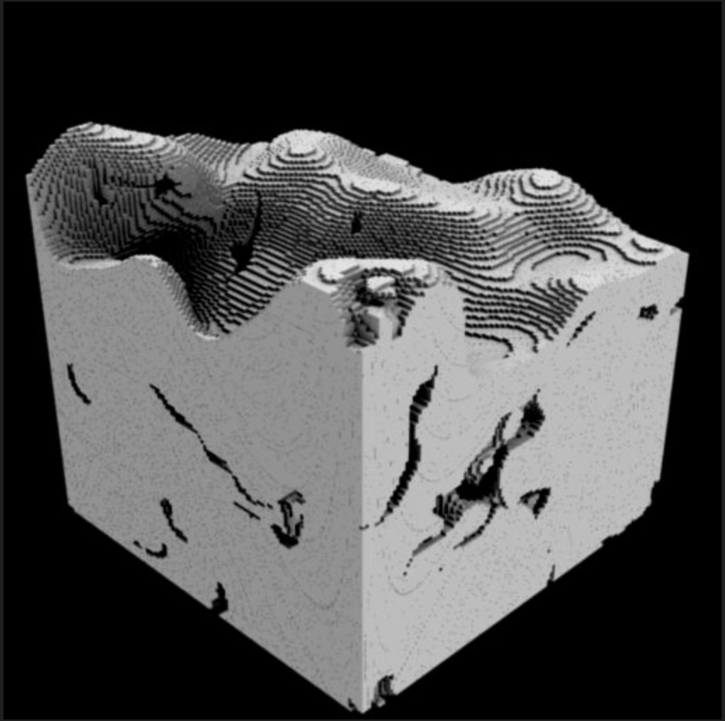
OpenGL Rendering

- Blocks mapped from position to block type
- OpenGL vertices generated dynamically
- Camera movement updated in game loop



```
x: [illegible]
y: [illegible]
z: [illegible]
Chunk: [illegible]
[illegible] fps
Blocks: [illegible]
Blocks shown: [illegible]
```

OpenGL Rendering - Version: Chunks

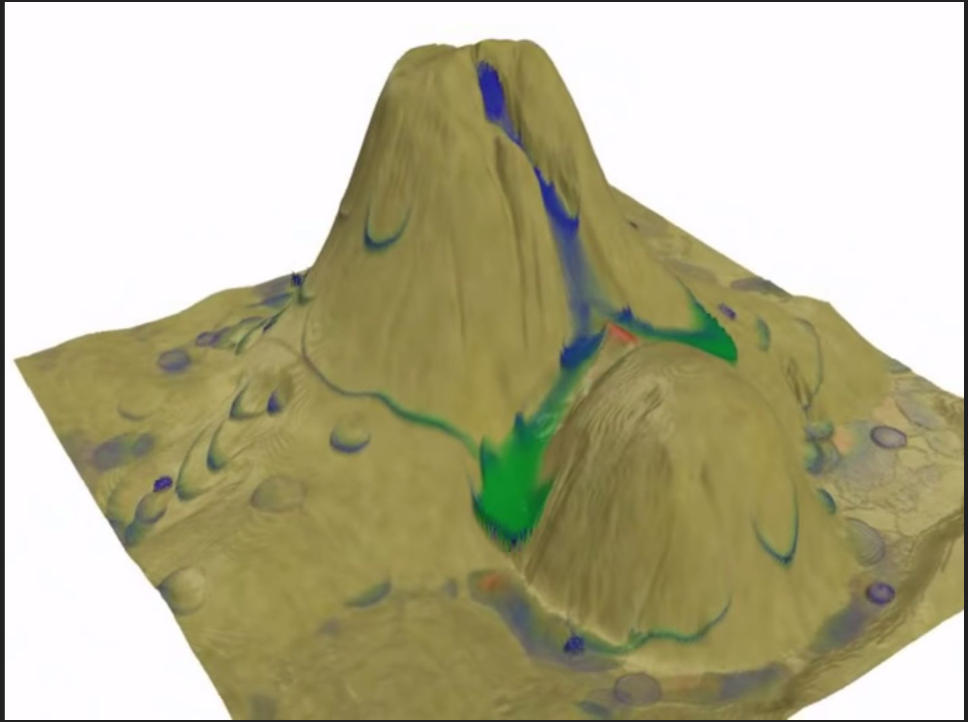


- Computers can’t handle enormous 3D matrices loaded into memory all at once

  ←Instead, the world is sectioned into 25x25 chunks, which are loaded individually based on proximity to the player

Erosion (Hydraulic Model)

- Hydraulic erosion models are pretty well-generalized
- Covers both river and rainfall erosion
- The process we're implementing (image shown) doesn't actually do fluid simulations (allows for faster processing)



Our Goal:

(we wish)

The Actual Plan:

- Erosion modeling
- Fractal detailing
- Holdridge life zones
- Rendering faster and farther! (maybe with meshes)
- File I/O for world storage