World Generation Through Tectonic Simulation

Dmytro Kosynskyy
School of Computer Science TU858

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Introduction

There are various methods of terrain generation for video games, including but not limited to Perlin noise, wave function collapse, or marching cubes. While each of these carry their own benefits and drawbacks, varying in speed, memory efficiency, and individual use cases, these methods don't usually follow realistic physics when generating the video game terrain. The goal of this project is to create a program that is able to create terrain that is based on a tectonic plate simulation that can be applied to both a fixed area map and a procedurally generated map.

Physics simulations

The physics simulation is done through a combination of rigidbody like simulation for the external hull of the plate, and softbody like simulation for the internal height mesh deformation. This is done to help seperate the two processes as they work in different ways to each other aswell as complete different objects to each other.
The height mesh can be seen in Fig 1 on the right.

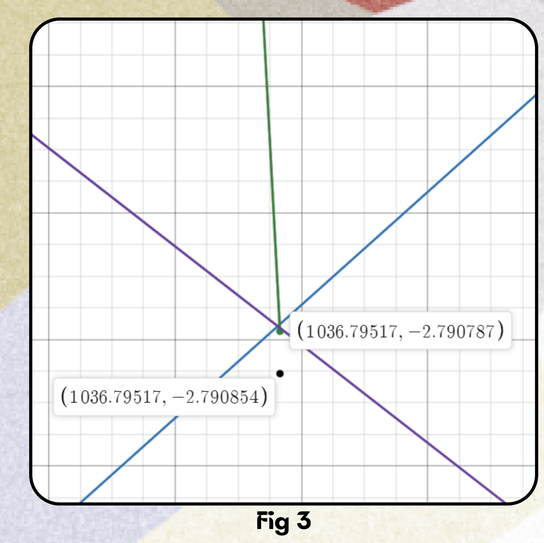

Fig 3

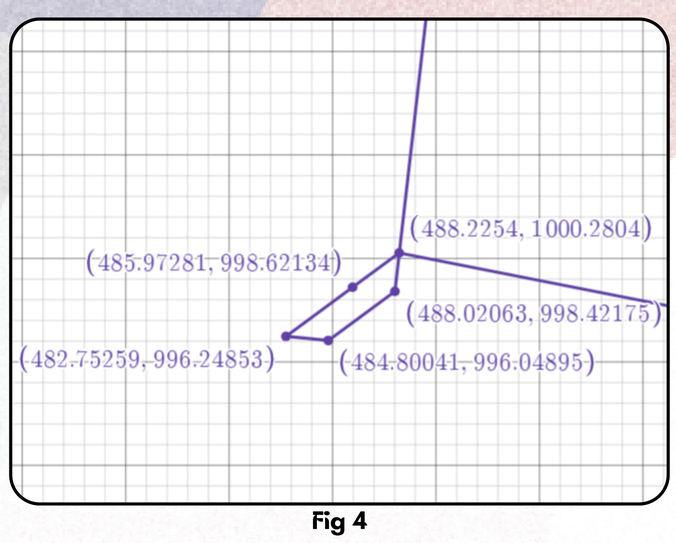

Fig 4

World Generation

The initial world generation is done through the use of Voronoi cells to create plates for the tectonic simulation. These plates are created through the scattering of seed points in the grid cells of the world. The program then finds the perpindicular bisectors through the midpoint of the seed and its neighbouring cell's seeds. This helps create the initial plates for the tectonic simulation as can be seen in Fig 2 below.

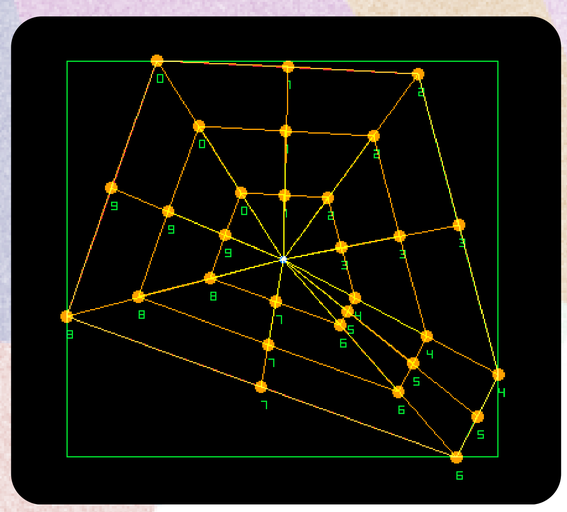

Fig 1

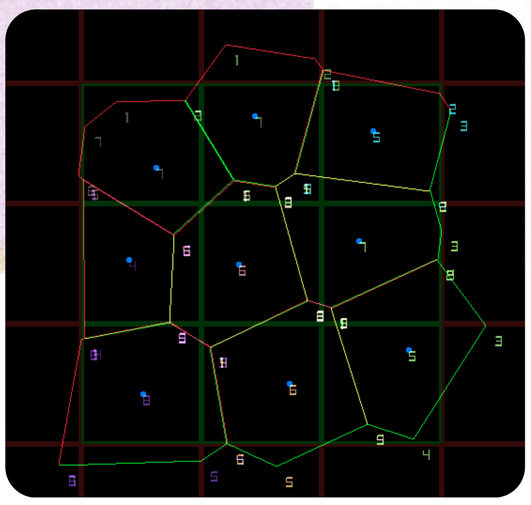

Fig 2

Pitfalls and Issues

One of the main issues that was constantly being ran into is the issue of floating point errors. As can be seen in Fig 3 on the left, the line end and the point of intersection is off by 0.000067 due to floating point errors. This error continues to grow until it causes issues such as the ones seen in Fig 4 where the plates hull has self intersected. This errors can't solved for completely but can be mitigated throught the use of angle filters, margin of error, and other options.

Uses

The main use for this program is in game development through the use of game engines such as Godot or Unity to generate terrain that relies on tectonic simulation rather than perlin noise. To do this depends on the specific game engine, for Godot it needs to be built alongside Godots code and create a gdnlib file by marking functions as external and registering them to godot.

LinkedIn

Github