Genetic Generation of 3D Models

Abstract

Anonymous Author(s)

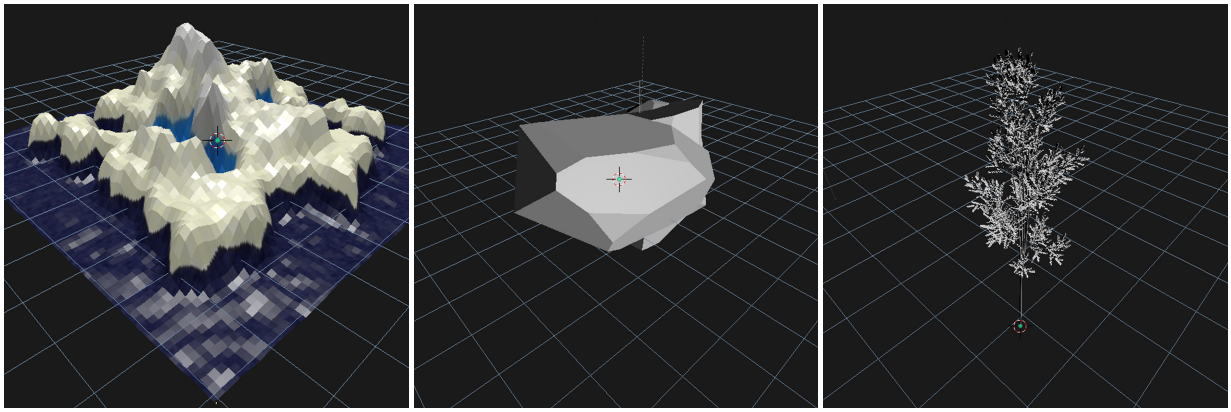

Figure 1: Example of objects generation

ABSTRACT

This abstract is a brief about a work made by a team of developers who have some interest for generative art.

KEYWORDS

LaTeX, genetic algorithm, random generation, interactive

ACM Reference Format:
Anonymous Author(s). 2018. Genetic Generation of 3D Models: Abstract. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, Article 4, **??** pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Within the context of a video game development, we wanted to generate a world which has some original decorative elements. We try to search a way to model objects we can find in the wild. The fact is that these objects seems to have either some random properties or some characteristics which had evolved over time. So, we decided to create a generator of natural objects based on genetic algorithms with random alterations, made on Blender software and with Python programming language.

1.1 Related Work

We made this project by following the work of Matthew Lewis[**?**] about the Aesthetic Evolutionary Design, especially we remade the Metavolve interface.

For the island generation, we studied the article of Amit Patel[**?**] about polygonal map generation.

2 OUR APPROACH

According to A. Michael Noll[**?**], "if 'creativity' is restricted to mean the production of the unconventional or the unpredicted, then the computer should instead be portrayed as a creative medium - an active and creative collaborator with the artist ...".

2.1 Pipeline

O. ur first approach for making these assets is to define some properties we can randomly set for the purpose of getting some original objects. The problem is to find ways to control the random aspect in such a way the user can lead the generation. So we decide in a second time to use the principle of genetic algorithm to combine the generated objects.

I. t consists of making some individuals evolving through several steps. As the biological genetic evolution the goal is to modify some characteristics for getting better and better individuals until the user find satisfying objects. These characteristics are defined as the genotype and the resulting objects are defined as the phenotype. In our case, we canâĂŹt assess this phenotype objectively, so we let the user evaluate his prefered individuals. Basing on best individuals we generate some new objects by combining their characteristics like in biology. And we go so on until the user gets the good one.

WOODSTOCK'97, 2018, Nowhere

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

Algorithm 1 Genetic Algorithm

```
Generate objects (genotypes and resulting individuals) randomly

repeat

  for each individual do

    Evaluate the individual
    if selection condition is not verified then
      Remove it
    else
      Change one of his genotype characteristics and update
      the individual
    end if
  end for
  Pick remaining objects and make new ones by crossing them
until user stop (Repeat the process as long as we want)
```

This pipeline is the same for all kind of objects. We just have to define the selection condition and the characteristics which evolve through the time.

2.2 Selection condition

We choose to put different types of selection based on the user evaluation. We let the user choose to favor quality or quantity, by keeping a certain number of the best objects or by keeping only the ones which have been rated more than a given threshold. We have also a third type of selection which uses the rating value of each object as the probality to be kept in the next round.

2.3 Generating island

To generate the island, we have to define two main maps, the height map and the biome map. The height map is generated with Perlin noise and some sinusoide. As a basic altitude map, the colors define the height of the area, dark represents deep area and light represents high one. For the biome map, we based our generation from the work of Amit Patel about polygonal map. We create a voronoi map, then we define which area is water and which is land. From this, we define a distance map from the water which is used for making the altitude and the moisture of the area. With these data and a global climate we can define the type of the area. As a result, the island generation depends on many characteristics which we can make evolve like noise, climate, moisture, height, land proportion, ...

2.4 Generating crystals and rocks

To create a crystal, we start from some *icospheres* and perform a bunch of cuts using Blender's bisect tool. The genotype of the crystal stores for each of his parts the information of each cut plane. Then, we adjust the scale, the orientation and the position of each crystal part to make them born from a starting point at the bottom of the crystal.

Rocks are simply crystals with just one part, that are more rounded, with a bigger number of cuts.

2.5 Generating Trees

Tree generation is performed by the Blender add-on *slapping tree*[**?**]. The genotype behaves like a remote that controls all the plugin's parameters.

3 FUTURE WORK AND LIMITS

3.1 Issues

- Generation is yet made too randomly. It's difficult to predict what's the next generation will be made of.
- Generation is too slow for the islands and the trees

3.2 Improvement

- We have to fit the characteristics of individuals for having palpable evolution and combination. We can also add an option to get only crossed individuals in the next generation.
- Object generation is synchronous and can take some time when the number of objects gets high. It could be more fluid by using asynchrone generation.

3.3 Future work

We can easily add other types of assets to be generated by just implementing their genotype generation and evolution. We plan to add flowers and ruins to our asset generator to help us in our video game projet. Also, terrain generation will be improved to get along with our artistic direction. We plan to make a entire landscape generation, by adding the differents other object in the island. With the definition of each biom, we can make, for instance, some forest or some cities.

REFERENCES

[] Matthew Lewis. 2008. Evolutionary visual art and design. In *The art of artificial evolution*. Springer Berlin Heidelberg, 3–37.

[] A. Michael Noll. 1967. *The digital computer as a creative medium* (10 ed.). Vol. 4. IEEE spectrum. 89–95 pages.

[] Amit Patel. [n. d.]. Polygonal Map Generation for Games. ([n. d.]). http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/

[] Andrew Hale (TrumanBlending). [n. d.]. *Blender Plugin Sapling Tree.* https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Curve/Sapling_Tree.