Semi-Random Terrain Generation

Mariatta Wijaya David Callele

Department of Computer Science
University of Saskatchewan

Abstract

We present a voxel based system for semi-random terrain generation. Cellular automata rules guide growth, erosion, and transformation phases. Emergent behavior of the cellular automata gives the appearance of randomness at the micro scale while producing repeatable results.

Key words: Voxel, terrain generation, cellular automata

1 Introduction

Asset creation is rapidly becoming the most costly element in video game production. Player expectations for virtual worlds, commonly referred to as *levels*, are constantly growing. The return on investment is questionable: portions of levels, sometime even entire levels, are never seen by many players yet if they are *not* included, the game is likely to receive weaker reviews.

This work presents a system for semi-random terrain generation. At the macro level, the terrain is strongly patterned – in this case, the terrain generation process is initialized with a 2D maze construct. At the micro level, the terrain presents the player with the appearance of natural randomness. Ideally, the player would be unable to detect the maze within the final terrain.

2 Related Work

While there are many possible approaches to terrain generation, we seriously considered only mesh and voxel based approaches. Mesh techniques, such as subdivision surfaces, multi-resolution meshes, and adaptive tessellation, typically deform an initial planar mesh to form the terrain. While we could achieve our goals using these techniques, the numeric complexity of these approaches (particularly with respect to managing texture coordinates for scenarios such as caves or tunnels) precluded them from further consideration.

The second approach is voxel based [2], [1], [3]. A voxel approach has the benefit of being a discrete approximation, the associated mathematics are in **N**. Texture coordinates are always locally scoped – with proper planning, all textures will tile independent of orientation. Unfortunately, a voxel approach generates significantly larger data sets than a mesh based approach and visualization can be challenging for large data sets.

In production, a combination of techniques is in order. The user can generate a discrete approximation to the terrain with our voxel based system. Then, as a final optimization phase, a continuous surface mesh could be generated from the discrete approximation [4].

3 Requirements

The system must automatically generate a discrete approximation to outdoor terrain for use in a real-time video game. We must be able to control the macro level features of the terrain. At the micro level, the system should generate terrain that appears similar to the natural variation found in the physical world. The terrain generation process must be repeatable – for a given set of inputs, we must generate the same sequence of outputs. True randomness is *not* acceptable for we must be able to analyze, play test, and verify the suitability of the resulting terrain. Finally, the terrain should be *interesting* to a player, the player should not be able to detect any patterns in the results.

4 The Terrain Generator

Our solution is a three pass terrain generator that applies Cellular Automata (CA) rules to create the terrain out of fundamental building blocks (Figure 1(e)). Each building block is bounded by the volume of a unit cube and can be in one of four possible orientations.

The system is seeded with an initial maze layout (Figure 1(a,c)). As the terrain is generated, the walls of the maze are transformed into hills and mountains. During the GROWTH phase, the walls of the maze are created by adding cubes beside existing cubes and stacking cubes on top of existing cubes. The resulting structures are strongly pyramidal in nature. During the EROSION phase, cubes are removed from the terrain to provide greater variation (Figure 1(g)). During the TRANSFORMATION phase, the type and orientation of the cubes on the surface of the terrain are changed to more closely approximate natural randomness.

The current system has approximately 100 CA rules. The application of each rule is controlled via an associated probability attribute. The rules are stored in an XML

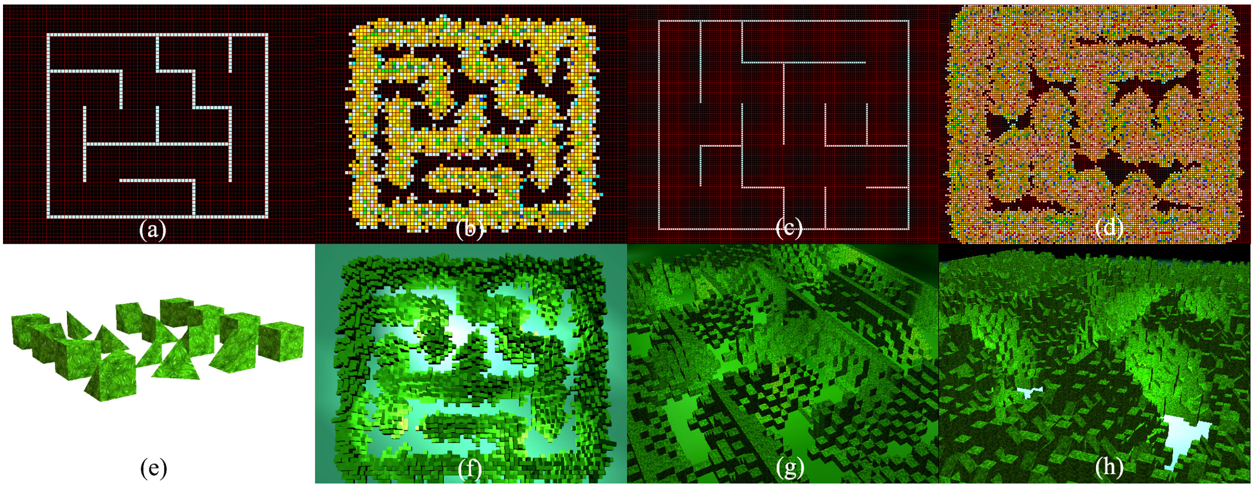

Figure 1: (a,b) Small Maze seed and final, (c,d) Large Maze seed and final, (e) Building blocks, (f) Small Maze top view, (g,h) Perspective view, Small Maze and Large Maze

format to promote experiential investigation.

CA are well-suited to our requirements. The macro scale attributes of the terrain are readily achieved by simple growth automata. The desired micro scale attributes appear as a byproduct of the *emergent behavior* exhibited by the interaction of so many cellular automata rules.

The emergent behavior comes at the price of reducing fine control over the resulting surface. However, if we assume that the output of this system is only a discrete approximation to the final terrain, any micro control deemed necessary can be achieved in subsequent operations.

5 Results

Typical results are shown in Figure 1. Figure 1(a) is the seed maze for a low resolution terrain. Figure 1(b) is a top view of the resulting CA output and Figure 1(f) is a 3D visualization of the same data. Figure 1(c,d) are the seed maze and top view of a high resolution terrain.

Figure 1(g) shows the low resolution terrain at the end of the erosion phase. Figure 1(h) shows the final result of the high resolution terrain generation of Figure 1(d).

6 Conclusions and Future Work

We have demonstrated the feasibility of generating semi-random terrain using voxel techniques controlled by cellular automata rules. The resulting terrain is a discrete approximation, suitable for conversion to a continuous surface or for use as inspiration to level designers and level modelers.

Since the CA rules are stored in XML, modifying them is easy which promotes experimentation and rapid prototyping.

In the future, we want to add more building blocks and their associated rules – the next phase will have a dozen geometric primitives and support for trees.

Finally, we will add mapping the discreet terrain to a continuous surface and strongly consider integration into a game engine.

Acknowledgements

Our thanks to nVIDIA for their support of our research.

References

[1] Stephane Gobron and Norishige Chiba. 3D Surface Cellular Automata and Its Applications. *The Journal of Visualization and Computer Animation*, 10:143–158, 1999.

[2] N. Greene. Voxel space automata: modeling with stochastic growth processes in voxel space. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 175–184, New York, NY, USA, 1989. ACM Press.

[3] Sidney W. Wang and Arie E. Kaufman. Volume sculpting. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–157, New York, NY, USA, 1995. ACM Press.

[4] John R. Wright and Julia C. L. Hsieh. A voxel-based, forward projection algorithm for rendering surface and volumetric data. In *VIS '92: Proceedings of the 3rd conference on Visualization '92*, pages 340–348, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.