Semi - Random Terrain Generation

Mariatta Wijaya **David Callele**

University of Saskatchewan, Department of Computer Science
176 Thorvaldson Building
110 Science Place
Saskatoon, Saskatchewan, S7N 5C9 CANADA

Abstract

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

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 scale, the terrain is strongly patterned – in this case, the terrain generation process is seeded with a 2D maze construct. At the micro level, the terrain presents the player with the appearance of natural randomness. Ideally, the player would not be able to readily detect the maze within the final terrain.

Background and Related Work

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.

A voxel approach [1,2,3] has the benefit of being a discrete approximation. Texture coordinates are always locally scoped – with proper planning, all textures will tile independent of orientation. Unfortunately, this approach generates significantly larger data sets than a mesh based approach.

In production, a combination of techniques is in order. We 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].

Tools and Technologies

Cellular Automata processor: **Mirek's Cellebration (MCell)**
Cellular Automata rules representation: **XML**
Terrain Visualization : **Virtools Dev**

Requirements

Semi-Random

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 analyse, play test, and verify the suitability of the resulting terrain.

Maze

The terrain must be interesting to the player. The player should not be aware of any pattern in the results. A maze-like terrain motivates players to traverse and explore the level.

Creation of a new maze can be automated using a random maze generator. An example generator can be found at:
http://www.mazeworks.com/mazegen

Three-Dimensional Building Blocks

Figure 1: Building blocks

Each building block is bounded by the volume of a unit cube and can be in one of four possible orientations.

The Terrain Generator

Our solution is a three pass terrain generator that applies cellular automata rules to create the terrain out of fundamental building blocks (**Figure 1**). The system is seeded with an initial maze layout. As the terrain is generated, the walls of the maze are transformed into hills and mountains.

Growth Phase
A new cube is stacked on top of existing cube.
A new cube is added into an empty cell which has an active neighbour.

Erosion Phase
A cube is deleted during this phase to vary the heights of the terrain.

Transformation Phase
Cubes are transformed into other shapes, such as slopes.

Growth and **Erosion** are alternately applied for the first N generations. After N generations, only **Transformation** rules are applied.

Rule application is controlled via probability attributes.

Results

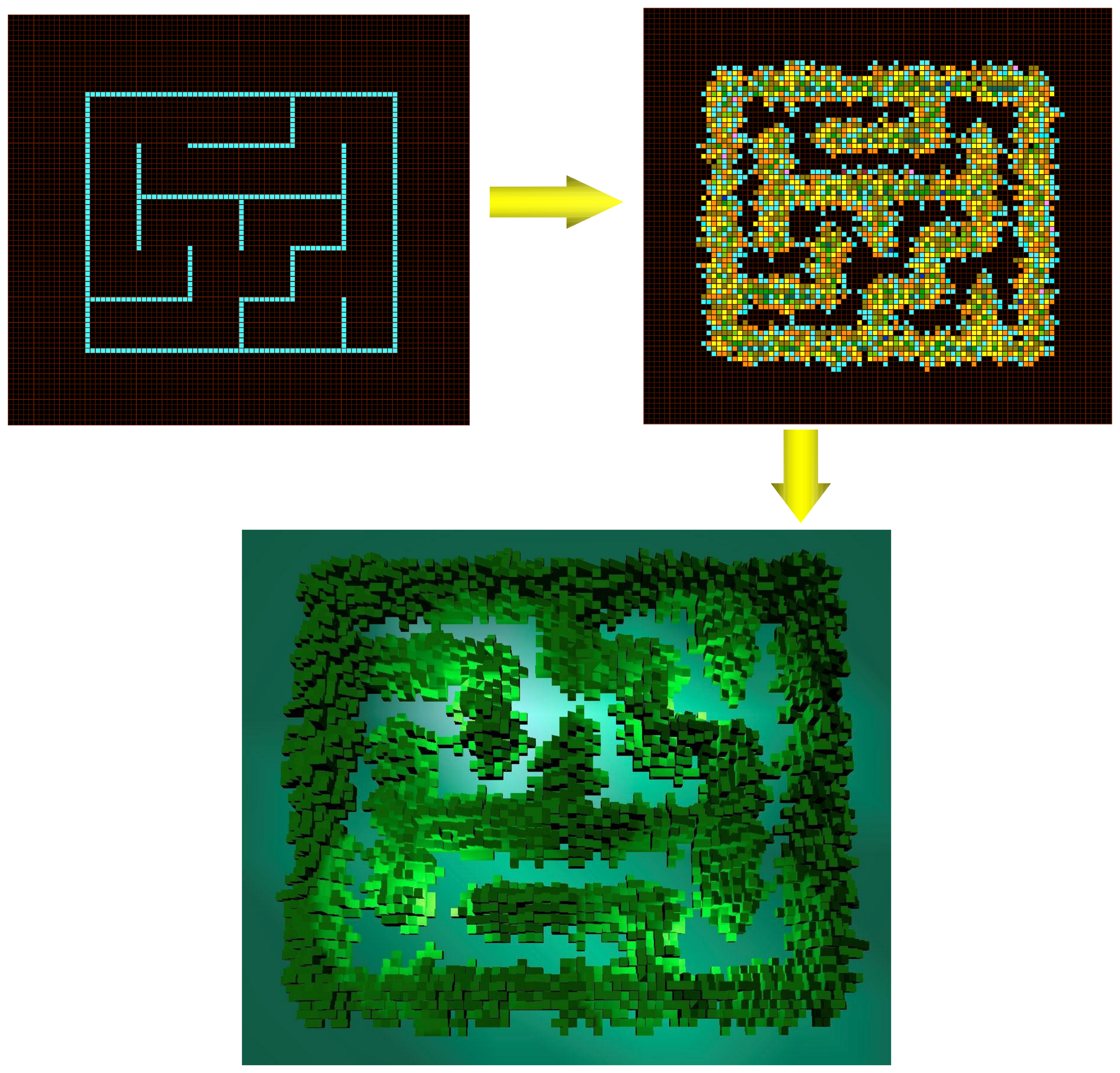

Figure 2: Transforming a small resolution maze into a terrain.

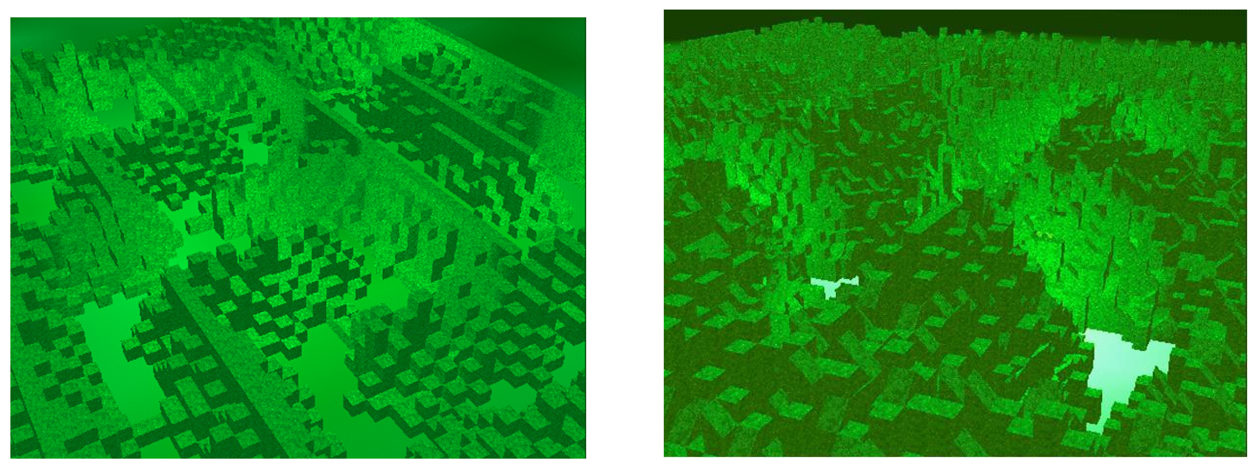

Figure 3: Perspective views of the final terrain.

Conclusions and Future Work

Creation of new terrain using the approach described in this project is inexpensive. New seed points can be automatically generated. Experimentation with new cellular automata rules can be performed by modifying the corresponding XML files. None of these steps require source code recompilation.

The next phase is to add more building blocks and their associated rules to the system to improve the realism of the generated terrain. The system can then be integrated with a game engine.

References

[1] Stephane Gobron and Norishige Chiba. 3D surface cellular automata and it's 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.