

# Real-time spatial growth model map generation

Ruud op den Kelder

January 22, 2009

## **Abstract**

The abstract of Real-time spatial growth model based map generation

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Procedural Modeling . . . . .	3
1.2	L-systems . . . . .	3
1.3	Map generation . . . . .	3
1.3.1	spatial growth based . . . . .	3
1.3.2	Interactivity . . . . .	4
1.4	Required Input . . . . .	4
1.5	Adding simple gameplay properties . . . . .	4
1.6	The growth model map generator tool . . . . .	4
<b>2</b>	<b>Method: realtime spatial growth models</b>	<b>4</b>
2.1	dynamic structures: polygonal surfaces . . . . .	4
2.2	l-systems and map l-system type spatial growth patterns . . . . .	6
2.3	Real-Time spatial growth Models For Outdoor Areas . . . . .	6
2.3.1	Land . . . . .	6
2.3.2	Water . . . . .	8
2.3.3	Roads . . . . .	8
2.3.4	Afforestation . . . . .	8
2.4	Real-time spatial growth models For Indoor Areas . . . . .	8
2.4.1	Caves . . . . .	8
2.4.2	Canarian Aboriginal style homes . . . . .	8
<b>3</b>	<b>Organic map generation tool</b>	<b>8</b>
3.1	Modifiers . . . . .	8
3.1.1	Spatial modifiers . . . . .	8
3.1.2	Timeflow modifiers . . . . .	9
3.2	User controlled map generation . . . . .	9
3.3	automatic map generation . . . . .	9
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Tools used . . . . .	9
4.2	Application Class Structure . . . . .	9
4.3	User Interface . . . . .	10
4.4	Communication system between models . . . . .	10
<b>5</b>	<b>Results</b>	<b>10</b>
<b>6</b>	<b>Conclusion</b>	<b>10</b>

# 1 Introduction

**Summary:** *Introduction zo snel mogelijk maken. saai werk en beschrijven van context /previous work geeft goeie basis om verder op te werken.*

**TODO:** *snel maken, dan ben je er voorlopig vanaf*

This thesis discusses and utilizes techniques from geometric and biological modelling for the procedural generation of 2D environments, with an emphasis on game environments. The techniques used in this paper are greatly influenced by the field of procedural city generation and the more mature field of procedural generation of plants and trees.

The research which preceded this document strived to construct a intuitive organic modelling method for the creation of 2D maps. The central focus of this research has been on adapting, constructing and implementing real-time *communicating* growth models using two types of dynamic structures. These two types can be described as network type structures, using l-systems and map l-systems, and 2D 2 manifold shaped structures, using techniques derived from cellular automata and cell systems.

## 1.1 Procedural Modeling

**Summary:** *beschrijf kort geschiedenis van procedural modeling en werk naar map generation [?]*

## 1.2 L-systems

**Summary:** *beschrijf toepassingen van L-systems*

## 1.3 Map generation

**Summary:** *beschrijf huidige games en tools die gebruik maken van procedural generation voor geometrie van virtuele werelden.*

### 1.3.1 spatial growth based

**Summary:** *verhaal over het feit dat spatial growth based-user controlled bouwen van maps relatief nieuw is.*

### 1.3.2 Interactivity

**Summary:** *beschrijf tools voor het manipuleren van modellen die afhankelijk zijn van l-systems en tools die worden gebruikt voor level design, zoals tools voor terrain editors.*

## 1.4 Required Input

**TODO:** *Belangrijk om snel vast te stellen hoeveel en welke input er nodig is.*

## 1.5 Adding simple gameplay properties

## 1.6 The growth model map generator tool

**TODO:** *discussieren dat gebruiker gelimiteerde controle heeft over het proces: nadelen en voordelen*

Resulting maps must look realistic but the growth model simulations which control the generation of the map to not necessarily have to  
tijdelijk:

# 2 Method: realtime spatial growth models

## 2.1 dynamic structures: polygonal surfaces

**Summary:** *2D polygonale expansie van een ondergrond model als land of water die wordt gestuurd door eigenschappen van het type maar ook door ruimtelijke manipulatie methodes zoals vectorfields. representatie(onder voorbehoud):*

- *Cellular automata*
- *Cell systems*
- *iets wat niet met cellen werkt*

*realtime hertriangulatie zal moeten worden toegepast. polygonal meshes:*

- *representation for irregular 2D volumes.*
- *suitable for water, land, cave like structures*
- *real-time demand, so need for efficient algorithms*
- *houdt rekening met triangulatie*



Figure 1: impressions  
5

- *houdt rekening met smoothing*
- *datastructure*

*citeer: On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling*

**TODO:** *dit kan al snel worden geïmplementeerd dus logisch om dit als eerste te beschrijven en methode uit te werken.*

**-Insert:** *beschrijf waarom ik dynamic modeling technieken nodig heb*  
**-Insert:** *review methodes for dynamic modeling*  
**-Insert:** *focus op cell systems*  
**-Insert:** *custom systeem dat alle eigenschappen heeft die nodig zijn voor de polygonale expansie*  
 [?] [?]

## 2.2 l-systems and map l-system type spatial growth patterns

[?]

## 2.3 Real-Time spatial growth Models For Outdoor Areas

**Summary:** *Per type spatial growth model bespreken wat de context is binnen outdoor areas en dus met welke andere spatial growth modellen het communiceert en welke effecten het spatial growth model heeft op de andere en vice versa. Ook bespreken op wat voor manier een vector field (en andere tools die invloed hebben op de ruimtelijke indeling) effect heeft op de uitdijning van dit type model.*

### 2.3.1 Land

**TODO:** *beschrijven hoe land expansie verloopt en hoe het communiceert met andere modellen*

A definition of land: "The part of Earth which is not covered by oceans or other bodies of water". (bron: <http://en.wiktionary.org/wiki/land> ...jaja ik vind nog wel een betere bron) In reality land obviously does not grow, so first we must ask ourselves whether growing land in our virtual world can be made useful and intuitive to use. There are no real rules for the bounding shape of a piece of land, the bounds can be configured in any way. However due to this fact almost any random configuration of the shape of land looks realistic. The question is, as with other growth models with a minimum amount of rules, whether generating land by means of a growth model, thereby having less control over the result compared to conventional methods, is something which we want.

Obviously when you want maximum control over the resulting geometry of a piece of land, using a growth model is not an option. I would like to prove however there are situations in which this method could prove useful.

**Summary:** *wanneer is het groeien van land handig:*

- *Obviously: In situaties waarin de gebruiker niet veel waarde hecht aan de precieze vorm van de resulterende geometrie.*
- *stimuleert de creativiteit: elaborate.*

Land provides the underlying surface of many of the growth models discussed in this paper.

**Summary:** *dus het bestaan van land op de seedpositie van een afhankelijk groeiproces is een eis. Wellicht moet het genereren van een map bestaan uit verschillende fases:*

- *fase 1: generatie van land en water.*
- *fase 2: plaatsen en activeren van groei processen die afhankelijk zijn van land.*

The field of biological modeling provides some interesting modeling techniques for the process of cell division. Cell division is interesting with respect to the concept of growing land because it is an expanding process with some nice properties. **-Insert:** *benoem properties of revise deze zin.*

### 2.3.2 Water

### 2.3.3 Roads

### 2.3.4 Afforestation

## 2.4 Real-time spatial growth models For Indoor Areas

### 2.4.1 Caves

### 2.4.2 Canarian Aboriginal style homes

## 3 Organic map generation tool

### 3.1 Modifiers

#### 3.1.1 Spatial modifiers

**Summary:**

- *vector fields: dit moet je wellicht helemaal aan het begin doen omdat alle spatial growth models gemanipuleerd worden door deze vector fields.*

- *obstruction with solid objects*
- *smudge-like tool*

### 3.1.2 Timeflow modifiers

#### Summary:

- *pause model*
- *speed up model*
- *slow down model*
- *reverse time flow*

## 3.2 User controlled map generation

## 3.3 automatic map generation

# 4 Implementation

## 4.1 Tools used

#### Summary:

- *programming language: c++ (for now)*
- *visualization library: OpenGL or Ogre3D (if it features easy vertex manipulation)*
- *physics engine: ODE (onder voorbehoud)*
- *collision detection: D-Collide (onder voorbehoud)*

## 4.2 Application Class Structure

**-Insert:** *klassen structuur(voorlopig):*

- *Environment or Canvas: mainloop*
- *Seedpoint*
- *GrowthModel*
- *ShapeModel*
- *NetworkModel*



- *ModelHistory*
- *ModelTopology*
- *ModelGeometry*
- *EventHandler*
- *Communicator*
- *Timeline: can place seeds on the timeline which will be handled by eventhandler when time comes.*

### **4.3 User Interface**

### **4.4 Communication system between models**

## **5 Results**

## **6 Conclusion**