

Real-time spatial growth model map generation

interactivity in procedural modeling tools

Ruud op den Kelder

April 1, 2009

Abstract

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

Contents

1	Introduction	3
2	Review of State of the Art in Procedural Modeling of Landscape	4
2.1	Procedural Modeling	4
2.2	Map generation	4
2.2.1	spatial growth based	4
2.2.2	Interactivity	4
2.3	Required Input	5
2.4	Adding simple gameplay properties	5
2.5	The growth model map generator tool	5
3	Method: realtime spatial growth models	5
3.1	dynamic structures: polygonal surfaces	5
3.2	l-systems and map l-system type spatial growth patterns	7
3.3	Real-Time spatial growth Models For Outdoor Areas	7
3.3.1	Land	7
3.3.2	Water	8
3.3.3	Roads	8
3.3.4	Afforestation	8
3.4	Real-time spatial growth models For Indoor Areas	8
3.4.1	Caves	8
3.4.2	Canarian Aboriginal style homes	8
4	Organic map generation tool	8
4.1	Modifiers	8
4.1.1	Spatial modifiers	8
4.1.2	Timeflow modifiers	9
4.2	User controlled map generation	9
4.3	automatic map generation	9
5	Implementation	9
5.1	Tools used	9
5.2	Application Class Structure	9
5.3	User Interface	10
5.4	Communication system between models	10
6	Results	10
7	Conclusion	10

1 Introduction

description: *This is a general introduction to what the thesis is all about – it is not just a description of the contents of each section. Briefly summarize the question (you will be stating the question in detail later), some of the reasons why it is a worthwhile question, and perhaps give an overview of your main results. This is a birds-eye view of the answers to the main questions answered in the thesis.*

This thesis is about the balance between interactivity by user control and efficiency by procedural generation in procedural modeling tools for urban areas and other types of terrain. **-Insert:** *waarom worden urban areas en terrain procedureel gegeneerd en door wie, wat kan er verbeterd worden*

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.

In order to demonstrate some of the ideas of realtime user control on procedural methods discussed in this paper I have developed a simple pseudo procedural modeling tool.

The main entities in this tool are growing surfaces which can be manipulated by the user in various ways, but also move on their own acting on the rules which have been given to them. The novelty in this approach lies in the fact that the vertices of the boundary of the growing surface are bodies in a physics simulation. User control over the growing polygons is provided with several handles:

Revise: *maak dit beter...*

1. a dynamic innerskeleton which is formed by the contour of the polygon.
2. the vertices of the polygon contour can be dragged.
3. scripting properties of the growing surface.
4. global properties

TODO: *arguments in favour of this method: ease of use. experimental curiosity*

The research which preceded this document strived to construct an intuitive organic modelling method for the creation of 2D maps. The central focus of this research has been on methods for construction of dynamic structures using real-time growth models. To enable a large degree of control over these dynamic structures there is a need for skeletons.

A large part of this thesis is about constructing skeleton representations for dynamic structures. There obviously is a realtime demand in order to enable realtime interactivity, which means building such skeletons should above all be an efficient process.

description: *2. Background Information (optional)* A brief section giving background information may be necessary, especially if your work spans two or more traditional fields. That means that your readers may not have any experience with some of the material needed to follow your thesis, so you need to give it to them. A different title than that given above is usually better; e.g., "A Brief Review of Frammi's Algebra."

2 Review of State of the Art in Procedural Modeling of Landscape

description: *3. Review of the State of the Art* Here you review the state of the art relevant to your thesis. Again, a different title is probably appropriate; e.g., "State of the Art in Zylon Algorithms." The idea is to present (critical analysis comes a little bit later) the major ideas in the state of the art right up to, but not including, your own personal brilliant ideas. You organize this section by idea, and not by author or by publication. For example if there have been three important main approaches to Zylon Algorithms to date, you might organize subsections around these three approaches, if necessary: 3.1 Iterative Approximation of Zylons 3.2 Statistical Weighting of Zylons 3.3 Graph-Theoretic Approaches to Zylon Manipulation

2.1 Procedural Modeling

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

Procedural modeling
[?]

2.2 Map generation

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

2.2.1 spatial growth based

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

2.2.2 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.*

2.3 Required Input

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

2.4 Adding simple gameplay properties

2.5 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:

3 The problem: User control in procedural generation

description: *Engineering theses tend to refer to a "problem" to be solved where other disciplines talk in terms of a "question" to be answered. In either case, this section has three main parts: 1. a concise statement of the question that your thesis tackles 2. justification, by direct reference to section 3, that your question is previously unanswered 3. discussion of why it is worthwhile to answer this question. Item 2 above is where you analyze the information which you presented in Section 3. For example, maybe your problem is to "develop a Zylon algorithm capable of handling very large scale problems in reasonable time" (you would further describe what you mean by "large scale" and "reasonable time" in the problem statement). Now in your analysis of the state of the art you would show how each class of current approaches fails (i.e. can handle only*



Figure 1: impressions
6

small problems, or takes too much time). In the last part of this section you would explain why having a large-scale fast Zylon algorithm is useful; e.g., by describing applications where it can be used. Since this is one of the sections that the readers are definitely looking for, highlight it by using the word "problem" or "question" in the title: e.g. "Research Question" or "Problem Statement", or maybe something more specific such as "The Large-Scale Zylon Algorithm Problem."

3.1 Rigidity of Procedural methods

Revise: *bedenk een betere titel*

3.2 User control

TODO: *elke van de volgende controls behandelen met behulp van bestaande tools die hier gebruik van maken*

1. drawing techniques
2. modeling tools
3. parameters

3.2.1 local control

3.2.2 global control

3.3 Growing surfaces: An alternative method for area generation

4 Using realtime dynamically growing surfaces

4.1 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*
- *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*

[?] [?]

4.2 l-systems and map l-system type spatial growth patterns

[?]

4.3 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.*

4.3.1 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.

When you want maximum control over the resulting geometry of a piece of land, using a growth model is not a very attractive method. However there are some situations in which this method is quite helpful.

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.*

4.3.2 Water

4.3.3 Roads

4.3.4 Afforestation

4.4 Real-time spatial growth models For Indoor Areas

4.4.1 Caves

4.4.2 Canarian Aboriginal style homes

5 Organic map generation tool

5.1 Modifiers

5.1.1 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*

5.1.2 Timeflow modifiers

Summary:

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

5.2 User controlled map generation

5.3 automatic map generation

6 Implementation

6.1 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)*

6.2 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.*

6.3 User Interface

6.4 Communication system between models

7 Results

8 Conclusion