

Real-time spatial growth model based map generation

Ruud op den Kelder

January 14, 2009

Abstract

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

Contents

1	Introduction	1
1.1	Procedural Modeling	1
1.2	L-systems	1
1.3	Map generation	1
1.3.1	spatial growth based	1
1.3.2	Interactivity	2
1.4	Adding simple gameplay properties	2
2	Method: realtime spatial growth models	2
2.1	dynamic polygonal surfaces	2
2.1.1	triangulation of polygonal area	2
2.1.2	extra: smoothing of polygons with curve fitting	3
2.2	l-system type spatial growth patterns	3
2.3	Real-Time spatial growth Models For Outdoor Areas	3
2.3.1	Land	3
2.3.2	Water	3
2.3.3	Roads	3
2.3.4	Afforestation	3
2.4	Real-time spatial growth models For Indoor Areas	3
2.4.1	Caves	3
2.4.2	Canarian Aboriginal style homes	3
3	Organic map generation tool	3
3.1	Modifiers	3
3.1.1	Spatial modifiers	3
3.1.2	Timeflow modifiers	3
3.2	User controlled map generation	4
3.3	automatic map generation	4

4	Implementation	4
4.1	Tools used	4
4.2	Application Class Structure	4
4.3	User Interface	4
4.4	Communication system between models	4
5	Results	4
6	Conclusion	4

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*

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 Adding simple gameplay properties

2 Method: realtime spatial growth models

2.1 dynamic 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):*

- *'Winged-Edge meshes'*
- *'Renderable Dynamic Mesh'*

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*
- *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.*

[?]

2.1.1 triangulation of polygonal area

[?]

Summary:

- *<http://mathworld.wolfram.com/Triangulation.html>*
- *<http://www.cs.unc.edu/dm/CODE/GEM/chapter.html>*
- *computational geometry boek*

2.1.2 extra: smoothing of polygons with curve fitting

2.2 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

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: *beschrijf XNA, C#, physics engine, collision detection: D-collide*

4.2 Application Class Structure

4.3 User Interface

4.4 Communication system between models

5 Results

6 Conclusion

References

- [1] M. v. K. Mark de Berg, Otfried Cheong and M. Overmars. *Computational Geometry: Algorithms and Applications Third Edition*, chapter 4. Springer-Verlag, 2008.
- [2] C. Smith. On vertex-vertex systems and their use in geometric and biological modelling. 2006.