Contribution title

Filip Rynkiewicz
Lodz University of Technology
Łódź, Poland
173186@edu.p.lodz.pl

Stanisław Nowak
MacGilll University
Montreal, Canada

Abstract

Since the beginning of computer era we are trying to imitate nature. The results of those attempts are mathematical equations describing weather, snow flakes, plant growth or influence of species in individual biomes. In computer graphics fractal structure are often used because they can be simple characterize in mathematics. Those forms are common in naure and can be found in example in broccoli, corals, lightnings or in trees. Self-similar fractals can be created in computer graphics using Lindermayer Systems. This article was created to analyze efficiency of this algorithm in modeling 3D tree triangle meshes for games. The general motivation of this topic is to produce way to procedural modeling and simplified edition of complicated 3D tree models. By combining L-systems with Bezier Curve there is a possibility to produce complicated 3D models. For example tree created with 468 branches having 87200 vertices and it is made in few seconds. In conclusion, time-consuming modeling in 3D programs can be replaced by solution described in this article.

1 Introduction

From beginning of our era we are trying to imitate nature. Results of this attempts are descriptions of natural phenomenon using mathematical equations. Those mathematical statements are used for example in weather forecast, simulation of plant growth, creating snowflakes or lightnings. In computer graphics fractal based structure adopted from nature are often used because they can be simply stored as mathematical formula. One of those formulas is *Lindenmayer System*. It allows user to create a mathematical description of appearance, behavior or grow of plant.

Using 3D modeling programs, such as *3ds Max* or *Blender*, to create a complex 3D model of tree is very time consuming, because every branch has to be shaped separately. To speed up and automatize this process L-system can be used. This article is description of implementation and usage of L-System in Unity3D Game Engine. Problem with procedural generated structures using L-System in it's basic description it's that all algorithms are generating static models. It's mean that user can't move, add or do anything with model without recalculating it all over again. The basic idea to add this feature is to use *Bezier Curve*, and generate the mesh based on it.

1.1 Fractal

One of the most important feature of fractal is self-similarity. It's mean that every part of fractal is similar to whole fractal and can be calculated by iterating over simple formula, which describe this fractal. Never ending patter is created by scaling fractal. Observation of plants brought some conclusion which shows that every plant can be treat as a very complex fractal, with multiple elements so it can be simplified using *Lindenmayer System*.

1.2 Lindenmayer System

L-system was created in 1968 by Hungarian theoretical biologist and botanist from University of Utrecht Artist Lindenmayer. System was originally devised to describe formal description of simple multicellular organisms. With fast development of computer graphics the L-system is used by the modern artist to create plant models and to simulate its growth. The system in based on rewriting rules, like productions. All system are based on string comprasion, evaluation and conversion. L-system is always includes axiom, rules and number of iterations. Axiom is starting word, rules are production precepts, and number of iterations is the number how many times rules must be applied to the word.

DOL System

Deterministic and contex free L-system is one of the L-system, where context of the letter have no affect on algorithm. Simple example. When axiom, starting word, is *b* and rules are: $b \rightarrow a$ and $a \rightarrow ab$ with 5 iterations, the example creation is shown at Figure1.

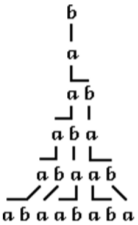
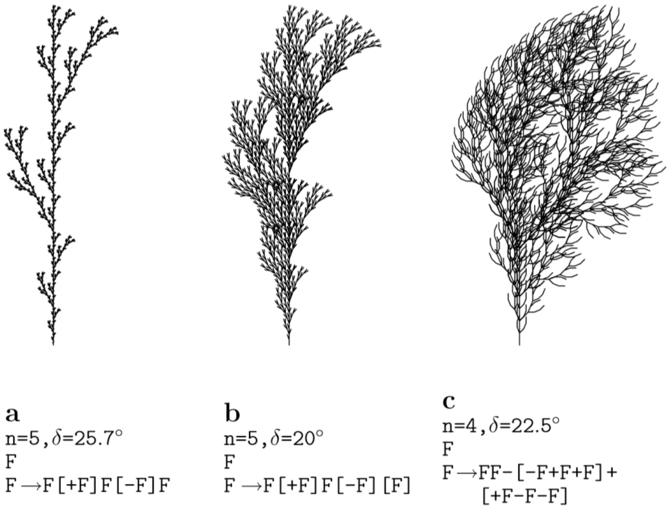


Figure 1. Example of DOL system[1]

Bracket L-System

Using additional character in words user can create extra behaviors. L-system always create one long continued line. Using *[* and *]* the *branches* can be created. First character is responsible for starting the branch and next for closing it. Branch is simply new line in geometric interpretation of string created by L-system.


a
n=5, δ=25.7°
F
F →F[+F]F[-F]F

b
n=5, δ=20°
F
F →F[+F]F[-F][F]

c
n=4, δ=22.5°
F
F →FF-[-F+F+F]+
[+F-F-F]

Figure 2. Example of Bracket L-system[1]

Parametric L-System

To create more complicated models the Parametric L-System was created. Thank to parameters L-system can change the values of variables, in example the steering angle of branch. When in DOL-System the sing + was strictly assigned to hard-coded value in this approach the value can be changed in every iteration using *+(r)* where r is some parameter. Every letter or string must have brackets immediatly behind it and within them there can be multiple parameters from *0* ... *n* seperated with comma. Every parameter can be a mathematical expression, parameters can be divided, multiplied, added, subtracted and etc. Example $C(x\cdot e, e\backslash 2, a+c, b-d)$

Turtle interpretation of string

L-System algorithm generating string. This string must be evaluated to create graphicals content. Created word contains steering, drawing and ommited characters. The sterring characterch are the one who controls the behavior of turtle. Drawin are those which simply adding a line between points. *Turtle graphics* is term where turtle have it's position and angle which describes the direction in which it is facing. Sterring characters :

+

-

&

ˆ

\

/

—

[

]

2 Methods

3 Results

Example 1

Rules:

$$F(l) \rightarrow F(l \cdot 2)$$

$$X(l) \rightarrow F(l)[/(r)X(l)]F(l)[\backslash(r)X(l)-(r)X(l)]F(l)[\backslash(r)X(l)+(r)X(l)]$$

Axiom:

X(1)

Number of iterations:

5

Variables:

$$r \rightarrow 25$$

Example 2

Rules:

$$F(l) \rightarrow F(l \cdot 2)$$

$$X(l,w) \rightarrow F(w \cdot l)[/(r \cdot l)X(l,w)-(r)X(w,w)]+(r)F(l)[\backslash(l)X(l,w)+(r)X(l,w)]-(r)F(w)$$

Axiom:

X(1)

Number of iterations:

5

Variables:

$$r \rightarrow 17.2$$

Example 3

Rules:

$$F(l) \rightarrow F(l \cdot 2)$$

$$\begin{aligned} X(l,w) \rightarrow F(w \cdot l)[/(r \cdot l)X(l,w) - (r)X(w,w)C(l)] \\ + (r)F(l)[G(l) \setminus (l)X(l,w) + (r)X(l,w)] - (r)F(w) - (r)F(w) \end{aligned}$$

$$G(l) \rightarrow F(l/5)[X(l,l) + (r \cdot k)]$$

$$C(l) \rightarrow F(l/5)[X(l,l)/(r \cdot k)]$$

Axiom:

X(1,2)

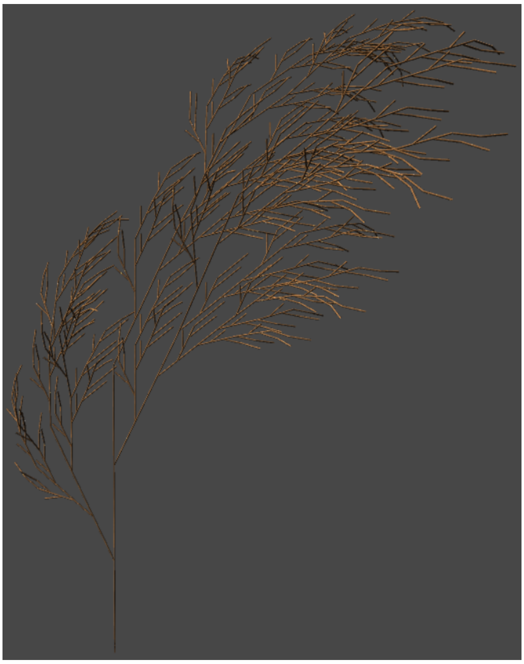
Number of iterations:

5

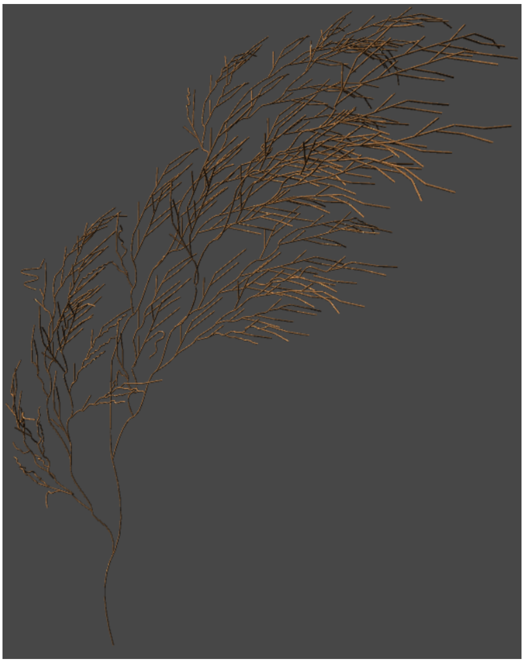
Variables:

$$r \rightarrow 23.5$$

$$k \rightarrow 0.707$$

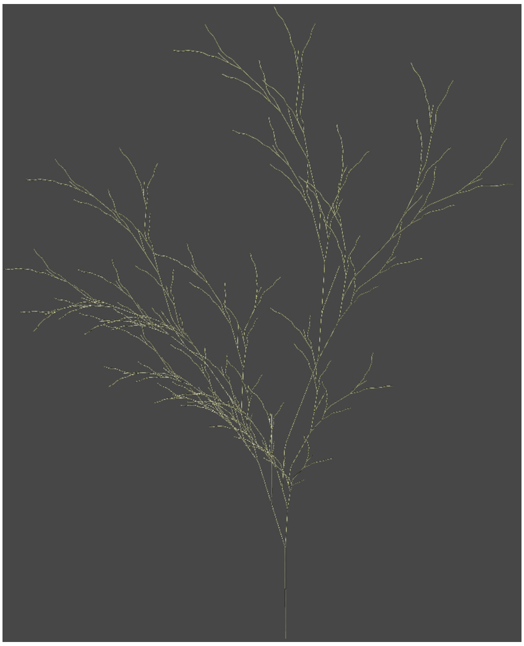


(a) Mesh without modification.
Own source.

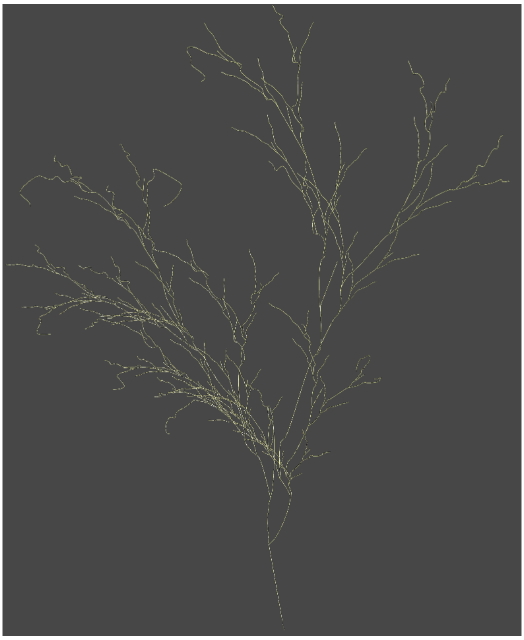


(b) Mesh after modification.
Own source.

Figure 3. Comparison of meshes for example 1.

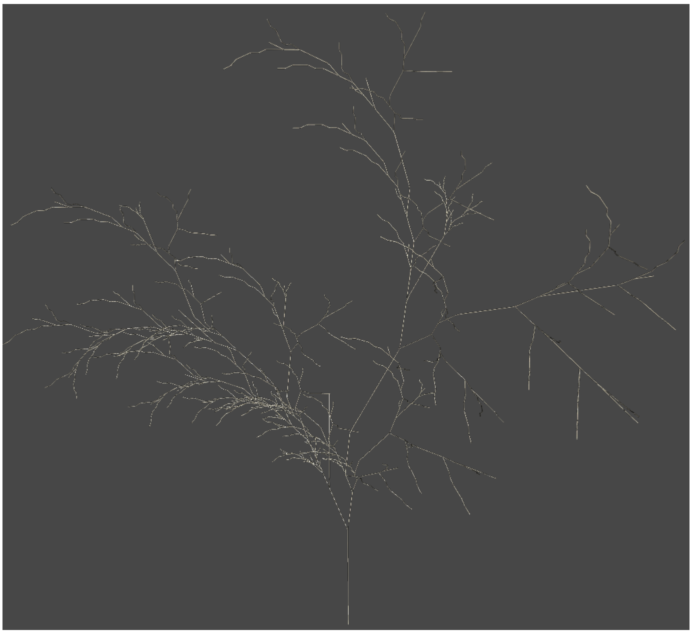


(a) Mesh withouth modification.
Own source.

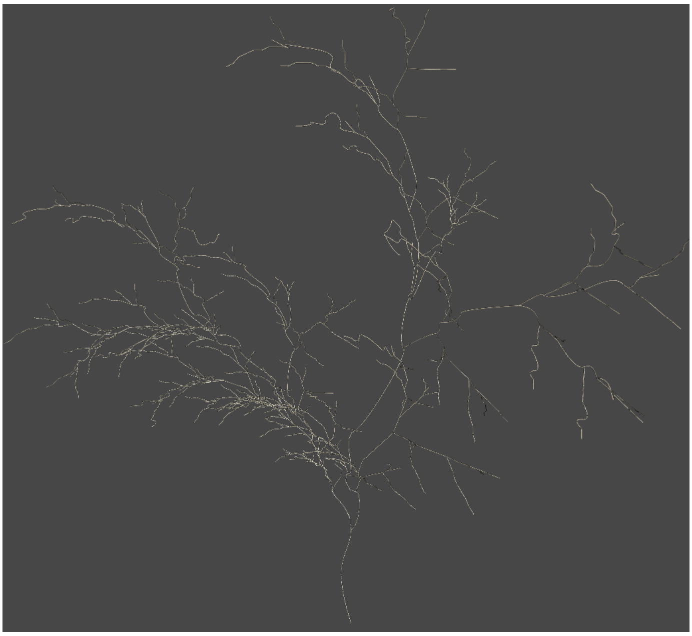


(b) Mesh after modification.
Own source.

Figure 4. Comparison of meshes for example 2.



(a) Mesh without modification.
Own source.



(b) Mesh after modification.
Own source.

Figure 5. Comparison of meshes for example 3.

4 Discussion

For each of the example the same values was used. Five segments on the side surface and five points on spline, it's mean that every curve was divided on five equal parts. Each result of implemented algorithm was divided into two pictures. Screenshot of model generated by algorithm and model after changes. Changes was made by transforming Bezier Points on corresponded to branch Bezier Curve. First all the example and it's description was shown, then the pictures.

In first example for 5 iterations the number of branches is 468, word forming tree have 46086 chars and number of vertices in model is 87200.

This example is modification of simple fractal plant described in [1]. Every branch is directed at 25 ° relative to parent branch. As it can be seen at the Figure3a, the model created by algorithm is complex, using only two rule and one variable. Model after changes have been shown at Figure3b. Every branch in tree have been altered according to user actions.

Next example have 171 branches, world forming tree have 24918 chars and 73700 vertices in model. In this case the custom and longer rules have been used to make model looks like non-deterministic and more chaotic. Model before changes was shown at Figure4a, and after changes at Figure4b. In first example every variable had exactly one parameter without any mathematical operations, except $F(l \cdot 2)$. But in the second example every operation on parameter had to be evaluated and changed to float value, so the time of the calculation had increased significaly. Thanks to usage of multiple parameters and operation between them user can create more complicated structures, using less rules, but for the price of time for parameter evaluations.

Last example have 214 branches, world world forming tree have 24918 chars and 73700 vertices in model. POOOOOOOPRAW!!!!! Figure5a presents unchanged model of last example, Figure5b same model after changes. Creating more complicated rule, based on those in example 2, and adding two more simple rules the model in example 3 is more non-deterministic and tree-like than other examples. Time for creating this tree is not greater than time for example 2, because to store all rules algorith use C# *Dictionary* where the *Key* is from what chars rule must be changed, and *Value* stores to what rule chars must be changed.

5 Conclusion

By using technique described before the user can simply generate tree-like model in short time, small amount of vertices, and with ability to change every point of model using Bezier handles. Creating model with 300 branches using normal

3d modeling software would be very time consuming.We can move our work to machine and automatize process of modeling, using simple parameterized mathematical equations, without even knowing how the 3d modeling software works. And we can do this inside game engine, without other programs.

To evaluate all string into float values the simple parser have been created. Parser created with this implamentation is simple, but creating possibility to evaluate more than 3 parameters, and operations betwen them, using approaches from this implemantation would be very difficult. To ensure that every parameter would be evaluated propertly whole implementation will be rewriten into new *Boost Spirit X3* C++ library, and created as *Dynamic-Link Library*. Unity,Enreal Engine and others game engines have ability to write plugins based on *DLL* written in **C++**, so every engine could have their own implementation of this algorithm, based on *DLL*.

References

[1] Prusinkiewicz P., Lindermayer A. (2004) The Algorithmic Beauty of Plants. wersja elektroniczna.

[2] Fuhrer M. Hairs, Textures, and Shades: Improving the Realism of Plant Models Generated with L-systems. M.Sc. thesis, University of Calgary, August 2005. wersja elektroniczna

[3] Runions A., Lane B., Prusinkiewicz P. Modeling Trees with a Space Colonization Algorithm,Department of Computer Science, University of Calgary, Canada ,Eurographics Workshop on Natural Phenomena (2007).

[4] MacMurchy P., The Use of Subdivision Surfaces in the Modeling of Plants, THE UNIVERSITY OF CALGARY, April, 2004

[5] Smith C., On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling, THE UNIVERSITY OF CALGARY, April, 2006

[6] http://www.digitaltutors.com/tutorial/570-Procedural-Tree-Growth-Using-L-systems-in-Houdini

[7] `http://www.speedtree.com/images/modeler_lrg.jpg`

[8] `https://unity3d.com/5`

[9] `http://www.mono-project.com/`

[10] `http://www.zobaczycmatematyke.krk.pl/025-Zolkos-Krakow/b-spline.html`

[11] TreeSketch:Interactive Procedural Modeling of Trees on a Tablet, Longay S., Runions A., Boudon F., Prusinkiewicz P.

[12] http://kacikzdrowia.pl/wp-content/uploads/2015/01/s%C5%82onecznikb.jpg

[13] https://sadzawka.pl/Convallaria_majalis_Aurea-Konwalia_majowa

[14] https://msdn.microsoft.com/pl-pl/library/system.text.stringbuilder%28v=vs.110%29.aspx

[15] http://net-informations.com/faq/general/dictionary-list.htm

[16] https://www.assetstore.unity3d.com/en/#!/content/11278

[17] https://www.youtube.com/watch?v=o9RK6O2kOKo

[18] http://docs.unity3d.com/Manual/class-MeshRenderer.html

[19] http://jayelinda.com/modelling-by-numbers-part-1a/

[20] Rozenberg G., Salomaa A. *The mathematical theory of L systems*, Academic Press, New York, 1980