3D Tree Generation using LSystems in Unreal Engine 4

Maths and Graphics for Computer Games
Lecturer: Prof Frederic Fol Leymarie

Witold Gawłowski
MSc in Computer Games Entertainment

November 2016

1 Approach

I have been considering a number of tools to apprach this task: *Octet*, *Unity3D* and *Unreal Engine*. *Octet* was an opportunity for learning *OpenGL* API. Using *Unity3D* would be easier as I already know it and therefore I could focus on some bonus features. Finally there was *Unreal Engine* at which I am a beginner. Knowing that it is widely used in the industry and is capable of producing stunning visual effects I have decided to use it.

To keep the code separated as much as possible from "the framework", my project uses pure *c++* to finalize the task.

2 Feature Overview

The code generates 3D, low-poly trees using L-Systems. For each tree segment a cuboid with decagonial base is created. The basis are skewed and rotated to smoothly merge with each other using *Unreal*'s *SplineMesh*s and *SplineComponent*s.

Six sets of rules form the assignment specification and two additional ones [1] were used to create eight input files. It is possible to switch between input files during generation using keys **1-8**. Each file consists of four separate sections: *variables*,*start*, *roll angle* and *rules*. Following rule symbols are used to describe the L-system turtle's actions:

F	+	-	&	^	[	]
go worward	turn right	turn left	pitch down	pitch up	save state	restore state

To give an example, file generating tree displayed by default looks like this:

```
variables:
F
start:
F
roll angle:
25
rules:
F -> F[^+&F]&F^[^-^F]F
```

The path to the specified file is displayed on the screen after pressing correspondng button.

Is is possible to adjust four parameters. To change parameter one first need to select it for later change: *roll angle* is selected with **R** key, *pitch angle* with **T**, *length multiplier* with **U** and *width multiplier* with **Y**. To adjust selected parameter one then needs to press **up arrow** or **down arrow**. The number of steps in tree generation is changed with **left arrow** and **right arrow**.

3 Code structure

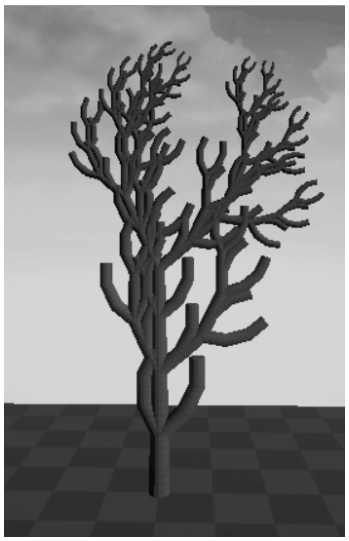
The *c++* part of the code is linked to *Unreal* by the TreePawn class. It handles input files, player input. Also the camera movement is implemented there. It holds reference to the Tree class. Finally it is also responsible for generating a string that is used for Tree initialization. Tree core functionality is to "walk" the string and generate points that form the skeleton of generated structure. The points are gathered by Branch class instances. Tree is consisting of an array of Branches. Each instance of a Branch class has a Draw method that is responsible for generating meshes for a elements, each defined by neighbouring points.

4 Results

Enclosed images depict reconstuction of the L-System trees from assignment specification. Rule sets for specific trees are provided below each picture.

F −> F[^+&F]&F^[^−^F]F

F −> F[^+F]F[−F][&F]



F −> FF−[^−F+F+F]+[+F−&F−F]

Two last pictures depict two additional rule sets.

starting symbol: X
F −> FF
X −> F[+X]F[−X]+&Xˆ

starting symbol: X
F −> FF
X −> F[++X][+X][−X][−−X]F&Xˆ

starting symbol: X
F −> FF
X −> F−[[&Xˆ]+ˆX&]+F[+FˆX&]−X

5 Conclusion

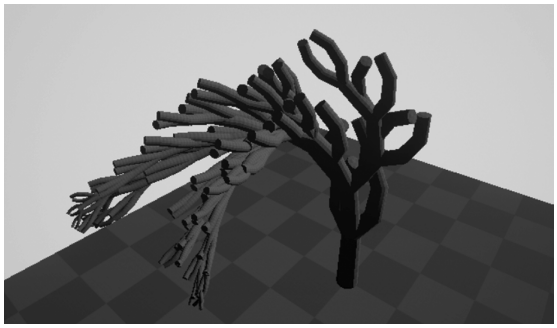
The demo of the project is available on youtube:

`http://www.youtube.com/watch?v=ieAfzjdyIcM`

also the repository with the code source is available on github:

`https://github.com/witold-gawlowski/LUnreal`

starting symbol: F
F −> FF
X −> F[+X]F[−X]+&Xˆ



starting symbol: F
F −> FF
X −> F[++X][+X][−X][−−X]F&Xˆ

Overall *Unreal engine* caused multiple problems. The engine is complex and documentation is spread across forums, official docs and youtube tutorials. Terefore its frequently hard to find information and examples. Also Unreal engines is created for big projects and therefore compilation times for the simple boilerplate code takes 6-10 seconds after optimization.

On the other hand, unreal engine has pleasant game-styled GUI and provides beautuiful rendering out-of-the box which made the process of learning it a pleasure experience.

References

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy.* San Val, 1995.