

L-system과 Interactive Genetic Algorithm을 이용한 웹 기반

게임 오브젝트 생성 기법

윤두밈^o 김경중

세종대학교 컴퓨터공학과

krad@hanmir.com, kimkj@sejong.ac.kr

Web-based Game Object Generating Method using L-system and Interactive Genetic Algorithm

DuMim Yoon^o KyungJoong Kim

Dept. of Computer Engineering, Sejong University

요 약

최근의 게임들은 비용에 비해 완성도가 떨어지고 플레이 타임이 짧아졌다. 이러한 문제의 해결책 중 하나로 게이머들은 Mod를 제작하여 그들의 욕구를 충족시켰고 이는 게임의 수명을 연장하는 길로 이어진다. 하지만 Mod의 제작은 전문 지식과 능력이 요구되기에 일부 유저만이 가능했고, 그것도 대부분이 게임의 수정에 관대한 게임들 뿐이어서 대부분의 유저들은 그저 다른 사람들이 만들어 놓은 Mod를 즐기는 단계있을 뿐이었다. 본 논문에서는 게임 Mod를 구성하는 많은 부분 중 게임의 오브젝트, 특히 배경 건물에 집중하여 L-system을 이용해 building footprint를 성장시킨 뒤, 3차원 공간좌표로 변환하는 방법과 이렇게 나온 건물 오브젝트들을 Interactive Genetic Algorithm을 이용해 유저가 원하는 형태를 얻을 수 있도록 하였다. 또 이 모든 것을 웹상으로 구현하여 다른 사람들과 공유할 수 있는 것은 물론, 오픈 소스 레이싱 게임인 TORCS에 실제로 적용한 결과를 보여주어 비전문가들도 특별한 도구 없이 기본 웹브라우저만으로도 게임 오브젝트를 생성할 수 있는 기법에 대해 제안한다.

1. 서 론

최근의 게임들은 기술의 발전과 함께 게임제작 비용이 늘어나는 추세이다. 게다가 좋은 게임이 만들어지기 위해서는 다양한 부분에서 질을 향상시켜야 하는데, 제한된 자원 내에서 갈수록 커지는 유저들의 요구를 모두 충족시키는 것은 결국 불가능하다. 이는 유저들로 하여금 직접 게임 콘텐츠에 손을 대는 계기가 되었다. 인터넷의 확산과 게임 산업의 부흥으로 게임 개발의 진입 장벽 또한 낮아지면서 게임을 수정할 수 있는 능력의 유저들이 늘어남에 따라 직접 기존의 게임을 바탕으로 Mod(추가적인 데이터 설치를 통해 게임의 내용을 추가 및 수정함)를 통해 게임의 콘텐츠를 스스로 늘려나가는 길을 선택하였다[1]. 그러나 이와 같은 방법은 게임에서 맵 에디터와 같이 콘텐츠 제작 툴을 직접적으로 제공해 주거나 Mod에 개방적인 게임, 혹은 게임 자체를 수정 가능한 능력이 있는 전문가들만이 가능했고 일반적인 게이머들은 다른 유저들이 제작한 Mod를 사용할 수 있을 뿐이었다. 본 논문에서는 이러한 비전문가들도 간단히 웹브라우저만을 통해 게임 오브젝트를 생성하는 기법에 대해 제안하고 실제로 게임에 적용하는 과정에 대해 설명한다.

2. 관련연구

2.1. L-system

L-system[2]은 Lindenmayer system이라고도 하며 형태를 표현하는 문법(Shape Grammar)으로 식물의 성장과정을 문법으로 표현하기 위해 만들어졌는데 매우 간단한 규칙과 초기 값으로 복잡한 형태를 표현할 수 있다는 것을 보여준다. 그 때문에 3D 그래픽적으로 자연스러운 나무를 다룰 때 자주 사용된다.[3]

2.2 Interactive Genetic Algorithm

IGA는 Genetic Algorithm을 기반으로 사람과 상호작용이 가능한 인터페이스를 추가한 것이다. GA는 오직 컴퓨터만을 이용해서 문제를 해결하는 알고리즘인데, 실제적인 문제에 적용하다 보니 컴퓨터만으로는 해결이 안 되는 문제가 발생했다. GA에서 중요한 것 중 하나가 평가 함수인데, 컴퓨터로는 평가가 불가능한, 수식으로 표현이 불가능한 문제들이 나타났기 때문이다. 이러한 문제 중에는 컴퓨터와 달리 인간이 손쉽게 평가할 수 있는 문제가 있었기 때문에 IGA를 사용하게 되었다. IGA는 컴퓨터는 평가가 불가능 하나 사람들은 손쉽게 평가 가능한 문제나 사람들이 갖고 있는 주관적인 기준이 필요할 경우에 사용된다[4][5].

3. 제안하는 기법

3.1. L-system을 이용한 건물 생성 방법

현재 3D 그래픽 쪽에서 주로 사용하는 건물 생성 방법은 Procedural Building Generator로 스크립트에서부터 상용프로그램까지 다양한 종류로 분포되어 있으면서도 결과물의 퀄리티 또한 좋은 편이다. 그러나 이러한 방법은 웹상에서 직접적으로 사용되기 힘들고 주로 도로변을 따라서 배경 건물을 생성하거나 다각형들의 병합을 통해 생성된 건물 기반(Building footprint)내에 건물을 생성해 준다[그림 1]. 본 논문에서는 L-system을 사용하여 building footprint의 성장을 통해 형태의 복잡도를 늘리는 방법을 이용하기로 했다.

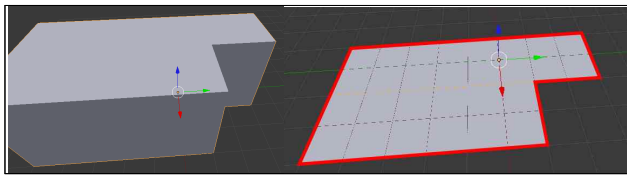


그림 1 Building (좌) 와 Footprint (우)

이 방법은 단 하나의 다각형을 기반으로 만들어지기 때문에 건물 표면의 복잡도나 사실성이 떨어지지만, 게임에 사용되어지기에는 충분한 복잡도를 갖고 있고 단위 길이의 직각의 형태를 이루고 있어 차후 수정이 용이하다. 또한 L-system을 통해 성장시키는 방식은 적은 리소스로 복잡한 표현을 가능케 한다.

L-system은 입력 값에 의해 결과물이 결정된다는 특징이 있기 때문에 자연스러우면서 복잡한 물체를 생성하기 위해서는 난수가 사용된다. 그러나 그 형태를 재현하기 위해서는 값의 변화를 재현할 수 있어야 한다. 따라서 단지 무작위적인 수를 출력하는 것이 아닌, 재현 가능한 난수를 제공할 수 있는 랜덤함수가 필요하다. 본 논문에서는 George Marsaglia가 제안한 Multiply-with-carry 기법을 응용하여 간단한 랜덤함수를 만들어 주었다. 이 함수는 seed 값을 기반으로 0.000 ~ 1.000 사이의 재현 가능한 난수를 생성해준다[6].

다음으로 L-system을 이용하여 building footprint를 표현하기 위해서는 형태 문법에 대해서 정의해주어야 한다. 직사각형은 직선과 ±90°의 회전으로 이루어져 있어서 F와 R, L의 3가지 변수로 표현이 가능하다. 따라서 L-system에는 세변수를 이용하여 성장시키도록 한다.

초기 값은 최소의 기본 크기, 즉 넓이가 1 제곱 단위인 정사각형으로 한다. 이는 길이를 정수로 표현 가능하게 해주며, 성장시킬수록 넓이가 넓어지기 때문에 성장 횟수를 통해 그 크기를 가늠할 수 있다. 이를 형태 문법으로 표현하면 “FRFRFRFR”이 되며 이를 기반으로 성장 규칙에 따라 성장하게 되는데, 성장 규칙은 단 하나의 규칙만을 갖는다.

- (0.0 ~ 1.0 , FRF → LFRFRFF) : “FRF”는 일정 확률로 “LFRFRFF” 변한다.

이 규칙은 building footprint를 한 칸 확장한다는 것을

의미한다. 그러나 이러한 확장을 반복함에 있어 테두리의 바깥 부분이 안쪽으로 파고드는 ‘외부가 내부가 꼬이는’ 문제가 발생하게 되는데, 이를 막기 위해서는 ‘단순화’를 수행해야 한다. 단순화 방법 자체는 확장시키는 방법과 다를게 없다. 단순화를 위한 규칙을 이용하면 되기 때문이다. 테두리가 중복되어 있는 부분을 형태 문법으로 표현하면 “FRRF”나 “FLLF”로 표현할 수 있고 “F”부분을 제거함으로써 테두리의 중복을 막을 수 있다. 그 외에도 “LR”의 경우 회전의 총합이 0이기 때문에 굳이 써줄 필요가 없다. 단 “L”과 “R”이 비슷한 비율로 사용되게 되면 상기한 문제가 발생되기 때문에 이 문제를 해결하기 위해 주요 회전을 시계방향(“R”)으로 집중시켜주어야 한다. 이런 식으로 단순화를 정리하면 다음과 같다.

조건	“FLLF”	“FRRF”	“RRRR”	“LLL”
변경	“LL”	“RR”	“”	“R”
조건	“RRR”	“LL”	“RL”	“LR”
변경	“L”	“RR”	“”	“”

표 1 L-system을 이용한 단순화 규칙 (중복의 삭제, 문자의 단순화, “R”로의 집중)

주의해야 할 부분은 단한번의 단순화로 전부 단순화된다는 보장이 없기 때문에 더 이상 변화가 없을 때까지 반복하여 단순화를 시켜주어야 하며, 기본적인 L-system과 달리 시작지점으로 돌아오는 순환구조를 취하고 있기 때문에 시작 위치에 중복이 발생하게 되면 검출이 불가능할 수 있다. 따라서 문자열을 쉬프트 시켜서 한 번 더 단순화 시켜주어야 하며, 이때 최적의 쉬프트 위치는 단순화 패턴에 포함되지 않는 구간(ex> “FF”)이다[표 2].

“LFLLFRFRFR” => “LLRRR” => “LR” => “” “LF...FF...FL” >> “F...FLLF...F” => “F...LL...F”

표 2 반복에 의한 단순화(상)와 쉬프트에 의한 단순화의 예시(하)

3.2. IGA를 이용한 웹상에서의 평가 방법

L-system를 이용하여 다양한 building footprint의 건물들을 생성할 수 있었지만, 이렇게 만들어진 오브젝트가 항상 유저를 만족시킬 수는 없었다. 사람들에 의해 인공적으로 만들어진 건물이 자연을 모방한 L-system으로는 자연스럽지 않았고 유저의 취향이나 만족감을 고려하지 않았기 때문이다. 따라서 이 문제를 극복하기 위해 IGA를 사용하였고, 다른 사람들과 결과물을 공유할 수 있도록 웹기반에서 동작하게 제작하였다.

L-system은 초기 값을 이용해서 성장시키는 시스템이기 때문에 복잡한 결과물에 비해 적은 변수만이 사용된다. 실제로 L-system에 영향을 주는 요인은 Axiom, 성장 규칙, 삭제 규칙 등 여러 가지가 있지만, 앞서 밝혔던 성장 규칙, 삭제 규칙을 사용하지 않고 제약 사항을 만족시키면서 모든 요인을 진화시키기에는 연산/평가량이 매우 많아져 IGA를 적용하기에 적합하지 않아 제외시키기

로 하였다. 따라서 진화는 서로 독립적인 3개의 gene (seed, growth, height)을 기반으로 이루어지는데, 본 논문에서는 선택압을 이용한 품질비례 룰렛휠 기법과 단순 crossover, mutation을 이용하였다.

본 논문에서는 비전문가들도 사용이 용이하도록 웹 브라우저를 이용하여 IGA를 적용하기로 하였다[그림 2]. 이전에도 웹상에서 평가를 하는 연구는 있어왔으나 단지 2D 객체나 단색의 3D 물체였을 뿐이었다. 이는 웹 기술의 성장과도 관련이 있는데, WebGL이 등장한 것은 얼마 되지 않았기 때문이다. 기본 인터페이스는 endlessforms[6]에서 형태를 따왔는데, 이는 최근에 널리 사용되고 있는 터치스크린 시스템에서도 적용될 수 있는 원버튼 인터페이스로써 적절했기 때문이다.



그림 2 웹기반 IGA 인터페이스

본 논문에서 구현한 웹기반 IGA 인터페이스 시스템은 <http://cilab.sejong.ac.kr/EC/>에서 체험해 볼 수 있으며, 만들어진 3차원 건물 오브젝트 데이터를 저장하고 TORCS (<http://torcs.sourceforge.net/>)를 설치한 다음 맵생성을 위해 <http://usuarios.multimania.es/fltorcs/tutorial/>를 참조하여 만들어 주면 실제로 게임에 오브젝트가 적용된 모습을 볼 수 있다[그림 3].

4. 결 론

본 논문에서는 비전문가들도 특별한 툴 없이 웹브라우저만으로도 손쉽게 게임 오브젝트를 생성할 수 있음을 보여주었다. 비록 전문가들이 직접 제작한 결과물에 비해서는 아직 많이 부족하나 이와 같은 툴을 이용하게 되면, 일반 유저들도 전문 지식 없이 자신이 원하는 오브젝트를 손쉽게 만들어서 곧바로 게임에 적용시킬 수 있게 되고, 이는 곧 게임의 질을 높이고 수명을 연장시키는 힘의 원동력이 된다. 현 단계는 매우 기초적인 단계로 간단한 건물 모델링을 생성하여 게임에 추가했지만, 추가적인 연구를 통해 더욱 세밀하고 현실적인 결과물을 낼 수 있게 된다면 게임 제작 시에도 사용될 수 있을 것

으로 보이며, 건물의 외부뿐만이 아니라 건물의 내부나 다른 오브젝트에 대해서도 적용하여 차후에는 게임의 모든 오브젝트를 생성할 수 있을 것으로 기대한다.



그림 3 TORCS에의 적용 사례

감사의 글

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2012-0001749) 및 뇌과학 원천기술개발사업임(2012-0005799).

참고문헌

- [1] B. Bostan, U.Kaplançali, "Explorations in Player Motivations: Game Mods," in Proceedings of GAMEON-ASIA, 2010.
- [2] P. Prusinkiewicz and A. Lindenmayer. The algorithmic beauty of plants, Springer, 1990.
- [3] P. Tan, G. Zeng, J. Wang, S. B. Kang and L. Quan, "Image-based tree modeling," in *Proceeding SIGGRAPH '07*, Trans. on Graphics, vol. 26, Issue3, no..87, 2007.
- [4] H. Takagi, "Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation," in *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275-1296, 2001.
- [5] J. Clune and H. Lipson, "Evolving three-dimensional objects with a generative encoding inspired by developmental biology," in *Proceedings of the European Conf. on Artificial Life*, pp. 144-148, 2011.
- [6] G. Marsaglia, "Random number generators," in *Encyclopedia of Computer Science 4th*, pp. 1499-1503, 2003.