

# 펼린 노이즈를 이용한 3D 하늘과 지형 생성 그래픽스 구현

이현정\*, 이종서\*, 심재창\*  
\*안동대학교 컴퓨터공학과  
e-mail : 6711hyun@naver.com

## 3D Sky and Terrain Generation Graphics Using Perlin Noise

Hyeonjeong Lee\*, Jongseo Lee\*, Jaechang Shim\*  
\*\*Dept. of Computer Engineering, Andong National University

### 요 약

본 논문은 프로세싱(Processing) 언어와 펼린 노이즈(Perlin Noise) 기법을 활용하여 가상의 3D 지형을 생성하고, 하늘을 구현하였다. 키보드를 이용하여 방향키에 따라 해의 위치가 표현되고, 값에 따라 명암과 밝기가 변하는 하늘과 지형을 그래픽스로 표현하였다. 이를 이용하여 현실 세계의 환경을 더욱더 정확한 데이터로 간접적으로 컴퓨터에서 지형과 하늘의 움직임을 자세하게 볼 수 있다.

### 1. 서론

오늘날 3차원 컴퓨터 그래픽스를 컴퓨터 그래픽스라고 부른다. 컴퓨터 그래픽스는 컴퓨터를 이용해 실제 세계의 영상을 조작하거나 새로운 영상을 만들어내는 기술을 가리킨다[1]. 컴퓨터 그래픽스 작업은 짧은 시간 내에 수작업으로 불가능한 표현이나 효과를 조작해낼 수 있기 때문에 시간과 비용을 줄일 수 있고, 또한 자유로운 수정, 반복, 변형 등을 통해 정확하고 사실성 있는 이미지 생산을 가능하게 한다. 자연현상 시뮬레이션, 가상현실, 의료 영상, 인간-컴퓨터 상호작용, 실시간 시뮬레이션, 영화 특수효과 등 다양한 연구 및 응용 분야를 파생시키며 점차 복잡한 지형도를 그려가고 있다. 국내외에서도 컴퓨터 그래픽스의 응용 분야는 과학, 공학, 예술, 인문의 전 분야에 걸쳐 광범위하게 분포되어 있다[2]. 본 논문에서는 가상의 지형과 하늘을 구현하여 키보드의 입력값에 의해서 명암과 밝기가 변하는 것이 가능하며 방향키에 따라 해의 위치가 표현되는 3D 하늘과 지형을 제작하려 한다. 또한, 하늘과 지형을 구현하기 위해 사용된 원리와 구현 과정에 대하여 다루어 보려고 한다.

### 2. 관련 연구

펼린 노이즈(Perlin Noise)는 Ken Perlin에 의해 고안된 노이즈(Noise) 기법이다. 절차적인 자연현상을 표현하는데 상당히 효율적인 기법이고, 자연적인 질서를 가진 일련의 임의 값을 생성하는 알고리즘이다[3]. 본 연구는 프로세싱(Processing)언어로 Daniel Shiffman의 펼린 노이즈 알고

리즘을 기반으로 3D 가상 지형을 생성하는 연구를 진행하였다.[4]

노이즈 함수는 본질적으로 seed 값에 의한 난수 생성기이다. 인자로 하나의 정수를 받아서 그 인자에 기반한 난수를 반환한다. 만약 똑같은 인자를 두 번 전달하면 동일한 수가 두 번 생성된다. 이런 식으로 동작한다는 건 매우 중요한 사실이다. 그렇지 않다면 펼린 함수는 쓸모없는 것을 생성할 것이다. 이 값들 사이를 부드럽게 보간함으로써 전달 인자를 정수로 받지 않는 연속적인 함수를 정의할 수 있게 되는 것이다[5].

펼린 노이즈 기법을 적용하려면 노이즈를 임의고자 하는 평면을 일정한 크기의 그리드로 구현한다. 평면을 격자화 한 후 각각의 점에서 랜덤한 벡터를 부여하고, 기준이 될 점을 각각의 그리드 가운데에 랜덤하게 찍는다. 각각의 모서리 점의 위치를 빼서 벡터를 만들고, 원래 모서리 점이 가지고 있던 벡터와 내적 시켜준다. 이 계산을 모든 모서리에 대해서 4번 해준다. 4개의 모서리에 각각 내적한 값이 들어갔으면 보간 함수를 사용하여 보간 해준다. 총 3번의 보간함수를 통해서 모든 그리드의 기준점에서 노이즈를 주면 펼린 노이즈가 된다[6].

보간이란 새로운 점을 만들기 위해 수많은 점을 평균화시키는 것이다. 이 방법은 샘플 점들을 직선으로 연결하지 않고 곡선으로 연결함으로써 본래 신호 파형에 대한 변형을 최소화해 준다[7]. 쉽게 말한다면 보간은 두 점을 연결하는 방법을 의미한다.

본 연구는 과학기술정보통신부 및 정보통신기획평가원에서 지원하는 SW중심대학사업(IITP-2019-0-01113)의 연구 결과로 수행되었음.

### 3. 지형 생성

그림 1은 펠린 노이즈 기법을 적용하기 위한 평면을 일정 크기의 그리드로 구성된 것이다.

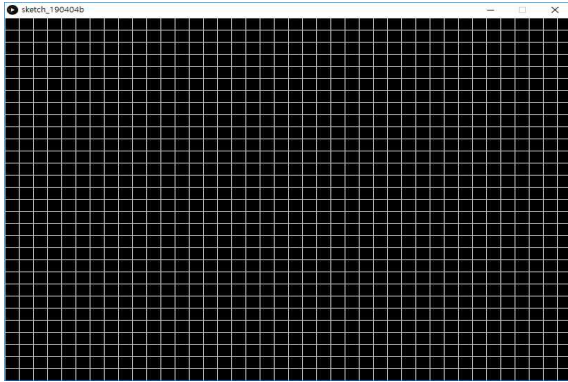


그림 1. 평면의 그리드(Grid) 생성

그리드 사이를 연결하여 지형을 쉽게 표현할 수 있도록 X 좌표와 Y 좌표를 이용해서 대각선으로 만들어 준 것이 그림 2이다. 이 평면은 앞으로 구현할 지형을 입체감 있게 표현할 수 있도록 해준다.

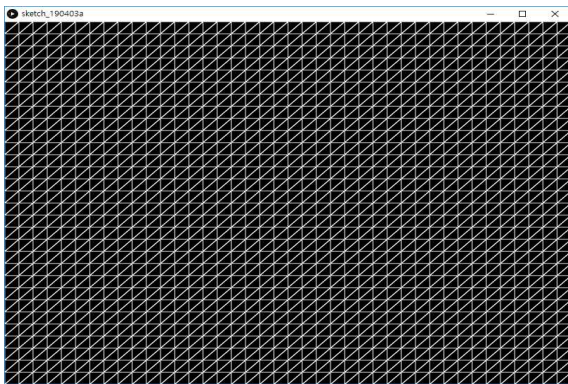


그림 2. 대각선으로 연결한 그리드(Grid) 구현

그림 3은 격자로 이루어진 평면을 X축으로 회전하여 입체감을 표현한 것이다.

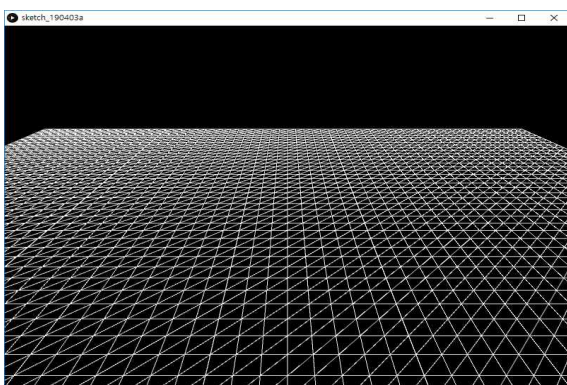


그림 3. X축으로 회전시킨 그리드(Grid) 구현

그림 4는 그리드의 Z 좌표의 값으로 높낮이를 설정하였다. Z 좌표값을 랜덤하게 생성하여 현실 세계의 모습과 흡사한 지형을 구현하기 위하여 펠린 노이즈 기법을 사용하

였다[8]. 노이즈 함수를 이용하여 연속성이 없는 값을 그리드의 Z 좌표에 대입하여 높낮이를 설정하고, 입체감을 표현하였다.

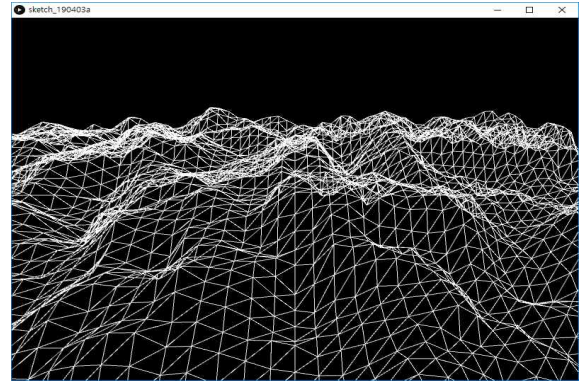


그림 4. 노이즈(Noise)를 이용한 그리드

### 4. 구현

위의 연구들과 본 논문에서 제시한 펠린 노이즈 기법을 바탕으로 3D 하늘과 지형을 그래픽스로 구현한 것이 그림 5이다

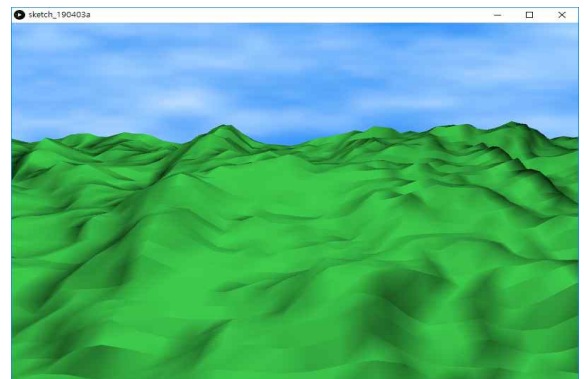


그림 5. 하늘과 지형 생성

실제 하늘과 유사하도록 색상을 입히고, 구름이 몽치듯 지나가는 것을 표현하기 위하여 노이즈 기법을 도입하였다[9]. 본 연구 내용의 지형 생성으로 노이즈 값을 조절하여 지형을 구현하였다. 지형에 색상을 입혀 실제 산맥들과 유사한 색상을 입혀서 간접적으로 맑은 하늘 밑 푸른 산을 시뮬레이션하였다. 이렇게 만들어진 지형은 키보드로부터 방향키와 숫자를 입력받아 밝기와 해가 뜨는 위치를 조절하여 해가 뜨는 방향을 설정할 수 있게 하였다.

그림 6은 키보드 숫자 1, 2, 3 중 하나의 값을 입력하면 입력한 숫자의 따라 밝기가 강, 중, 약으로 표현되는 시뮬레이션을 구현하였다.

그림 7은 방향키를 해의 위치를 조정하여 100씩 값이 커지고, 작아지면서 해가 뜨는 방향과 밝기가 조절된다. 여기서 해는 도형 원을 삽입하여 해를 표현하였다.

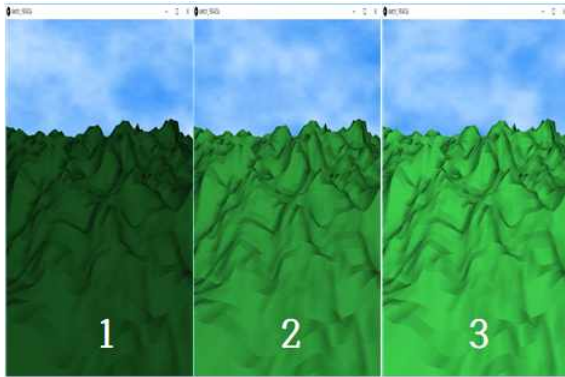


그림 6. 숫자 값에 따라 조절된 밝기

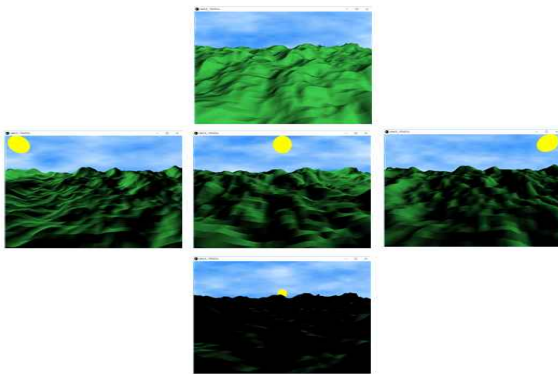


그림 7. 해의 위치에 따른 지형 모습

## 5. 향후 과제

본 연구는 현재 해의 위치에 따라 빛만 표현하였으며 실제 시간의 흐름은 고려하지 않았다. 시간대로 해의 위치를 조절하고, 밤이 되면 하늘이 어두워지는 효과를 추가한다면 지금보다 좀 더 사실에 가까운 3D 하늘을 구현할 수 있을 것이다. 하늘을 시간의 흐름에 따라 구름이 이동하고, 햇살이 들도록 구현한다면 우리의 연구가 실제 산업 현장이나 날씨에 관련된 업무에 많은 도움이 될 것이다.

## 6. 결론

본 논문에서는 자바 기반의 프로세싱 언어를 사용하여 3D 하늘과 지형을 그래픽으로 구현하였다. 지형을 구현하기 위하여 펄린 노이즈 기법을 사용하였고, 기법을 사용하기 위해 공부를 해보았다. 평면은 격자화 하여 노이즈를 발생시켜 높낮이가 일정하지 않은 지형을 만들어 냈다. X 축을 회전시켜 원근감을 만들었고, 일정하지 않은 높낮이를 위해서는 Z 좌표의 값을 랜덤하게 주었다. 자연스러운 지형 그래픽을 위해서 Z 좌표의 값을 조절해가며 현실 세계의 지형과 같은 모습의 시뮬레이션을 만들었다. 본 연구에서 구현한 지형뿐만 아니라 하늘을 통해서 밝기를 조절하고, 빛을 비추었다. 해가 뜨는 위치에 따라 밝아졌다가 어두워질 수 있도록 하였다. 시뮬레이션을 통해 해가 가

워지면 그 주위에만 빛이 들고, 해가 멀어지면 전체 지형에 해가 비추어져서 밝아진다는 것을 알 수 있을 것이다. 이 시뮬레이션으로 어린이들을 위한 학습 또는 연구에도 상당히 많은 활용과 도움이 될 것이다. 실제 현실에서는 자세히 볼 수 없었던 것을 가상 공간에서 실제적인 데이터로 정확하고 자세한 정보를 얻을 수 있을 것이다.

## 참고문헌

- [ 1 ] 위키백과, [https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0\\_%EA%B7%B8%EB%9E%98%ED%94%BD%EC%8A](https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0_%EA%B7%B8%EB%9E%98%ED%94%BD%EC%8A)
- [ 2 ] 고희석, 김정현, 우운택, 윤경현, 이인권, 이제희, “컴퓨터 그래픽스 연구 현황:2부 융합하는 응용 기술”, 한국컴퓨터그래픽스학회논문지, v.18, no.2, 2012년, pp.35-53
- [ 3 ] 네이버블로그, 비치, <https://blog.naver.com/daehuck/220457203383>
- [ 4 ] Daniel Shiffman, 3D terrain Generation with Perlin Noise in Processing, The coding Train, <https://www.youtube.com/watch?v=IKB1hWWedMk>
- [ 5 ] 펄린노이즈 1탄(Perlin Noise), 3DMP, <https://3dmpengines.tistory.com/170>
- [ 6 ] 네이버블로그, 어느 테크니컬 아티스트의 블로그, <https://atelierseera.blog.me/221099942778>
- [ 7 ] 티스토리, ikfluencer, <https://iskim3068.tistory.com/35>
- [ 8 ] Processing, “noise()”, <https://processing.org/reference/libraries/#3d>
- [ 9 ] Processing, Noise3D, <https://processing.org/examples/noise3d.html>