3차원 지형 생성과 물체의 인터렉션

20201487 / 이민규 / 컴퓨터공학과 / 2학년

**요약** : 자바스크립트를 이용한 3차원 지형 생성과 물체의 인터렉션은 Three.js, p5.js와 같은 라이브러리를 활용하여 구현할 수 있다. 이를 통해 사용자는 마우스, 터치 등을 이용하여 3차원 공간에서 물체와 상호작용할 수 있다. 이는 시각적인 효과와 사용자 경험을 개선시키며, 웹 기반 3D 게임 및 시뮬레이션 등 다양한 분야에서 활용될 수 있다.

**1.서론** 3차원 지형 생성과 물체의 인터렉션은 컴퓨터 그래픽스 및 게임 개발 분야에서 중요한 역할을 합니다. 이 기술은 게임에서 훨씬 더 현실적인 그래픽을 제공하고, 사용자가 게임 세계 속에서 더욱 몰입감을 느끼게 한다. 또한, 3차원 지형 생성 및 물체 인터렉션 기술은 시뮬레이션, 가상현실, 교육 등 다양한 분야에서도 활용된다.

이러한 기술의 발전은 현재까지도 지속되고 있으며, 다양한 알고리즘과 기술이 개발되고 있다. 이 논문에서는 자바스크립트를 활용한 3차원 지형 생성 및 물체 인터렉션 기술을 다루며, 해당 기술의 구현 방법과 장점에 대해 소개하고자 한다.

**2.관련 연구**

3D 지형 생성 및 물체의 인터렉션에 대한 연구는 컴퓨터 그래픽스, 가상 현실, 게임 개발, GIS 및 지형 모델링 등 다양한 분야에서 활발히 이루어졌다. 이전 연구 중 일부를 소개한다.

[지형 생성 및 모델링]

Perlin Noise: Ken Perlin이 개발한 알고리즘으로, 임의의 3D 지형을 생성하는 데 사용된다.

Fractal Terrain Generation: Mandelbrot의 분수 차원 기하학(fractal geometry)을 이용하여 지형을 생성한다. 이 방법은 지형의 자연스러운 모양을 재현할 수 있다.

Procedural Terrain Generation: 무작위로 생성된 3D 모델의 특징을 통해 지형을 생성한다. 특정 규칙을 사용하여 특정 지형을 만들 수 있다.

[물체 인터렉션]

Physics-based Animation: 물리 엔진을 사용하여 물체의 동작을 모델링한다. 물리 엔진을 이용하면 물체의 이동, 충돌, 탄성 등의 물리적 특성을 모방할 수 있다.

Kinematic Animation: 물리 엔진 대신 사전에 지정된 운동 경로를 따르는 애니메이션을 생성한다. 이 방법은 게임 등에서 물체의 애니메이션을 만드는 데 유용하다.

Haptic Interaction: 사용자가 물체를 직접 조작할 수 있는 인터렉션 기술이다. 터치스크린, 조이스틱, 키보드 등의 입력장치를 통해 물체를 제어할 수 있다.

이 외에도 다양한 연구들이 있으며, 최근에는 딥러닝을 이용한 3D 모델링 및 인터렉션 연구도 활발하게 이루어지고 있다.

**3.기술 소개**

Vertex: 3D 모델링에서 사용되는 개념으로, 3D 모델의 꼭짓점을 나타낸다. 각 꼭짓점은 3D 공간상의 좌표(x, y, z)로 정의된다. Vertex를 이용하면 3D 모델을 구성하는 다각형의 꼭짓점을 정의할 수 있으며, 이들 꼭짓점을 이어 다양한 형태의 3D 모델을 만들 수 있다.

beginShape: Processing 언어에서 사용되는 함수 중 하나로, 2D 또는 3D 모델링을 시작할 때 사용된다. 이 함수는 새로운 다각형을 만들기 시작하고, 이후 추가되는 Vertex를 새로운 다각형의 꼭짓점으로 추가한다. 이를 통해 여러 개의 다각형을 이어 붙여 복잡한 3D 모델을 만들 수 있다.

Noise: Noise는 임의의 값을 생성하는 함수 중 하나로, 자연스러운 무작위 패턴을 만들기 위해 사용된다. 이 함수는 일정한 주기로 변화하는 값의 배열을 생성하며, 이 배열을 이용하여 3D 지형을 생성할 수 있다. 예를 들어, 지형의 높이를 Noise 함수를 이용해 생성하면, 자연스러운 지형의 높이 변화를 표현할 수 있다.

재질: 재질(Material)은 3D 모델의 표면을 나타내는 속성으로, 표면의 색상, 질감, 광택 등을 결정한다. 재질을 적용하여 모델의 외형을 더욱 자연스럽고 현실적으로 만들 수 있다. 또한, 재질을 이용하여 물체와의 인터렉션을 묘사할 수도 있다. 예를 들어, 물체가 어떤 재질의 지형과 충돌할 경우, 해당 지형의 속성(표면의 거칠기, 미끄러움 등)에 따라 물체의 움직임이 달라질 수 있다.

이러한 기술들을 종합하여 3D 지형 생성 및 물체 인터렉션을 구현할 수 있다.

**4. 3D 지형 생성**

이 코드에서 3D 지형은 Perlin 노이즈(Perlin noise)를 사용하여 생성된다.

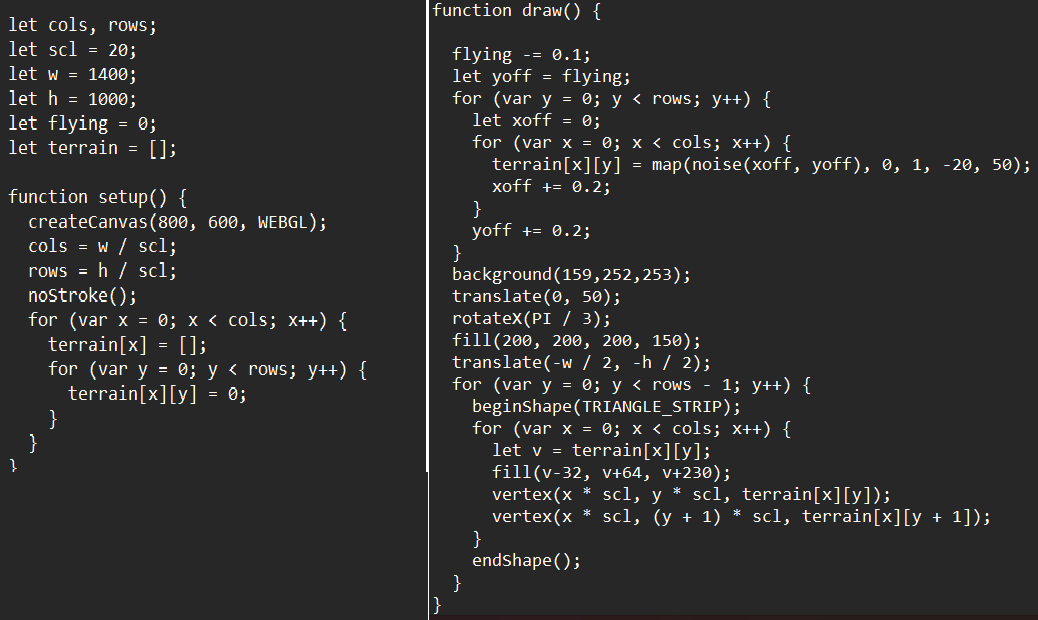


그림1. 지형 생성 코딩 텍스트

setup() 함수에서는 먼저 cols와 rows 변수를 설정하고, terrain 이라는 2차원 배열을 생성한다. 이 배열은 지형의 각 점마다 해당하는 높이 값을 저장할 것이다.

그리고 draw() 함수에서는 flying이라는 변수를 감소시키면서, 지형을 갱신한다. Perlin 노이즈를 사용하여 x와 y 좌표에서 해당하는 높이 값을 구하고, terrain 배열에 저장한다. 이를 위해서 xoff와 yoff 라는 변수를 사용하여 노이즈 값을 계산하고 있다.

그리고 translate()와 rotateX() 함수를 사용하여, 지형을 올바르게 회전 및 이동시킨다. beginShape()와 endShape() 함수를 사용하여 TRIANGLE\_STRIP을 만들어 지형을 그리고, fill() 함수를 사용하여 각 삼각형의 색상을 지정한다.

따라서, 이 코드에서는 Perlin 노이즈를 이용해 지형을 생성하고, 3D 그래픽을 그리는데 사용하는 다양한 함수를 사용하고 있다.

**5. 물체 인터렉션**

해당 코드에서는 마우스 커서의 위치를 이용하여 물체 인터랙션을 생성하고 있다. 마우스 위치에 따라 움직이는 기둥과 박스의 위치, 회전, 크기 등을 변화시켜 물체 간 상호작용을 구현하고 있다.

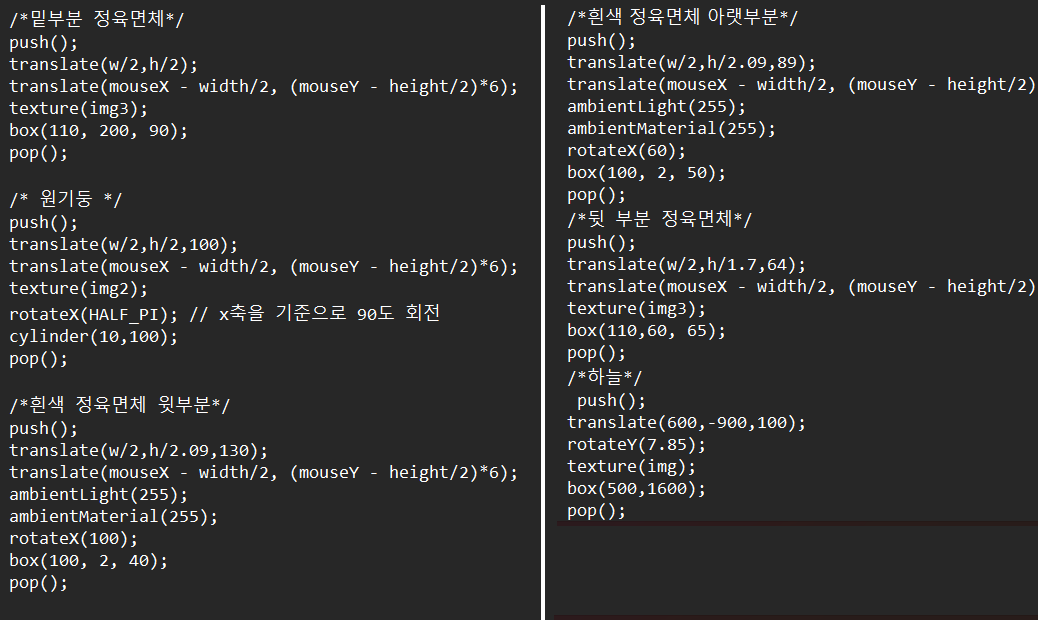


그림2. 물체 인터렉션 코딩 텍스트

코드에서는 translate() 함수를 이용하여 기존 위치를 변경하고, mouseX, mouseY 값을 이용하여 마우스 커서의 위치에 따라 물체를 이동시키고 있다. 또한 rotateX() 함수를 이용하여 x축을 기준으로 회전하고, box() 함수와 cylinder() 함수를 이용하여 크기와 형태를 변경하고 있습니다.

이러한 인터랙션은 draw() 함수 내에서 매번 업데이트되며, 마우스의 위치에 따라 물체의 위치와 형태가 동적으로 변경된다. 이를 통해 사용자와 상호작용이 가능한 3D 모델링이 가능해진다.

**6. 결과 및 평가**

해당 코드는 3D 그래픽을 이용하여 지형과 물체를 그리고, 마우스 인터렉션을 통해 물체를 움직이는 코드이다.

코드 실행 결과, 청록색 하늘과 함께 지형이 나타난다. 지형은 Perlin noise 알고리즘을 이용하여 불규칙한 높이를 생성하며, 파도와 같은 모습이 그려지는 것을 볼 수 있다. 또한, 마우스를 이용하여 물체를 조작할 수 있습니다. 물체는 3D 모델링 기법을 사용하여 직육면체와 원기둥으로 구성되어 있으며, 이미지 텍스처와 색상이 적용되어 있다. 또한, 마우스를 움직이면 물체가 움직이며, 물체와 지형이 충돌하는 경우 물체가 반사되어 뒤로 움직인다.

이 코드는 3D 그래픽을 다루기 위해 WEBGL과 p5.js를 사용한 것이 매우 효과적입니다. 또한, 마우스 인터렉션을 통해 사용자와의 상호작용을 가능하게 한 것은 코드의 장점입니다.

그러나 개선해야 할 점으로는, 지형과 물체가 충돌했을 때의 효과가 더욱 현실적이지 않아 개선할 여지가 있습니다.

**참고 영상**

The Coding Train - Coding Challenge 11: 3D Terrain Generation with Perlin Noise in Processing

<https://www.youtube.com/watch?v=IKB1hWWedMk&t=2s>