유니티3D를 이용한 3D 랜덤 맵 제네레이팅 기법

정동진[1] · 이임건[2*]

[1]동의대학원 · [2]동의대학교

3D Random Map Generator with Unity3D

Dong-jin Jung[1] · Imgeun Lee[2*]

[1]Dong-eui University Graduate School · [2]Dong-Eui University

E-mail : kojdj@naver.com / iglee@deu.ac.kr

요 약

소규모 게임 개발사가 많아지고 있는 지금, 금전적 자금이 부족한 개발사는 다양하고 퀄리티 있는 맵을 제작하기 어려움이 따른다. 그러나 무작위로 맵을 생성하여 게임에 적용하면, 시간적 비용과 인적 비용을 현저히 절감할 수 있다. 본 연구에서는 랜덤하게 생성한 스무스 노이즈맵을 활용하여 게임에서 사용 가능한 3D 맵을 제네레이팅 하는 방법을 기술한다.

ABSTRACT

To day, the number of small game developer has increased, developers who lacked financial resources have difficulty in creating various and quality maps. However, if applied in game that randomly generated maps, development time and human costs will be significantly reduce. In this paper, we describe a method for generating 3D maps with smooth noise maps, can be used in games.

키워드

유니티, 3D 맵, 랜덤 맵, 제네레이터, 제네레이팅

Ⅰ. 서 론

기존의 랜덤 맵 제네레이터의 경우, 먼저 스무스 노이즈맵을 생성한다. 그리고 이렇게 생성된 노이즈맵의 픽셀값을 이용하여, 지오메트리가 될 메쉬에 사용자가 정의한 버텍스의 수와 매쉬의 너비를 기준으로 하여 CPU를 사용한 연산을 통해 매쉬에 존재하는 각각의 버텍스의 위치를 변경하는 방법으로 무작위적인 지오메트리를 생성한다. 이는 오픈월드형 게임과 같이 광활한 맵을 랜덤으로 생성하기에 굉장히 부적합 한데, 이는 CPU를 사용한 연산이 넓은 지오메트리를 생성하기 위하여 수많은 버텍스를 짧은 시간 안에 계산하기 때문에 프레임드랍을 유발시키는 주요 원인기 되기 때문이다. 따라서 이러한 기존의 맵 제네레이팅 기법을 사용할 경우, 플레이어가 아직 생성되지 않은 지역에 진입 할 때 마다 프레임 드랍 현상을 경험하도록 만든다.

이에 본 논문은, CPU가 아닌 GPU를 이용한 병렬 연산을 사용하는 랜덤 맵 제네레이팅 기법을 제안하며, 이를 통하여 기존의 CPU를 이용한 랜덤 맵 제네레이팅 기법과 비교하여 플레이어에게 보다 개선된 게임 플레이 환경을 구축하기 위한 방법을 기술한다.

Ⅱ. 기존 기술

스무스 노이즈맵을 이용한 일반적인 형태의 랜덤 맵 제네레이터는 CPU 연산을 사용한다[1].

유니티에서 작동하는 랜덤 맵 제네레이터의 기본적인 원리는, 먼저 지오메트리가 될 오브젝트를 생성하여, 사용자 정의 사각 매쉬를 컴포넌트로 추가한다. 이 매쉬는 CPU연산을 통하여 생성되며,

* corresponding author

사용자가 미리 입력한 폴리곤의 수를 기준으로 하여 생성된다.

생성된 매쉬를 가지는 오브젝트에서 임의의 키 값을 이용하여, CPU 연산을 통해 스무스 노이즈맵을 랜덤으로 생성한다. 이렇게 생성된 스무스 노이즈맵의 픽셀의 색상값은 앞으로 만들어질 지오메트리의 높이값으로 사용된다.

매쉬에 존재하는 각각의 버텍스의 로컬 좌표와 대응되는 스무스 노이즈맵의 UV 좌표상의 픽셀값을 높이 삼아, 픽셀값에 사용자 정의 델타값을 곱한 결과를 픽셀에 대응하는 버텍스의 높이로 설정한다.

마지막으로, 오브젝트에 머테리얼 컴포넌트를 추가하여, 생성된 노이즈맵을 기준으로 높이에 알맞은 자연스러운 색상의 이미지를 생성해서 머테리얼의 텍스쳐로 설정한다.

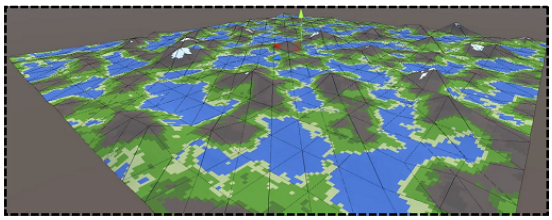

그림 1. 랜덤 맵 제네레이터의 결과물[1]

III. 구현 방법

Unity3D에서 GPU를 이용하여 병렬연산을 사용하는 방법은 쉐이더를 이용하는 것이다. Unity3D는 HLSL과 GLSL을 제공하며, HLSL는 DirectX에서 사용하는 쉐이더 언어이며, GLSL는 OpenGL에서 사용하는 쉐이더 언어이다.

쉐이더에는 프레그먼트 쉐이더, 버텍스 쉐이더, 지오메트리 쉐이더 등으로 나뉜다. 본 연구에서는 랜덤 맵 제네레이터의 텍스쳐와 버텍스를 연산하기 위한 도구로 쉐이더를 사용할 것 이므로, 텍스쳐 연산에 사용되는 프레그먼트 쉐이더와 버텍스 연산에 사용되는 버텍스 쉐이더를 사용한다[2].

프레그먼트 쉐이더를 이용한 GPU의 병렬 연산을 통하여 노이즈맵을 생성하고, 생성된 노이즈맵을 기준으로 하여 출력할 텍스쳐를 만든다. 버텍스 쉐이더는 프레그먼트 쉐이더에서 전달하는 노이즈맵의 픽셀의 색상값에 델타값을 곱하여 버텍스의 높이를 설정한다.

IV. 기대 효과

이 모든 작업이 GPU의 병렬 연산으로 처리된다. 따라서 기존의 랜덤 맵 제네레이팅에 걸리는 CPU의 부하를, 매쉬를 생성하는 과정에만 CPU가 사용되도록 하고 노이즈맵의 생성과 텍스쳐의 생성, 그리고 지오메트리의 생성에 사용되는 연산량을 GPU로 분산시킨다.

따라서 프로그램의 퍼포먼스 향상을 기대 할 수 있다.

프로그램 개발시 프레임 드랍 이슈 등 CPU와 GPU의 연산 과부하로 인한 문제가 발생할시, Unity3D의 프로파일러 창을 이용하여 CPU와 GPU의 연산량을 분석하여 CPU와 GPU의 연산량을 조절하는 요소로서 응용할 수 있는 이점이 있다.

V. 결 론

기존에 사용되어 왔던 CPU만을 이용한 랜덤 맵 제네레이팅 기법은 게임에서 실시간으로 사용하기에는 한번에 처리해야 하는 연산량이 많아, 로딩 지연과 프레임 드랍과 같은 퍼포먼스 이슈가 발생할 가능성이 높았다.

본 연구로 인하여, GPU를 이용한 랜덤 맵 제네레이터를 사용할 경우 CPU에 부담되는 연산량을 GPU로 분산 시키는 것이 가능하게 되었으며, 이에 따라서 게임에 런타임 동안 동적으로 랜덤한 맵을 생성할 때 발생할 수 있는 퍼포먼스 이슈의 발생 가능성을 낮추는 것이 가능해 졌다.

또한 퍼포먼스 이슈 발생시, CPU와 GPU에 부담되는 연산량을 적절히 조절하여 퍼포먼스의 향상을 기대 할 수 있으므로 기존의 CPU만을 사용한 랜덤 맵 제네레이팅 기법보다 최적화에 유리하다는 결론을 도출하게 되었다.

이 연구의 다음 단계로서, GPU를 사용함으로서 가지는 이점인 컴퓨트 쉐이더와 지오메트리 쉐이더 등을 이용하여, 랜덤으로 생성된 지오메트리의 퀄리티 향상과 불필요한 연산의 소거에 대한 연구가 진행되어야 할 것이다.

References

[1] Procedural Landmass Generation (Unity 5) [Internet] Available : https://www.youtube.com/playlist?list=PLFt_AvWsXl0eBW2EiBtl_sxmDtSgZBxB3

[2] Render Hell – Book I [Internet] Available : https://simonschreibt.de/gat/renderhell-book1/