

Blender 4.5 Fire Simulations 기본

[Francesco Milanese](#)

이 PDF는 무료이며 자유롭게 배포할 수 있습니다. 내용이 마음에 드신다면, 제 [YouTube](#) 채널을 구독하고 영상에 '좋아요'를 눌러 주시면 가장 큰 감사의 표시가 됩니다. 감사합니다!

1. [기초](#)
2. [Principled Volume, Blackbody](#)
3. [Cache, Baking, Offset](#)
4. [Geometry, Inflow, Outflow](#)
5. [Domain Settings, 1/2 \(Gas, Fire, Collections\)](#)
6. [Domain Settings, 2/2 \(Weights, Data, Render, Display\)](#)
7. [Emission, Attributes, Volume Info](#)
8. [Effectors \(Guides, Colliders\)](#)
9. [Force Fields \(Turbulence, Wind, Curve Guide\)](#)
10. [Motion Blur, Filmic, Compositing, Eevee](#)

Blender 4.5 Fire Simulations 기본 | 1/10: 기초

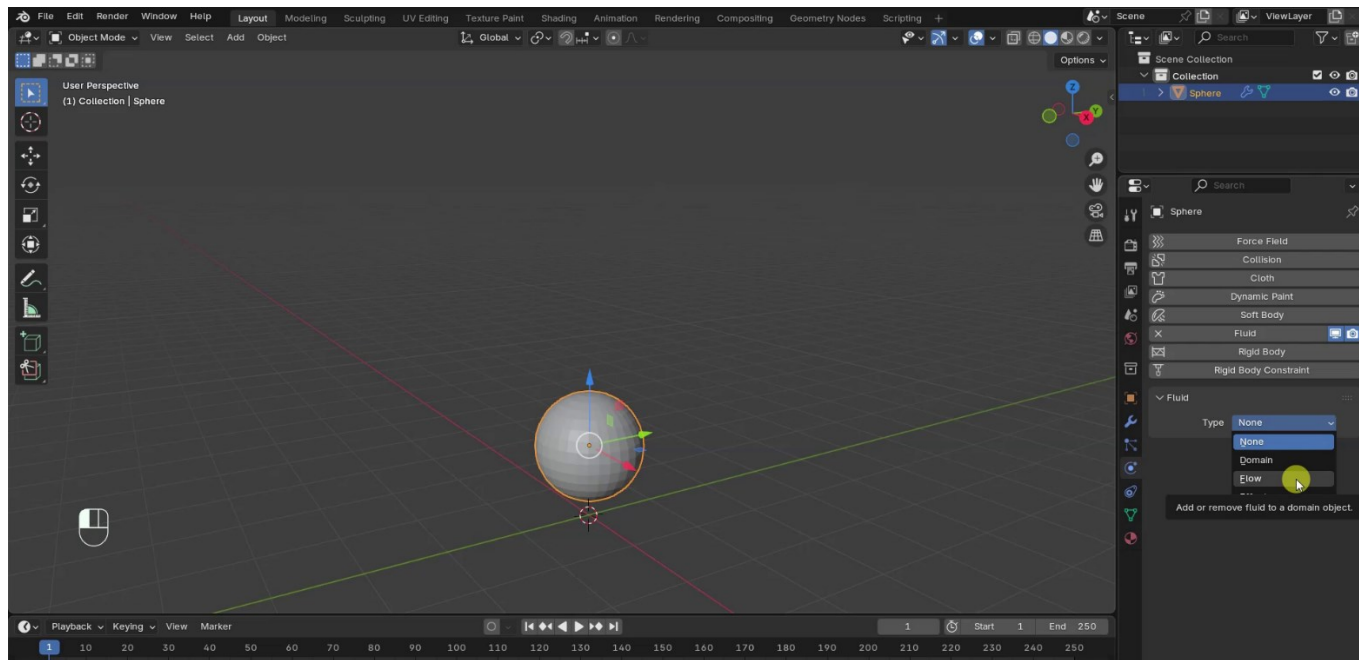
<https://youtu.be/q7pYluwvzr0>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 첫 번째 튜토리얼에서는, Blender에서 시뮬레이션을 만들기 위해 필요한 두 가지 기본 요소를 살펴보겠습니다.

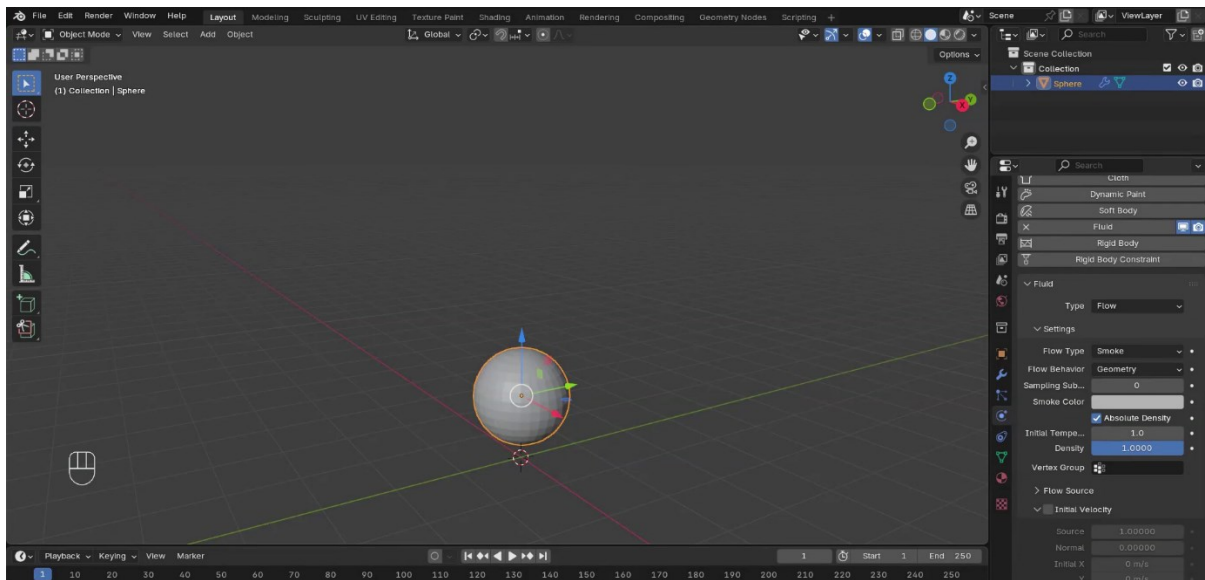
빈 Blender 씬에서 시작해 보겠습니다. 씬에 Fire나 Smoke를 추가하려면, 가장 먼저 이러한 요소를 방출하는 오브젝트를 넣어야 한다고 생각할 수 있습니다. 그 생각은 맞지만, Blender에서 시뮬레이션을 실행하기 위해 필요한 기본 요소의 절반만 포함하고 있습니다. 어쨌든 우선 Fire와 Smoke의 방출체 역할을 할 오브젝트를 씬에 추가해 보겠습니다.

Blender의 Fire와 Smoke 시뮬레이션 도구는 각 오브젝트의 Physics 탭에 있습니다. 우리가 추가해야 할 컴포넌트의 유형은 Fluid인데, 이는 Liquid뿐만 아니라 Fire와 Smoke 시뮬레이션도 처리합니다. 처음에는 Type이라는 메뉴만 보이는데, 이 단계에서는 아직 Fluid나 다른 카테고리를 구분하지 않고, 단순히 시뮬레이션에서 가능한 액터 역할을 나열합니다.

각 요소 유형의 Tooltip은 그 기능에 대한 힌트를 줍니다. 특히 Flow 타입은 씬에 요소를 추가하거나 제거하는 데 사용된다는 것을 알 수 있습니다. 우리가 만든 첫 번째 오브젝트는 Fire와 Smoke의 방출체가 되어야 하므로, 이 유형을 선택하겠습니다.



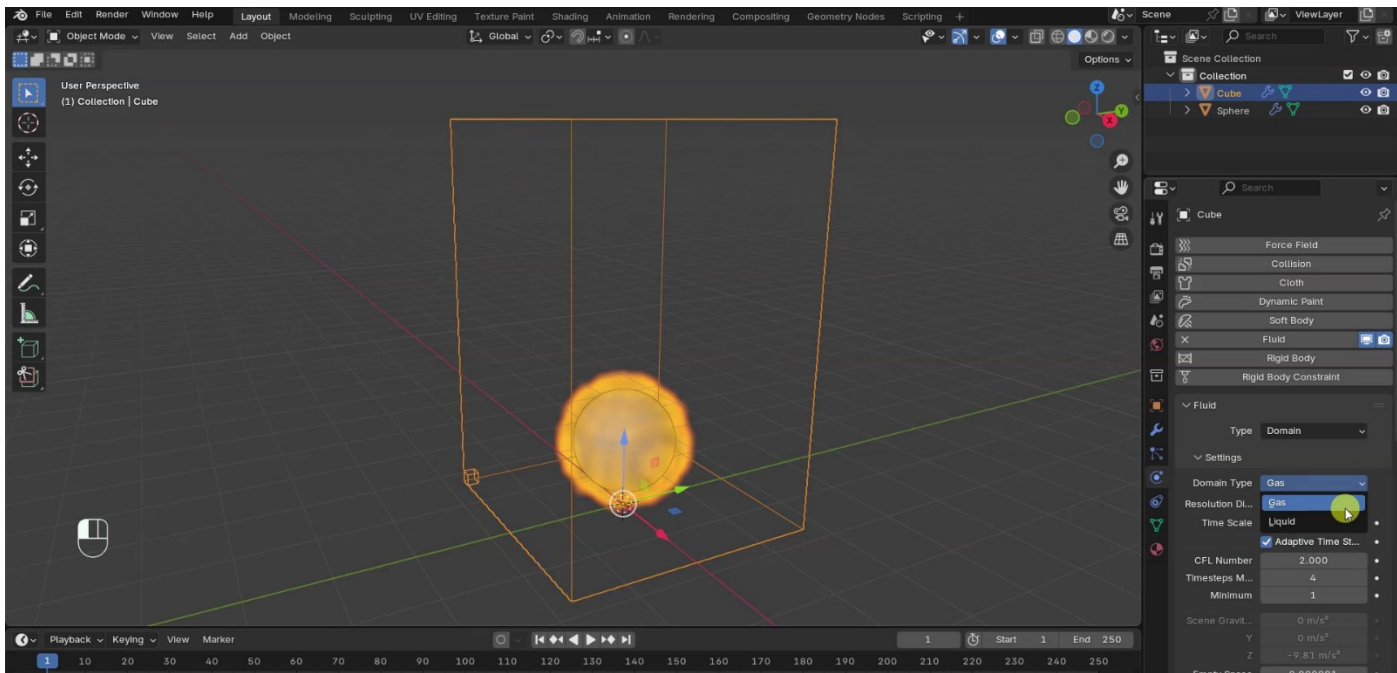
이제 컴포넌트 패널에 여러 새로운 옵션이 나타납니다. 첫 번째는 Flow Type으로, 기본값은 Smoke입니다. 이 메뉴를 열어보면 Liquid에는 하나의 옵션만 있는 반면, Fire와 Smoke에는 세 가지 선택지가 있다는 것을 알 수 있습니다. 두 요소를 개별적으로 사용할 수도 있고, 조합해서 사용할 수도 있습니다. Fire And Smoke 옵션에는 두 요소 모두의 파라미터와 설정이 포함되어 있으므로, 이 튜토리얼 시리즈의 첫 부분에서는 이 옵션을 선택하겠습니다.



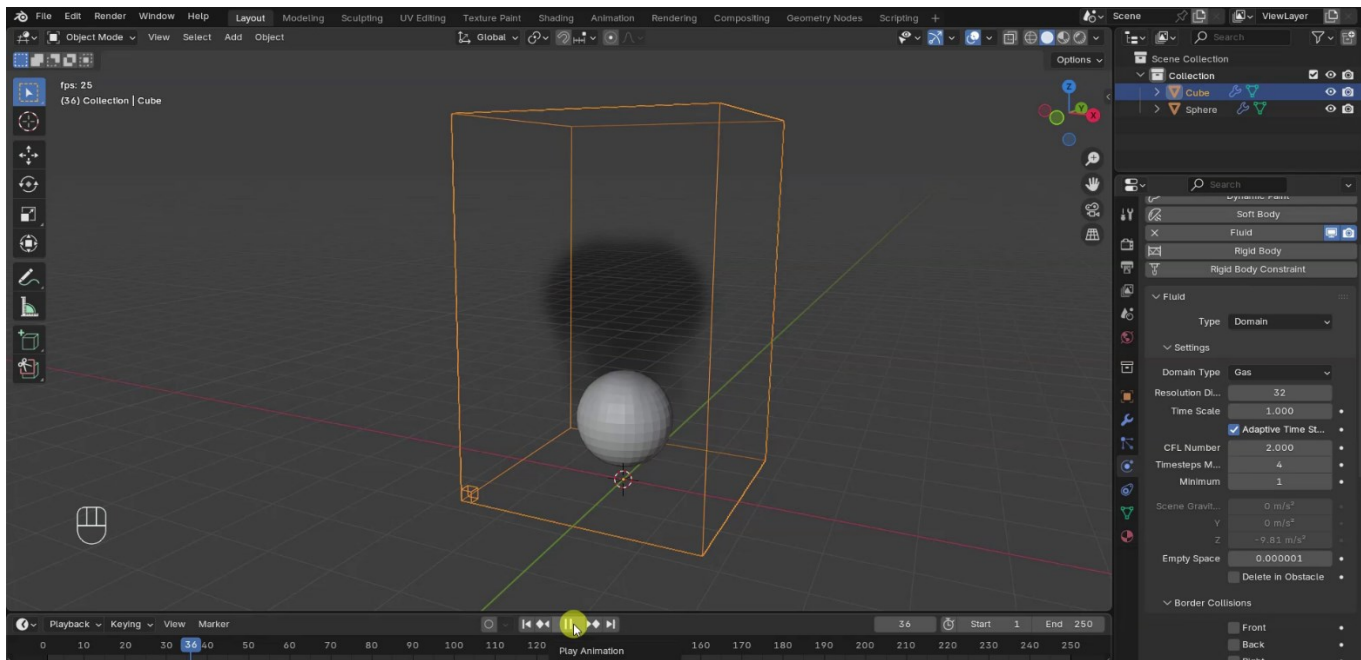
Cloth나 Collision 같은 Blender의 다른 물리 시뮬레이션에 익숙하다면, 이제 Timeline의 첫 프레임으로 가서 Play 버튼을 눌러 Fire와 Smoke 시뮬레이션이 시작되길 기대할 수 있습니다. 하지만 이렇게 하면 아무런 결과가 나오지 않습니다. 따라서 Timeline의 첫 프레임으로 돌아가, Fluid 시뮬레이션에 필요한 두 번째 기본 액터인 Domain을 추가해야 합니다.

모든 Fluid 시뮬레이션은 메쉬로 정의된 3D 공간, 즉 Volume 안에서 이루어져야 합니다. Blender에서는 이 Volume을 시뮬레이션의 Domain이라고 부릅니다. 이는 주로 성능상의 이유 때문인데, 이러한 시뮬레이션은 많은 메모리와 연산을 요구하기 때문입니다. Domain은 어떤 형태의 오브젝트라도 될 수 있지만, Blender는 결국 Bounding Box를 사용합니다. 따라서 가장 명확한 선택은 Cube를 적절하게 스케일링해서 사용하는 것입니다.

이제 방출체 오브젝트를 Domain 역할을 할 Cube 안에 넣었으니, Cube를 선택하고 Fluid 컴포넌트를 추가한 다음 Type을 Domain으로, Subtype을 Gas로 설정하겠습니다. Cube는 즉시 투명하게 변하면서 내부 구조가 보이게 됩니다. 또한 한쪽 하단 모서리에 작은 Cube가 나타나는데, 곧 그것이 무엇인지 보게 될 것입니다. 그러나 가장 큰 변화는 Domain 안의 Inflow 오브젝트가 이제 불꽃에 둘러싸이게 되었다는 점입니다!



Timeline에서 첫 번째 프레임을 현재 프레임으로 설정한 후 Play를 눌러 시뮬레이션을 시작해 보겠습니다. 무언가 분명히 일어나지만, 첫 번째 Fire와 Smoke 시뮬레이션치고는 다소 실망스러워 보입니다. 불꽃은 처음 몇 프레임에서만 나타나고, 그 이후에는 Smoke만 남기 때문입니다.



이런 동작의 원인은 방출체 오브젝트의 설정 중 하나에 있습니다. 해당 오브젝트를 선택해 보겠습니다. 우리가 확인해야 할 Fluid 컴포넌트 파라미터는 Flow Behavior로, 기본값은 Geometry입니다. Tooltip에 따르면 Geometry는 현재 지오메트리를 사용해 Fire를 생성하는 반면, Inflow 모드는 씬에 Fire를 지속적으로 추가합니다. 따라서 모드를 Inflow로 전환한 뒤 Timeline의 첫 프레임으로 돌아가 다시 Play를 눌러 보겠습니다.

Inflow의 “adds fire”라는 표현이 핵심입니다.

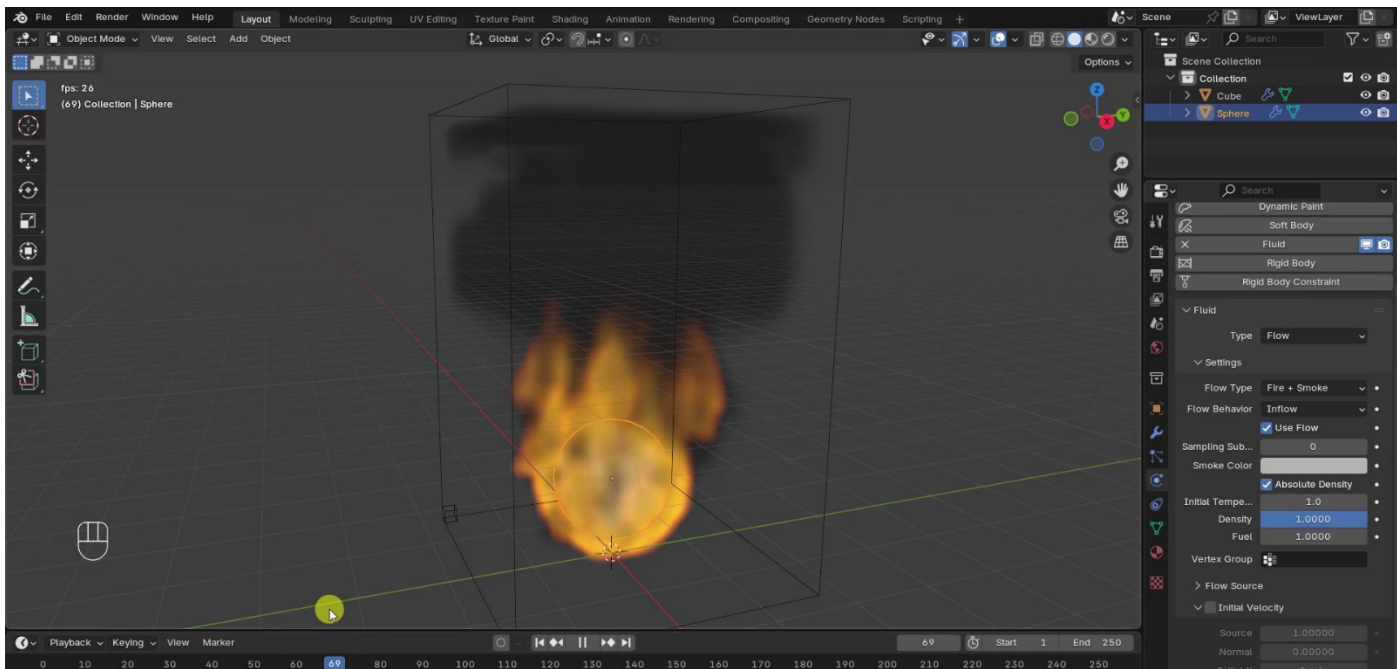
Inflow 오브젝트는 매 프레임마다 지오메트리로부터 새로운 시뮬레이션 요소를 계속 생성합니다.

반면 Geometry 모드는 첫 프레임에서만 지오메트리를 고려하고 즉시 소모해 버립니다.

비유하자면, Geometry 모드는 작은 종잇조각을 태우거나 스모크 머신이 한 번만 연기를 내뿜는 것과 같고, Inflow는 가스레인지 버너나 계속해서 연기를 내뿜는 스모크 머신과 같습니다.

오브젝트의 하단 부분에서 특히 불꽃이 약간 블록처럼, 정확히는 3D이므로 큐브처럼 보인다는 것을 눈치채셨을 수도 있습니다. 이러한 모양은 Domain 파라미터 중 하나인 Resolution 때문인데, 이는 Domain의 볼륨이 작은 큐브로 얼마나 세밀하게 분할되는지를 결정합니다.

실제로 이 Resolution은 Domain 하단 모서리에 나타나는 작은 Cube로 시각적으로 표시되며, 특히 낮은 값일 때는 불꽃과 Smoke가 그 크기의 큐브로 나뉘어 보이는 것을 쉽게 확인할 수 있습니다.



Fluid 시뮬레이션은 Domain의 볼륨을 작은 Cube로 분할하여 계산하는데, 이를 Voxel이라고 부릅니다.

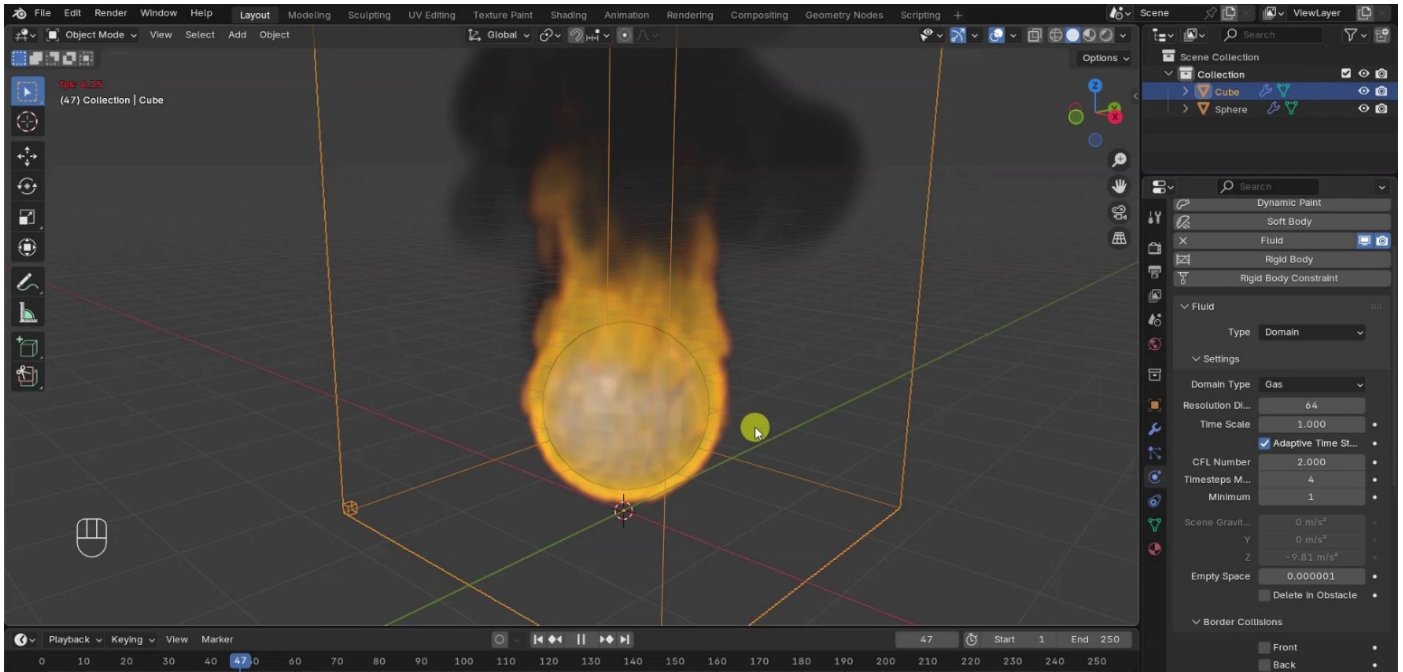
Voxel은 시뮬레이션 계산에 사용되는 가장 작은 단위이며, 2D 이미지의 Pixel에 해당하는 3D 개념입니다.

Pixel과 마찬가지로, 동일한 볼륨에 대해 Voxel 수가 많을수록 품질은 좋아지지만, 더 많은 연산 자원과 디스크 메모리가 필요합니다. 가장 좋은 접근 방법은 기본 Resolution 값인 32에서 시작하여 다른 파라미터를 조정하면서 원하는 Fire와 Smoke 형태를 얻은 다음, Resolution을 점차 높여 테스트하고 최종 결과물에는 아주 높은 값을 사용하는 것입니다.

이제 메모리 사용과 관련해, Fluid 시뮬레이션 데이터가 어디에 저장되는지도 살펴보겠습니다. 특히 저는 아직 프로젝트 파일을 저장하지 않았습니다. 이 정보는 Domain 오브젝트의 Cache 섹션에서 확인할 수 있으며, 거기에 표시된 경로를 보면 Blender가 시뮬레이션 파일을 저장할 디스크 폴더를 생성한 것을 알 수 있습니다. 현재 프로젝트 파일을 저장하지 않았기 때문에, 이 폴더는 임시 위치에 만들어졌습니다. 그러니 제가 한 것처럼 하지 마시고, 항상 Blender 프로젝트 파일을 Domain과 다른 Fluid 시뮬레이션 요소를 만들기 전에 저장하세요. 그래야 Cache 폴더가 프로젝트 파일과 같은 디렉토리에 생성되어, 디스크에서 따로 찾아 헤맬 필요가 없습니다!

이 시점에서 뷰포트 셰이딩 모드를 전환하거나 Timeline의 중간 프레임에서 렌더링을 실행해도 Fire나 Smoke는 보이지 않습니다. 그 이유는 현재 3D Viewport의 Solid 모드에서 보이는 것은 단지 3D 공간에서 Fire와 Smoke의

Voxel 시각화일 뿐이며, 아직 렌더링용 Material을 정의하지 않았기 때문입니다. 이 부분은 다음 에피소드에서 다루겠습니다!



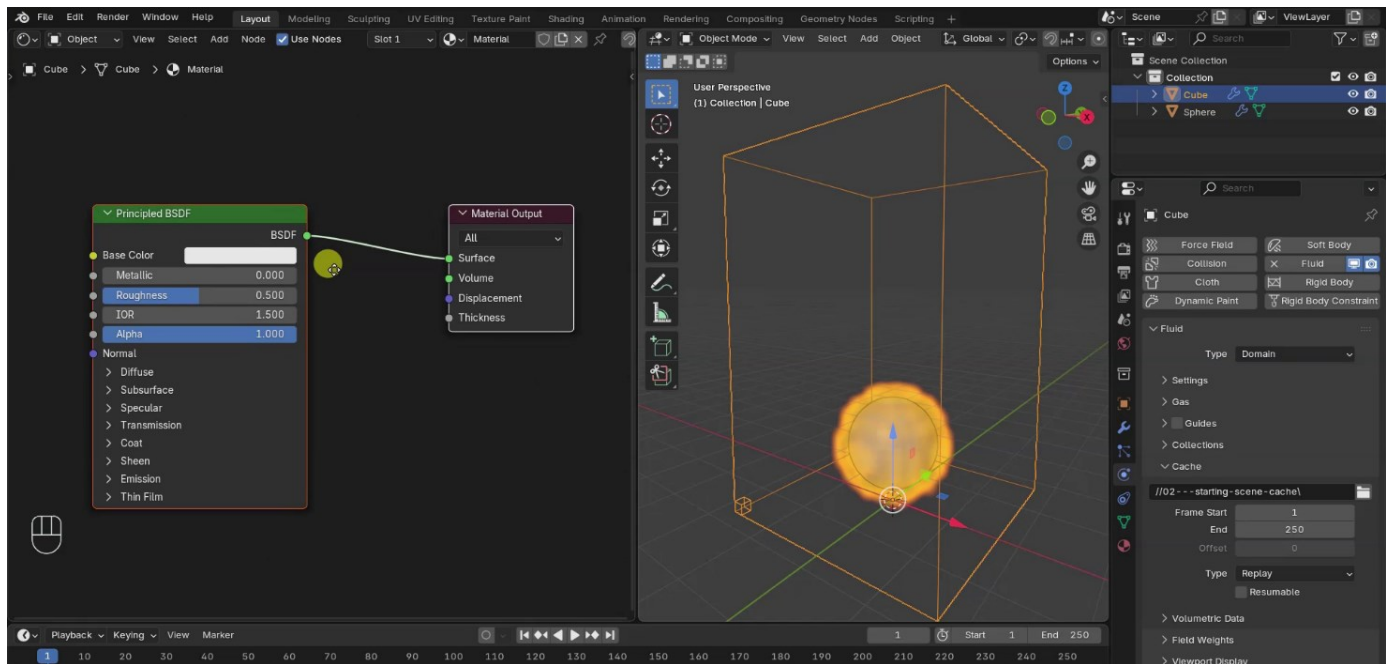
Blender 4.5 Fire Simulations 기본 | 2/10: Principled Volume, Blackbody

https://youtu.be/X6ba1ZP8_dk

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 두 번째 튜토리얼에서는, Cycles에서 Fire와 Smoke를 렌더링하기 위한 기본 Material 설정 방법을 살펴보겠습니다. 우리는 Principled Volume Shader와, 이 Material이 제공하는 두 가지 모드인 Emission과 Blackbody를 살펴보겠습니다.

저는 이전 튜토리얼 마지막에 사용했던 씬에서 이어가겠습니다. 이번에는 Blender 파일을 저장하고, 프로젝트 파일이 있는 동일한 디렉토리에 Cache 폴더를 설정해 두었습니다.

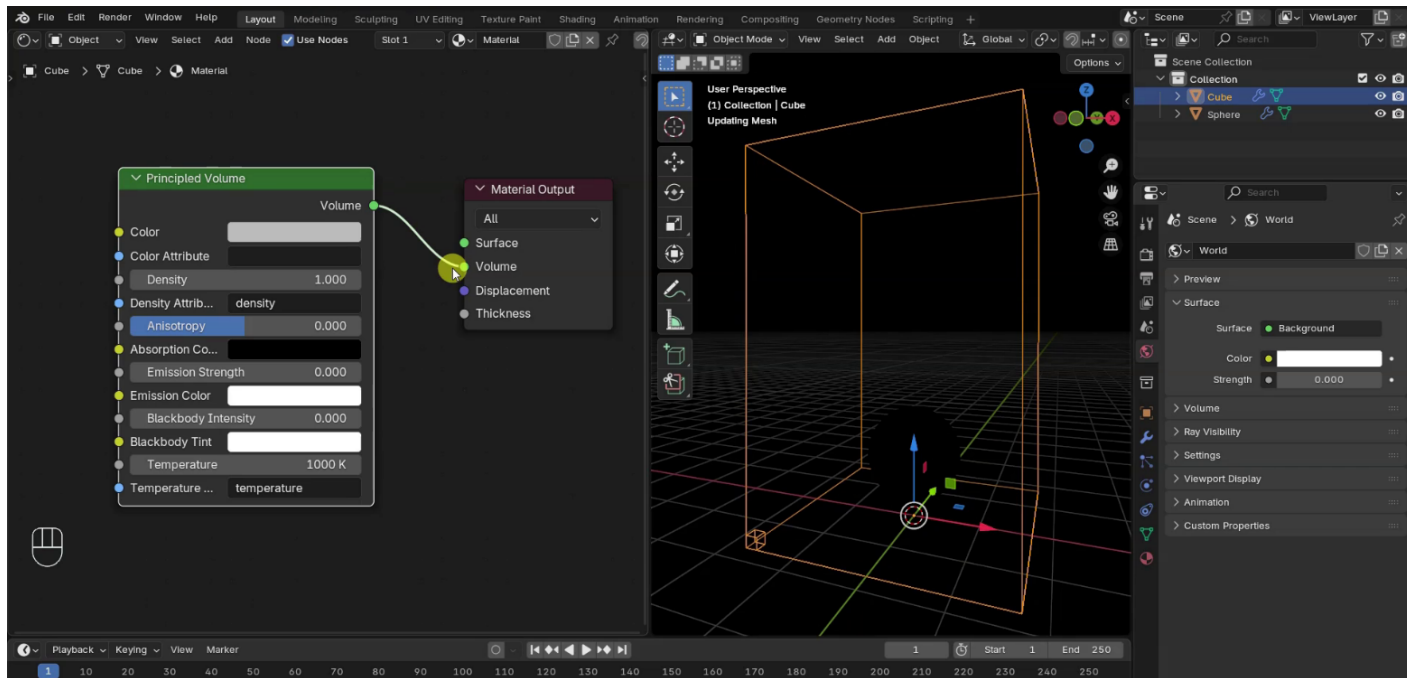
Fire와 Smoke용 Material은 씬의 Flow 오브젝트나 다른 오브젝트에 정의하는 것이 아니라 Domain에 정의해야 합니다. 따라서 Domain을 선택하고, 인터페이스에 Shading Editor를 추가한 다음 Rendered 뷰 모드로 전환하여 기본 Material을 생성합니다.



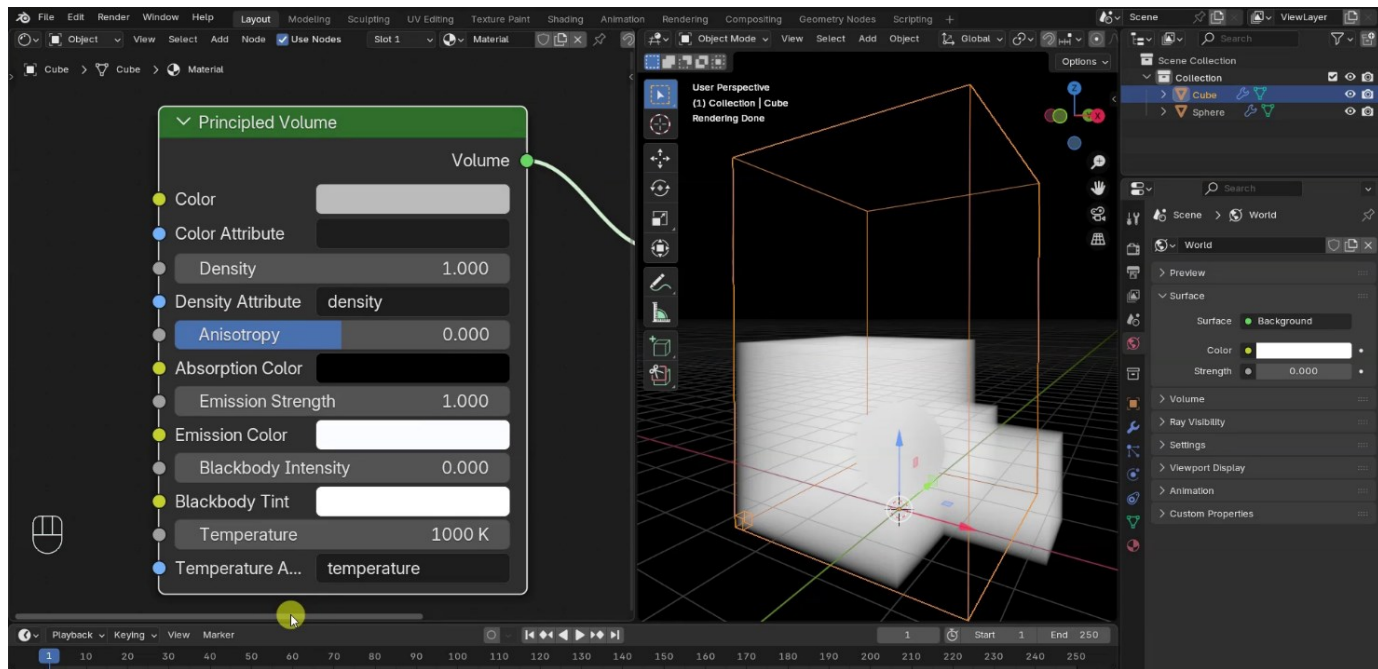
World 탭에서 Background의 Strength를 0으로 설정해 불꽃이 잘 드러나도록 합니다.

애니메이션을 재생해 보면 아직 아무것도 보이지 않는데, 이는 기본 Material로 생성된 Principled Shader가 불륨용 Material에는 적합하지 않기 때문입니다.

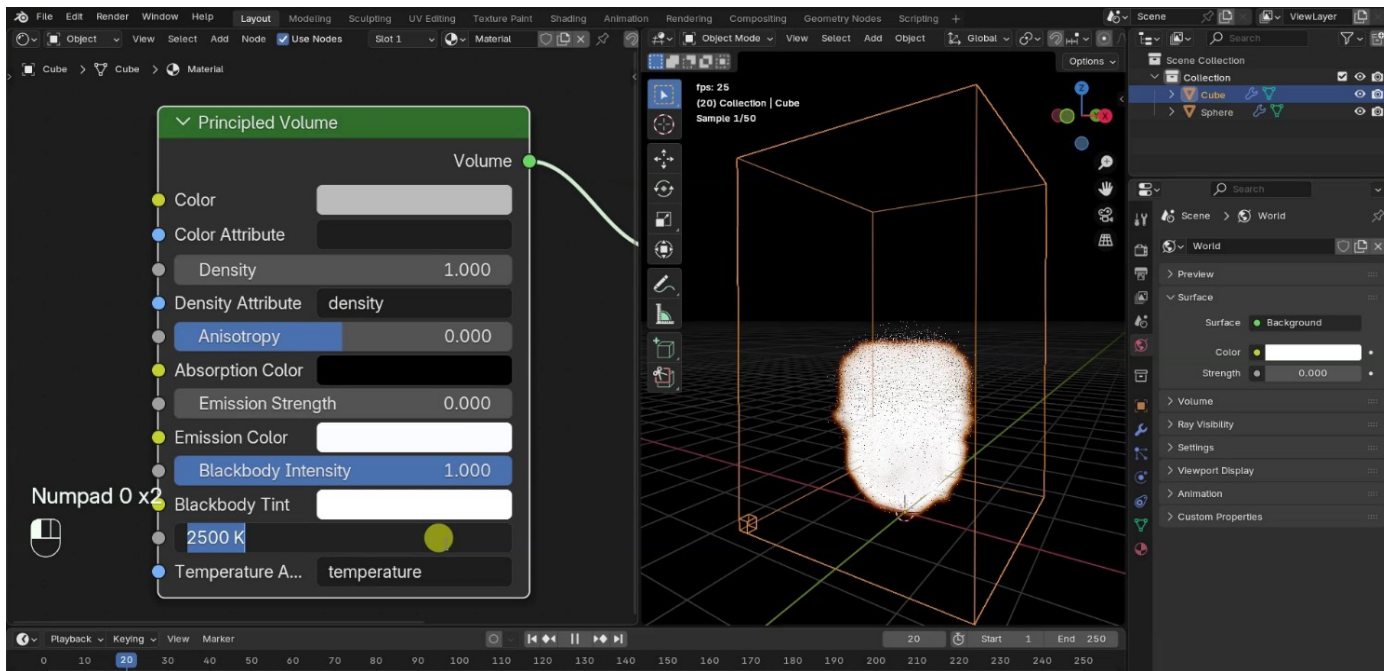
Principled Shader를 제거하고 Principled Volume Shader를 삽입한 뒤, 그 출력을 Material Output 노드의 Volume 입력에 연결합니다. 그러나 애니메이션을 재생해도 여전히 Rendered 미리보기에는 아무것도 보이지 않습니다. 이제 불꽃에 색을 입히는 방법을 살펴보겠습니다.



Blender는 불꽃을 색칠하는 두 가지 접근 방식을 제공합니다. 첫 번째는 Emission이라고 불립니다. 이는 물리적으로 정확하지 않고 처음 결과는 다소 실망스러울 수 있지만, 실제로는 불꽃의 강도와 색상을 관리하는데 큰 유연성을 제공합니다. 이 모드와 그 활용에 대해서는 이후 에피소드에서 더 자세히 다루겠습니다.



두 번째 접근 방식은 Blackbody라고 불리며, Strength를 1로 설정하고 해당 필드에 적절한 온도 값을 지정하면 물리적으로 정확합니다. 이 도구를 사용하면 Rendered 미리보기에서 즉시 멋진 불꽃을 확인할 수 있습니다. Temperature 값은 Kelvin 단위로 표현되며, 실제 세계 값과 그에 대응하는 불꽃 색상을 보여주는 많은 예시를 온라인에서 찾을 수 있습니다.



계속하기 전에, Render 탭에서 렌더 미리보기를 위해 Denoise를 활성화합니다.

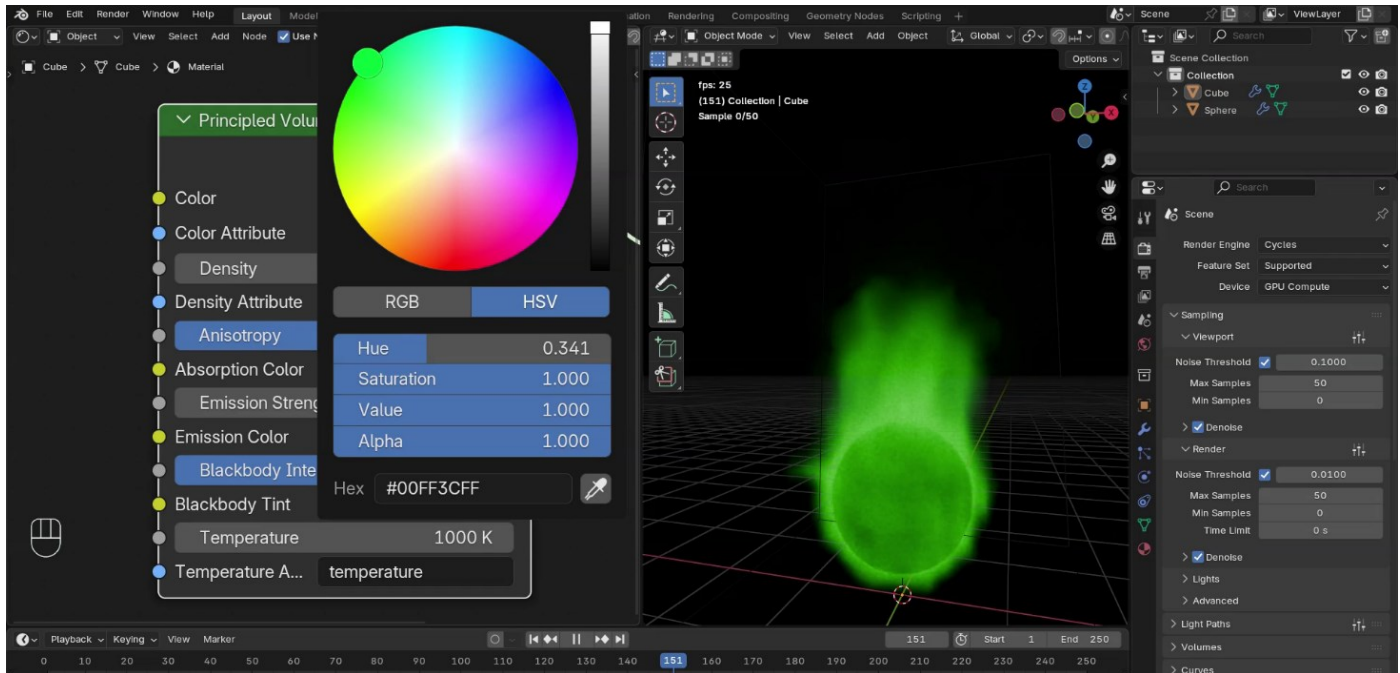
물리학에서 Blackbody는 모든 전자기 복사를 흡수하고, 온도에만 의존하는 스펙트럼(여기서는 단순히 색상이라고 부르겠습니다)을 다시 방출하는 이상적인 물체입니다. 이는 물체의 재질과는 무관합니다. 온도가 올라가면, 빨강계 달궈진 난로의 붉은색에서 노란-흰색으로, 더 올라가면 파란색으로 이동합니다. 실제로 표면 온도가 가장 높은 별들은 파란-흰색 거성입니다.

Intensity와 Tint 파라미터에 대해 말하자면, Intensity는 직관적인 의미를 가지고 있습니다.

앞서 언급했듯이 Intensity 값이 1일 때 물리적으로 현실적인 값이며, 1보다 크게 설정하면 불꽃이 더 밝아지지만 물리적으로는 정확하지 않습니다.

Tint 필드를 사용하면 불꽃의 색상을 변경할 수 있습니다. 물론 이것은 물리적으로 정확한 표현은 아닙니다. 물리적으로 정확한 결과를 원한다면 Temperature 값을 사용하고 Tint는 흰색으로 두는 것이 좋습니다. 하지만

Emission 모드에서 노드와 파라미터를 일일이 설정하지 않고 빠르게 색깔 있는 불꽃을 만들고 싶다면 이 옵션이 유용할 수 있습니다.

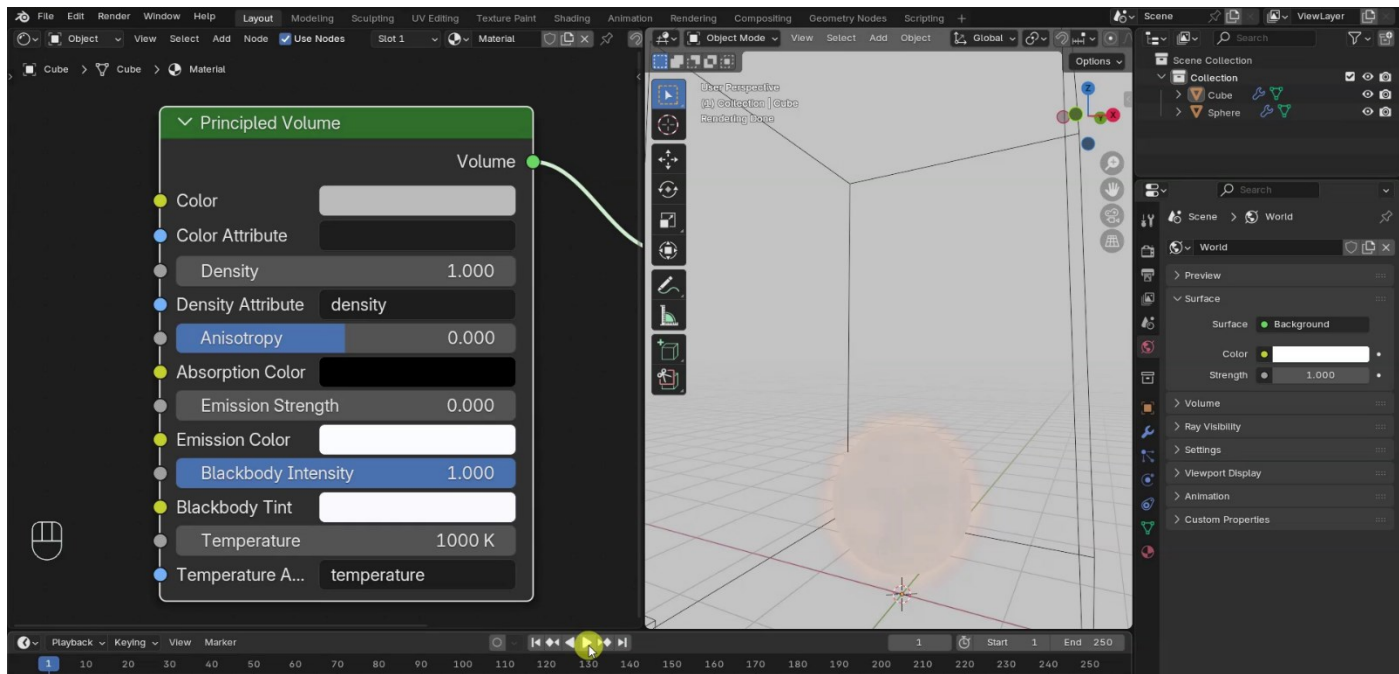


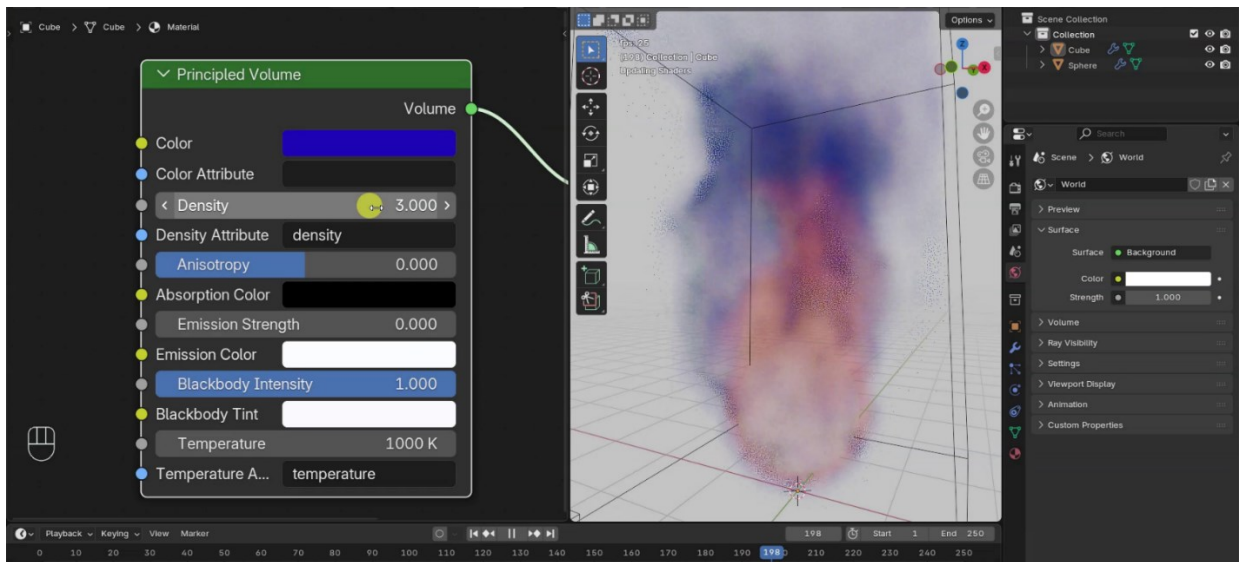
이제 아마도 Inflow 오브젝트마다 서로 다른 불꽃 색상을 어떻게 추가할 수 있는지 궁금하실 겁니다. 왜냐하면 색상은 Inflow 오브젝트가 아니라 Domain Material에서 정의되기 때문입니다. 기본적으로 두 가지 방법이 있습니다. 첫 번째는 Domain을 따로 만드는 것이지만, 이것은 불꽃들이 서로 상호작용할 필요가 없을 때만 잘 작동합니다. 두 번째 방법은 방출체 오브젝트의 파라미터와 Attribute를 사용하여 불꽃에 다른 색상을 주는 것입니다. 하지만 이 방식은 항상 잘 작동하지는 않습니다. 다른 오브젝트 파라미터와 Principled Volume을 더 자세히 다룬 후, 나중 에피소드에서 가능한 해결 방법을 살펴보겠습니다.

Principled Volume 노드 상단에는 기본값이 회색인 Color 필드가 있습니다. 이것은 Smoke의 색상입니다. Principled Volume은 하나의 노드에서 Smoke와 불꽃의 외형을 각각의 파라미터로 정의할 수 있게 해줍니다. Density 값도 Smoke를 가리킵니다. Smoke를 보여드리기 위해 World Background Strength를 다시 1로 설정하겠습니다.

Smoke 입자를 통과하는 빛의 산란은 Anisotropy와 Absorption Color로 조정할 수 있습니다. Anisotropy는 산란의 방향에 영향을 주며, 값이 음수인지 양수인지에 따라 뒤쪽으로 혹은 앞쪽으로 더 많이 산란하게 됩니다. 차이를

더 쉽게 확인하기 위해 Smoke Density를 2로 설정하여 밀도를 높이고, 불꽃이 Smoke 뒤에 오도록 화면 구도를 잡았습니다. Absorption Color 필드는 빛이 Smoke에 흡수될 때 색조를 줄 수 있습니다.





노드에서 사용할 수 있는 Attribute에 대해 이야기하려면 먼저 Domain에 저장되는 정보들을 설명해야 합니다. 그래서 이번 에피소드는 여기서 마무리하겠습니다. 지금까지 본 것은 Fire와 Smoke의 렌더링 외형을 정의하는 첫 단계일 뿐입니다. 하지만 이전 에피소드와 이번 내용을 통해 이미 기본적인 불꽃을 만들고 렌더링하는 데 필요한 기초 지식은 갖추었습니다. 몇 에피소드 후에는 Material이 제공하는 다른 도구들, 특히 Emission을 사용하는 방법을 분석하겠지만, 그 전에 불꽃과 Smoke를 더 잘 제어하기 위해 시뮬레이션의 몇 가지 기초를 먼저 다뤄야 합니다.

Blender 4.5 Fire Simulations 기본 | 3/10: Cache, Baking, Offset

<https://youtu.be/nqvVbKxH2oo>

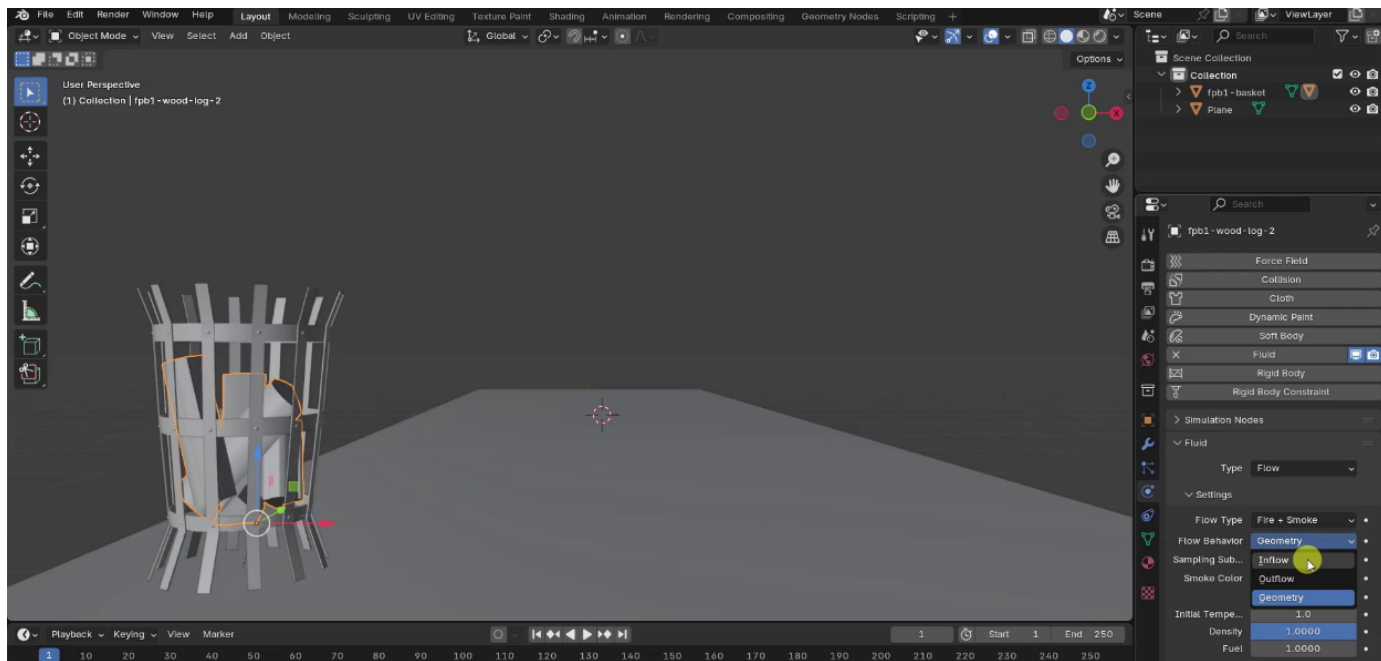
안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 세 번째 튜토리얼에서는, Blender가 Bake를 실행하고 저장하는 방식, 즉 Fire와 Smoke의 물리 시뮬레이션에 대해 이야기하겠습니다.

특히 하나의 Domain에서 시뮬레이션을 한 번만 계산한 후, 그 복사본들을 썬의 다른 위치에 배치해도 모든 복사본에 대해 다시 계산할 필요가 없는 방법을 살펴보겠습니다.

마지막으로 시뮬레이션 재생에 Offset을 설정해, 애니메이션이 불꽃이 처음 점화되는 장면이 아니라 이미 타고 있는 상태에서 시작하도록 만드는 방법을 보겠습니다.

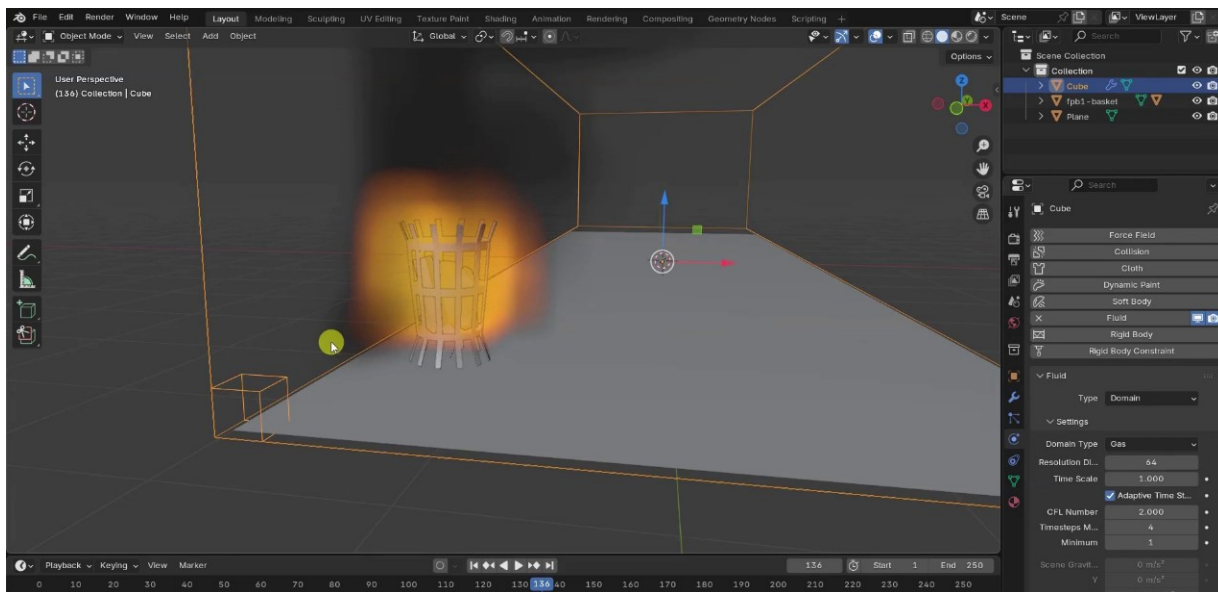
이번 튜토리얼의 시작 썬은 장작이 들어 있는 화로(brazier)로 구성되어 있습니다. 이들은 두 개의 분리된 메쉬이며, 장작은 하나의 메쉬로 이루어져 있습니다. 화로는 직사각형 Plane 위에 놓여 있는데, 이 Plane은 바닥 역할을 하며 나중에 다른 화로들을 여기에 배치할 예정입니다. 가상 World는 Strength 값이 0이고 광원이 없기 때문에 Rendered 모드에서는 썬이 완전히 검게 보입니다.

이전 에피소드에서 배운 것처럼, 오브젝트가 불꽃을 내도록 만들기 위해서는 Fire And Smoke Flow 타입의 Fluid 컴포넌트를 할당해야 합니다. 그래서 장작에 Physics Fluid 컴포넌트를 추가하고, Behavior를 Inflow로 설정해 매 프레임마다 불꽃을 생성하도록 합니다.

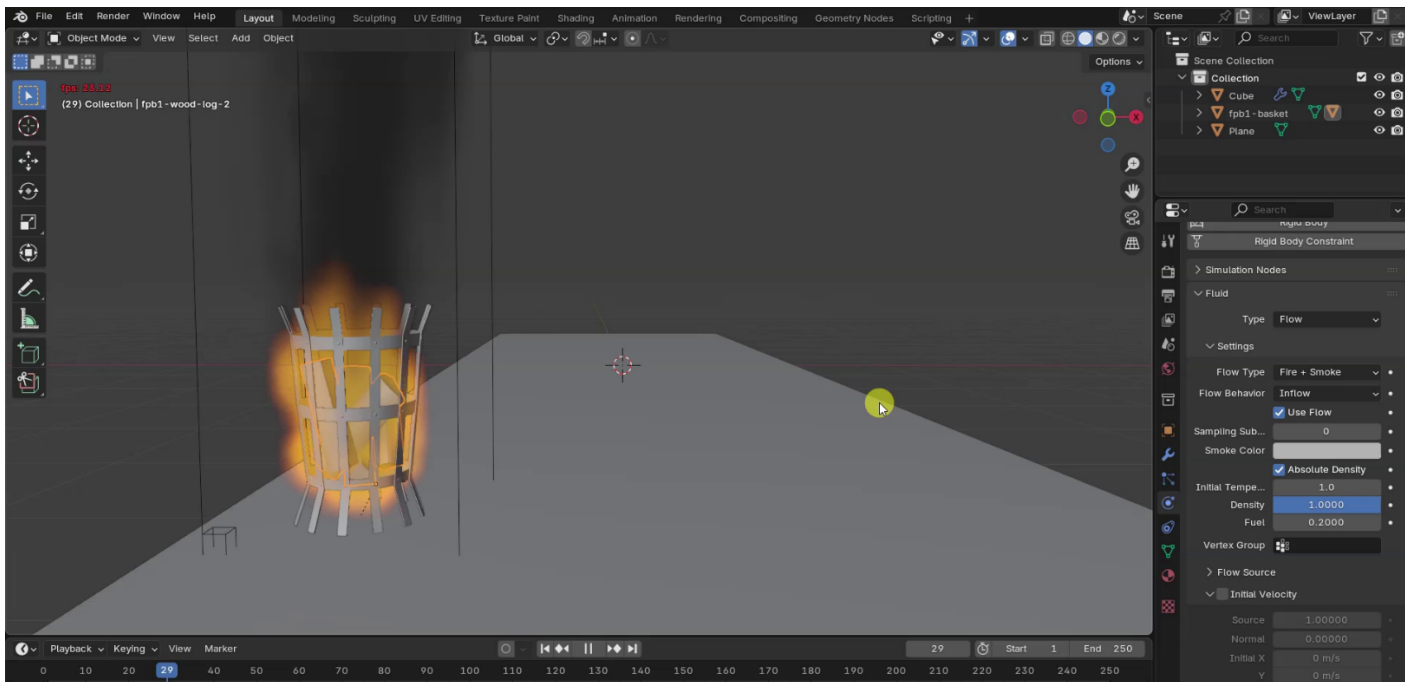


그런 다음 Cube를 생성해 Flow 컴포넌트를 Domain 타입으로 할당하고, 씬 전체를 감싸도록 배치합니다. 곧바로 문제를 발견할 수 있는데, 기본 Resolution 값이 너무 낮아 Voxel이 지나치게 커지고, 불꽃이 방출체 오브젝트에 비해 지나치게 크게 보입니다.

Resolution을 64로 높여도 크게 개선되지 않고, 시뮬레이션 계산 시간만 길어집니다. Resolution을 더 올리는 것은 이 접근법이 잘못된 것임이 분명하므로 시도조차 하지 않겠습니다.



가장 명확한 해결책은 Domain을 화로와 불꽃 및 Smoke가 위로 퍼질 수 있는 약간의 공간만 포함하도록 크기를 조정하는 것입니다. 이제 Resolution이 조금 과할 정도로 잘 작동합니다. 사실 너무 과해서 다시 32로 낮춰도 잘 정의된 불꽃을 얻을 수 있습니다. 오히려 문제는 장작이 불꽃을 지나치게 많이 생성한다는 점입니다.



생성되는 불꽃의 양을 줄이기 위해 Inflow 오브젝트의 Fuel 파라미터를 조정할 수 있습니다. 직관적으로 Fuel 값이 클수록 매 프레임마다 더 많은 연료가 소모됩니다. 따라서 불꽃이 적절한 크기에 도달할 때까지 이 값을 낮추는 것이 해결책입니다.

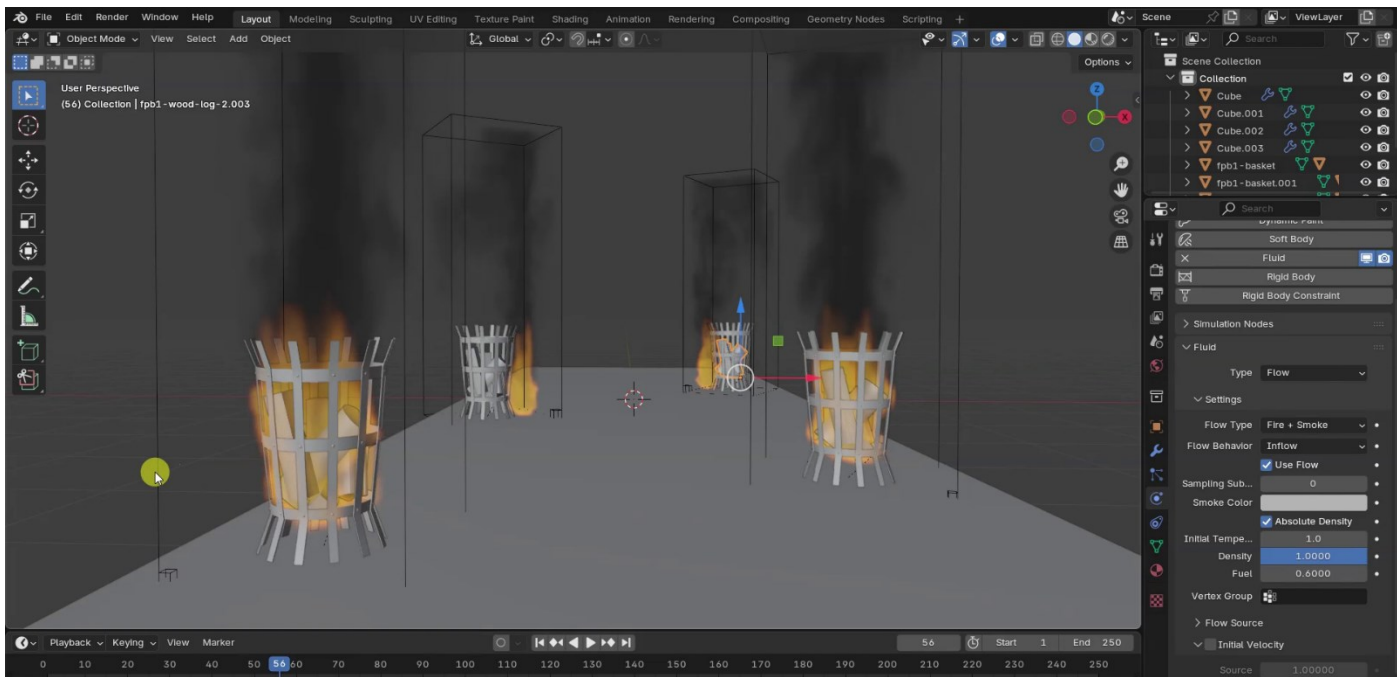
앞서 언급했듯이, Baking 즉 물리 시뮬레이션 계산에 관련된 데이터는 Domain 오브젝트의 Cache 섹션에 지정된 경로에 있는 전용 폴더에 저장됩니다. Domain을 생성하기 전에 파일을 디스크에 저장했기 때문에, Cache 폴더는 Blender 프로젝트 파일과 동일한 디렉토리에 생성됩니다.

Domain의 Cache 섹션에서는 Domain 시뮬레이션을 계산할 프레임 범위를 설정할 수 있습니다. 이는 씬의 애니메이션 프레임 범위와 일치할 필요는 없습니다. 이 기능은 전체 Bake를 실행하기 전에 불꽃의 모양을 대략 확인하기 위해 제한된 프레임만 Baking하는 데 유용합니다. 또 다른 활용 예는 Fluid 오브젝트가 카메라 뷰에 없는 구간으로, 이 경우 해당 프레임에 대해 물리 시뮬레이션을 계산할 필요가 없습니다.

Cache 섹션의 다른 파라미터를 살펴보기 전에, Domain과 화로, Inflow 역할의 장작으로 구성된 세트를 복사해 씬에 여러 개 배치하면 어떻게 되는지 보겠습니다. 앞서 본 것처럼 씬 전체를 덮는 단일 Domain을 사용하려면 매우 높은 Resolution이 필요하고, 많은 공간이 낭비됩니다. 따라서 여러 화로를 추가하려면 전체 세트를 복제하는 수밖에 없습니다.

문제는 Timeline에서 Play를 누르면 Blender가 모든 Domain에 대해 동일한 Cache 폴더를 사용한다는 점입니다. 즉, 모든 Domain이 동일한 시뮬레이션 데이터를 공유하게 됩니다. 모든 시뮬레이션에 동일한 파라미터를 유지하려는 경우라면, Domain을 약간 회전시키거나 각각 다른 Material을 부여하여 렌더링 시 불꽃의 외형을 다양하게 만드는 식으로 관찰자를 속이는 방법을 쓸 수도 있습니다.

진짜 문제는 Domain의 Resolution이나 다른 파라미터를 변경한 뒤 다시 Bake를 실행할 때 발생합니다. 예를 들어, 배경 오브젝트에는 높은 Resolution이 필요하지 않으므로 더 멀리 있는 Domain들의 Resolution을 낮췄다고 합시다. 이제 Blender는 데이터가 완전히 뒤섞이게 되는데, 이는 서로 다른 특성을 가진 오브젝트들이 동일한 볼륨 데이터 세트를 공유하고 있기 때문입니다.



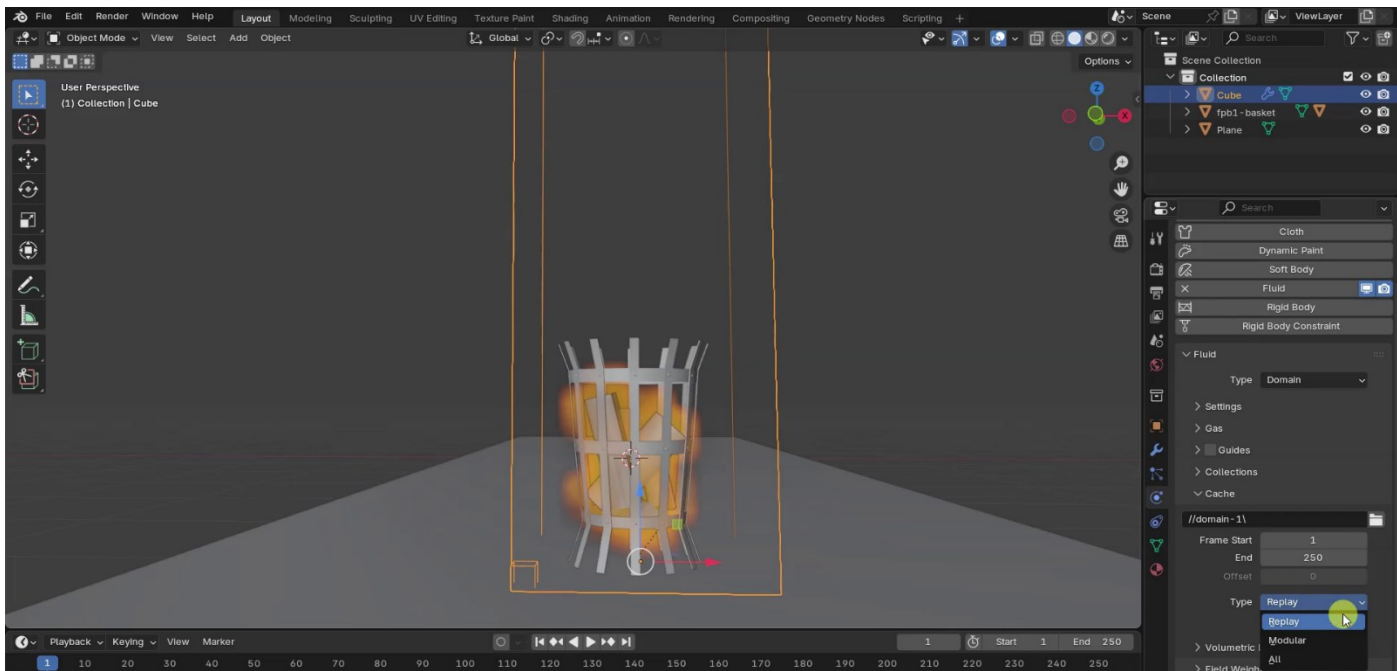
각 Domain이 다른 Domain과 명확히 분리된 데이터를 가지도록 하려면, 시뮬레이션을 실행하기 전에 각 Domain마다 별도의 Cache 폴더를 지정해야 합니다. 물론 이렇게 하면 다양한 Domain 데이터 때문에 디스크 사용량이 크게 늘어납니다. 따라서 필요에 따라 전략을 선택해야 합니다. 하나 또는 몇 개의 아주 큰 Domain을

사용해 같은 시뮬레이션을 복제할지, 아니면 Domain마다 별도의 Cache를 생성할지를 결정하는 것은 여러분에게 달려 있습니다.

지금까지 Blender는 우리가 Timeline에서 Play를 누를 때마다 시뮬레이션을 Bake해 왔습니다. 하지만 이것만이 가능한 모드는 아닙니다.

사실, 또 다른 모드가 있으며, 이를 통해 매우 유용한 도구를 사용할 수 있습니다. 간단히 하기 위해, 씬에는 화로와 그 Domain만 남기고 다른 것들은 삭제하겠습니다. 주의할 점은, Domain을 삭제해도 디스크의 데이터는 삭제되지 않는다는 것입니다. 사실 이것은 좋은 점이기도 한데, 나중에 시뮬레이션 데이터를 저장하고 재사용할 수 있기 때문입니다. 하지만 이번 경우에는 그 데이터가 필요 없으므로 직접 수동으로 삭제하겠습니다.

Domain의 Cache 섹션에는 Type 메뉴가 있으며, 기본값은 Replay입니다. 지금까지 사용해 온 모드로, Blender는 프레임 1로 돌아가 Play를 누를 때마다 새로운 Bake를 다시 계산하고 이전 데이터를 덮어씹습니다.

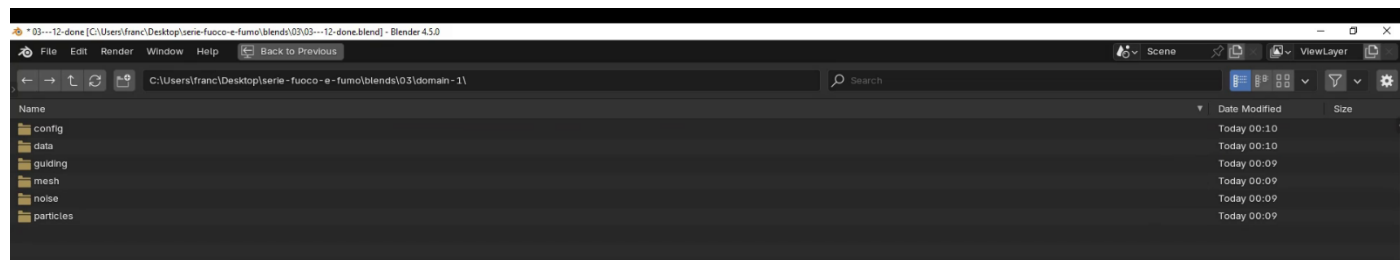


Modular 모드에서는 시뮬레이션의 다양한 측면을 분리하여 개별적으로 Bake할 수 있습니다. 이 방식은 전체 시뮬레이션을 다시 하지 않고 특정 부분만 따로 확인하거나 교체할 수 있어서 유용한 경우가 있습니다.

마지막으로 All 모드는 전체 시뮬레이션을 Bake하지만 Timeline이 아니라 백그라운드에서 계산합니다.

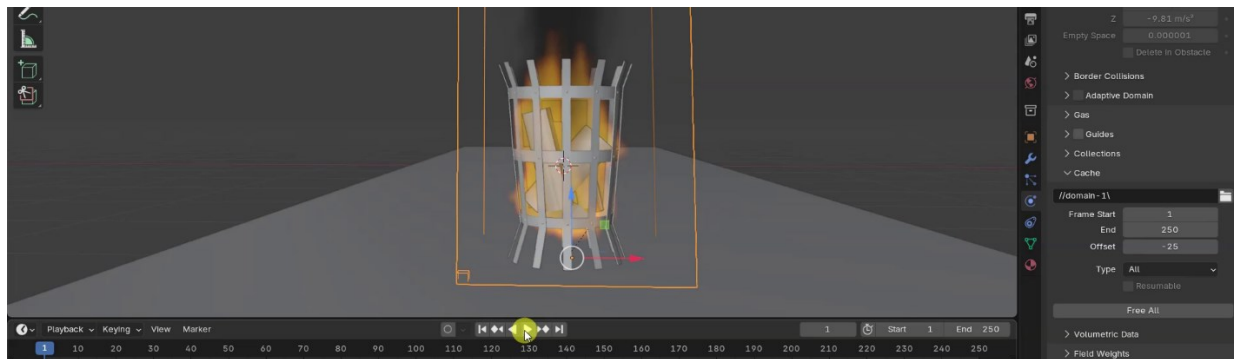
Bake를 시작하려면 패널에서 Bake All 버튼을 클릭해야 합니다. Blender는 진행 상태를 보여주는 진행 바를 표시합니다. 계산이 완료되면, Timeline에서 Play를 눌러 훨씬 부드럽게 결과를 확인할 수 있습니다.

Blender가 이미 시뮬레이션을 미리 계산했기 때문입니다. 시뮬레이션 설정을 변경하려면 먼저 Free All 버튼을 클릭해 기존 데이터를 지워야 합니다. 그러면 파라미터를 다시 수정할 수 있고, 변경 후에는 다시 Bake All을 클릭해 Blender가 시뮬레이션을 새로 계산하도록 해야 합니다. 사용 중인 폴더를 살펴보면 Modular와 All 모드에서 계산되고 따로 저장되는 다양한 모듈 또는 Bake 단계가 있다는 것을 확인할 수 있습니다.



All 모드에서는 Offset 파라미터가 활성화되는데, 이것은 이미 눈치챘을지도 모를 문제를 해결하는 데 도움이 됩니다. 지금까지 모든 Fire 시뮬레이션은 애니메이션 첫 프레임에서 불꽃이 점화되는 장면으로 시작했습니다. 하지만 때로는 불이 이미 타고 있는 상태에서 애니메이션을 시작하고 싶을 때가 있습니다.

이 문제를 해결하려면 Offset 필드에 음수를 입력하면 됩니다. Tooltip의 설명에 따르면, 이것은 Baking이 더 일찍 시작된다는 뜻이 아니라, 애니메이션 첫 프레임에서 시뮬레이션이 이미 지정된 프레임 수만큼 진행된 것처럼 재생된다는 의미입니다. 애니메이션 프레임 길이는 여전히 Frame Start와 Frame End 사이에 지정된 값으로 유지됩니다. 이 경우 Timeline의 마지막 25프레임에는 시뮬레이션이 없는데, Offset을 통해 Timeline의 첫 프레임에서 시뮬레이션의 25번째 프레임을 재생하도록 Blender에 지시했기 때문입니다.



이를 해결하려면 Bake에서 더 많은 프레임을 계산해 시뮬레이션이 Timeline 전체를 덮도록 하면 됩니다. 따라서 Domain의 Cache 섹션에서 Frame End를 275로 설정하고, Free All을 클릭한 다음 Bake All을 실행해 다시 Bake를 진행합니다.

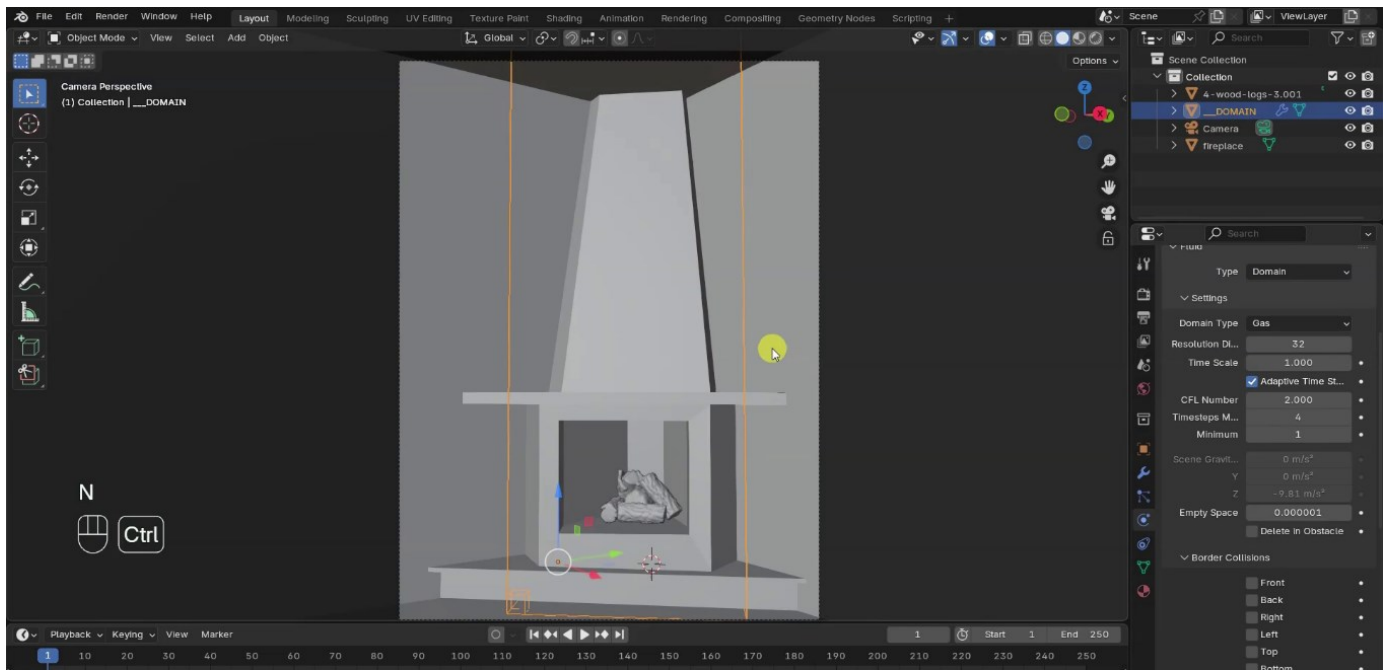
Blender 4.5 Fire Simulations 기본 | 4/10: Geometry, Inflow, Outflow

<https://youtu.be/xSN2Ez-dgl0>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 네 번째 튜토리얼에서는, Fire와 Smoke의 물리 시뮬레이션에 사용되는 세 가지 Flow 오브젝트 유형의 파라미터를 자세히 살펴보겠습니다.

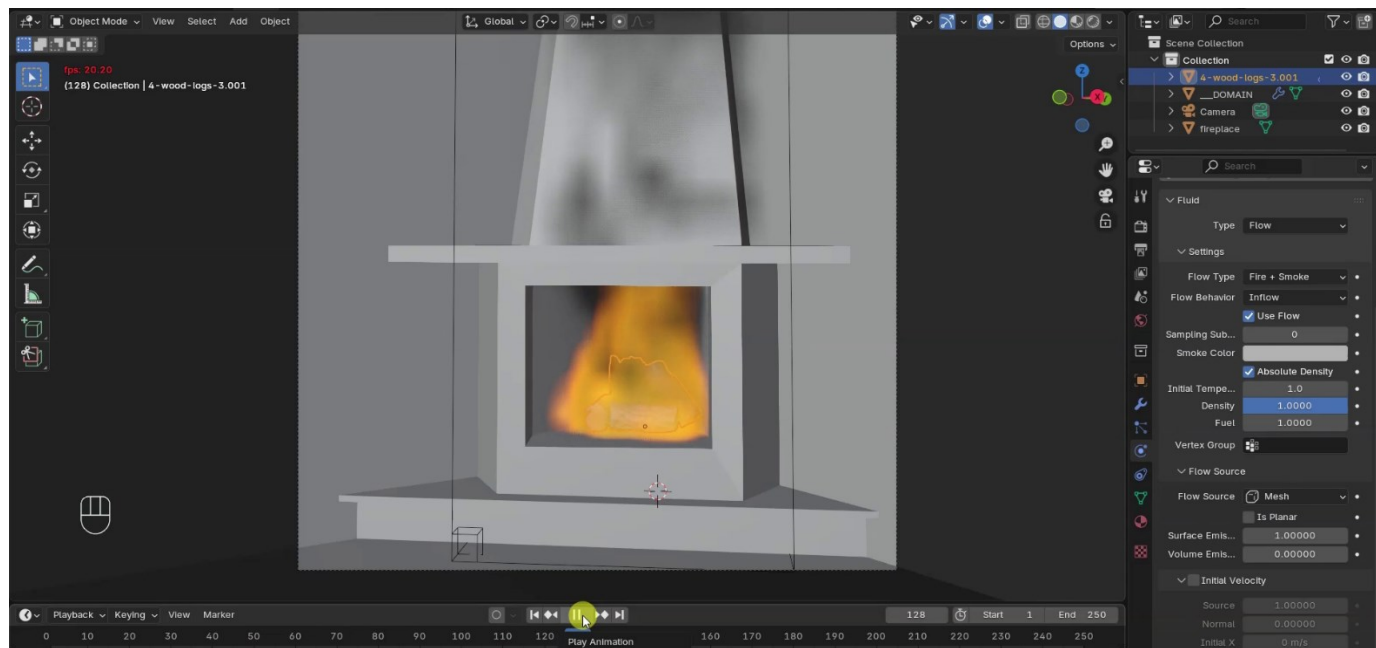
제가 사용하는 씬에는 벽난로 안에 장작이 배치되어 있습니다. 이 장작은 하나의 메쉬 오브젝트로 구성되어 있으며, 이를 Inflow로 설정하겠습니다. 시뮬레이션용 Domain은 이미 생성해 두었으며, 벽난로의 볼륨만 포함하도록 설정했습니다.

장작에 Fluid 컴포넌트를 Inflow 타입으로 할당합니다. Fire And Smoke 모드는 Smoke와 Fire 파라미터를 모두 포함하고 있으므로, 모든 파라미터를 분석하기 위해 이 조합을 선택합니다. 이후에는 Fire 전용 시뮬레이션에서는 사용할 수 없는 Smoke 전용 파라미터를 따로 짚어드리겠습니다. 보시는 것처럼 파라미터 수가 그렇게 많지는 않습니다.



기본적으로 Flow Behavior는 Geometry로 설정되어 있습니다. 앞서 본 것처럼 이 모드는 처음에만 불꽃과 연기를 내고, 그 이후에는 더 이상 Fire나 Smoke를 방출하지 않습니다. 폭발이나 단발적인 연기에는 유용하지만, 이번 경우에는 Inflow 모드가 필요합니다. 어쨌든 Use Flow 체크박스를 제외하면 Geometry와 Inflow는 같은 파라미터를 공유하므로, 우리는 Inflow를 기준으로 모든 파라미터를 살펴보겠습니다.

이전 에피소드에서 배운 것처럼 불꽃의 양은 Fuel 파라미터로 조절할 수 있습니다. 이번 경우에는 값을 약간 낮춰 활발하지만 과하지 않은 불꽃을 만들거나, 더 크게 낮춰 잔불을 표현할 수 있습니다.



Domain의 크기를 조금 줄이고 Resolution을 높였습니다. Inflow 파라미터를 보기 전에, Outflow 오브젝트를 사용해 벽난로 안의 연기를 사라지게 하는 방법을 살펴보겠습니다.

Outflow 오브젝트는 Fire나 Smoke와 같은 Fluid 시뮬레이션 요소를 충돌 시 사라지게 하는 메쉬입니다. 단순한 Plane을 추가해 Outflow로 사용하고, 연기가 굴뚝으로 빠져나가는 위치에 배치했습니다. Outflow 오브젝트는 Inflow 오브젝트처럼 렌더링에서 반드시 보일 필요는 없습니다. 따라서 3D Viewport 미리보기와 최종 렌더링에서 가시성을 꺼두겠습니다. 이제부터는 Outliner에서만 선택하겠습니다.

오브젝트에 Fluid 컴포넌트를 추가하고, Type을 Smoke Flow, Behavior를 Outflow로 설정합니다.

이 유형의 오브젝트는 파라미터가 매우 적으며, 사실 Geometry와 Inflow 오브젝트에도 존재하는 것들입니다.

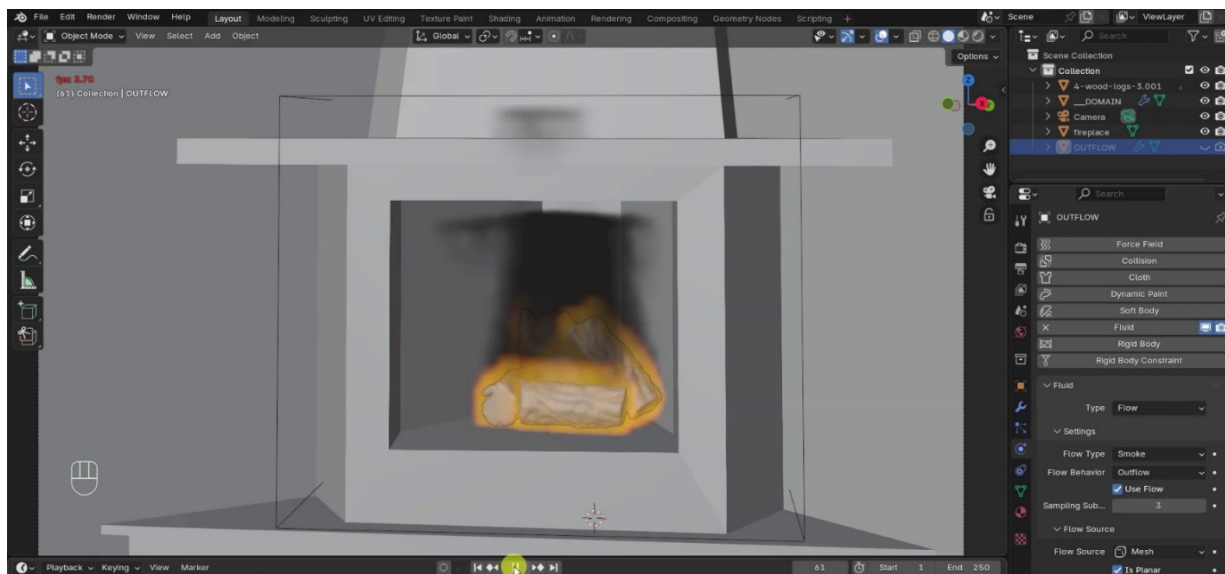
간단히 설명드리겠습니다. Is Planar 옵션은 오브젝트에 볼륨이 없거나 일반적으로 닫혀 있지 않을 때 선택해야 합니다. Plane은 정확히 이런 경우이므로 이 옵션을 활성화합니다. Use Flow 파라미터는 Inflow나 Outflow를 활성화하거나 비활성화할 수 있으며, 애니메이션이 가능해 간헐적이거나 비연속적인 Fire와 Smoke의 방출 또는 제거를 만들 때 유용합니다.

Sampling Substeps 파라미터는 프레임 사이에 계산되는 추가 시뮬레이션 단계 수를 정의하며, 시뮬레이션의 품질을 향상시킵니다.

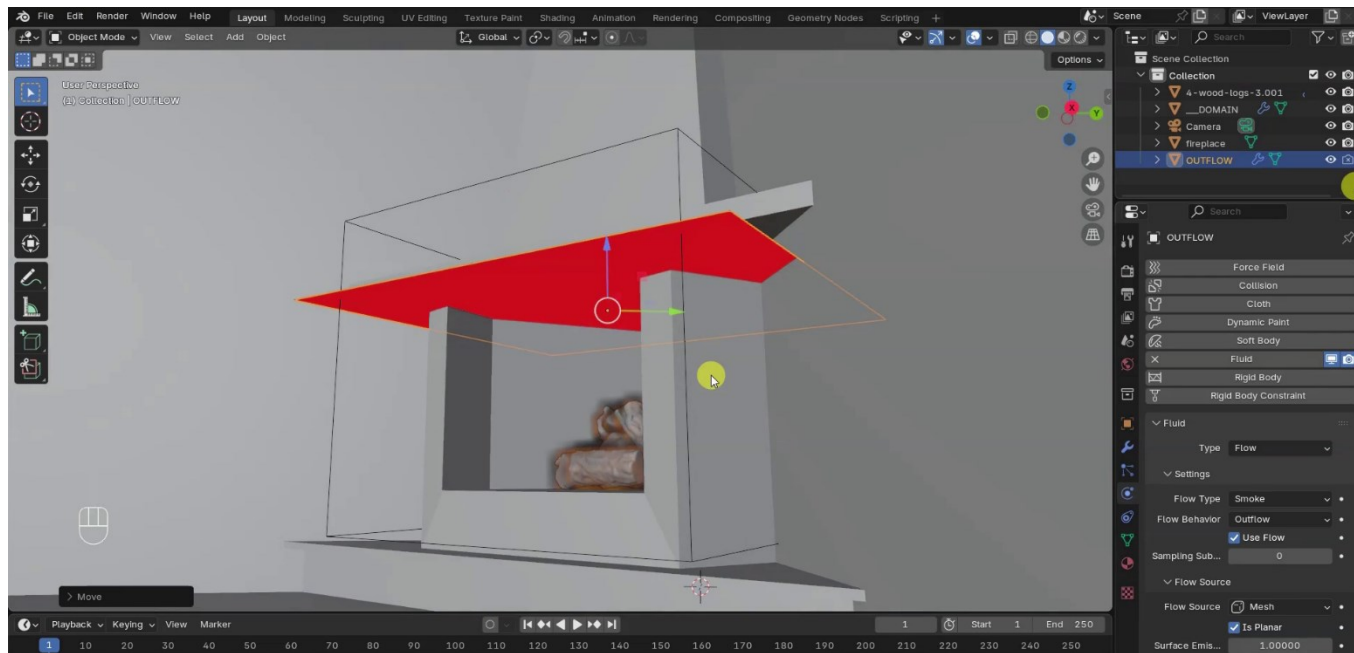
빠른 시뮬레이션에서 특히 유용하며, 3 정도의 낮은 값만으로도 눈에 띄는 차이를 만듭니다.

Surface Emission과 Volume Emission은 Fire와 Smoke의 방출 또는 흡수 지점을 오브젝트의 표면이나 볼륨으로부터 오프셋할 수 있지만, 이번 경우에는 필요하지 않습니다. Flow Source는 나중에 다루겠습니다.

시뮬레이션을 실행해 보면 처음에는 괜찮아 보이지만, 어느 순간 약간의 연기가 위로 빠져나가는 것을 확인할 수 있습니다. Substeps 수를 늘려도 상황은 변하지 않습니다.



Plane을 다시 3D View에서 보이게 하고 시점을 바꿔보면, 해결 방법이 분명해집니다. Plane이 너무 작아서 어느 순간 일부 Smoke가 바깥으로 빠져나가 위로 올라가는 것입니다. 유일한 해결책은 Plane의 크기를 적절히 조정하고 다시 시도하는 것입니다.



이제 Outflow에서 이미 다른 항목들을 제외하고, Inflow 오브젝트의 파라미터를 자세히 살펴보겠습니다! Smoke Color는 사실 렌더링 시 Smoke의 색상이 아니라 3D Viewport 미리보기에서의 색상을 의미합니다. 여기서 다른 색상을 지정하는 것은 복잡한 씬에서 Smoke를 더 쉽게 식별하거나, 이 오브젝트의 Attribute를 Shading에서 사용해 Domain Material의 색상이나 다른 요소를 변경하는 데 유용할 수 있습니다.

Absolute Density 필드는 기본적으로 활성화되어 있으며, 일반적으로 그대로 두는 것이 좋습니다. 이렇게 하면 Domain 안에서 Fire와 Smoke의 최대 밀도가 일정하게 유지되기 때문입니다. 비활성화하면 Fire와 Smoke가 급격히 증가하여 통제 불능의 불길 같은 결과가 나올 수 있습니다. 여기서는 Outflow의 존재로 인해 효과가 제한되므로, 이를 Domain에서 일시적으로 제거해 불꽃이 증가하는 모습을 보여드리겠습니다. Absolute Density를 끄는 것이 유용한 경우도 있지만 대부분은 그렇지 않으므로 다시 켜두겠습니다.

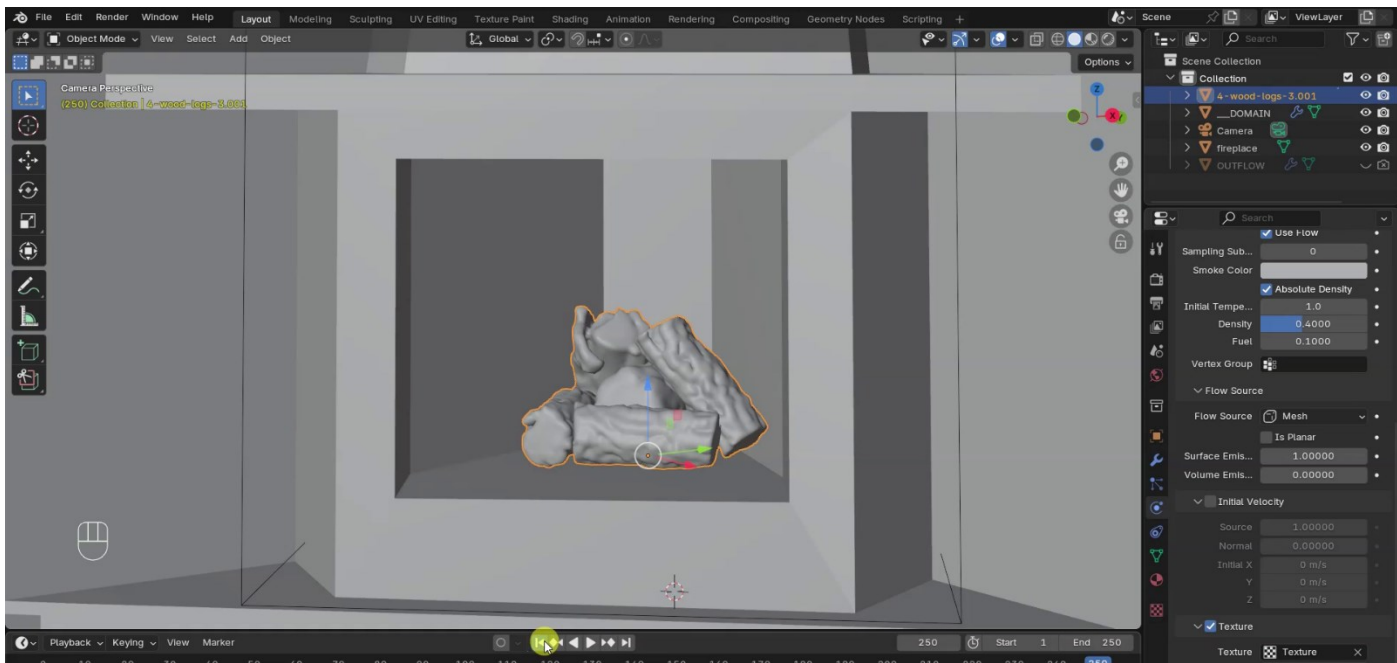
Initial Temperature와 Density 파라미터는 Smoke에 관한 것이므로 Fire 전용 시뮬레이션에서는 사용할 수 없습니다. Density는 방출할 Smoke의 양을 의미하며, 값이 높을수록 부피가 커집니다. 실제로 이번 씬에서는 Smoke가 너무 많이 방출되고 있어서 이 값을 낮추겠습니다.

Initial Temperature 파라미터는 Smoke의 온도와 주변 환경의 온도 차이를 정의하며, 기체가 상승하는 속도를 제어합니다. 그러나 그 효과는 Domain 파라미터 중 하나인 Buoyancy Heat와 연결되어 있으므로, 이 부분은 다른 에피소드에서 더 자세히 살펴보겠습니다.

지금까지는 Inflow 오브젝트가 표면 전체에서 불꽃을 방출하도록 했지만, Vertex Group에서 특정 버텍스를 선택하고 Flow 컴포넌트의 해당 필드에 이 Vertex Group의 이름을 지정하면 방출 영역을 제한할 수 있습니다.

불꽃 방출에 변화를 줄 수 있는 또 다른 도구는 Inflow 패널 하단에 있는 Texture입니다. 여기에서 Texture를 할당할 수 있으며, 특히 중요한 것은 Offset 필드입니다. 이 필드는 Procedural Texture의 어떤 “슬라이스”를 사용할지를 정의합니다. Clouds나 Noise와 같은 Procedural Texture는 2D 이미지가 아니라 실제 3D 볼륨입니다.

Timeline에서 Offset 값을 애니메이션하면, 화면에서 보듯 Texture가 수직으로 이동하여 매번 다른 패턴을 사용하게 되고, 이로 인해 불꽃이 덜 균일해집니다. 이 방식의 문제는 적절한 Texture 유형과 Offset 변화의 조합을 찾기까지 여러 번 시도해야 한다는 점입니다.



Initial Velocity 그룹의 도구들은 불꽃에 초기 방향과 속도를 줄 수 있습니다. 필드 이름이 직관적이어서 더 설명할 필요는 거의 없지만, 다른 씬을 사용해 실제 예제를 나중에 보여드리겠습니다.

마지막으로 Flow Source 필드가 있는데, 여기서는 Mesh와 Particles 중 선택할 수 있습니다. 의미는 단순합니다.

지금까지는 Mesh를 사용해 오브젝트에서 Fire와 Smoke를 방출했으며, Vertex Group으로 방출 표면을 제한하기도 했습니다. 하지만 오브젝트에는 파티클 시스템을 추가할 수 있고, 때로는 방출체가 아니라 그 파티클이 Fire와 Smoke를 생성하도록 하고 싶을 수 있습니다. 이 경우 Flow Source를 Particles로 선택하고, 시뮬레이션을 구동할 파티클 시스템을 지정해야 합니다. 예상할 수 있듯이, 선택한 오브젝트의 지오메트리나 특정 파티클 시스템에 Flow 컴포넌트를 할당할 수 있습니다.

Blender 4.5 Fire Simulations 기본 | 5/10: Domain Settings 1/2 (Gas, Fire, Collections)

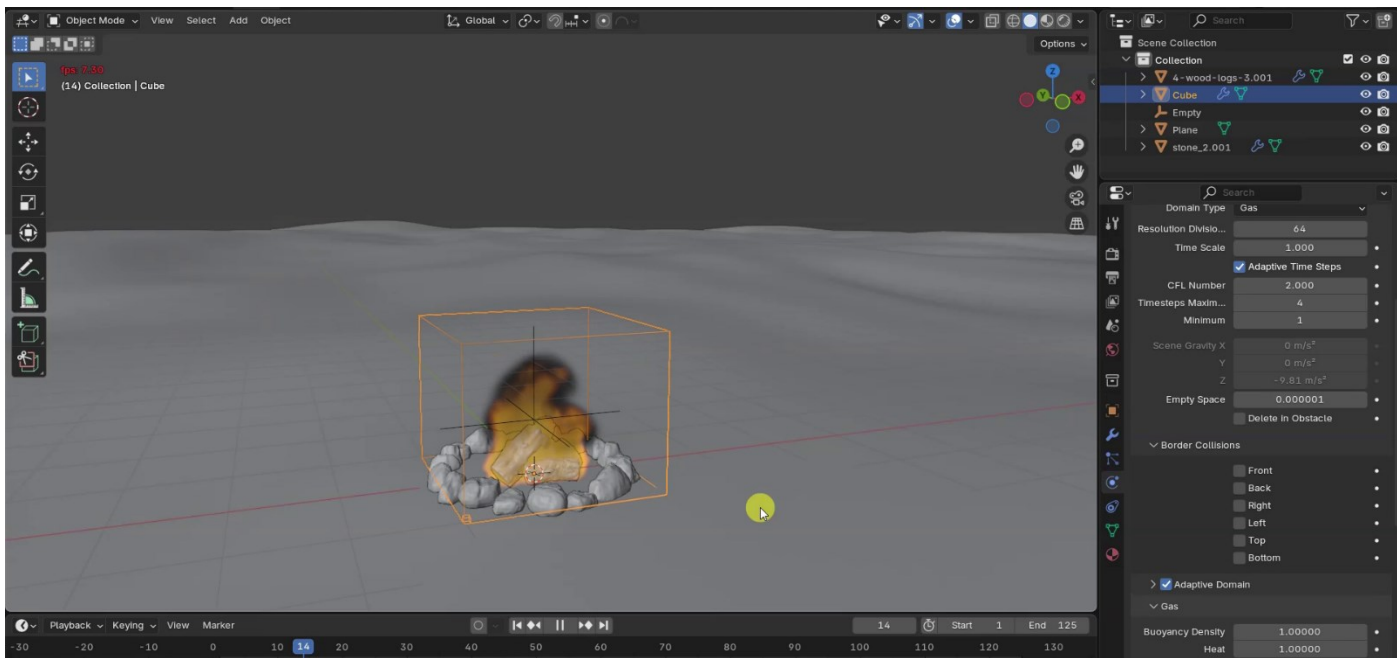
<https://youtu.be/Shp52CZ4bW0>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 다섯 번째 튜토리얼에서는 Domain 파라미터 중 Settings, Gas, Fire, Collections 섹션의 주요 항목들을 살펴보겠습니다.

이 파라미터들 중 일부는 불꽃과 Smoke를 더 생동감 있고 덜 균일하게 보이도록 만들어 줍니다.

Settings 섹션의 주요 요소부터 시작하겠습니다. 지금까지 우리는 Resolution Divisions 파라미터에 익숙해졌으며, Domain 하단에 보이는 작은 Cube가 나타내는 Voxel의 크기가 매우 작고 Domain 자체는 큰 경우, 시뮬레이션 실행에 더 많은 계산과 메모리가 필요하다는 것을 알고 있습니다.

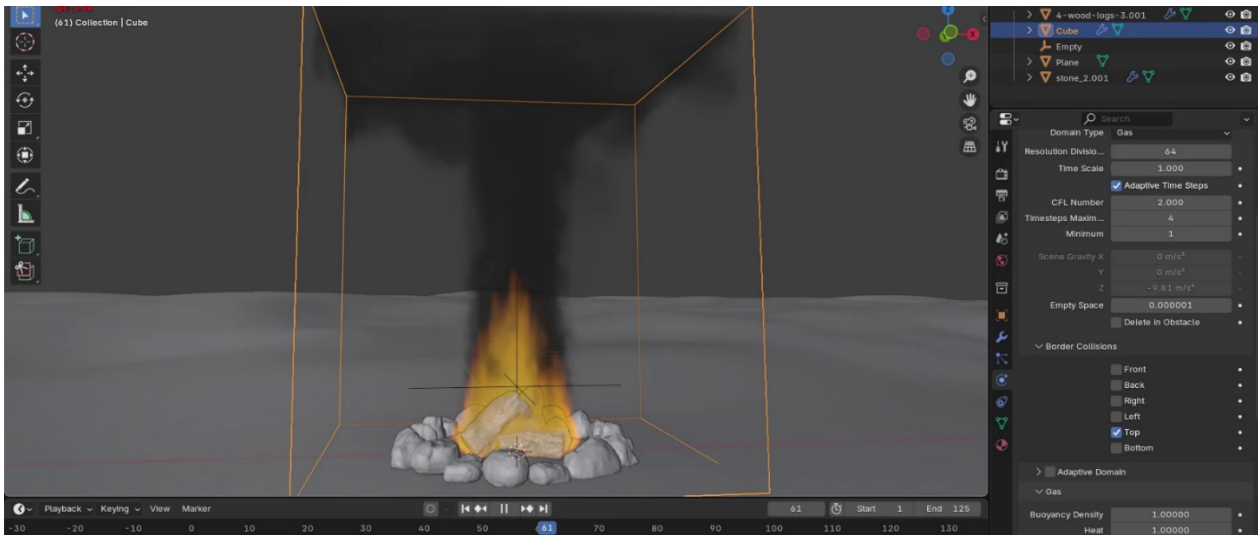
Settings 섹션 하단에 있는 Adaptive Domain 옵션을 활성화하면, Blender가 시뮬레이션 중에 Domain 크기를 동적으로 변경할 수 있습니다.



하지만 Adaptive를 활성화해도 Domain은 일정 높이까지만 확장됩니다. 그 이후에는 Smoke가 Domain의 천장에 닿으면 Outflow처럼 취급되어 사라지게 됩니다. Smoke를 더 빨리 소멸시키는 옵션이 Gas 섹션에 따로 있는데, 곧 살펴보겠습니다. 우선 Adaptive Domain을 비활성화하고 Border Collisions 섹션을 보겠습니다.

Border Collisions 파라미터를 활성화하면 Bounding Box의 면들이 Outflow가 아닌 장애물처럼 동작하여 불꽃과 Smoke를 튕겨내게 됩니다. 비활성화 상태가 기본값인데, 이때는 불꽃과 Smoke가 경계에 닿는 즉시 사라집니다. Domain의 천장에 Border Collision을 활성화하면 Smoke가 위쪽에 쌓이고 옆으로 퍼지다가, 측면에 닿으면 Border Collision이 비활성화된 상태이므로 사라지게 됩니다.

다른 면들에도 Border Collision을 활성화하면 Domain은 닫힌 상자처럼 동작하여, Fire와 Smoke가 벽에 닿아도 사라지지 않고 계속 축적됩니다.



Settings 패널 상단으로 돌아가면 Time Scale과 Adaptive Time Steps 항목을 볼 수 있습니다. Time Scale은 단순히 시뮬레이션 속도를 늦추거나 빠르게 조절할 수 있습니다.

Adaptive Time Steps 체크박스는 기본적으로 활성화되어 있으며, 알고리즘이 애니메이션의 각 프레임에 대해 더 많은 중간 단계(Steps)를 계산할 시점을 결정하도록 합니다. 이 체크박스 아래에 있는 세 가지 파라미터는 알고리즘이 매번 몇 개의 Steps를 수행할지 결정하는 방식과 직접적으로 관련이 있습니다. Maximum과 Minimum은 이름 그대로 직관적이고, CFL Number는 조금 더 복잡합니다. 구현 세부 사항까지 알 필요는 없지만, 중요한 점은 값이 높을수록 Steps 수가 줄어들어 계산 시간이 단축되지만 정확도는 떨어진다는 것입니다. 불꽃과 Smoke가 특히 빠르게 움직이는 시뮬레이션에서는, 더 낮은 CFL Number를 사용해 품질을 향상시키는 것이 좋습니다.

Gravity 파라미터는 기본적으로 비활성화되어 있는데, 이는 Scene 탭의 전역 설정이 사용되기 때문입니다. Scene 탭에서 Gravity를 전역적으로 비활성화하면, Physics에서 이 파라미터들이 다시 활성화되어 수정할 수 있습니다.

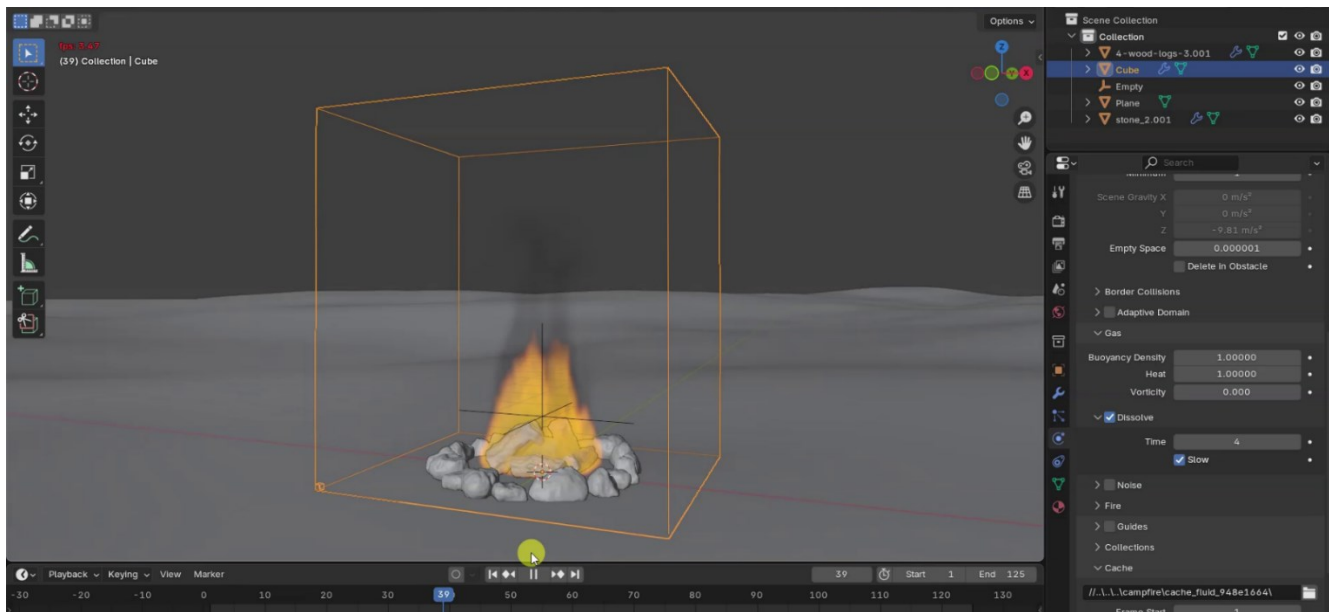
Empty Space는 Smoke 요소에만 작동합니다. 구체적으로, 이 임계값 아래로 채워진 Voxel은 빈 것으로 간주되어 시뮬레이션에서 무시되며, 그만큼 계산 시간을 절약할 수 있습니다.

이제 Gas 섹션으로 넘어가면, Buoyancy Density와 Heat 파라미터를 볼 수 있습니다. 이 값들은 기체에 작용하는 상승력으로, 첫 번째는 기체 밀도에, 두 번째는 기체 온도에 기반합니다. 이 두 값은 온도와 밀도를 직접 나타내는 것이 아니라, 이로 인해 발생하는 상승력을 나타냅니다. 다행히 이 파라미터들의 Tooltip이 매우 명확하여, 값을 높이면 기체가 더 빨리 상승한다는 것을 쉽게 이해할 수 있습니다. Buoyancy Heat는 특히 Smoke의 열로 인한 상승력입니다. Smoke의 초기 온도와 주변 환경의 온도 차이는 각 Inflow 오브젝트에서 Initial Temperature 파라미터로 정의됩니다. Inflow 오브젝트 패널에는 Density 파라미터도 있는데, 이는 Domain의 Buoyancy Density와 연결되어 있습니다.

Vorticity는 Smoke의 난류를 제어하며, 이는 불꽃에도 간접적으로 영향을 줍니다. Vorticity 값이 높을수록 Smoke는 더 요동치게 됩니다. 주의해야 할 점은, 이 파라미터의 낮은 값만으로도 매우 강한 난류가 발생할 수 있다는 것입니다! 화면에서는 Buoyancy Density 값을 조정해 Smoke가 더 빨리 상승하는 것도 함께 보여드리고 있습니다.

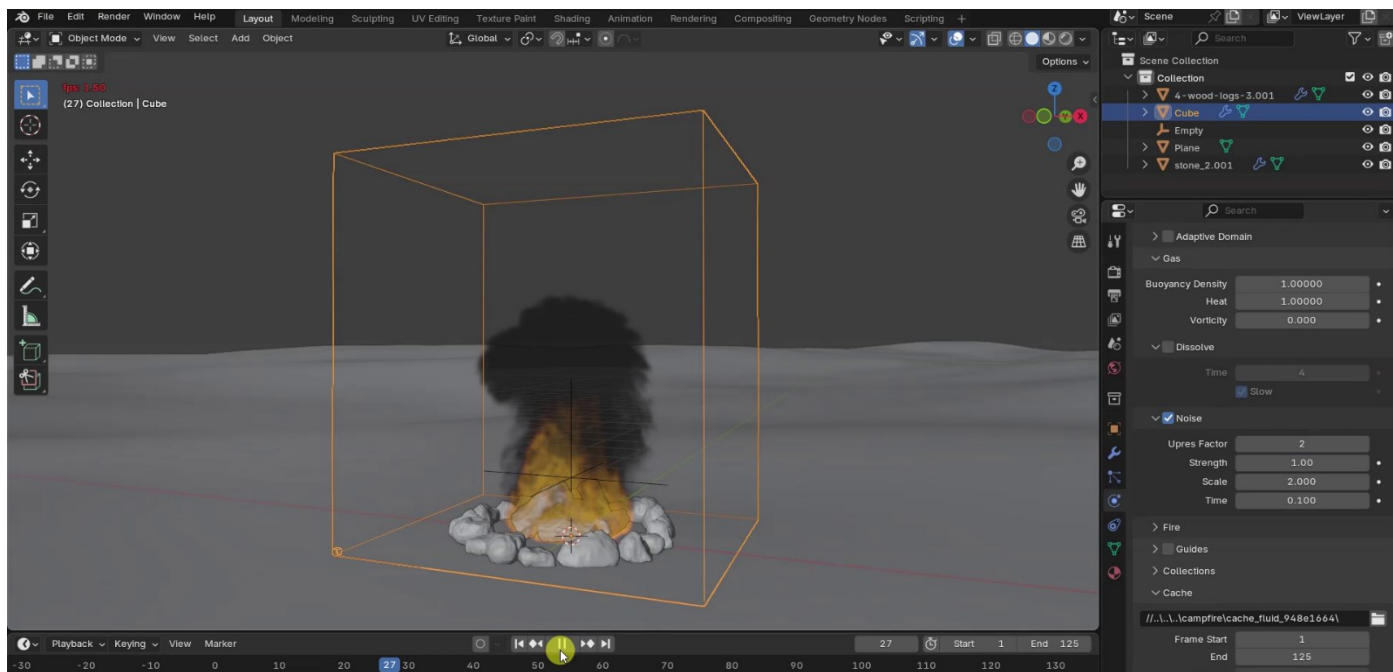


기체가 천장에 쌓이는 것을 방지하고, 일반적으로 더 빨리 사라지도록 하려면 Dissolve를 활성화하고 Dissolve Time 파라미터를 조정하면 됩니다. 이 값은 프레임 단위로 측정됩니다. 당연히 Dissolve Time 값이 낮을수록 Smoke는 더 적은 프레임 동안만 보이므로 더 빨리 사라집니다. Dissolve Slow 옵션을 사용하면 기체가 로그 곡선처럼, 즉 처음에는 빨리 그리고 시간이 지남에 따라 점점 천천히 사라집니다.



Gas 섹션에는 Noise와 Fire도 포함되어 있어 불꽃에 교란을 추가할 수 있습니다. 특히 Noise는 불꽃의 외곽에 많은 디테일을 추가하여, 모닥불이나 벽난로 불처럼 더 요동치고 덜 균일한 불꽃을 만드는 데 유용합니다. Strength와 Scale 파라미터는 교란의 강도를 직관적으로 제어합니다. Upres Factor 파라미터는 Domain의 기본 Resolution에 추가로 적용되는 세분화 계수를 의미하며, Noise 디테일을 생성하는 데 사용됩니다. 그렇기 때문에 시뮬레이션 속도가 느려지는 것인데, 이는 Noise가 적용된 영역이 더 높은 Resolution으로 계산되기 때문입니다.

Fire 섹션의 파라미터를 보면, 여기에도 Gas 섹션과 마찬가지로 Vorticity가 있다는 것을 바로 알 수 있습니다. Smoke와 마찬가지로 Vorticity는 난류를 도입하는데, 이번에는 불꽃에 적용됩니다. 불꽃에도 온도가 있으며, 값이 높을수록 더 빠르게 상승합니다. 두 파라미터를 함께 사용하면 더 크고 생동감 있는 불꽃을 만들 수 있습니다.



Collections 섹션에는 Flow와 Effector라는 두 개의 Collection을 지정할 수 있는 필드가 있습니다.

이를 통해 지정된 Collection에 속한 Flow 오브젝트와 Effector 오브젝트만 시뮬레이션에 포함되도록 할 수 있습니다. Effector는 Fire와 Smoke의 경로를 바꿀 수 있는 특수한 요소입니다.

Collider와 Guide라는 두 가지 유형의 Effector에 대해서는 나중 에피소드에서 다룰 것입니다. 기본적으로는 어떤 Collection도 지정되지 않으므로, 모든 Flow 오브젝트와 Effector 오브젝트가 시뮬레이션에 포함됩니다.

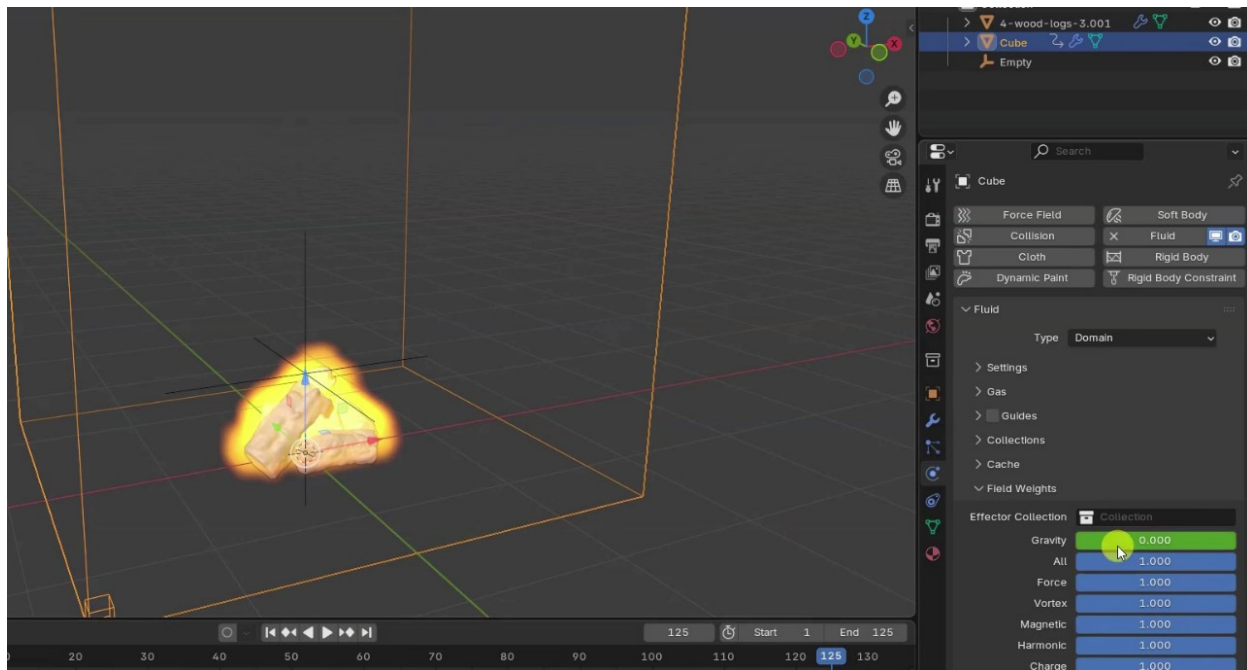
Blender 4.5 Fire Simulations 기본 | 6/10: Domain Settings 2/2 (Weights, Data, Render, Display)

<https://youtu.be/YQpdnXawx3o>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 여섯 번째 튜토리얼에서는 Domain 오브젝트 설정에 대한 개요를 마무리하겠습니다. 구체적으로 Field Weights, Volumetric Data, Viewport Display, Render 섹션의 주요 파라미터를 살펴보겠습니다. 이 중 일부 파라미터는 Attributes라고 불리는 특정 Material 속성을 더 잘 이해하는 데 도움이 되며, 이는 다음 에피소드에서 다룰 예정입니다.

Field Weights에 관해서는, 지금은 시뮬레이션이 외부 힘의 영향을 받을 수 있으며 여기에서 이러한 힘이 시뮬레이션 요소에 얼마나 영향을 미칠지를 정의할 수 있다는 정도만 말해두겠습니다.

이러한 외부 힘 중 하나는 중력(Gravity)으로, 기본적으로 씰 레벨에서 활성화되어 있으며 Fire와 Smoke가 위로 상승하게 만드는 요인입니다. Field Weights 섹션에서 Gravity 값을 조정하면 이러한 힘이 어떻게 작용하는지 감을 잡을 수 있습니다. 또한 Field Weights 파라미터는 애니메이션이 가능하므로, 힘 자체의 설정을 바꾸지 않고도 이 패널에서 애니메이션 도중에 힘의 영향을 변경할 수 있습니다.



All이라는 컨트롤도 있는데, 이는 시뮬레이션에 영향을 주는 모든 Force Field에 동일한 값을 적용할 수 있습니다. Effector Collection 필드는 지정된 Collection에 속한 Effector만 시뮬레이션에 포함되도록 제한할 수 있습니다. 기본적으로는 비어 있으므로 Domain 안의 모든 Effector가 시뮬레이션에 영향을 줍니다. Force Field와 Effector에 대해서는 나중 에피소드에서 다룰 예정이므로, 지금은 이 섹션을 닫겠습니다.

Cache에 대해 이야기할 때는 어떤 정보가 저장되고 이를 어떻게 재사용하는지만 다뤘지만, 저장 방식 자체는 다루지 않았습니다. 이 파라미터들은 Cache 패널의 Volumetric Data 섹션에 있으며, 디스크에 Voxel 데이터를 저장할 때 사용하는 포맷과 압축 및 정밀도 설정을 지정할 수 있습니다.

디스크 저장에 가장 적합한 압축은 Zip이지만, 기본 옵션은 Blosc입니다.

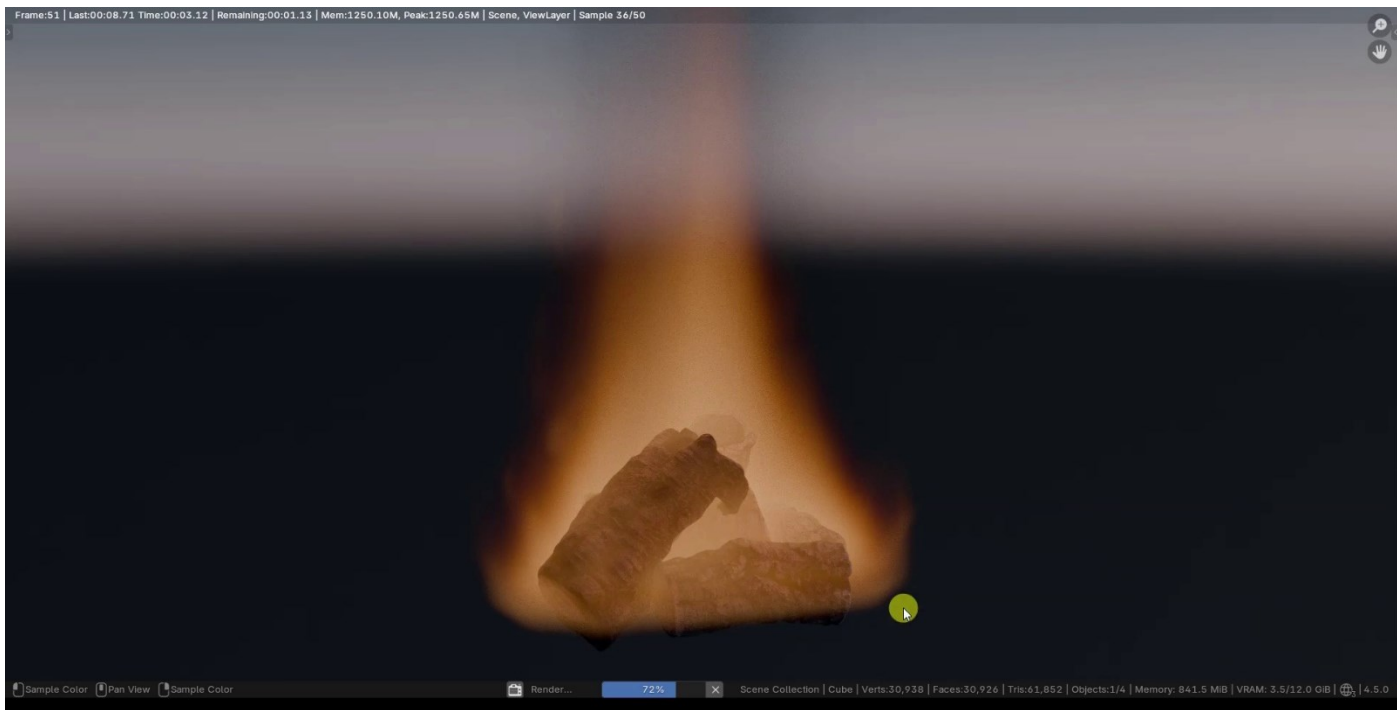
이는 멀티스레드를 지원하므로 더 빠릅니다.

Half precision은 16비트로 데이터를 저장하고, Full은 32비트로 저장하는데, 어떤 경우에는 과도할 수 있습니다.

기본 설정은 Open VDB, Blosc, Half입니다. 대부분의 경우 이 기본값이면 충분합니다.

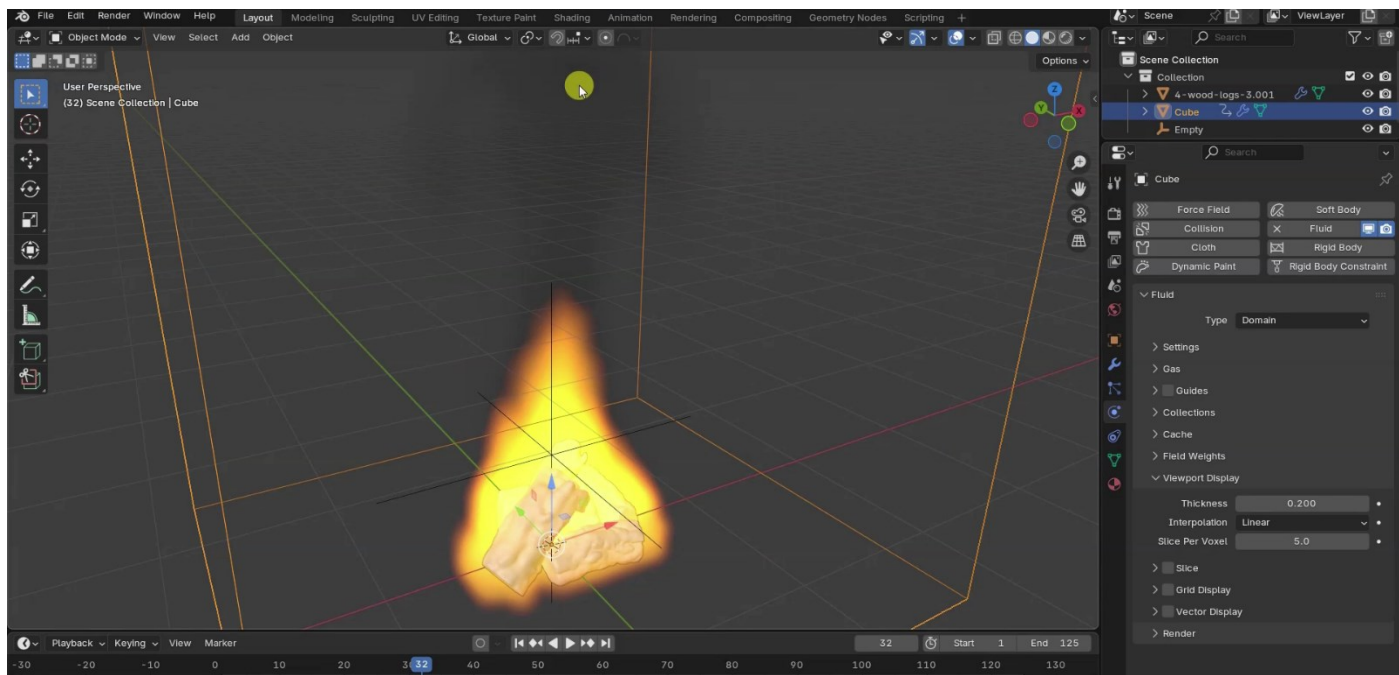
Render 섹션에는 사실 Motion Blur와 관련된 하나의 파라미터만 있습니다.

Motion Blur는 Render 탭에서 프로젝트 레벨에서 활성화해야 하며, 나중에 렌더링 튜토리얼에서 보겠지만 많은 경우 매우 매력적인 효과를 줄 수 있습니다.

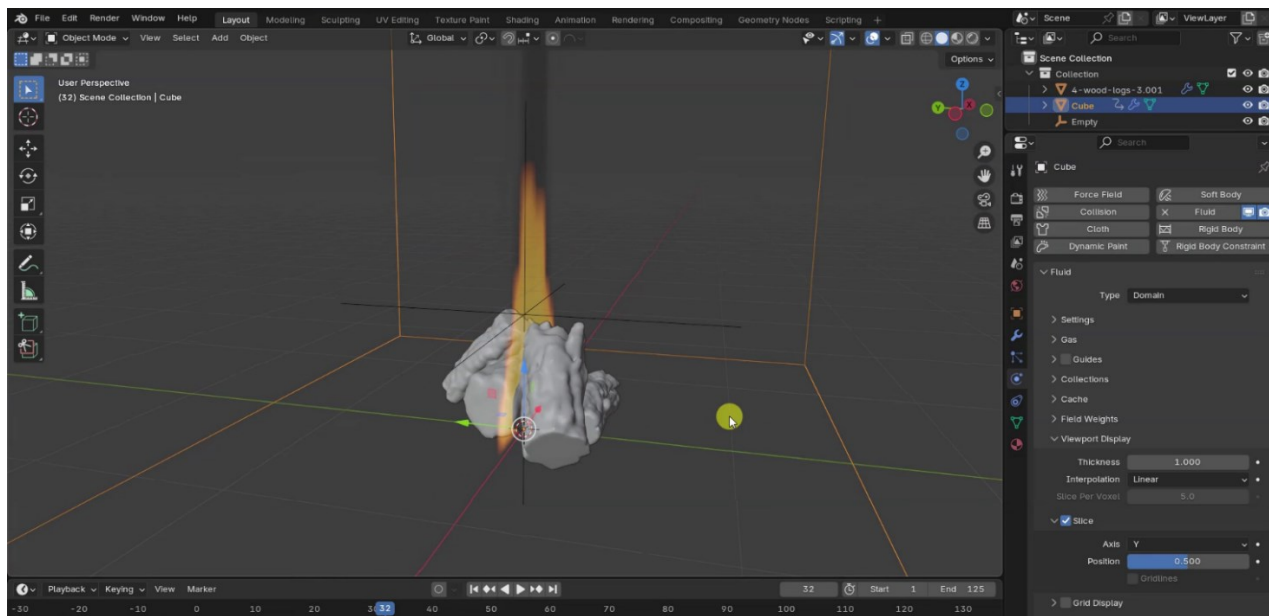


그러나 이번 튜토리얼의 핵심은 Viewport Display 섹션입니다. 여기에는 개별 Voxel에 포함된 정보를 시각화할 수 있는 여러 흥미로운 옵션이 있습니다. 이를 통해 다음 에피소드에서 Fire와 Smoke의 Material을 수정하는 데 사용할 수 있는 Domain Attributes를 더 잘 이해할 수 있습니다.

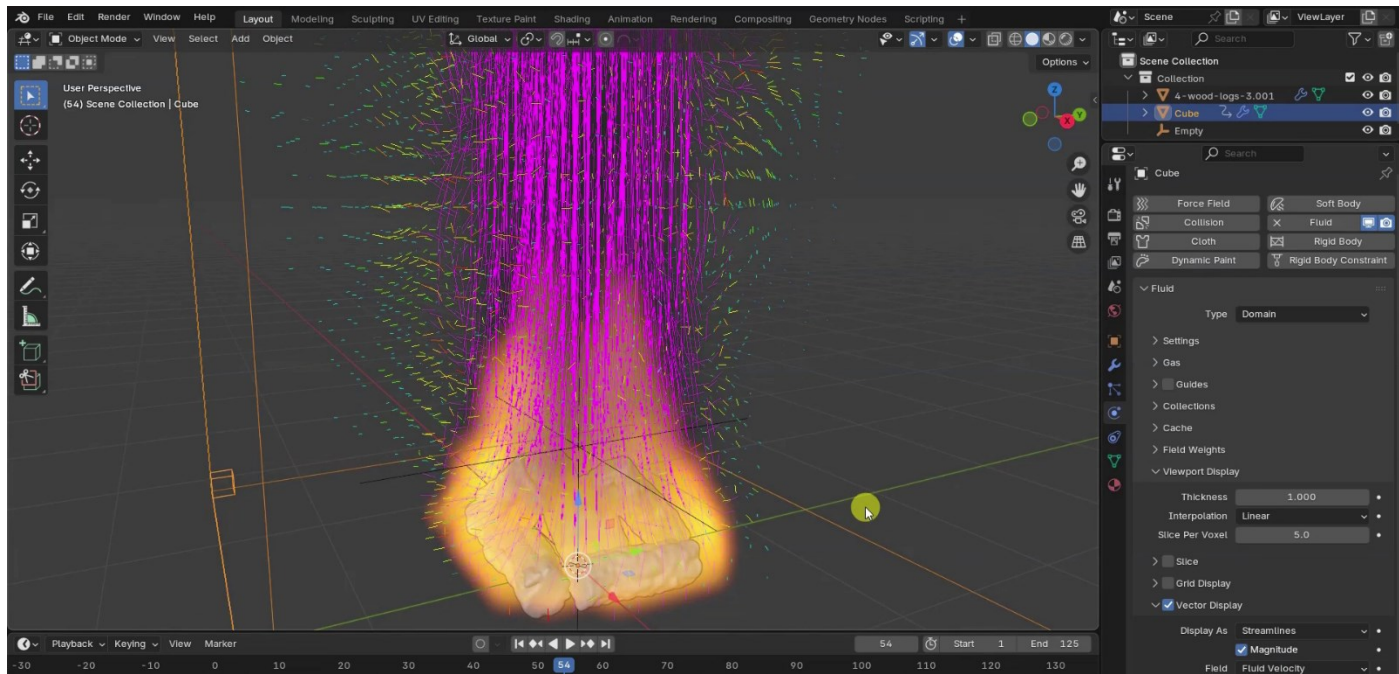
먼저, 3D View에서는 각 Voxel마다 여러 슬라이스가 표시된다는 점을 알아야 합니다. 여기서 Smoke 두께, 보간 품질, Voxel당 계산할 슬라이스 수와 같은 파라미터를 지정할 수 있습니다. Smoke 두께를 늘리면 더 뚜렷하게 보이게 되어 동작을 관찰하기가 쉬워집니다. 다만 이 섹션의 파라미터는 3D Viewport의 Solid 모드에서의 시각화에만 영향을 주며, 최종 렌더링에는 영향을 주지 않는다는 점을 명심해야 합니다.



Domain 전체의 단일 슬라이스를 보고 싶다면 Slice 옵션을 활성화하고 축을 선택하면 됩니다. 이는 Domain을 해당 축을 따라 잘라 단면을 보는 것과 같습니다. 처음에는 3D 씬에서 카메라를 움직일 때마다 표시가 바뀌는데, 기본적으로 슬라이스 축이 Auto로 설정되어 관찰자의 시점에 자동으로 맞춰지기 때문입니다. Position 값은 0에서 1 사이이며, 선택한 축을 따라 Domain의 크기에 비례합니다. 화면에서는 Y축을 사용해 Position 값을 조정하는 예시를 보여드리겠습니다.

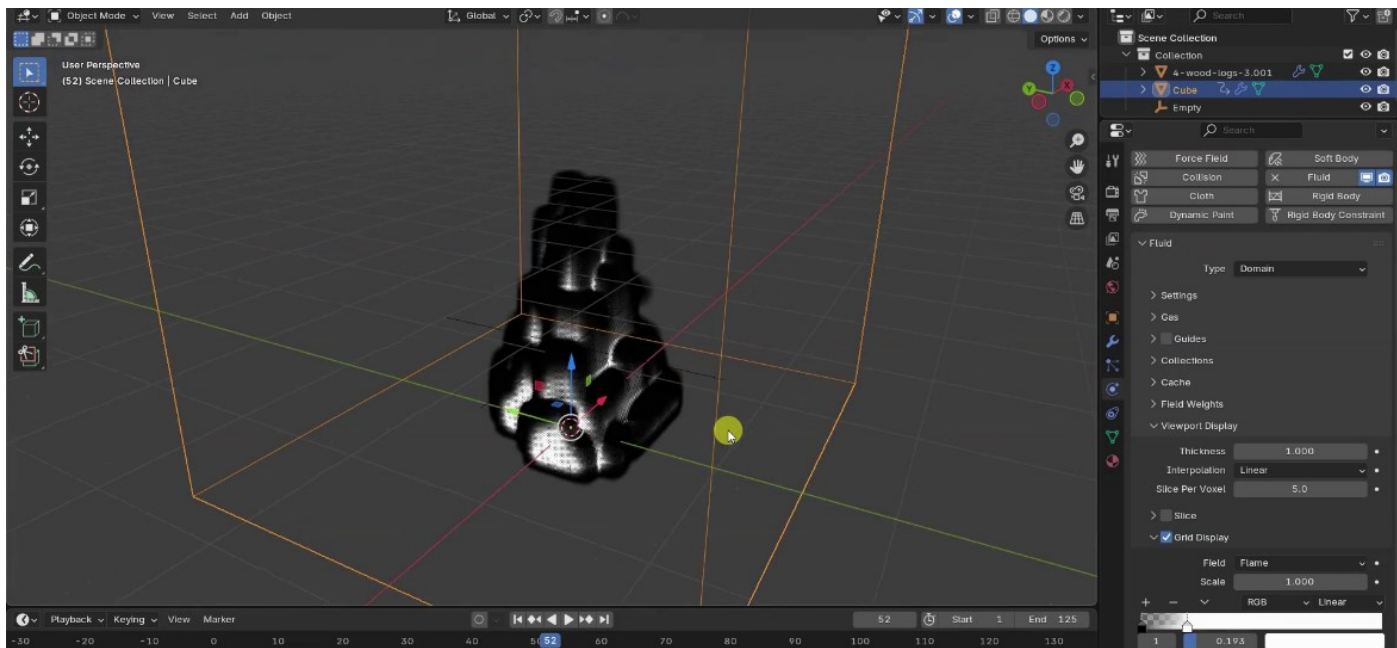


Vector Display는 각 Voxel 안에 Velocity, Guides, Forces의 벡터를 표시합니다. Guides는 Force Field처럼 Fire와 Smoke의 거동에 영향을 주는 특수한 오브젝트입니다. 현재 씬에는 Guides나 Forces가 없으므로 유체 자체에 고유한 Velocity만 볼 수 있습니다. 이 섹션의 다른 옵션들은 시각화에 사용되는 그래픽 요소의 모양과 크기를 조정하여 더 쉽게 읽을 수 있도록 합니다.



마지막으로 Grid Display 섹션을 살펴보겠습니다.

개인적으로 가장 흥미로운 부분이라고 생각합니다. 이 기능을 사용하면 각 Voxel에 대한 Heat나 Density 같은 시뮬레이션 속성을 시각화할 수 있습니다.



Domain은 Voxel로 나뉘며, Resolution은 Grid Display에서 보이는 Cube 분할로 명확히 드러납니다.

각 Voxel에는 온도와 밀도 같은 정보가 포함되어 있습니다.

Color Ramp를 조정하면 모든 Voxel에 걸쳐 이러한 속성들의 값 범위를 확인할 수 있습니다.

이러한 시각화는 특히 유용한데, 일부 속성은 Shading 섹션에서 Attribute로 저장되어 있기 때문입니다.

다음 에피소드에서는 이 Attribute를 불러와 Fire와 Smoke Voxel의 Material을 예를 들어 온도에 따라 수정하는 방법을 살펴보겠습니다.

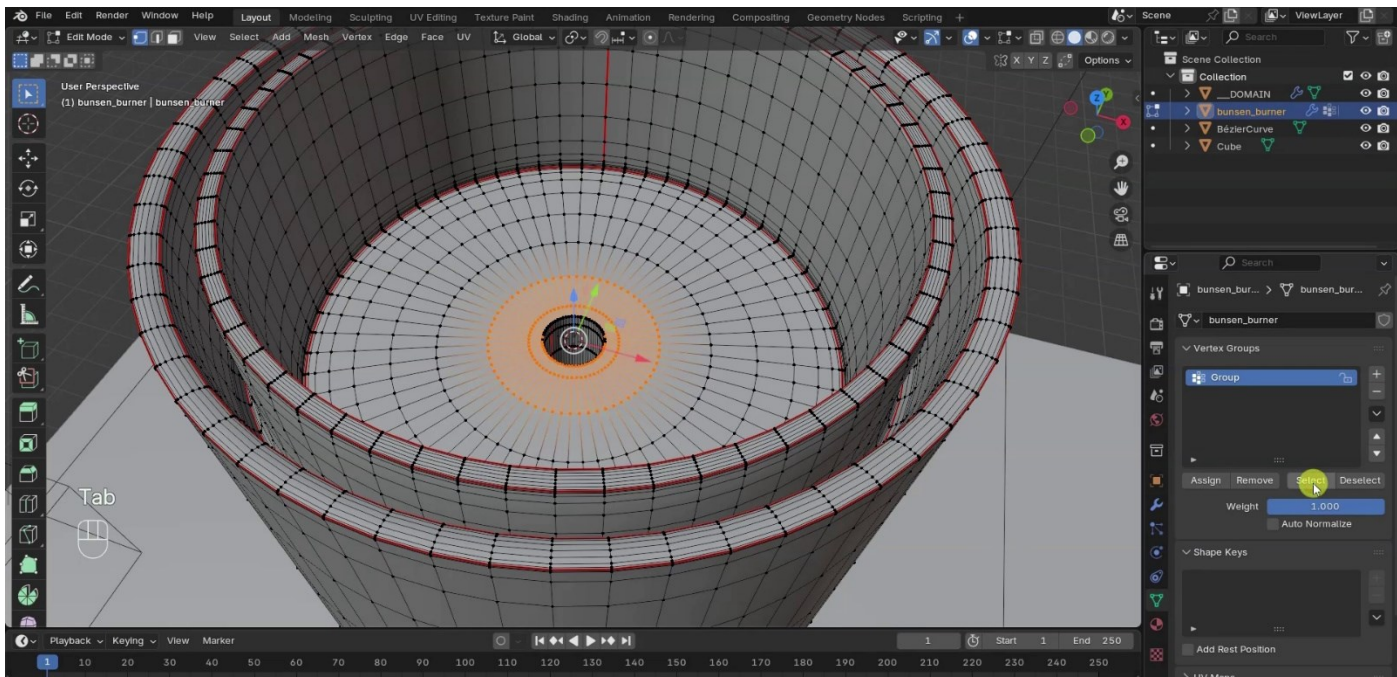
Blender 4.5 Fire Simulations 기본 | 7/10: Emission, Attributes, Volume Info

<https://youtu.be/GwM-rsXtH2k>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 일곱 번째 튜토리얼에서는, 지난번에 설명한 Voxel Attributes를 사용해 렌더링용 Fire와 Smoke Material을 수정하는 방법을 살펴보겠습니다.

이번 에피소드의 첫 번째 예시 씬에서는, Inflow Fire로 설정된 Fluid 컴포넌트를 가진 오브젝트의 Vertex Group에서 Fire가 방출됩니다. 시뮬레이션의 Domain은 버너 상단 부분만 감싸는데, 불꽃이 그곳에서 발생하기 때문입니다.

Vertex Group에서의 방출과 관련해, 불꽃이 보이지 않는다면 문제는 Domain의 Resolution일 수 있습니다. Resolution이 너무 낮아 Voxel 크기가 너무 크면, 방출 버텍스를 포함한 Voxel이 빈 것으로 간주되어 불꽃을 방출하지 않을 수 있습니다. 따라서 Domain Resolution을 높여 보시기 바랍니다.



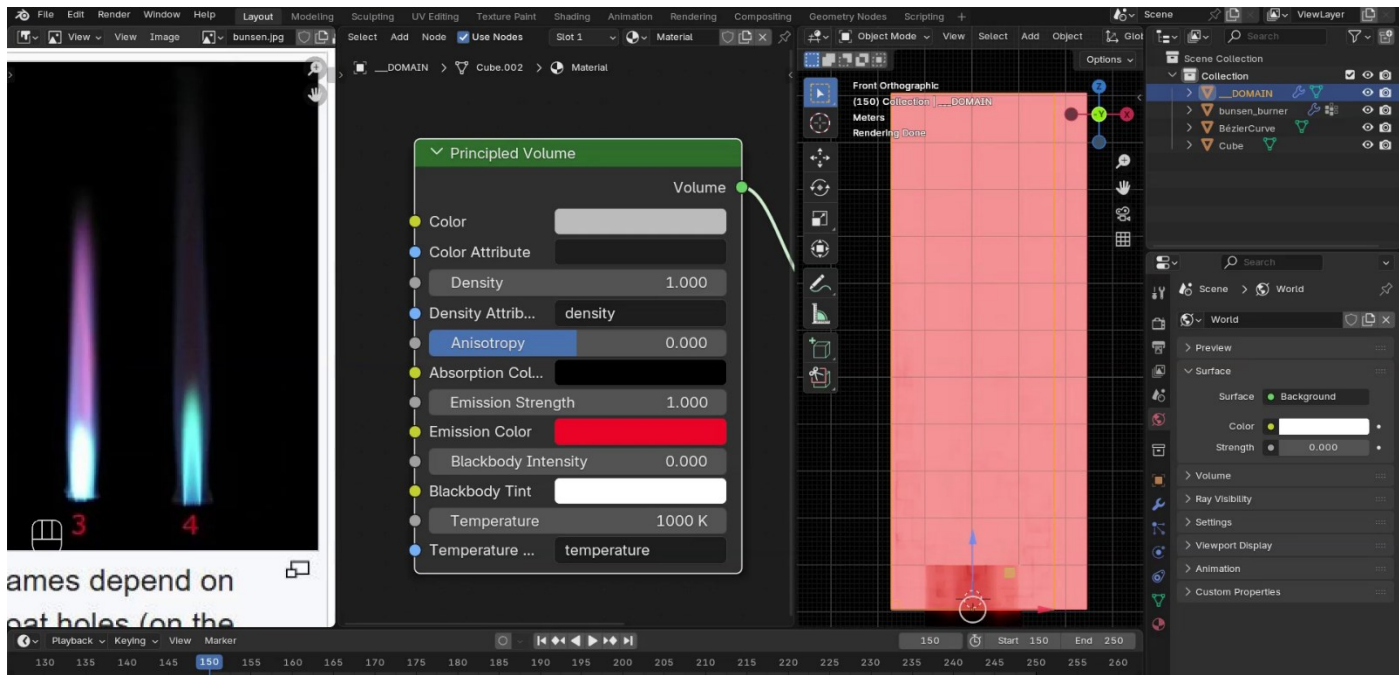
기본 설정에서는 초기 불꽃이 매우 낮게 나타나거나, 옆에서 보면 보이지 않을 수도 있습니다. 하지만 이제 우리는 그 외형을 조정할 수 있는 여러 방법을 알고 있습니다. 예를 들어, Inflow 오브젝트의 Initial Velocity를 조정해 불꽃에 상승력을 주거나, Domain의 반응 속도를 조정해 불꽃을 더 크게 만들 수 있습니다. 이 버너에서

생성되는 불꽃 유형과 관련해, 난류와 색상이 달라질 수 있습니다. 거의 난류가 없는 네 번째 유형의 불꽃을 얻기 위해 Domain의 Fire 섹션에서 Vorticity 값을 0으로 낮추겠습니다.

Material을 정의하기 전에 마지막으로, 방출 Vertex Group에 버텍스를 더 추가하고 Timeline 애니메이션을 시뮬레이션의 마지막 100 프레임으로 제한하겠습니다. 즉, 불꽃이 이미 충분히 발달하고 안정된 구간입니다.

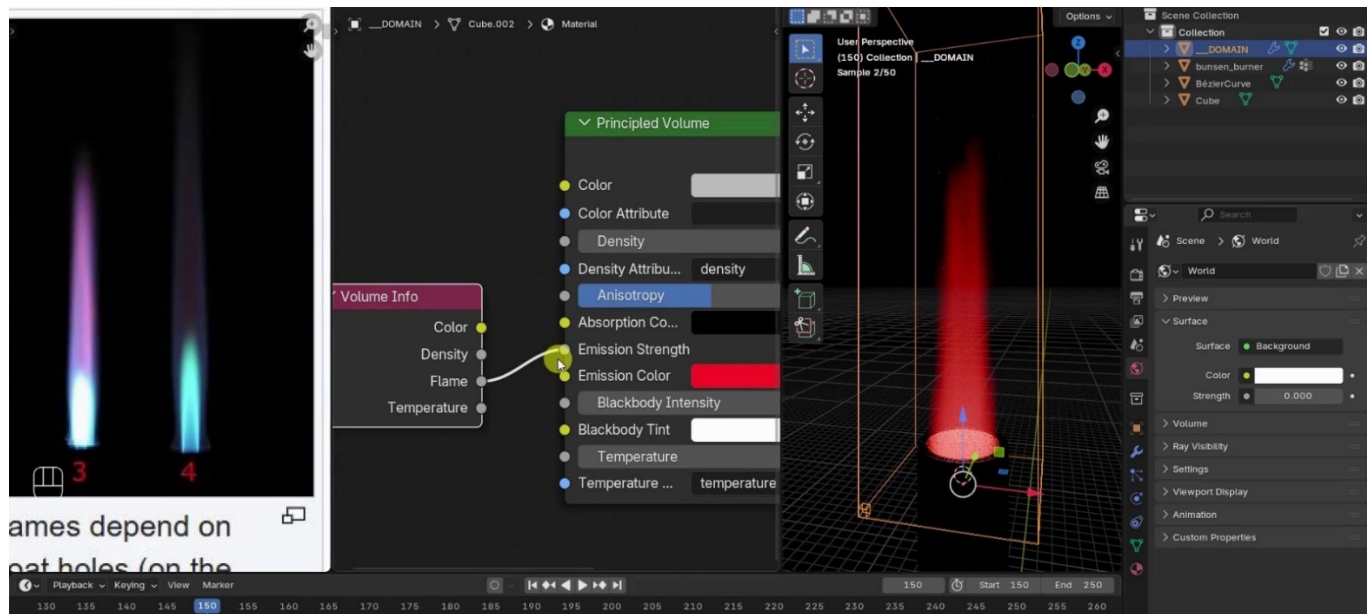
이제 Material 정의로 넘어가겠습니다! 3D 사이드 뷰를 바로 Rendered 모드로 설정하고, World Background Strength를 0으로 두었기 때문에 처음에는 미리보기가 검은색으로 보입니다. 왼쪽의 Image Editor에는 참조 이미지를 배치했고, 가운데에는 Domain의 Principled Volume 노드가 있는 Material Editor를 열어두었습니다.

이전 튜토리얼에서 본 것처럼, Emission Strength를 높이고 Emission Color에 단색을 지정하면 결과가 매우 실망스럽습니다. Blackbody를 사용하는 방법은 이미 살펴보았지만, 이제는 Emission 모드를 배우는 시간이 왔습니다.



지난 에피소드에서 본 것처럼, Fluid 시뮬레이터는 각 Voxel에 대한 정보를 저장합니다. 이러한 데이터는 Attribute 노드를 사용해 해당 필드에 Attribute 이름을 입력하면 불러올 수 있습니다. 하지만 볼륨 Material에는 Volume Info라는 전용 노드가 있으며, 이 노드는 이러한 Attribute를 출력 소켓으로 제공합니다.

Flame 출력이나 Temperature 출력을 Emission Strength 입력에 연결하면 불꽃의 외형이 극적으로 변합니다! Flame 출력은 Temperature 출력보다 보간된 값이 더 잘 제공되며, Temperature 출력은 Domain Resolution을 더 뚜렷하게 드러냅니다. 이런 이유로 저는 Flame을 사용하겠습니다. Flame과 Temperature 출력은 Emission Color에도 사용할 수 있으며, 특히 Color Ramp 노드를 중간에 넣으면 더욱 유용합니다. 이렇게 하면 Domain Voxel에 포함된 정보에 따라 불꽃의 색상을 위치별로 다르게 바꿀 수 있습니다.



저는 Flame 출력을 Color Ramp에 연결하여 Material의 Emission Color 파라미터로 사용하고 있습니다. 이제 Color Ramp의 스톱 색상과 투명도를 정의하는 작업이 필요합니다.

이때 Blender의 Color Picker를 사용해 참조 이미지에서 직접 색상을 샘플링할 수 있습니다.

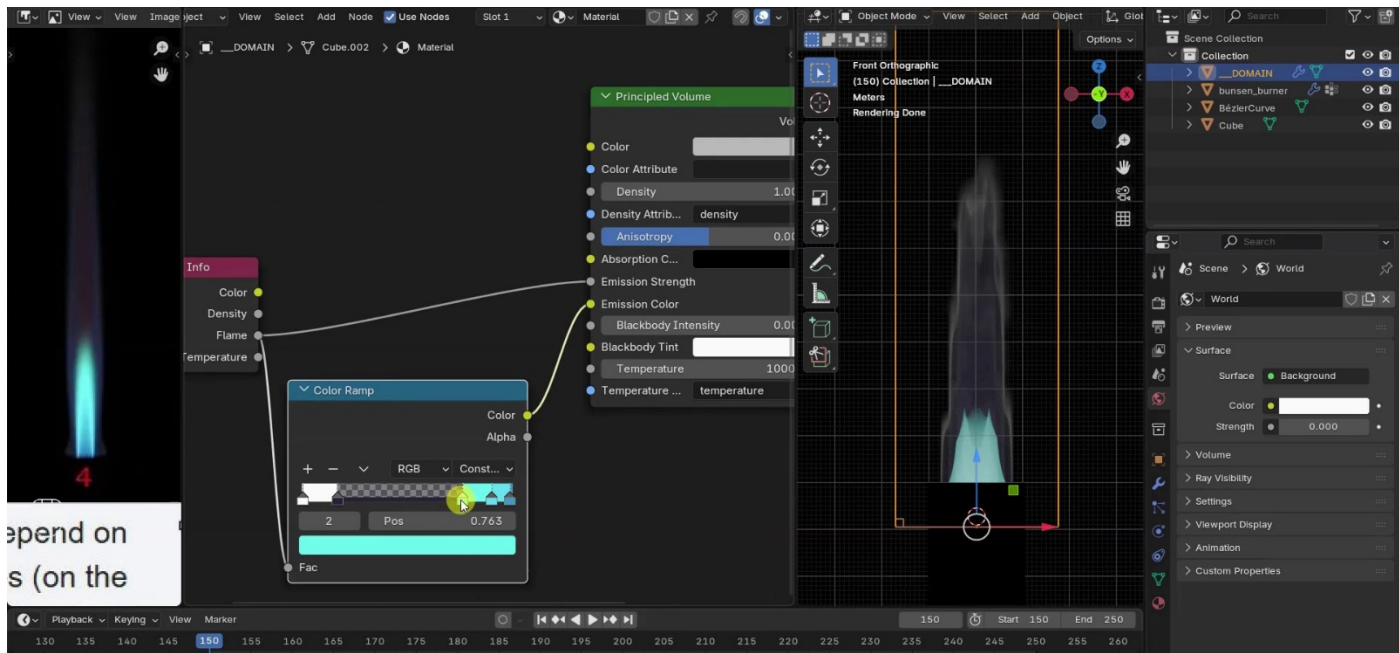
각 Color Stop에는 투명도도 포함할 수 있는데, 이는 불꽃 외곽에 자주 나타나는 반투명 영역을 표현하는 데 매우 유용합니다. 또한 Color Stop 간의 보간 방식을 신중히 선택해야 합니다.

옵션 중 가장 극단적인 것은 Constant인데, 이는 Color Stop 사이를 갑자기 전환합니다.

현실적이지는 않지만, 더 예술적인 표현에는 사용할 수 있습니다.

이번 Material 튜토리얼이 짧게 느껴진다면, 이는 우리가 이미 이전 튜토리얼에서 학습한 정보와 파라미터를 기반으로 하고 있기 때문입니다.

이 시점에서 우리는 Domain과 Flow 오브젝트의 Fluid 파라미터로 다양한 불꽃 형태를 정의하고, Blackbody나 Emission과 Blender의 여러 노드를 사용해 시각적 외형을 조정할 수 있는 충분한 지식을 갖추고 있습니다.



제가 방금 보여드린 세팅에만 국한되지 마세요. 다른 Material 노드도 실험해 보시기 바랍니다. 예를 들어, Math 노드를 Multiply 또는 Power로 설정해 Emission Strength를 수정함으로써 Flame에서 오는 정보를 바꿀 수 있습니다. 게다가 이러한 노드들의 파라미터에는 키프레임을 넣을 수 있으므로, 시뮬레이션의 Fluid 컴포넌트 필드를 조정하지 않고도 Material을 변경하여 불꽃의 강도를 애니메이션할 수 있습니다.

튜토리얼을 마무리하기 전에, 동일한 Domain 안에 서로 다른 색상을 가진 세 개의 Inflow 오브젝트를 만든 방법을 보여드리겠습니다. 결과가 완벽하지는 않지만, 노드의 또 다른 활용 사례, 특히 Color Ramp를 보여드리기 위한 좋은 기회가 됩니다.

아시다시피, Material은 Domain의 Voxels에서 생성되므로 Domain 레벨에서 정의되어야 합니다. 따라서 그 안의 Inflow 오브젝트에 서로 다른 Blackbody 노드를 할당할 수는 없습니다.

그러나 Inflow 오브젝트에는 Inflow 패널에서 정의되는 Smoke Color라는 필드가 있습니다. 이 색상은 Volume Info 노드를 통해 Domain Shader에서 가져올 수 있으며, 특히 Color 출력으로 불러올 수 있습니다.

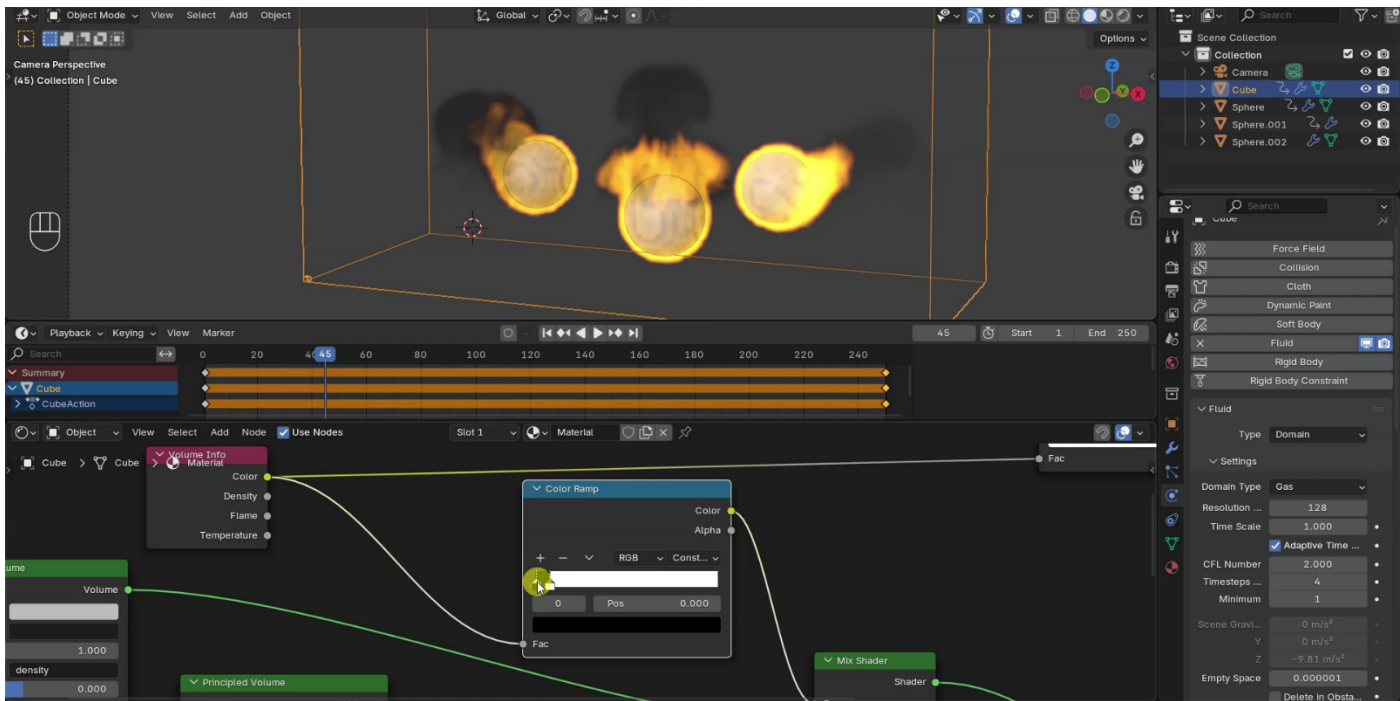


이 필드를 Mix Shader 노드의 Factor로 사용하면, 각 Voxel에 대해 정의된 Smoke Color에 따라 서로 다른 Principled Volume 노드를 할당할 수 있습니다.

하지만 Mix Shader만 사용하면 단순히 두 Shader를 반반 섞기 때문에, Volume Info에서 오는 색상들을 날카롭게 구분하기 위해 Color Ramp를 Constant로 설정해야 합니다.

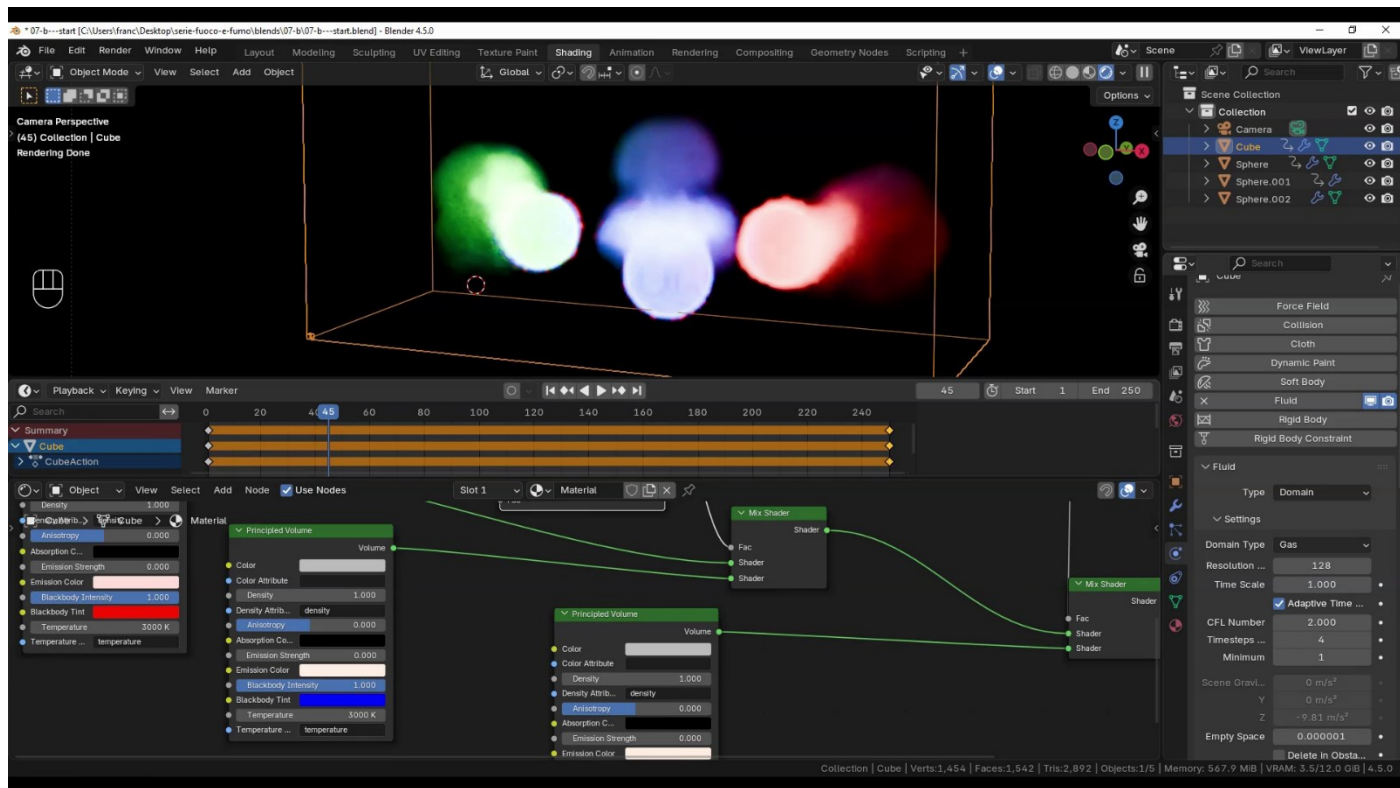
그런 다음 세 개의 Inflow 오브젝트에 각각 흰색, 회색, 검은색 Smoke Color를 왼쪽에서 오른쪽으로 할당했습니다.

첫 번째 Color Ramp는 검은색과 다른 색상을 구분합니다. 검은색은 0에 해당하므로 첫 번째 Mix Shader 노드의 첫 번째 Shader 입력으로 전달되며, 이 입력은 빨간 Blackbody Tint가 적용된 Principled Volume 노드에 연결되어 있습니다. 또한 Rendered 뷰에서 요소의 색상을 더 잘 확인하기 위해 3D 프리뷰에서 Denoise를 활성화했습니다.



Mix Shader의 두 번째 Shader 입력은 파란 Blackbody Tint가 적용된 Principled Volume 노드에 연결되어 있습니다. 따라서 Smoke Color가 검은색이면 빨간 Blackbody가 나오고, 검은색이 아니면 파란 Blackbody가 나옵니다.

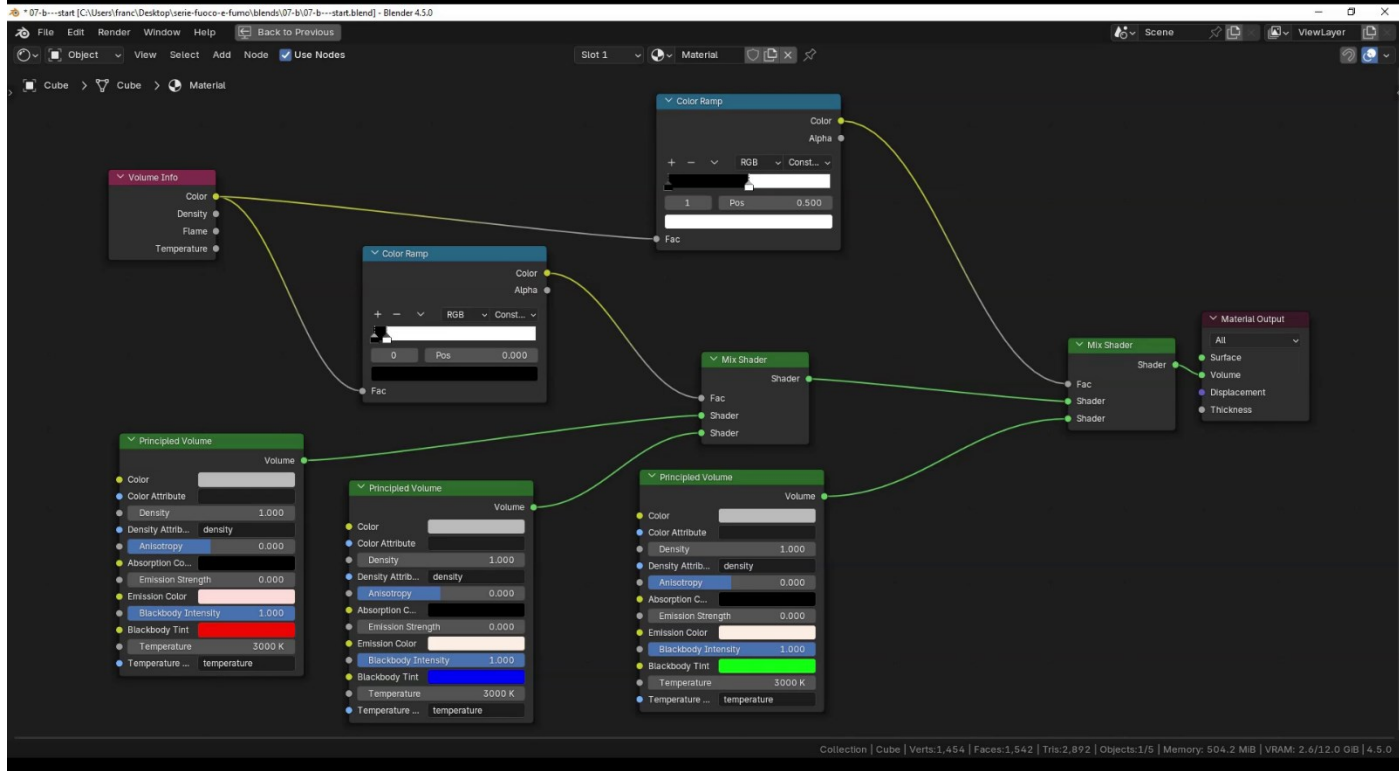
그 다음은 또 다른 Mix Shader 노드인데, 이 노드는 Constant로 설정된 또 하나의 Color Ramp를 입력으로 사용하며, Color Stop이 Ramp의 정확히 중간 지점에 배치되어 있습니다.



이 경우 Smoke Color가 검은색과 회색 사이에 있으면 두 번째 Mix Shader의 첫 번째 Shader 입력이 사용됩니다. 이 Shader는 앞서 검은색과 회색을 구분한 결과이므로, 최종 색상은 빨강 또는 파랑이 됩니다.

반대로 Smoke Color가 회색과 흰색 사이에 있으면 두 번째 Mix Shader의 두 번째 Shader 입력이 사용됩니다. 이 입력은 초록 Blackbody Tint가 적용된 Principled Volume 노드에 연결되어 있습니다. 따라서 Smoke Color가 흰색인 오브젝트는 초록색으로 렌더링됩니다.

물론 두세 가지 색상이 섞인 불꽃 구체를 만드는 방법은 이것만 있는 것이 아니지만, 사례로 보여드리고 싶었습니다. 앞서 말씀드린 것처럼 구분이 완벽하지 않으므로, 더 나은 설정이나 다른 방법을 찾으시면 댓글로 공유해 주시길 권장합니다!



Blender 4.5 Fire Simulations 기본 | 8/10: Effectors (Guides, Colliders)

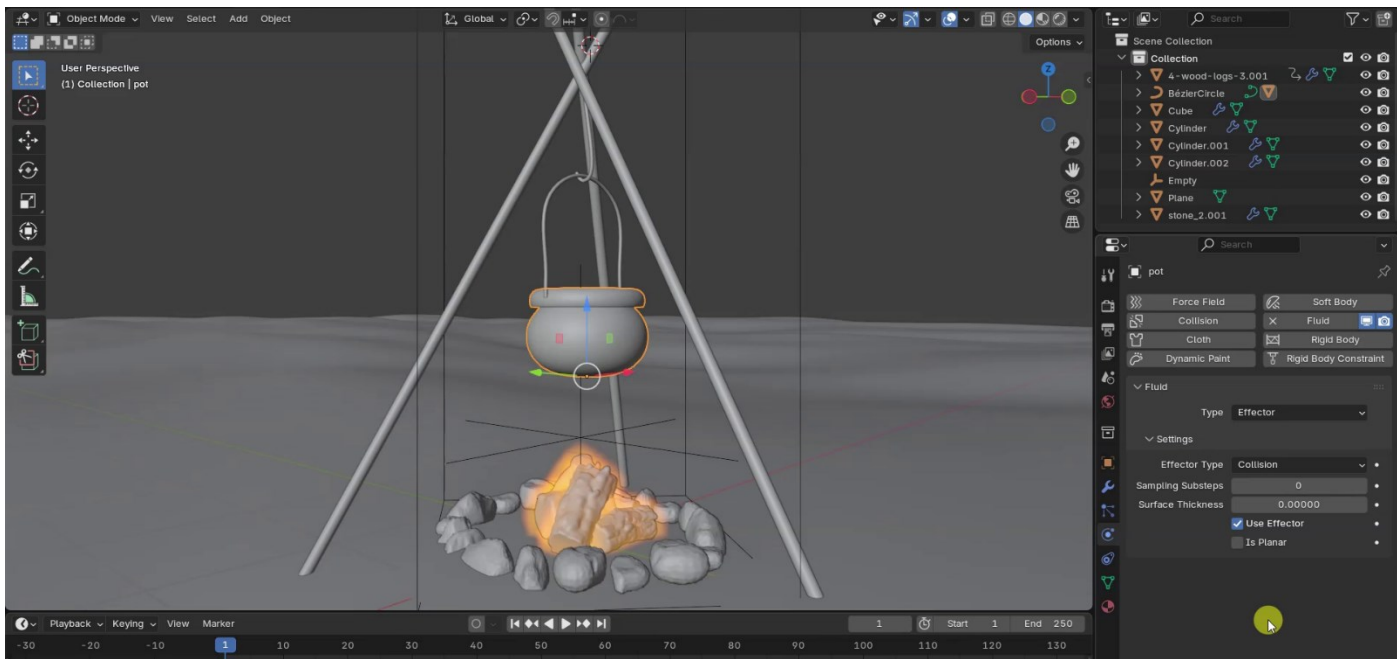
<https://youtu.be/evQP4nxeQRw>

안녕하세요!

이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 여덟 번째 튜토리얼에서는 Blender에서 제공하는 두 가지 Effector 오브젝트, 즉 Collision과 Guides를 살펴보겠습니다.

Effector 오브젝트는 물리 시뮬레이션 요소를 굴절시키거나 그 흐름에 영향을 주는 데 사용됩니다. 화면에 보이는 씬에서는 불꽃과 Smoke가 냄비를 그대로 통과하는데, 이는 냄비가 장애물로 인식되지 않기 때문입니다.

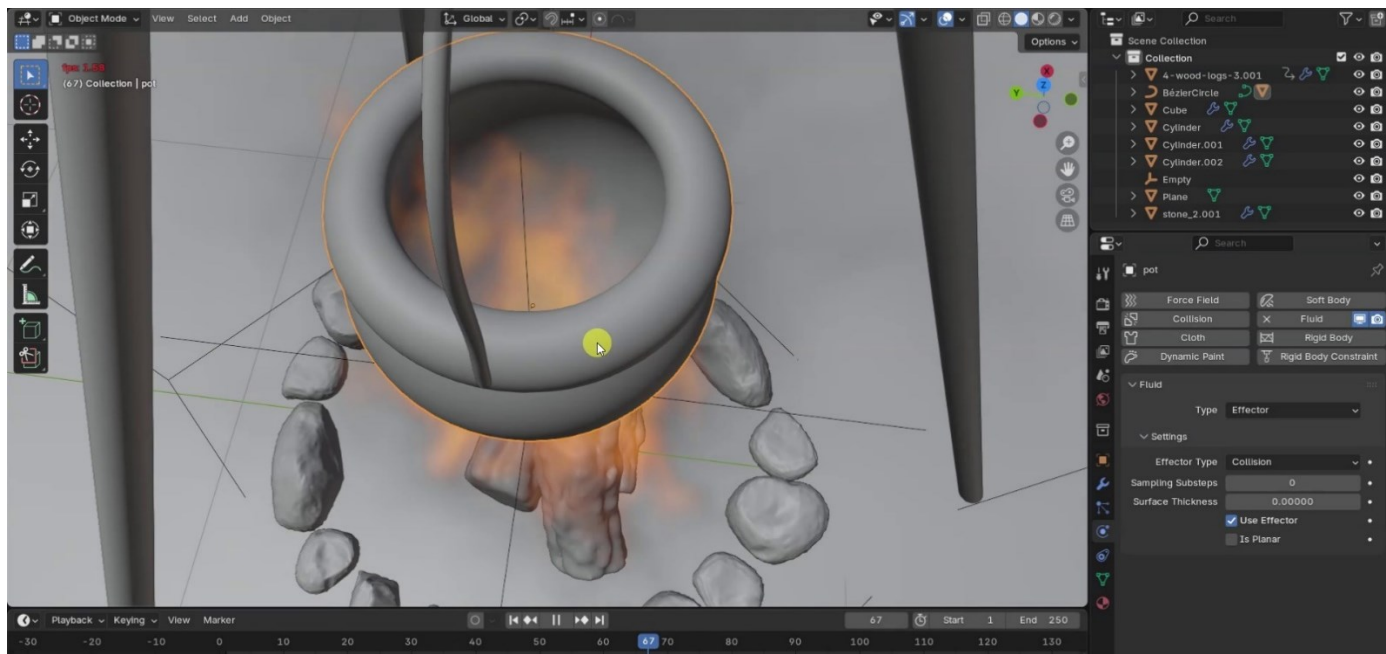
이 문제를 해결하려면 냄비에 Fluid 컴포넌트를 Effector 타입으로 할당한 뒤, 전용 메뉴에서 Collision을 선택해야 합니다.



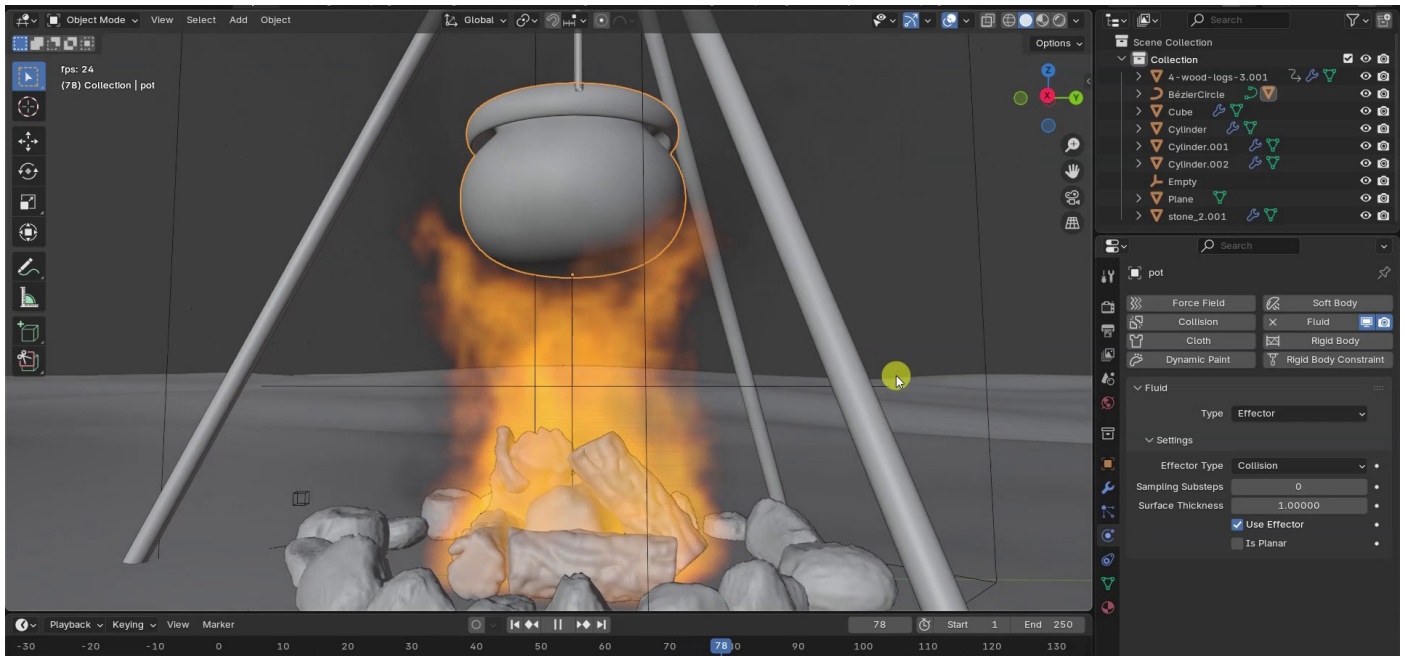
보시는 것처럼 파라미터는 아주 적으며, 사실 대부분은 Flow 오브젝트에서 사용된 것과 비슷한 항목들입니다. 예를 들어 Sampling Substeps는 프레임 사이에 더 많은 중간 계산을 추가합니다. Use Flow와 유사한 Use Effector 옵션도 있는데, 이는 애니메이션이 가능하여 원하는 대로 충돌을 활성화하거나 비활성화할 수 있습니다. Is Planar는 볼륨이 없거나 안쪽과 바깥쪽을 정의할 수 없는 지오메트리에 유용합니다. 여기서 새로 등장하는

파라미터는 Surface Thickness인데, 이는 오브젝트 주변에 일종의 외피를 만들어 줍니다. 시뮬레이션에서 불꽃, Smoke, 오브젝트 표면 사이에 교차 현상이 발생할 때 유용할 수 있습니다.

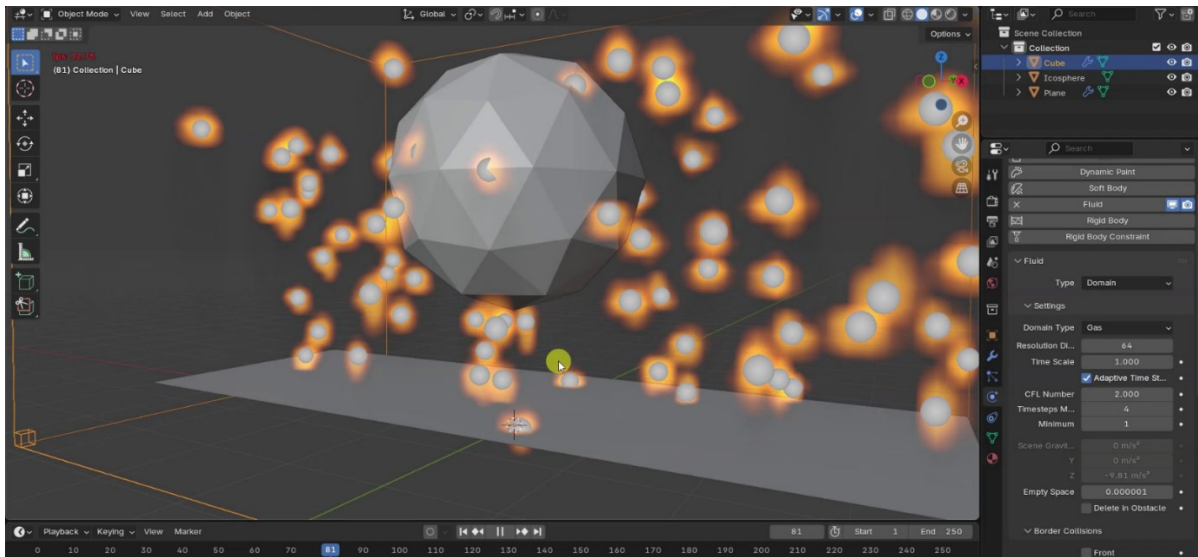
하지만 시뮬레이션을 재생해 보면, 몇몇 프레임에서는 불꽃이 여전히 오브젝트를 통과하는 것을 확인할 수 있습니다.



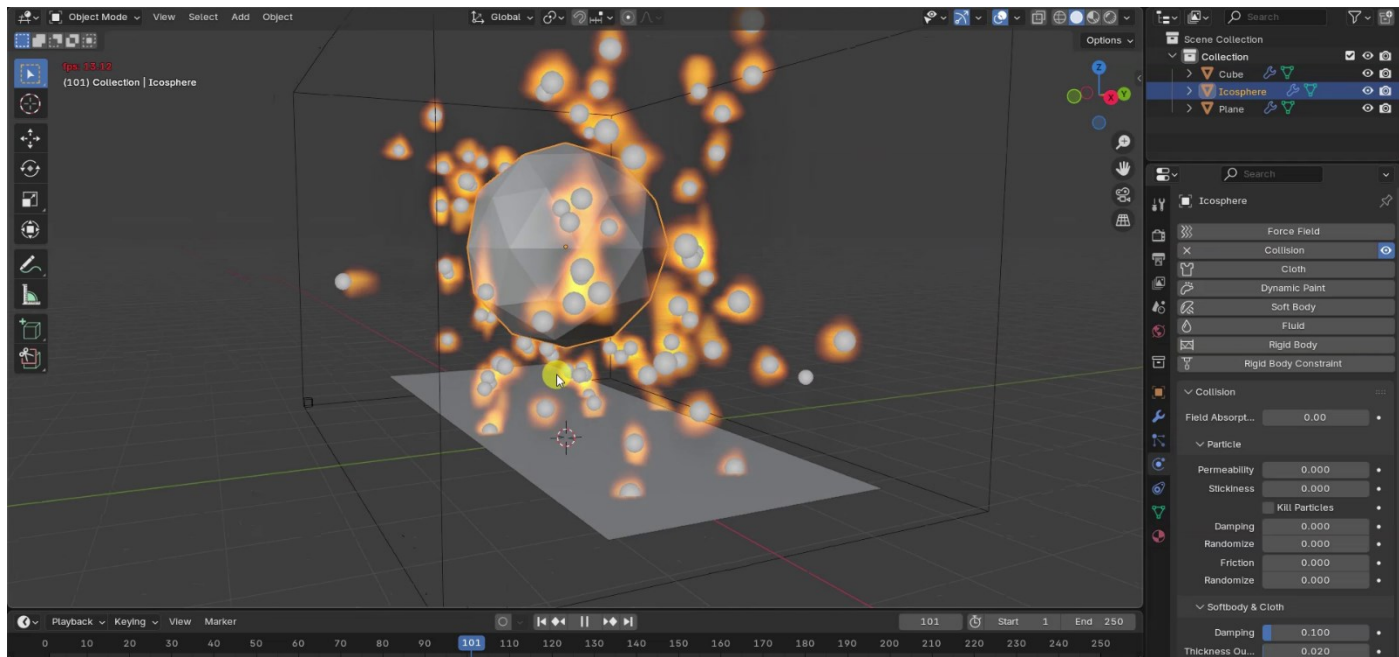
이 문제를 해결하기 위해 Domain Resolution은 그대로 두고, Surface Thickness 값을 점차 높이며 여러 번 테스트했습니다. Thickness를 1로 설정하니 화면에서 보이는 결과를 얻을 수 있었습니다. Substeps 값을 변경하지 않았는데, Tooltip에 따르면 이 옵션은 주로 빠르게 움직이는 Effector에서 유용하기 때문입니다. 우리의 오브젝트는 정적이므로 Thickness만 조정했고, 이것이 최종 결과입니다.



Guide 오브젝트로 넘어가기 전에, Collision에 관한 또 다른 점을 명확히 해야 합니다. 제가 사용하는 씬에는 위쪽으로 파티클을 방출하는 Plane이 있습니다. 파티클이 상승하도록 하기 위해 파티클 시스템 설정에서 Gravity를 비활성화했습니다. 이 방출체 오브젝트는 Fire 시뮬레이션의 Inflow 오브젝트이기도 합니다. 그래서 Flow Source를 Particle System으로 바꾸고, 해당 필드에 제가 만든 파티클 시스템을 지정했습니다. Fire 파티클이 시스템의 파티클보다 훨씬 크고 블록처럼 보이지 않도록 Domain Resolution을 높였습니다. 하지만 보시는 것처럼 파티클은 Domain 안의 오브젝트와 전혀 상호작용하지 않습니다.



이 경우, Fire 파티클이 오브젝트에 닿았을 때 튕기게 하려면 Collision 컴포넌트를 추가해야 합니다. 이 컴포넌트는 파티클을 튕기게 하지만, Smoke와 일부 불꽃은 여전히 오브젝트 안으로 들어가 버립니다. 따라서 Fire와 Smoke 시뮬레이션을 제대로 처리하려면 Collision Effector 타입의 Fluid 컴포넌트도 추가해야 합니다.

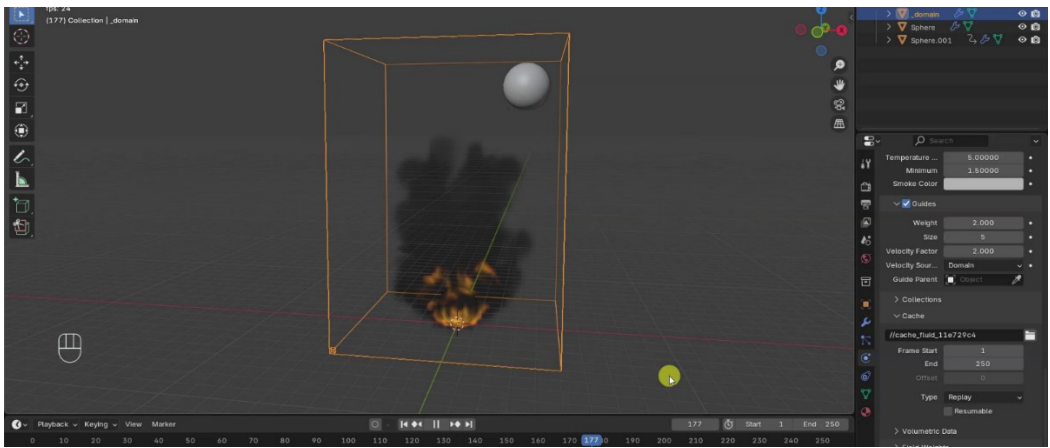
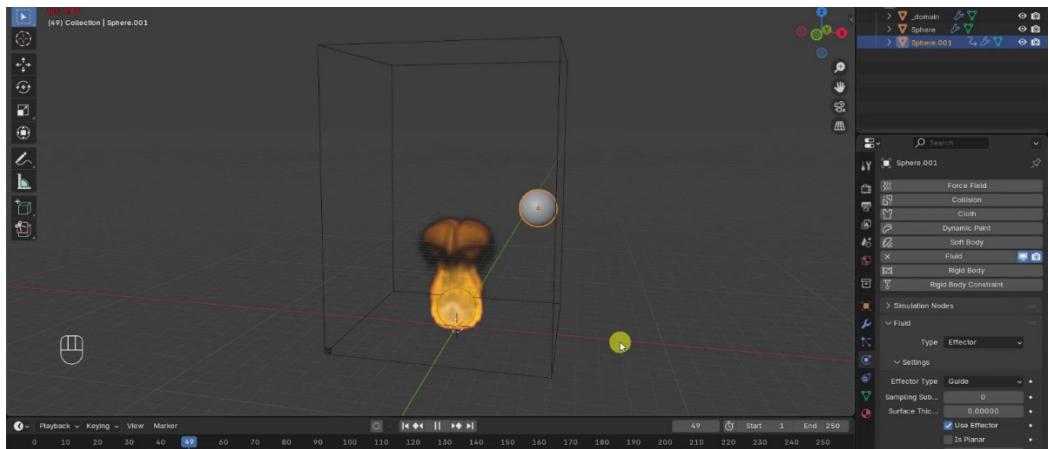


이번 에피소드를 마무리하면서 Guide 타입 Effector 오브젝트에 대해 이야기해 보겠습니다. 이들은 속도를 가져야 하는 특수 오브젝트로, 시뮬레이션에서 불꽃과 Smoke의 움직임에 영향을 줍니다. 예시를 보여드리기 위해, 저는 시뮬레이션 Domain 안에서 특이한 궤적을 따라 움직이는 구체를 추가했습니다. 이 구체에는 아직 Fluid 컴포넌트가 없으므로 시뮬레이션에 아무런 영향을 주지 않습니다.

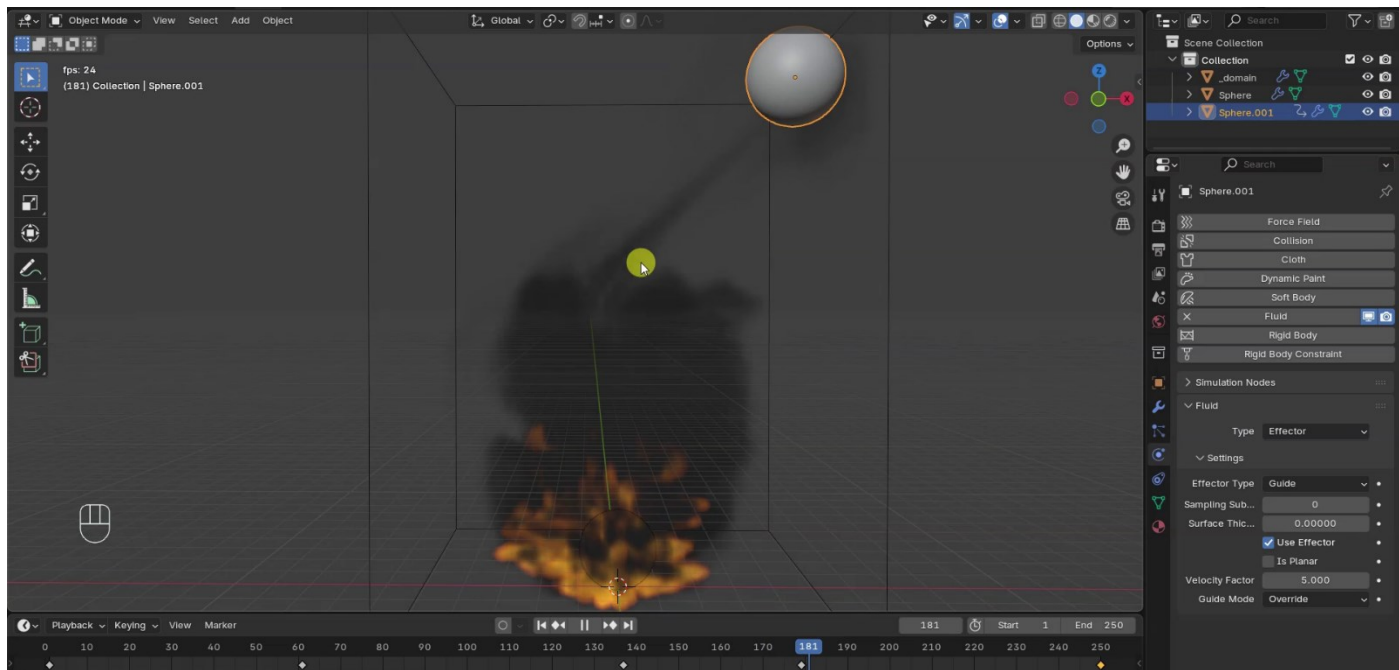
Fluid Effector 컴포넌트를 추가하고 타입을 Guide로 선택하면, 익숙한 파라미터 몇 개와 새로운 항목 두어 개가 보입니다. 그중 Velocity Factor가 이후 우리가 가장 주목할 항목입니다.

시뮬레이션을 재생해 보면, 성능이 느려지긴 하지만 Fire나 Smoke에 오브젝트의 가시적인 영향은 없습니다.

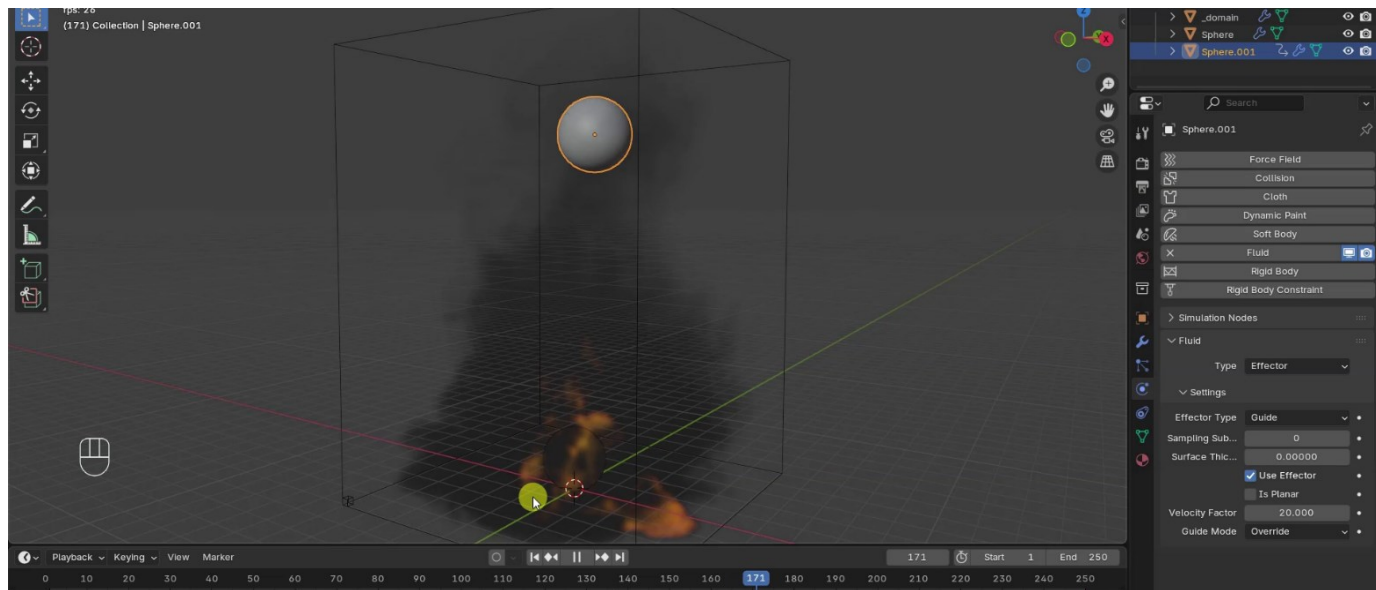
이는 Guides 사용이 계산적으로 매우 비용이 큰 작업이므로, Domain 레벨에서도 Guides를 활성화해야 하기 때문입니다. 그렇지 않으면 효과가 적용되지 않습니다. Domain의 Guides 섹션을 활성화한 후 시뮬레이션을 다시 시작했습니다. 이제 계산 시간이 훨씬 오래 걸리므로 Blender가 이를 처리하도록 두고, 그 결과를 보여드리겠습니다.



결과는 Guides 없이 본 것과는 이미 꽤 다르지만, 아직 인상적이지는 않습니다. 오브젝트가 시뮬레이션에 미치는 영향을 더 강하게 만들기 위해, 해당 오브젝트를 선택하고 Velocity Factor 값을 더 높은 수치로 테스트했습니다. 이것은 Velocity Factor를 5로 설정한 시뮬레이션입니다. Smoke가 끌려가는 방식이 더 눈에 띄게 나타납니다.



이것은 Velocity Factor를 20으로 설정한 시뮬레이션입니다. Guide 오브젝트는 Collider가 아니라 시뮬레이션 요소를 교란하고 끌어가는 특수한 타입임이 분명합니다.



그러나 Fire와 Smoke의 움직임에 영향을 주는 또 다른 방법이 있는데, 그것은 Force Fields를 사용하는 것이며, 이는 다음 튜토리얼에서 다룰 것입니다.

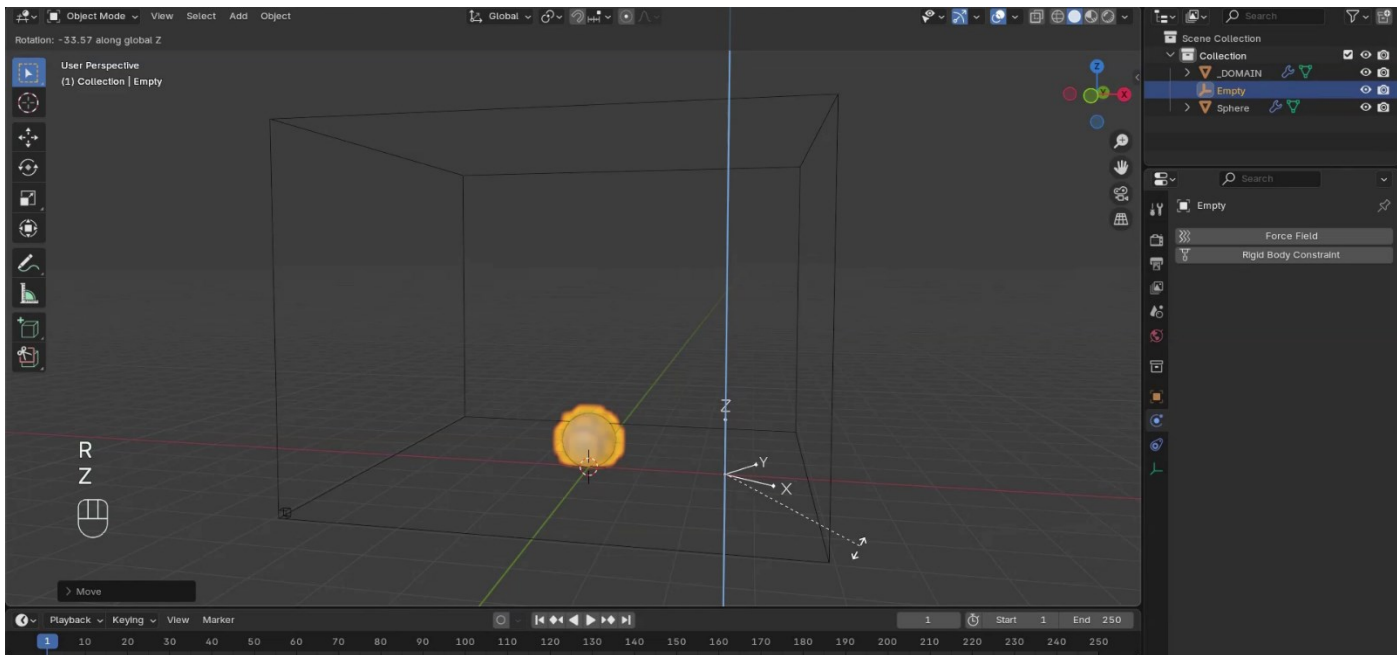
Blender 4.5 Fire Simulations 기본 | 9/10: Force Fields (Turbulence, Curve Guide)

<https://youtu.be/AsrYsZcwpqU>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 아홉 번째 튜토리얼에서는 Turbulence와 Curve Guide 같은 Force Fields를 사용해 시뮬레이션 요소들의 거동과 궤적에 영향을 주는 방법을 살펴보겠습니다.

이번 튜토리얼에서 사용할 시작 씬은 매우 단순하며, Domain과 Fire와 Smoke를 Inflow 모드로 방출하는 구체만으로 구성되어 있습니다.

씬에 Empty를 추가하면 여기에 Force Field를 할당해 시뮬레이션 요소를 교란하거나 유도할 수 있습니다. Force Fields는 Mesh 오브젝트의 Fluid 시뮬레이션과 마찬가지로 Physics 패널에서 찾을 수 있습니다. 저는 일부 Force Fields가 특정 방향으로 작동하기 때문에, 이를 시각적으로 바로 이해하기 쉽도록 Empty 타입을 Arrows로 선택했습니다.



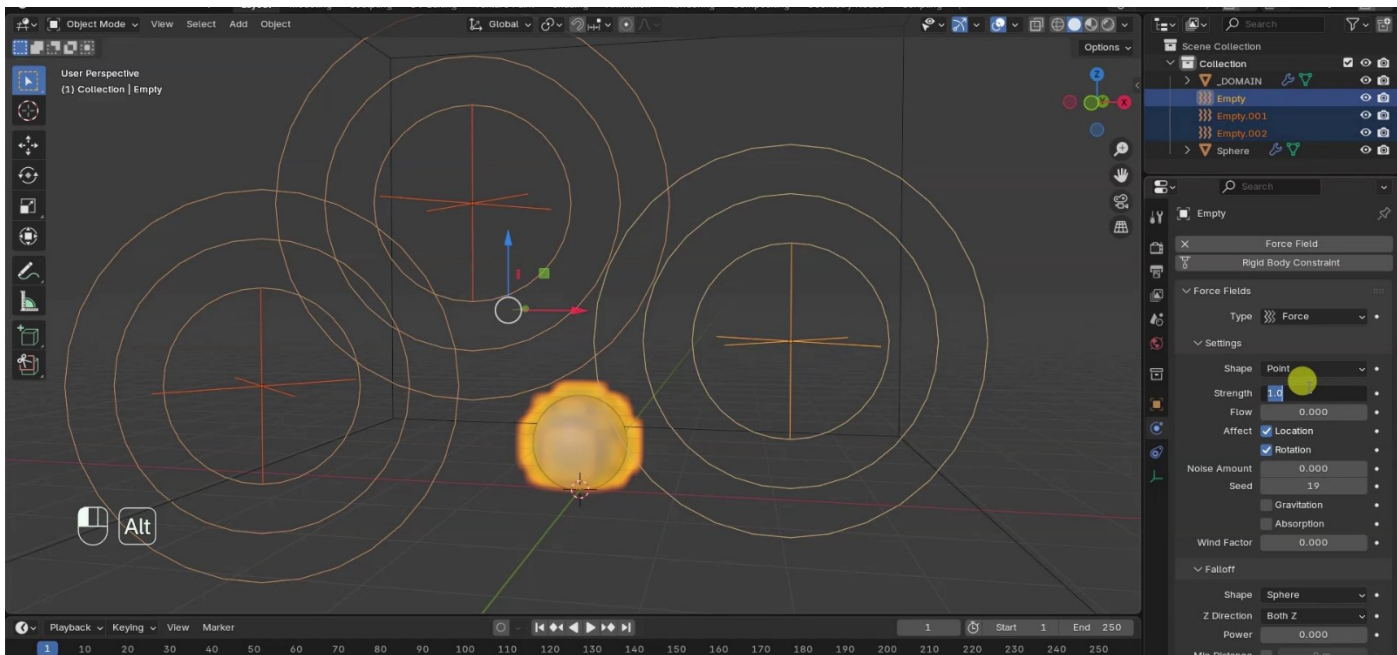
그런 다음 Force Field를 추가했는데, 기본 타입은 Force입니다. 각 Force Field에는 Tooltip이 있어 해당 작용과 영향을 주는 방향을 설명해 줍니다. Force는 기본적으로 Point 모양을 가지며 중심에서 방사형으로 작용합니다. 그러나 시뮬레이션을 재생해 보면, Inflow 오브젝트가 Force Field 범위 안에 있는 것처럼 보여도 눈에 띄는 효과는 없습니다.

이런 경우 작동하는 절차 중 하나는 다음과 같습니다. 먼저 프로젝트 파일을 저장하고 캐시 폴더를 지정합니다. 그다음 Domain의 Baking 모드를 All로 전환하고, Blender가 최소한 몇 프레임이라도 시뮬레이션을 계산하도록 둡니다. 이 과정이 끝나면 Play를 눌러 시뮬레이션이 제대로 계산되었는지 확인합니다.

제대로 실행되면 Domain Cache를 비우고 Baking 모드를 다시 Replay로 설정합니다. 프로젝트의 애니메이션 길이와 Domain의 시뮬레이션 프레임 범위를 조정한 뒤 Timeline에서 Play를 누릅니다.

시뮬레이션이 이제 정상적으로 실행되면, 이후부터는 Replay 모드에서 다양한 효과를 테스트할 수 있습니다.

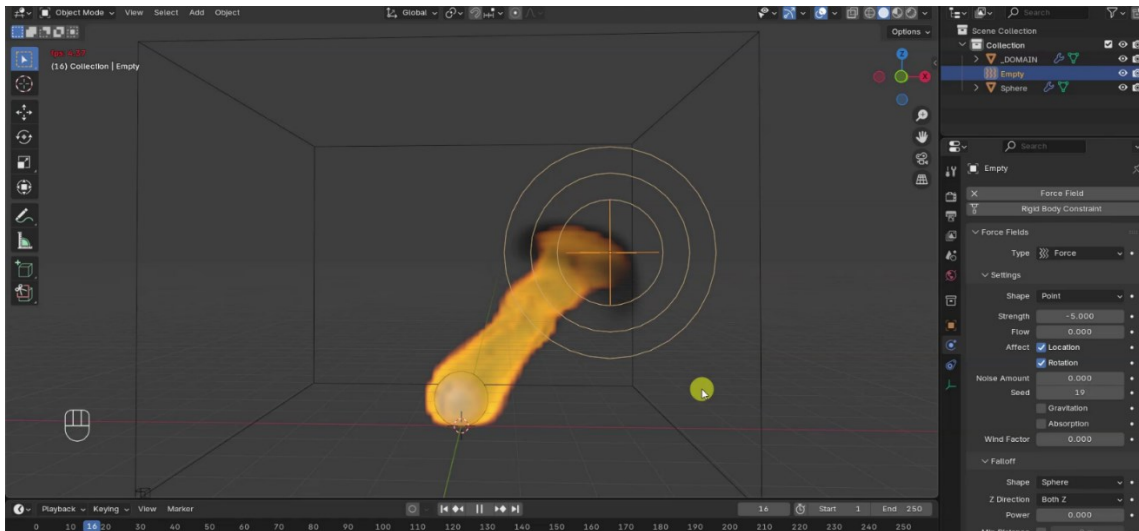
효과의 강도를 제어하기 위해 Force Fields에는 Strength 필드가 있으며, 이는 선택한 Force Field에 적용되고 애니메이션이 가능합니다. 참고로, ALT 키를 누른 상태에서 필드를 클릭해 값을 변경하면 여러 개의 선택된 오브젝트에 동시에 적용할 수 있습니다.



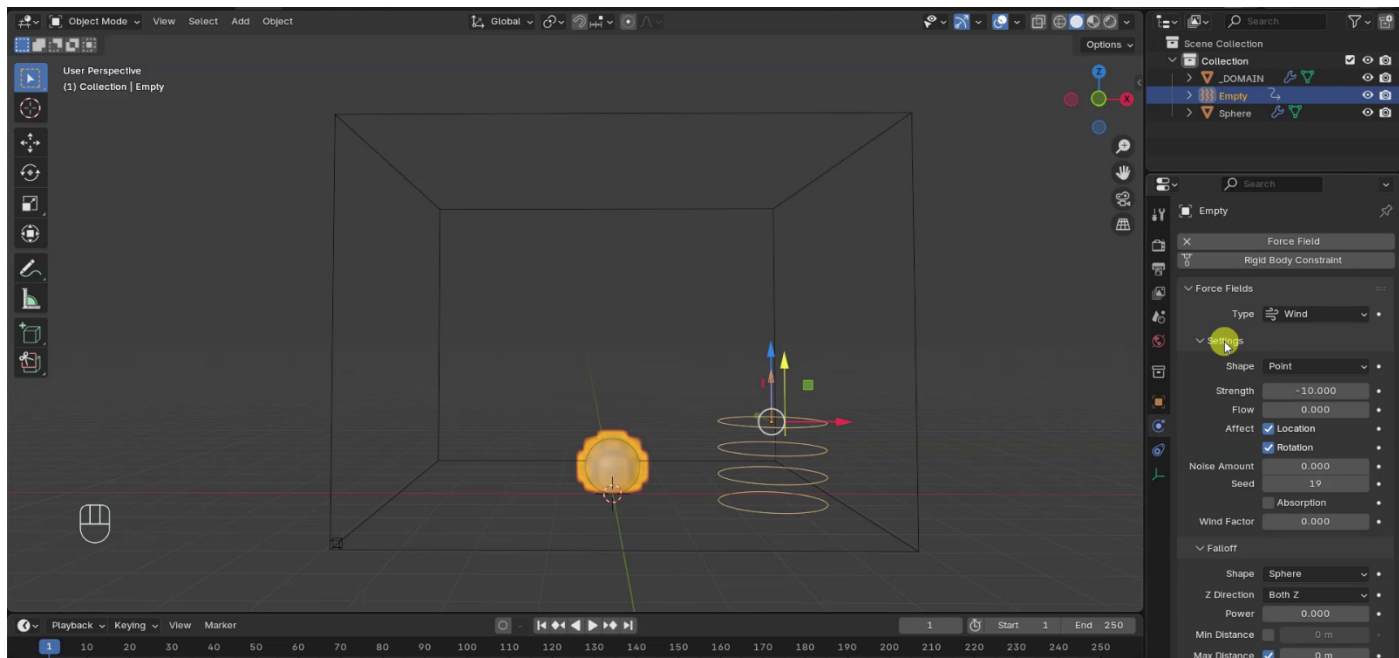
Strength는 개별 Force Field의 강도를 조정할 수 있을 뿐만 아니라, 애니메이션을 통해 그 값을 시간에 따라 변화시킬 수도 있습니다. 하지만 Domain 튜토리얼에서 배운 것처럼, 씬 안의 모든 Force Fields 또는 특정 타입의 Force Fields에 대한 영향을 제어할 수도 있습니다. 이러한 설정은 Domain의 Field Weights 섹션에 있으며, 이 또한 키프레임을 가질 수 있어 하나 또는 여러 Force Field의 효과를 시간에 따라 애니메이션할 수 있습니다.

Force를 포함한 많은 Force Fields의 Strength 필드는 음수 값도 허용합니다. Force의 경우, 이는 Fire가 Force Field 쪽으로 끌려가도록 만듭니다. 화면에는 Empty의 움직임을 애니메이션하고 Force를 음수 값으로 설정한 예시를 보여드리고 있습니다. 보시는 것처럼 Fire와 Smoke를 유도하는 흥미로운 방법이 될 수 있습니다.

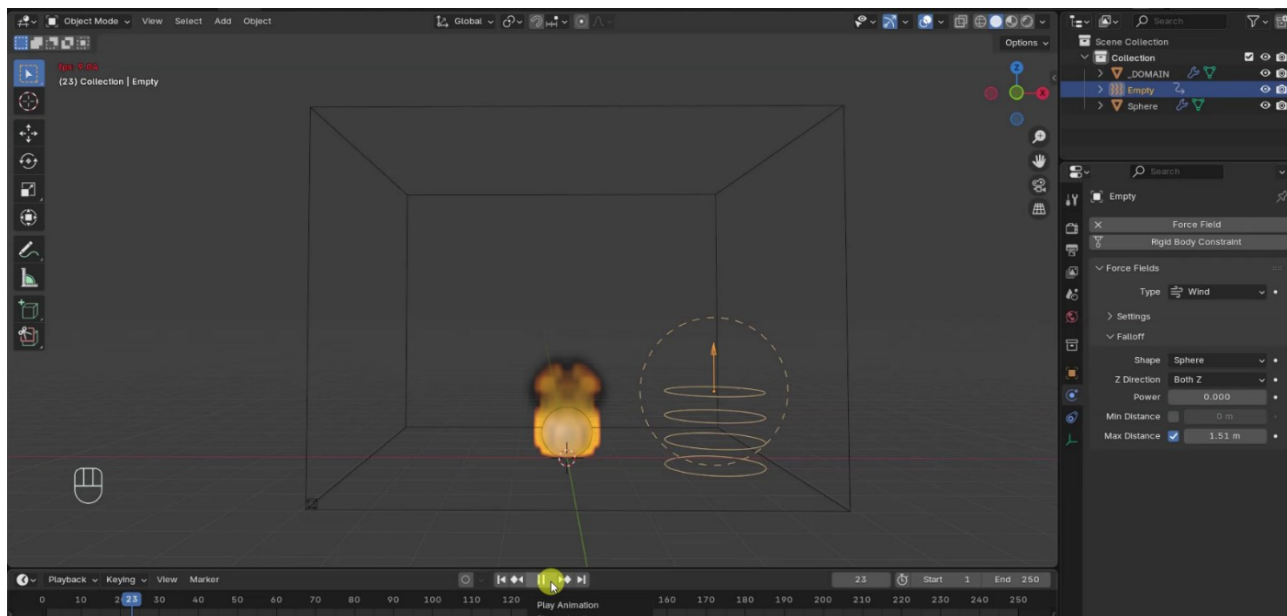
Blender에서 제공되는 모든 Force Field의 파라미터를 전부 다루려면 별도의 미니 시리즈가 필요할 정도라서 여기서는 다루지 않겠습니다. 하지만 살펴볼 만한 중요한 섹션이 하나 있는데, 바로 Falloff입니다. 이 옵션은 Force Field의 영향을 미치는 범위를 설정할 수 있습니다.



현재 Force Field는 감쇠 없이 전체 가상 공간에 영향을 미칩니다. 더 명확한 예시를 위해, Force를 Wind로 변경하고 Strength를 음수로 둔 상태에서 확인해 보겠습니다. Wind는 z축에서만 작동하고 Inflow 오브젝트와는 꽤 떨어져 있음에도 불구하고, 여전히 영향을 주는 것을 볼 수 있습니다.



Force Field의 영향 범위를 제한하려면 Falloff 섹션의 Max Distance 필드를 사용할 수 있습니다. 기본적으로 이 파라미터는 비활성화되어 있습니다. 활성화하면 Force Field의 범위가 3D Viewport에서 원으로 표시됩니다. 다른 많은 파라미터와 마찬가지로 Max Distance에도 키프레임을 넣을 수 있어, Force Field의 효과를 애니메이션에서 제어하는 또 다른 방법이 됩니다. 이 경우 Wind는 Strength를 음수로 두었기 때문에 불꽃과 Smoke를 끌어당깁니다.



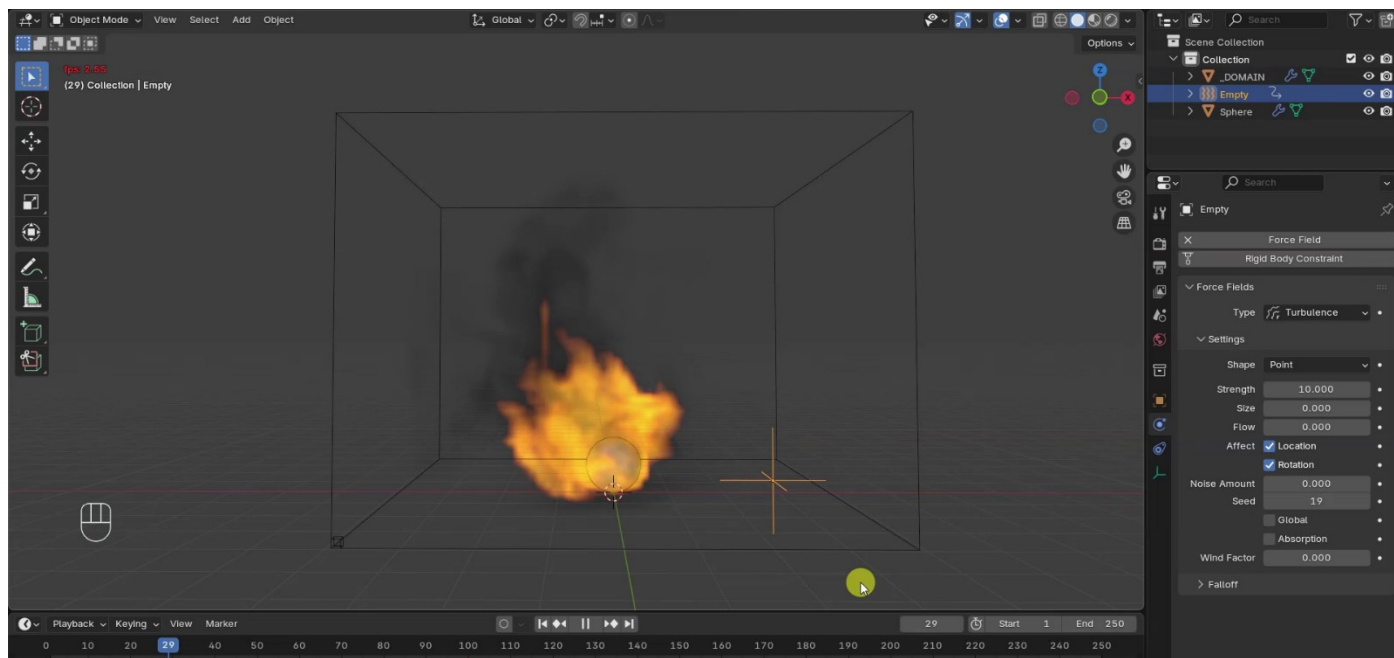
이제 Max Distance를 비활성화하고, Force Field를 원래 방향으로 되돌린 뒤 Strength를 양수 값으로 설정해 다른 Force Field 몇 가지를 빠르게 살펴보겠습니다.

불꽃에 움직임을 추가하는 데 매우 유용한 Force Field는 바로 Turbulence입니다. 이 Force Field가 만들어내는 교란은 매우 강력해서 Smoke가 불꽃을 가려버릴 정도입니다. 그래서 Domain에서 Dissolve 옵션을 사용해 이를 줄였습니다.

Turbulence는 Noise 파라미터를 높여 더 무작위적으로 만들 수 있습니다. 이 Noise는 사실 특정 패턴을 따르는 의사 난수이며, Noise Seed 파라미터를 통해 패턴을 선택할 수 있습니다. 불꽃이 Vorticity를 조정하거나 Texture를 추가한 뒤에도 여전히 너무 균일해 보인다면, 작은 Turbulence Force Field가 바로 필요한 해결책일 수 있습니다.

마지막으로 Curve Guide Force Field를 살펴보겠습니다. 이는 시뮬레이션 요소가 경로를 따라가도록 할 수 있습니다. 이 예시를 위해 저는 초기 씬을 처음부터 다시 만들고, 새 파일에 프로젝트를 저장한 후 Domain을 위한

새로운 캐시 폴더를 설정했습니다. 이렇게 하면 씬에 배치된 Domain과 Inflow 오브젝트의 초기 설정은 새로 생성할 때의 기본값이 됩니다. 다만 Smoke가 시야를 너무 가리지 않도록 Dissolve를 활성화하고, 기본값보다 많은 프레임 수로 설정했습니다. 또한 Inflow 오브젝트를 옆으로 옮겨 경로 곡선을 위한 공간을 확보했습니다.



Bezier Curve를 추가하고 위치를 조정해 Curve Guide Force Field를 부여했지만, 결과는 상당히 실망스러웠습니다.

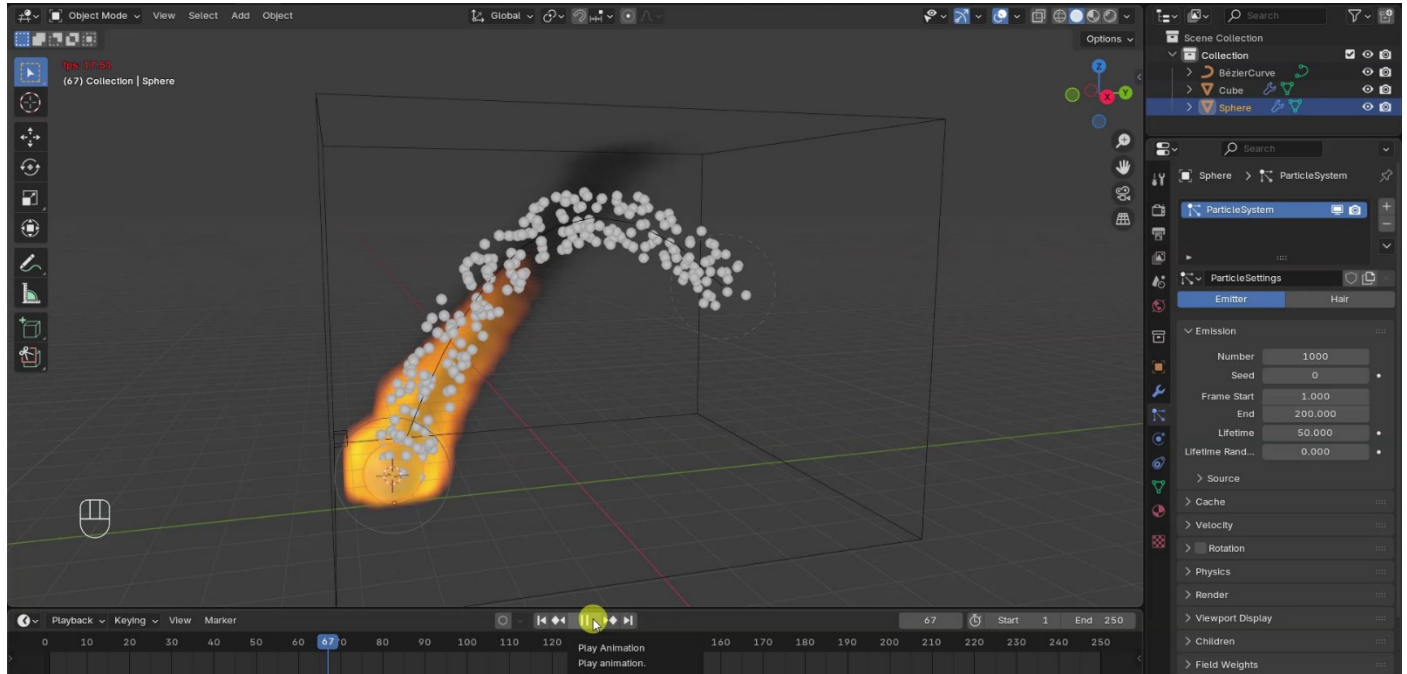
Curve와 Curve Guide Modifier에 익숙하다면, 경로의 시작점에 Curve의 Origin을 설정하는 것이 좋은 습관이라는 것과, 무엇보다 Curve가 올바른 방향을 가져야 한다는 것을 알고 있을 것입니다. 이는 Edit Mode에서 Curve Overlays로 확인할 수 있으며, 방향이 잘못되었을 경우 Switch Direction으로 수정할 수 있습니다. 시뮬레이션을 다시 시작하면 상황이 약간 나아지지만 여전히 완벽하진 않습니다.

저는 Domain에서 Gravity를 비활성화하거나 Field Weights에서 Curve Guide 값을 높이는 등 여러 설정을 시도했습니다. 물론 Vorticity와 Reaction Speed를 낮추거나 Resolution을 높이는 등의 방법도 시도했습니다.

무언가 변화는 있었지만 충분하지 않았습니다.

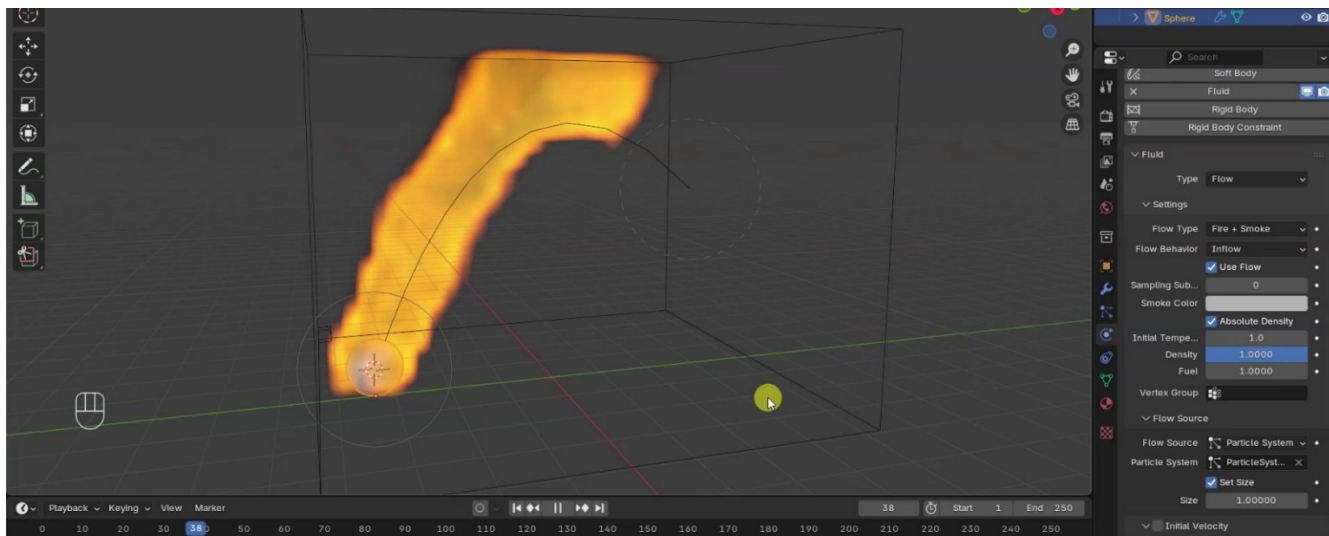
그래서 제가 원하는 결과를 얻기 위해 보통 사용하는 방법을 보여드리겠습니다.

Inflow가 있는 오브젝트에 Emitter 타입의 Particle System을 추가하고, 파티클이 제대로 방출되도록 기본 파라미터를 설정합니다. 파티클은 이미 Force Field의 범위 안에 있기 때문에 곡선을 바로 따라가는 것을 볼 수 있습니다.

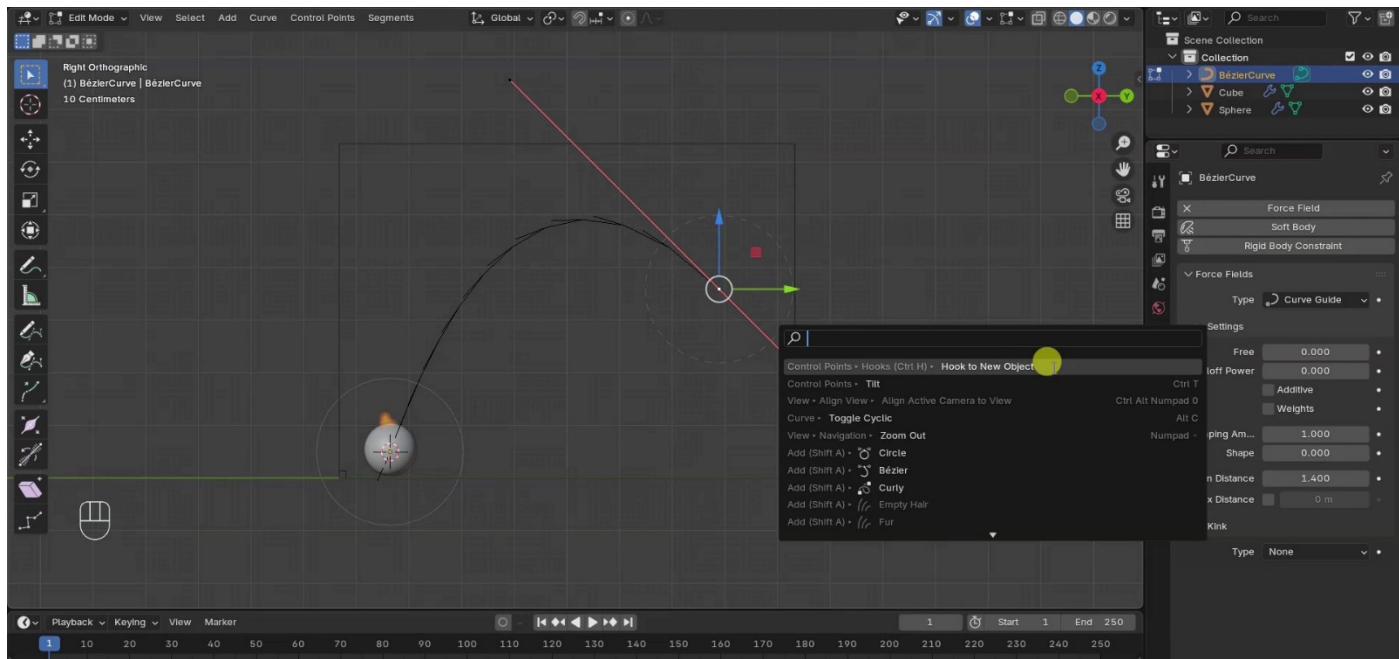


Force Field의 범위는 해당 패널에서 Minimum Distance 파라미터로 설정할 수 있습니다. 여기에서 Clumping을 1로 설정하면 모든 파티클이 곡선의 끝점으로 모이게 됩니다. 이 시점에서 파티클은 렌더링되지 않아야 하므로, Emitter의 Particle System에서 Render를 None으로 설정한 뒤 다시 Inflow 패널로 돌아갑니다.

이 패널에서 Flow Source를 Mesh에서 Particle System으로 바꾸고, 방금 만든 파티클 시스템을 지정합니다. 이렇게 하면 경로를 따라 움직이는 파티클에서 Fire와 Smoke가 방출됩니다. 여기서부터는 Fuel, Vorticity 등의 파라미터를 조정해 원하는 불꽃 형태를 만들 수 있습니다.

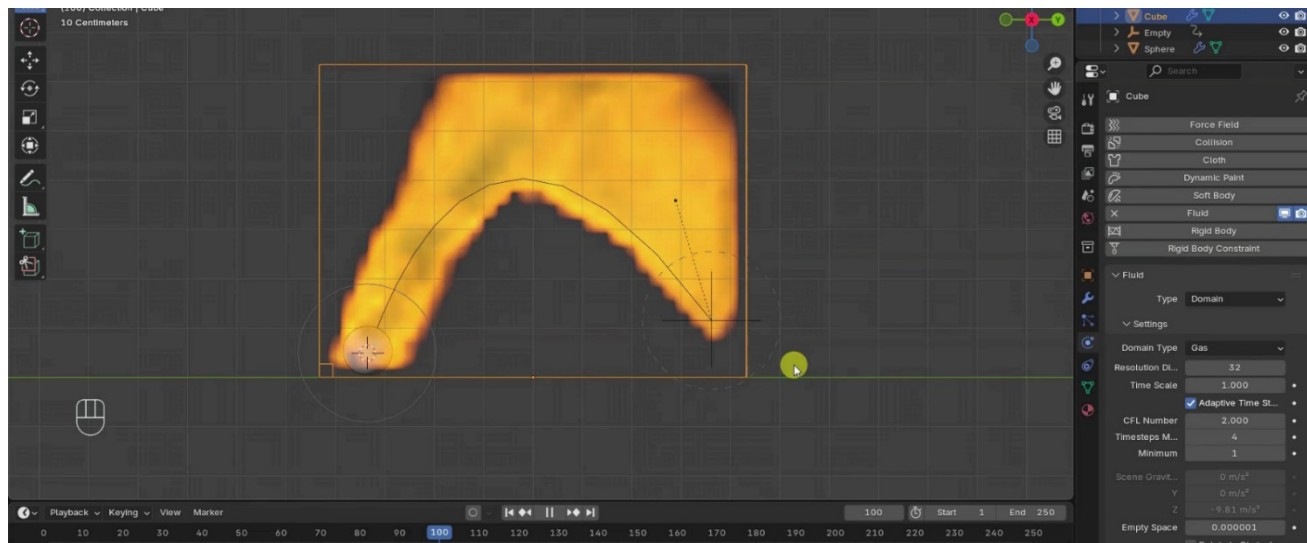


튜토리얼을 마무리하기 전에, 곡선의 컨트롤 포인트를 외부 오브젝트에 연결해 곡선 형태를 쉽게 애니메이션할 수 있는 Hooking이라는 기법을 보여드리겠습니다. 먼저 곡선을 선택하고 Edit Mode로 들어가 마지막 컨트롤 포인트를 선택합니다. 그런 다음 Hook 연산자를 사용해 Hook To New Object 옵션을 선택하면, 이 목적을 위한 Empty가 생성됩니다.

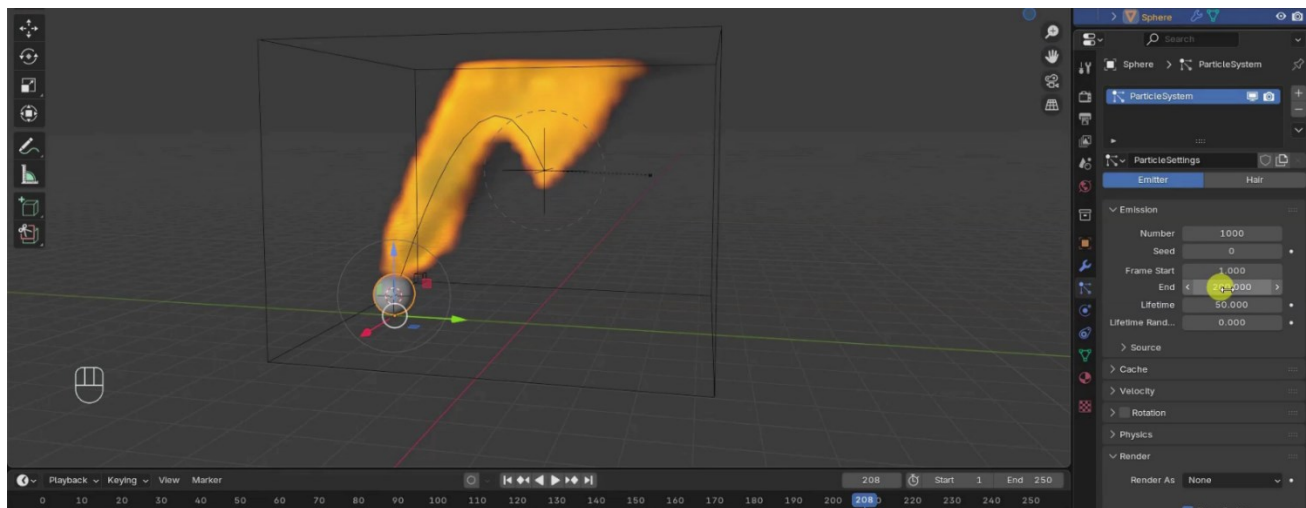


Object Mode에서는 이 Empty를 이동하거나 애니메이션할 수 있으며, 그에 따라 곡선과 불꽃도 변형됩니다.

물론 다른 컨트롤 포인트에도 더 많은 Empty를 추가해 동일한 기법을 사용하면 더 복잡한 애니메이션을 만들 수 있습니다.



이미 눈치채셨겠지만, 어느 시점에서 파티클 방출이 멈춥니다. 이 파라미터는 Frame End라고 하며 Particle System 패널에 있습니다. 같은 패널의 Lifetime 파라미터는 파티클이 곡선을 따라 이동하는 속도를 제어합니다.



마무리하기 전에 몇 가지 더 고려해야 할 점이 있습니다. Emitter 오브젝트의 크기는 방출 면적의 크기를 결정합니다. 따라서 파티클을 더 좁은 곡선을 따라 집중시키고 싶다면 이 오브젝트의 크기를 조정하세요. 파티클 수는 불꽃 곡선이 얼마나 잘게 나뉘어 보이는지에 영향을 주므로, 불꽃이 더 부드럽고 연속적으로 보이게 하려면 수를 늘려야 할 수도 있습니다.

이와 관련해, 다음 튜토리얼에서 다른 Motion Blur를 사용하는 것도 도움이 될 수 있습니다.

Blender 4.5 Fire Simulations 기본 | 10/10: Motion Blur, Filmic, Compositing, Eevee

<https://youtu.be/FByVQ-X7Z2c>

안녕하세요! 이번 Blender 4.5의 Fluid 시뮬레이터를 이용한 Fire와 Smoke 시리즈의 열 번째이자 마지막 에피소드에서는, 전체 튜토리얼보다는 렌더링 중과 후에 Fire의 외형을 수정하는 팁과 방법을 공유하겠습니다. Motion Blur, Color Management, Node Compositing을 사용한 후반 작업뿐만 아니라, Eevee에서 Fire와 Smoke를 렌더링할 때의 설정도 함께 다루겠습니다.

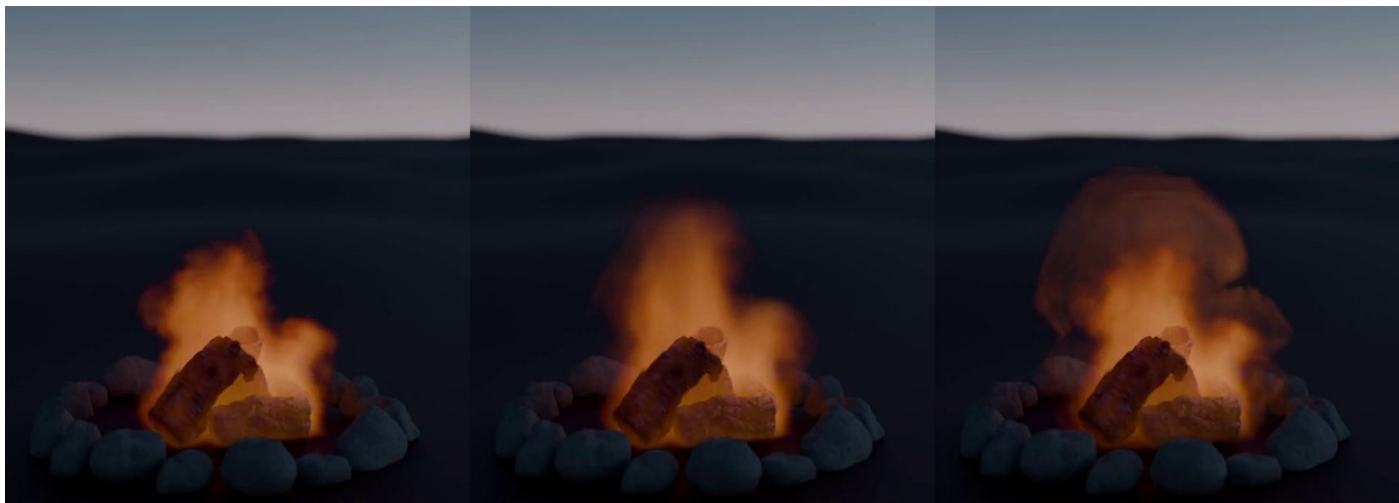
후반 작업이 아닌, 렌더링 자체에 영향을 주는 설정부터 시작하겠습니다. 이는 Render 탭에서 선택할 수 있는 Color Management입니다. 특히 Filmic 모드는 더 채도가 높은 불꽃을 만들어 주는 경향이 있습니다. High Contrast를 선택하면 불꽃이 더 강렬하게 보입니다.

화면에서는 비교 예시를 보여드리고 있습니다. 왼쪽은 AgX로 렌더링한 것이고, 오른쪽은 Filmic과 High Contrast로 렌더링한 것입니다. Color Management는 Fire뿐 아니라 이미지 전체에 영향을 주는데, 이는 하단의 돌을 보면 확인할 수 있습니다. 잠시 후 Compositing에서 불꽃만 따로 조정하는 방법을 살펴보겠습니다.

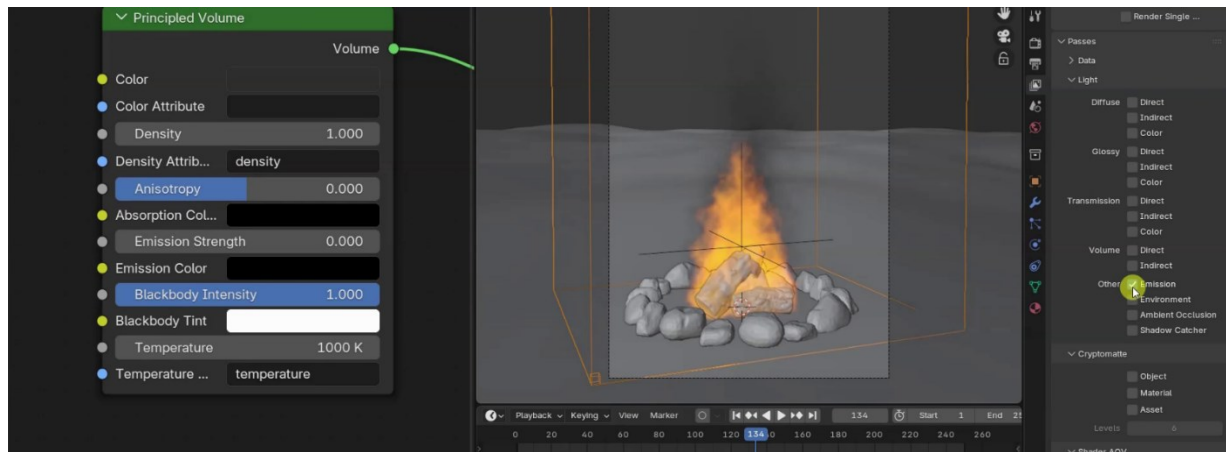


Render 탭에서 활성화할 수 있는 또 다른 옵션은 Motion Blur입니다. 이는 불꽃이 프레임 사이에서 절대 완전히 정지하지 않기 때문에 이러한 시뮬레이션을 렌더링할 때 매우 유용합니다. 이전 에피소드에서 Domain에는 Motion Blurring을 위한 Velocity Scale과 관련된 필드가 있다고 말씀드렸지만, Motion Blur는 Render 탭의 렌더링 레벨에서도 활성화해야 합니다. 이 탭의 Shutter 파라미터는 현재 프레임 전후 몇 프레임이 고려되는지를 정의합니다. 값이 높을수록 Motion Blur가 강해집니다.

화면에서는 Motion Blur를 비활성화한 렌더, Motion Blur를 활성화하고 Domain의 Velocity Scale을 1로 설정한 렌더, 그리고 Velocity Scale을 10으로 설정해 차이를 강조한 렌더를 비교해 보여드리고 있습니다. 안타깝게도 Shutter나 Velocity Scale의 최적값을 찾는 마법 같은 공식은 없으며, 현재 프로젝트의 요구에 따라 달라집니다. 따라서 테스트가 필요합니다. Motion Blur를 활성화하고 Shutter 또는 Velocity Scale 값을 높이면 렌더링 시간이 상당히 늘어난다는 점을 유념하세요.

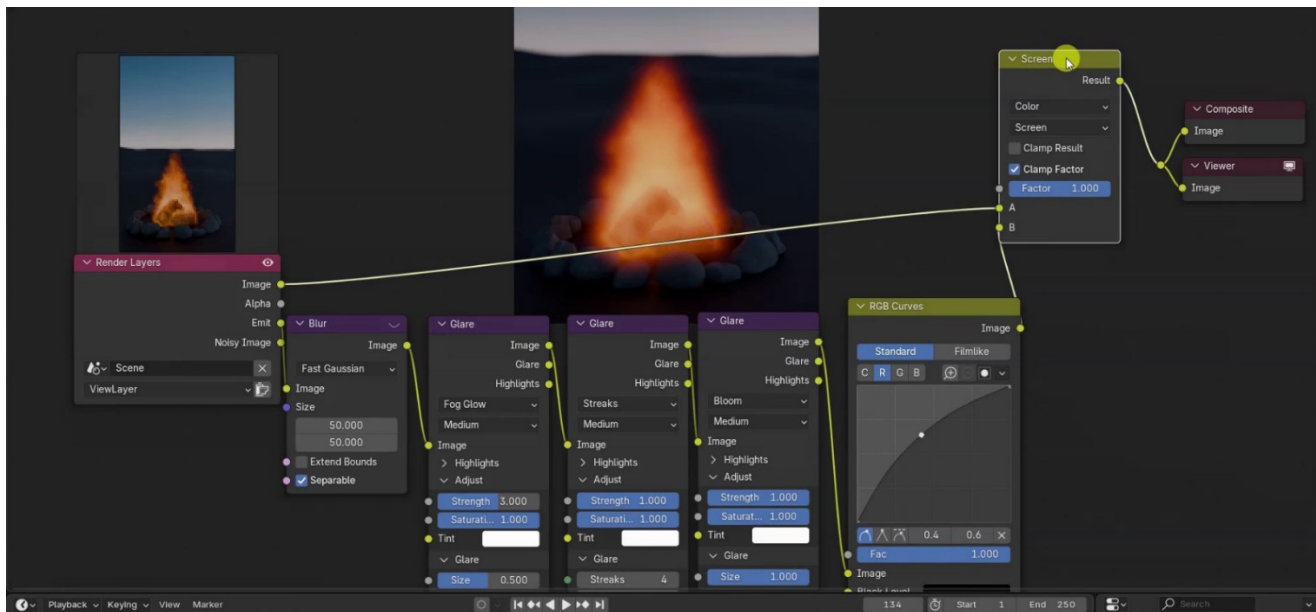


어떤 렌더링에서도 최종 결과는 후반 작업에서의 Node Compositing에 크게 의존합니다. Fire도 예외는 아니며, 이 경우 Emission Render Pass를 활용할 수도 있습니다. 이는 Emission이나 Volumetric Materials 같은 빛을 방출하는 소스의 픽셀을 합성용으로 분리할 수 있게 해줍니다. 따라서 테스트 렌더를 실행하기 전에 이 Render Pass를 전용 패널에서 활성화한 뒤 Compositing Nodes 에디터로 이동했습니다.



Emission 정보는 검은 배경 위의 컬러 불꽃을 제공하며, 이를 사용해 다양한 Compositing 노드를 적용해 Blur, Glow, Contrast, 심지어 Color까지 개선할 수 있습니다. 이 작업의 결과는 다시 원본 렌더에 적용할 수 있습니다. Blender는 이를 위한 여러 가지 Blending 모드와 Factor 옵션을 제공합니다. 예를 들어, 저는 불꽃을 따로

처리하여 Blur를 적용하고 여러 가지 Glare 효과를 추가한 뒤, 결과를 Mix Color 노드를 Screen 모드로 설정해 원본 렌더에 적용했습니다. 또한 RGB Curves 노드의 C 채널을 사용해 밝기를 빠르게 조정하고, 같은 노드에서 Red 채널만 수정해 보기도 했습니다. 최종 결과가 다소 강하게 느껴질 수도 있지만, 필요에 따라 Factor 값을 조정할 수 있습니다. 또한 Mix Color 노드를 선택하고 M 키를 눌러 Mute 기능을 사용하면 전체 세팅을 활성화하거나 비활성화할 수 있습니다.



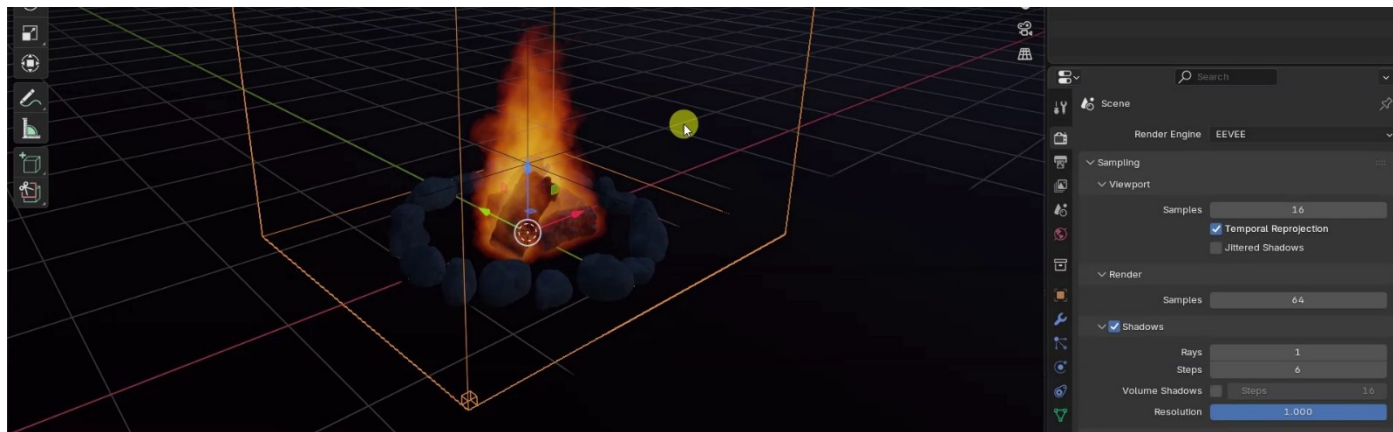
Eevee 렌더링 설정 팁으로 넘어가기 전에, 지금까지 다룬 옵션들을 최종적으로 비교해 보겠습니다. 왼쪽은 Color Management를 AgX로 설정하고 Motion Blur와 Compositing을 적용하지 않은 렌더입니다. 오른쪽은 Filmic을 High Contrast로 설정하고, Motion Blur에서 Shutter를 1로, Velocity Scale을 10으로 설정한 뒤, 앞에서 설명한 노드 세팅으로 약간의 후반 작업을 한 렌더입니다.



자, 이제 Eevee의 렌더링 설정으로 넘어가겠습니다. 지금까지는 항상 Cycles를 사용해 렌더링했지만, Eevee는 실시간 렌더링이 핵심 기능이기에 때문에 몇 가지 필요한 차이점이 있습니다.

Eevee에서는 Fire와 Smoke 시뮬레이션이 썬의 슬라이스로 표현되고, 이 슬라이스들이 반투명 시트처럼 겹쳐집니다. 여기서 설정해야 하는 파라미터는 관찰자의 위치에 대한 슬라이스의 시작과 끝 지점입니다.

최종 결과의 품질을 제어하는 설정은 주로 Render 탭, 특히 Volumes 섹션에 있습니다.

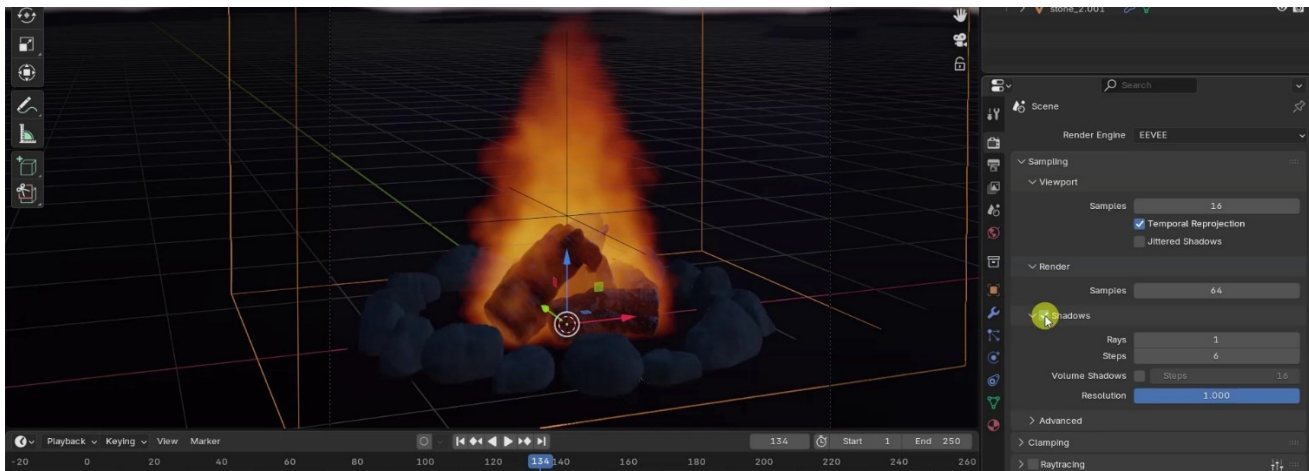
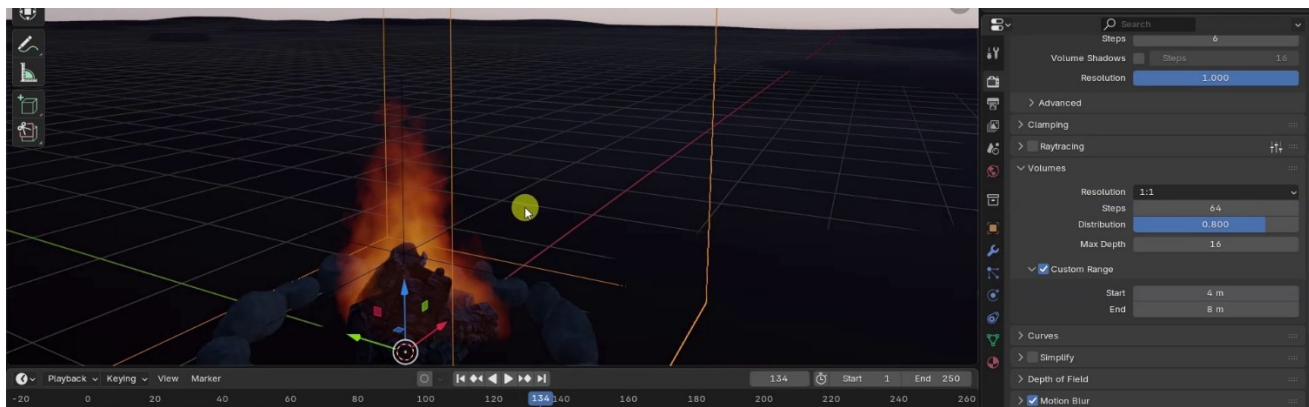


이 섹션의 첫 번째 파라미터는 Resolution입니다. 직관적으로, 이 값이 1:1 비율에 가까울수록 품질이 더 높아집니다. 비율이 낮아지면 품질은 떨어지지만 렌더링 속도는 빨라집니다.

앞서 언급했듯이, 뷰잉 볼륨은 Steps라고 불리는 슬라이스로 나뉩니다. Steps 파라미터는 기본적으로 활성화된 Custom Range Start와 End 옵션과 함께 작동합니다. Start 지점부터 End 지점까지, 즉 관찰자의 위치에서 전방으로 나아가는 공간이 Steps에 지정된 수만큼 슬라이스로 나뉩니다. 각 슬라이스는 렌더링되고, 결합된 결과가 최종 이미지를 만듭니다. 기본값인 100미터는 불필요하므로, Start와 End 값을 시뮬레이션 Domain에 맞게 줄여 해당 공간 안에 Steps를 집중시키면 더 나은 결과를 얻을 수 있습니다.

이 섹션에서 이해해야 할 가장 중요한 파라미터는 바로 이것들입니다. 특히, 씬 안에서 움직일 때 시뮬레이션이 사라지는 이유를 설명해 줍니다!

그림자와 그 품질에 대해서는 Render 탭의 Sampling Shadows 섹션에서 기본적으로 비활성화된 Volume Shadows 체크박스를 찾을 수 있습니다. 이를 활성화하면 Steps 값을 설정할 수 있으며, 당연히 이 값은 결과의 품질과 연결됩니다.



이렇게 해서 Blender 4.5에서 Fire와 Smoke를 위한 Fluid 시뮬레이션 기초 시리즈의 마지막 튜토리얼이 마무리되었습니다! 이 시리즈가 여러분께 유용했기를 바랍니다. 저에게 감사 인사를 하고 싶으시다면, 채널을 구독하고 영상에 좋아요를 눌러주시는 것이 가장 좋은 방법입니다. 감사합니다, 곧 다시 만나요!