Bifrost Extension에 오신 것을 환영합니다.

Bifrost는 3D 아티스트와 TD가 새로운 시각적 프로그래밍 환경을 이용해 마야에서 복잡한 효과를 쉽고 빠르게 생성할 수 있게 해주는 플랫폼입니다. TD는 Bifrost Graph Editor에서 맞춤형 그래프를 생성하여 이를 스튜디오 아티스트에 구성한 다음 여러 가지 쇼나 장면 및 사진에 사용할 수 있습니다. 아티스트 역시 즉시 사용할 수 있는 다양한 그래프를 이용하여 불, 연기, 폭발, 모래, 눈과 같은 효과를 즉시 생성할 수 있습니다.

최신 버전 내려받기



🖉 시작하기
다운로드 센터
설치 가이드
릴리즈 노트
튜토리얼

🚺 기본 정보
냄플 그래프 로드
그래프 편집기
공기 및 연소

국 공유하기 커뮤니티 허브 토론 포럼

Bifrost Extension 릴리즈 노트

Bifrost 플러그인을 다운로드해주셔서 감사합니다.

각 Bifrost 버전의 릴리즈 노트(아래 링크)에서는 해결된 문제, 발견된 제한사항과 해결방법 외에도 각 릴리즈에 대한 중요한 정보를 설명합니다.

설치 도움말

Bifrost 최신 플러그인은 AREA에서 상시 다운로드할 수 있습니다. 자세한 설치 방법은 Bifrost Extension for Maya 설치를 참고하십시오.

🖋 참고:

Maya 2019.2 Update를 설치하면, Maya 설치관리자에 Bifrost Extension for Maya가 기본적으로 포함되어 있습니다. Maya의 모든 이전 버전은 AREA(https://area.autodesk.com/bifrost/downloads/)에서 해당 설치관리자를 다운로드하면 됩니다.

추가 자료

- Community Hub on the AREA 여기서는 포럼을 통해 지원 및 문제해결 방법뿐만 아니라 튜토리얼과 기타 학습 자료까지 찾아볼 수 있습니다.
- 최소 시스템 사양 Bifrost를 설치하는 데 필요한 최소 시스템 사양은 Maya와 같습니다.
- Tutorials on AREA and YouTube Bifrost 온라인 도움말은 Bifrost 최신 플러그인 버전을 지원할 수 있도록 지속적으로 업데이트되며 유지관리되고 있습니다.
- Maya Help에는 각 Maya 릴리즈와 함께 들어 있는 Bifrost 버전의 설명서도 포함되어 있습니다. (예를 들어, Maya 2019.2 설명서는 업데이트되어 Bifrost 2.0.1.0에 대한 정보를 포함하고 있습니다.)

아래에서 각 Bifrost 플러그인 버전에 대한 릴리즈 상세 노트를 확인해 보십시오.

이 섹션의 페이지들

- Bifrost 2.0.1.1 릴리즈 노트
- Bifrost 2.0.1.0 릴리즈 노트

Bifrost 2.0.1.1 릴리즈 노트

절차식 그래프로 만든 Bifrost 콘텐츠를 렌더링할 수 있는 Arnold for Maya의 최소 버전은 3.2.2입니다.

Arnold for Maya의 최신 설치관리자를 다운로드 하려면, Arnold for Maya Download Area를 방문하십시오.

해결된 문제

잘못된 데이터를 write_OpenVDB에 연결할 때 충돌 BIFROST-3022

캡슐화: 컴파운드 생성 시 충돌(이름에 숫자가 있는 노드 문제 해결) BIFROST-2963

오류: 참조된 컴파운드나 그래프 중 컴파운드가 아닌 오류 상태의 노드를 포함하고 있거나 레거시 노드가 누락된 것을 가져올 수 없습니다. BIFROST-2655

즉시 값: 컴파운드를 가져올 때 입력 포트의 즉시 값들이 삭제되어 포트 유형을 변경하면 데이터 손실 또는 그래프 손상이 발생합니다. BIFROST-2933

즉시 값: 일부 노드를 캡슐화할 때 즉시 값들이 사라집니다. BIFROST-2536

즉시 값: 참조된 컴파운드를 덮어쓰면 즉시 값들이 삭제됩니다. BIFROST-3026

상호작용: Parameters Editor와 Info 탭에서 활성 상태가 불명확합니다. BIFROST-2820

상호작용: 탭을 복제하고 장면을 열 때 충돌 발생합니다. BIFROST-2708

Maya 플러그인: vnn 플러그인을 처음 로드할 때 Script Editor에서 Maya 2019.2 Update macOS 및 Linux: VNN / libaic 오류가 발생합니다. *BIFROST-313*8

게시: 최근에 덮어쓰기한컴파운드를 참조하는 컴파운드를 참조하면 충돌이 발생합니다. BIFROST-3013

발견된 문제점과 제약사항

BIFROST-3144 Aero: 파란색 불꽃 타일 인공물

파란색 불꽃 안에 타일 또는 복셀 인공물이 나타날 때가 있습니다(연소 속도).

BIFROST-2776 Aero: 기본 적응도 옵션이 basic_aero에서 적응도를 이용하지 않을 때보다 더 느립니다.

특정 상황에서 자동 적응도를 설정하면 시뮬레이션이 느려질 수 있습니다.

BIFROST-2982 Aero: Influence 필드에서 공기 객체의 복사본을 방지합니다.

Influence 필드를 이용할 때 발생하는 공기 객체 복사본이 성능을 떨어뜨립니다.

BIFROST-3150 Aero: 그을음 변화 문제

연소 시뮬레이션에서는 경우에 따라 변화가 너무 심한 그을음이 만들어지기도 합니다.

BIFROST-2823 Aero: boost_detail_with_points가 연소를 없앱니다.

연소 시뮬레이션에서 "boost detail with points"를 활성화하면 그을음 방출이 사라집니다.

BIFROST-2903 Aero: 장면의 스케일과 함께 경계가 정해지는 적응도 문제

장면 단위(m)와 함께 1 이외로 경계가 정해지는 적응도를 이용하면 경계의 크기를 조절할 수 없습니다.

BIFROST-822 Aero: 한 볼륨에서 방출 시 이미터 경계 밖 일부 구간에서 방출이 발생합니다.

경우에 따라, 공기 소스 지오메트리 밖의 영역에서 유령 방출이 생길 수 있습니다.

BIFROST-1498 Attraction_Repulsion은 이미터의 복셀화에 따라 파티클에 영향을 미칩니다.

Attraction-repulsion influence는 소스 geo 복셀화에 따라 파티클에 영향을 미칩니다.

BIFROST-2827 Compound Publishing: 시그니처를 변경해 다시 게시한 후, 참조된 컴파운드 복사본의 상태가 나빠집니다.

컴파운드가 참조되는 바로 그 장면에서 시그니처를 변경하고 컴파운드를 다시 쓴 후에는 그 장면을 저장하지 말아야 합니다. 그렇지 않으면, 참조된 컴파운드가 손상된 상태로 저장됩니다. 해당 장면을 저장하지 않고 다시 로드하면 참조가 올바르게 다시 로드됩니다.

BIFROST-2109 Compound Publishing: 기본 값을 갖는 자동 포트를 이용할 경우 컴파운드를 게시할 수 없습니다.

BIFROST-2765 Compound Publishing: Linux 플랫폼에서, 컴파운드를 사용자 위치에 게시할 때 .json 확장자가 파일 이름에 자동으로 붙지 않습니다.

해결방법: 게시할 때 파일 이름에 확장자 .json을 붙여야 합니다.

BIFROST-2767 Compound Publishing: 시그니처를 변경할 경우 그리고 자동 포트가 없는 경우 다시 게시하는 것이 불가능합니다.

해결방법: 컴파운드를 불러와, 디스크에서 json 파일을 삭제하고, Maya를 다시 시작해 다시 게시합니다.

BIFROST-2532 Compound publishing: 오류가 있을 때 네임스페이스 필드에 입력이 불가하고 키를 누를 때마다 포커스가 사라집니다.

해결방법: 게시 편집기에서 컴파운드의 이름을 임시로 변경해 오류를 제거하고, 네임스페이스를 편집한 다음, 이름을 원래대로 다시 변경합니다.

BIFROST-2769 Compound publishing: 컴파운드 게시는 대소문자를 구분하지만 탭(Tab) 생성은 대소문자를 구분하지 않습니다.

My_Compound와 my_compound처럼 대소문자만 다른 컴파운드를 동시에 만들지 않아야 합니다.

BIFROST-2279 Encapsulation: 자동 루핑 되는 노드 1개로 컴파운드를 만들면 그래프 반응이 바뀝니다.

BIFROST-2581 Encapsulation: 가져온 컴파운드를 다수의 오버로드로 수정할 때 예기치 않은 결과가 발생합니다.

get_geo_property와 같은 오버로드된 컴파운드를 가져온 다음 포트를 추가하거나 제거하거나 이름을

변경하거나 그룹화할 경우, 컴파운드가 손상되거나 충돌이 발생할 수 있습니다.

BIFROST-2542 Errors: 침묵의 컴파일 오류(예: 프로모션이 없음)

가끔 컴파일 오류가 발생할 때도 있는데 그래프에는 나타나지 않습니다.

그래프가 평가를 하지 않는 경우가 발생하면 Output Window에서 컴파일 오류가 없는지 확인해야 하는데, 오류가 있다면 오류 상태의 노드를 알려주고 문제 해결을 도와줄 것입니다.

BIFROST-3024 Errors: write_OpenVDB와 같은 쓰기 노드를 이용할 경우, 쓰기에 실패할 때에도 노드의 성공 포트가 항상 1을 반환합니다.

해결방법: 쓰기 작업을 할 때 폴더와 파일의 이름이 올바른지 확인합니다.

BIFROST-3121 Graph Publishing – Maya 파일 안에 있는 그래프들은 해당 문서의 링크를 절대 경로로 저장합니다.

따라서, 그 문서가 그래프의 예시 장면에 나타나지 않습니다.

BIFROST-3035 Graph Publishing: 게시 UI에 기본 브라우저 카테고리가 없습니다.

게시할 때 카테고리 안으로 들어가야 합니다.

BIFROST-2887 Graph publishing: 게시되는 그래프에 Tab 키가 나타나지 않습니다.

BIFROST-3154 속도 특성이 없을 때 볼륨으로부터 방출되는 파티클의 위치가 잘못 설정됩니다.

속도 데이터 없이 볼륨에서 파티클을 방출시키면 잘못된 결과가 나타납니다.

BIFROST-1581 Influences: 높이를 가진 지오메트리를 이용할 때 방사 필드에 예기치 않은 결과가 발생합니다.

BIFROST-3028 각 면마다 음영이 있는 입력 메쉬에는 머티리얼을 할당할 수 없고, 가장 먼저 찾아낸 컴파운드가 사용되지 않습니다.

현재, 각 면마다 음영이 있는 입력 메쉬는 지원되지 않습니다.

BIFROST-2882 Input: 입력 노드에서 Normals 옵션을 변경하면 Maya 장면과의 연결이 끊어집니다.

Input by Path 노드 Parameter Editor에서 Normals per point와 Normals per face-vertex 옵션을 선택하거나 선택 해제하면 경로가 재평가되어 Maya 객체와의 연결이 끊길 수도 있습니다.

이 문제를 해결하려면, 해당 메쉬를 그래프로 다시 끌어다 놓거나 Parameter Editor에서 새로운 경로를 입력하는 방식으로 입력 노드와의 연결을 재설정하면 됩니다.

BIFROST-2722 Input: 가져온 그래프에서 경로별 입력이 끊어집니다.

해결방법: Maya 지오메트리를 입력으로 이용하는 그래프는 게시하지 마십시오.

그 대신, create_mesh_cube나 create_mesh_plane처럼 사용자가 나중에 자신이 원하는 것으로 직접 바꿀 수 있는 자리 표시자 지오메트리를 만드십시오.

BIFROST-928 Input: 메쉬를 그래프의 더 심층 레벨로 끌어다 놓을 때 쓸모 없는 입력 노드가 나타납니다.

BIFROST-3237 Inputs: 한 그룹에 속해 있는 네임스페이스가 있는 메쉬를 네임스페이스 없이 끌어다 놓으면 올바른 경로가 생성되지 않고 기본적으로 그 메쉬를 그래프로 가져올 수도 없습니다.

해결방법: 입력 노드를 통해 메쉬에 올바른 경로를 입력합니다. 여기서는 "긴" Maya 이름도 사용 가능합니다.

BIFROST-3164 Inputs: 머티리얼 참조의 경우, 머티리얼 이름 변경으로 인한 업데이트 문제가 발생합니다.

BIFROST-2889 Installers: SDK: 설치 관리자에 BifrostFluids SDK가 없습니다.

BIFROST-3198 Interaction: 와치포인트가 들어있는 그래프의 경우 새로운 Maya 장면을 생성할 때 이따금씩 충돌이 발생합니다.

BIFROST-2799 Interaction: 작업공간을 전환할 때 Bifrost Graph Editor가 깜빡입니다.

Graph Editor가 도킹된 상태에서 Ctrl+스페이스바를 사용할 때에도 Bifrost Graph Editor가 깜빡일 수 있습니다. (BIFROST-2861)

BIFROST-3070 Interaction: Shelf 아이콘을 클릭할 때 BifrostGraph 플러그인이 자동으로 로드됩니다.

Bifrost Graph Editor 또는 Shelf에 있는 Bifrost Browser 버튼을 클릭할 때 이 오류가 발생하는 경우:

// Error: line 1: Cannot find procedure "openBifrostGraphEditorFromSelection".

bifrostGraph 플러그인을 로드해야 합니다. 그러기 위해서 Windows > Settings/Preferences > Plug-in Manager를 차례로 클릭합니다.

BIFROST-3021 Interaction: 옵션이 해제되어 있는데도 생성 중인 노드가 연결됩니다.

해결방법: Connect Nodes on Creation 옵션이 해제되어 있는 상태에서도 Tab 생성 노드가 노드들을 자동으로 연결할 때에는 Bifrost Graph Editor를 닫았다가 다시 여십시오.

BIFROST-3151 Interaction: 유형이 Any인 값 포트를 마우스 오른쪽 버튼으로 클릭할 때 충돌이 발생합니다. (예: set_property 노드에서)

유형이 Any인 포트를 마우스 오른쪽 버튼으로 클릭해 값 노드를 생성하려고 하면 충돌이 발생합니다.

해결방법: Value 노드를 하나 만든 다음, 이것을 set_property에 연결합니다.

BIFROST-574 Interaction: 포커스 문제 – Parameter Editor에서 값을 편집할 때 컴파운드로 마우스를 가져가서 Enter를 누르면 그 컴파운드 안으로 들어가게 됩니다.

BIFROST-1802 Interaction: 노드나 컴파운드의 이름을 특수 문자로 바꾸면 이름 필드가 사라져 해당 노드의 이름을 더 이상 변경할 수 없습니다.

노드나 컴파운드의 이름을 바꿀 때, ! @ # \$ % ^ & () + = \/ > < . , " : ; ' ~ `와 같은 특수 문자나 ASCII 이외의 문자(예: 다른 언어의 문자)를 사용해서는 안 됩니다. 이런 문자를 사용할 경우 예기치 못한 반응이 나타날 수 있습니다.

BIFROST-3065 Interaction: 메뉴에서 생성된 입력 또는 출력이 스크린 밖의 안 좋은 위치나 기존 입력 위에 나타납니다.

그 대신, Tab을 이용해 노드를 만드십시오.

BIFROST-2194 Interaction: 컴파운드를 만들고 안으로 들어가면 출력 노드가 잘못된 위치에 나타납니다.

BIFROST-2428 Interface: Bifrost UI를 이용할 때 DPI 값이 커집니다.

BIFROST-3062 Malformed geometry: face_vertex 데이터 입력과 construct_mesh의 연결을 끊으면 Maya와 충돌합니다.

BIFROST-3115 Maya 2018: bifrostshellnode 플러그인을 로드하지 않거나 설치하지 않을 경우, Attribute Editor가 비정상적으로 나타나고 오류가 발생합니다.

이런 오류는 Maya 2018, 2018.1 Update, 2018.2 Update에서 발생할 수 있습니다.

이 문제를 해결하려면, bifrostshellnode 플러그인을 설치하고 로드해야 합니다.

BIFROST-3166 Maya Plug-in: "Delete unused nodes"가 머티리얼 참조를 삭제합니다.

BIFROST-2334 Maya Plug-in: Bifrost 플러그인이 완벽히 언로드하지 않아 충돌이 발생할 수 있습니다.

동일 Maya 세션 안에서 Bifrost 플러그인을 언로드했다가 다시 로드하면 불안정성이나 기타 문제가 발생할 수 있습니다.

BIFROST-2444 Maya Plug-in: 뷰포트에서 bifShapes를 선택할 수 없습니다.

BIFROST-2997 Maya Plug-in: 포트 유형을 array에서 single로 바꿀 때 Attribute Editor가 올바로 업데이트되지 않습니다.

BIFROST-1869 Maya Plug-in: 시뮬레이션 도중 ESC가 중단되기 전에 Maya가 별도의 프레임을 지연시킵니다.

BIFROST-3171 Maya Plug-in: 그래프를 포함하는 장면을 여러 번 참조할 때 한 개의 bifrost 그래프만 작동합니다.

BIFROST-2330 Maya Plug-in: 장면을 로드해도 mayapy에서 vnn 플러그인이 로드되지 않습니다.

Bifrost는 mayapy에서도 작동하지만 VNN 플러그인은 자동으로 로드되지 않기 때문에 VNN Python 명령이 효과가 없을 것입니다.

해결방법: mayapy에서 VNN 명령을 사용하려면, 먼저 loadPlugin 명령을 이용해 플러그인을 로드하십시오.

BIFROST-3073 Maya Plugin-in: bifrostGraph 플러그인이 없을 때 bifrostGraph 노드를 포함하는 장면을 로드하면 충돌이 발생합니다.

해결방법: Bifrost 노드를 포함하는 장면을 열기 전에 먼저 bifrostGraph 플러그인이 설치되어 있는지 확인합니다.

BIFROST-3243 BIFROST_LIB_CONFIG_FILES가 원격 네트워크 위치를 가리키는 경우 Maya가 매우 느리게 실행됩니다.

BIFROST-3152 Merge geometry는 첫 번째 객체를 이용해 병합 반응을 설정하나 그 대신 비어 있지 않은 첫 번째 객체의 스키마를 이용해야 하는 경우

따라서, 첫 번째 객체가 비어 있다면, merge geometry는 작동하지 않을 것입니다.

해결방법: 첫 번째 객체가 비어 있고 유형이 올바른지 확인하십시오.

BIFROST-985 Missing overloads: 최소/최대 노드는 호환되는 모든 유형에서 작동해야 합니다.

BIFROST-1655 Node library: 입력과 출력의 유형이 같을 경우, Conversion 노드가 올바르게 작동하지 않습니다.

BIFROST-3023 Node library: 폴더 이름의 프레임 토큰(예: #)이 write_OpenVDB와 같은 쓰기 파일 노드용으로 대체되지 않습니다.

이것은 폴더 이름이 아니라 파일 이름의 토큰과 함께 올바로 작동합니다.

BIFROST-2729 Simulation: 느린 시뮬레이션으로 경계 성능

시뮬레이션 경계에 여전히 성능 개선이 필요합니다.

BIFROST-3195 Translation: 파일 로드 및 Maya 배치에서 'Type specified for new attribute's data type is unknown'과 같은 경고가 나타날 수 있습니다.

이런 경고는 무시해도 무방합니다.

BIFROST-2230 Value node: 구성원 출력에서 자동 루핑이 작동하지 않습니다.

BIFROST-1937 Value node: 전개된 구조(벡터, 행렬)의 구성원 연결을 끌어올 수 없습니다.

BIFROST-3108 Viewport Updating: 오류가 있는 그래프가 뷰포트 도면을 지우지 못합니다.

그래프가 오류 상태인 경우 뷰포트가 자동으로 업데이트되지 않아, 마지막으로 유효했던 상태의 도면이 계속 나타날 수 있습니다.

해결방법: 장면을 저장한 다음 다시 여십시오.

BIFROST-3059 Viewport Updating: 가닥의 색상을 설정할 때 뷰포트에서 자동으로 업데이트되지 않습니다.

BIFROST-2604 Viewport: 메쉬의 면-정점, 면 구성요소, 불변 사용자 데이터가 뷰포트에서 작동하지 않습니다.

BIFROST-2429 Viewport: Float4 컬러가 뷰포트 내 메쉬와 가닥에서 Z-Fighting을 유발합니다.

해결방법: 다른 Transparency Algorithm (Depth Peeling 또는 Alpha Cut)을 선택하십시오.

BIFROST-2638 Viewport: 인스턴스 뷰포트 렌더링 문제

노멀 없이 메쉬를 인스턴스화하면 검정색으로 그려집니다.

그래도 메쉬들은 정확히 그려지지만 인스턴스화하면 검정색으로 그려집니다.

인스턴스에서 머리티얼을 재지정하면 bif 모양이 자동으로 업데이트되지 않을 수도 있습니다. 장면을 저장한 후 다시 열어 뷰포트를 업데이트하십시오.

BIFROST-1982 Viewport: 뷰포트 내 다수의 볼륨들이 경우에 따라 투명도 순서대로 렌더링되지 않습니다.

BIFROST-2349 Viewport: Point Clouds 뷰포트 렌더링 문제

선택 항목 강조 표시 기능을 지원하지 않습니다.

Maya 2018에서는 스크린 정렬된 육면체와 구형을 그릴 수 없습니다. Maya 2019와 2020에서는 그릴 수 있어도 스크린 정렬이 불가능합니다.

Maya 2018에서는 포인트 숫자 값이 항상 0으로 그려집니다.

BIFROST-2706 Viewport: 뷰포트에서 사후 효과를 설정한 경우에는 포인트, 가닥, 메쉬가 올바로 그려지지 않을 수도 있습니다.

BIFROST-2380 Viewport: 가닥 뷰포트 렌더링 문제

육면체 그리기는 지원되지 않습니다.

"Use default materials"가 활성화되어 있으면, 리본이 와이어로 그려집니다.

Maya 2018에서는 'Ribbons with two sided lighting'이 활성화되어 있어도 기대한 대로 그려지지 않을 수도 있습니다.

스크린 정렬 가닥은 스큐잉(skewing) 때문에 너비가 잘못 그려집니다.

그래프에 standard_hair shader가 적용된 상태에서는 낱 가닥은 아예 그릴 수 없습니다.

그래프에 standard_hair shader가 적용된 경우, 가닥 어레이는 리본이 아닌 와이어로 그려집니다.

BIFROST-2067 Viewport: 볼륨 뷰포트 렌더링 문제

뷰포트에서 볼륨을 렌더링할 때에는 변형을 무시해도 좋습니다. Arnold에서는 올바로 렌더링됩니다.

볼륨 어레이를 렌더링할 때, 어레이의 첫 번째 요소만 그려집니다. Arnold에서는 나머지도 올바로 렌더링됩니다.

음영 모드에서는 선택 항목 강조 표시가 지원되지 않습니다.

BIFROST-2353 Viewport: OpenGL - Legacy 뷰포트에서 볼륨이 아예 나타나지 않습니다.

BIFROST-2573 Viewport: 뷰포트에서 사후 효과를 설정한 경우에는 볼륨이 올바르게 그려지지 않을 수도 있습니다.

뿐만 아니라, Viewport 2.0에서 'Use Default Material'을 활성화했다면 볼륨과 기타 bifShape들이 올바르게 그려지지 않습니다.

BIFROST-3054 Viewport: 쉐이더 특성에 따라 set_geo_property_reference가 작동되지 않는 경우도 있습니다.

BIFROST-1593 Watchpoint: 포트 유형 변경 후 와치포인트를 제거할 수 없습니다.

BIFROST-3162 bifcmd: BIFROST_LIB_CONFIG_FILES 환경 변수에서 config 파일을 로드할 수 없습니다.

ARNOLD-101 Bifrost Graph에서 2 GB 이상의 bob 파일을 읽을 때 렌더링이 되지 않습니다.

Bifrost Graph에서 2GB 이상의 .bob 파일을 읽을 때 Arnold를 이용한 렌더링에 실패할 수도 있습니다.

ARNOLD-93 선택한 bifShape들을 따로 분리할 때 Arnold Render View 충돌이 발생합니다.

ARNOLD-99 Arnold에서 Bifrost 볼륨을 인스턴스화할 수 없습니다.

ARNOLD-96 배치 및 시퀀스 렌더에서는 모션 블러를 포함하는 라이브 파티클 시뮬레이션에 실패합니다.

모션 블러를 포함하는 라이브 시뮬레이션의 렌더링은 지원되지 않습니다.

해결방법: 시뮬레이션을 .bob 파일 또는 .pdc로 캐싱한 다음 파일을 다시 불러와 렌더링합니다. 또는, file_cache 노드를 이용해 시뮬레이션을 읽고 쓸 수도 있습니다.

ARNOLD-67 Arnold에서 가닥을 렌더링하려면 반드시 set_strand_shape를 설정해야 합니다.

ARNOLD-92 On Windows 7: 볼륨을 킥 렌더링할 때 dll 오류를 알리는 대화상자가 나타나는 경우

이 대화상자를 무시하고 닫으면 렌더링에 성공할 것입니다.

ARNOLD-87 Arnold에서 방향이 있는 가닥이 렌더링되지 않습니다.

"camera facing"이 비활성화되어 있고 노멀들이 꼬여 있으며 노멀들의 접선을 중심으로 회전하는 가닥은 뷰포트에서는 올바로 나타나지만 Arnold에서는 카메라를 향하도록 렌더링됩니다.

"set_Arnold_strands_settings"에서 "oriented"로 설정된 가닥은 렌더링되지 않습니다.

ARNOLD-97 일부 볼륨이 렌더링되지 않거나 상당 부분 잘려집니다.

Arnold로 Bifrost 볼륨을 직접 렌더링할 때 잘림 현상이 나타날 수 있습니다.

해결방법: 최종 렌더링을 위해 OpenVDB로 캐싱합니다.

ARNOLD-84 Arnold IPR 세션 도중 bif들을 조작할 때 Script Editor에 'Warning: inputSurfa'라는 메시지가 나타납니다.

ARNOLD-91 Arnold에서 볼륨 렌더링 시간이 OpenVDB보다 최대 2.5배 더 길어집니다.

볼륨을 OpenVDB 파일로 캐싱하고 최종 렌더링을 위해 Arnold 스탠드 인을 이용합니다.

ARNOLD-45 bifshape를 포함하는 장면을 렌더링할 때 Arnold 신규 세션을 각각 시작할 때마다 ~6초 지연됩니다. (Arnold에서 Bifrost를 로딩하기 때문)

상위 페이지: Bifrost Extension 릴리스 노트

Bifrost 2.0.1.0 릴리즈 노트

절차식 그래프로 만든 Bifrost 콘텐츠를 렌더링할 수 있는 Arnold for Maya의 최소 버전은 3.2.2입니다.

최신 Arnold for Maya 설치관리자를 다운로드 하려면, Arnold for Maya Download Area를 방문하십시오.

발견된 문제점과 제약사항

BIFROST-3144 Aero: 파란색 불꽃 타일 인공물

파란색 불꽃 안에 타일 또는 복셀 인공물이 나타날 때가 있습니다(연소 속도).

BIFROST-2776 Aero: 기본 적응도 옵션이 basic_aero에서 적응도를 이용하지 않을 때보다 더 느립니다.

특정 상황에서, 자동 적응도를 설정하면 시뮬레이션이 느려질 수 있습니다.

BIFROST-2982 Aero: Influence 필드에서 공기 객체의 복사본을 방지합니다.

Influence 필드를 이용할 때 발생하는 공기 객체 복사본은 성능을 떨어뜨립니다.

BIFROST-3150 Aero: 그을음 변화 문제

연소 시뮬레이션에서는 경우에 따라 변화가 너무 심한 그을음이 만들어지기도 합니다.

BIFROST-2823 Aero: boost_detail_with_points가 연소를 없앱니다.

연소 시뮬레이션에서 "boost detail with points"를 활성화하면 그을음 방출이 사라집니다.

BIFROST-2903 Aero: 장면의 스케일과 함께 경계가 정해지는 적응도 문제

장면 단위(m)와 함께 1 이외로 경계가 정해지는 적응도를 이용하면 경계의 크기를 조절할 수 없습니다.

BIFROST-822 Aero: 한 볼륨에서 방출 시 이미터 경계 밖 일부 구간에서 방출이 발생합니다.

경우에 따라, 공기 소스 지오메트리 밖의 영역에서 유령 방출이 생길 수 있습니다.

BIFROST-1498 Attraction_Repulsion은 이미터의 복셀화에 따라 파티클에 영향을 미칩니다.

Attraction-repulsion influence는 소스 geo 복셀화에 따라 파티클에 영향을 미칩니다.

BIFROST-2827 Compound Publishing: 시그니처를 변경해 다시 게시한 후, 참조된 컴파운드 복사본의 상태가 나빠집니다.

컴파운드가 참조되는 바로 그 장면에서 시그니처를 변경하고 컴파운드를 다시 쓴 후에는 그 장면을 저장하지 말아야 합니다. 그렇지 않으면, 참조된 컴파운드가 손상된 상태로 저장됩니다. 해당 장면을 저장하지 않고 다시 로드하면 참조가 올바르게 다시 로드됩니다.

BIFROST-2109 Compound Publishing: 기본 값을 갖는 자동 포트를 이용할 경우 컴파운드를 게시할 수 없습니다.

BIFROST-2765 Compound Publishing: Linux 플랫폼에서, 컴파운드를 사용자 위치에 게시할 때 json 확장자가 파일 이름에 자동으로 붙지 않습니다.

해결방법: 게시할 때 파일 이름에 확장자 .json을 붙여야 합니다.

BIFROST-2767 Compound Publishing: 시그니처를 변경할 경우 그리고 자동 포트가 없는 경우 다시 게시하는 것이 불가능합니다.

해결방법: 컴파운드를 불러와, 디스크에서 json 파일을 삭제하고, Maya를 다시 시작해 다시 게시합니다.

BIFROST-2532 Compound publishing: 오류가 있을 때 네임스페이스 필드에 입력이 불가하고 키를 누를 때마다 포커스가 사라집니다.

해결방법: 게시 편집기에서 컴파운드의 이름을 임시로 변경해 오류를 제거하고, 네임스페이스를 편집한 다음, 이름을 원래대로 다시 변경합니다.

BIFROST-2769 Compound publishing: 컴파운드 게시는 대소문자를 구분하지만 탭(Tab) 생성은 대소문자를 구분하지 않습니다.

My_Compound와 my_compound처럼 대소문자만 다른 컴파운드를 동시에 만들지 않아야 합니다.

BIFROST-2279 Encapsulation: 자동 루핑 되는 노드 1개로 컴파운드를 만들면 그래프 반응이 바뀝니다.

BIFROST-2581 Encapsulation: 가져온 컴파운드를 다수의 오버로드로 수정할 때 예기치 않은 결과가 발생합니다.

get_geo_property와 같은 오버로드된 컴파운드를 가져온 다음 포트를 추가하거나 제거하거나 이름을 변경하거나 그룹화할 경우, 컴파운드가 손상되거나 충돌이 발생할 수 있습니다.

BIFROST-2655 Errors: 참조된 컴파운드나 그래프 중 컴파운드가 아닌 오류 상태의 노드를 포함하고 있거나 레거시 노드가 누락된 것은 가져올 수 없습니다.

참조된 컴파운드가 오류 상태의 노드를 포함하고 있다면, 경우에 따라, 참조된 컴파운드를 가져와서 UI를 통해 문제를 해결하는 것이 불가능할 수도 있습니다.

해결방법: 이 컴파운드 문제는 .json 파일로 해결할 수 있습니다.

BIFROST-2542 Errors: 침묵의 컴파일 오류(예: 프로모션이 없음)

가끔 컴파일 오류가 발생할 때도 있는데 그래프에 나타나지 않습니다.

그래프가 평가를 하지 않는 경우가 발생하면, Output Window에서 컴파일 오류가 없는지 확인해야 하는데, 오류가 있다면 오류 상태의 노드를 알려주고 문제 해결을 도와줄 것입니다.

BIFROST-3024 Errors: write_OpenVDB와 같은 쓰기 노드를 이용할 경우, 쓰기에 실패할 때에도 노드의 성공 포트가 항상 1을 반환합니다.

해결방법: 쓰기 작업을 할 때 폴더와 파일의 이름이 올바른지 확인합니다.

BIFROST-3121 Graph Publishing – Maya 파일 안에 있는 그래프들은 해당 문서의 링크를 절대 경로로 저장합니다.

따라서, 그 문서가 그래프의 예시 장면에 나타나지 않습니다.

BIFROST-3035 Graph Publishing: 게시 UI에 기본 브라우저 카테고리가 없습니다.

게시할 때 카테고리 안으로 들어가야 합니다.

BIFROST-2887 Graph publishing: 게시되는 그래프에 Tab 키가 나타나지 않습니다.

BIFROST-2536 Immediate values: 일부 노드를 캡슐화할 때 즉시 값들이 사라집니다.

다음과 같은 경우에 즉시 값(컴파운트 포트에 설정하는 값)들이 사라질 수도 있습니다.

컴파운드를 만들 때, 폐쇄형 컴파운드를 전개할 때와 편집 가능하게 만들 때, 참조된 컴파운드를 다시 게시할(쓸) 때. (BIFROST-2933, BIFROST-3026)

BIFROST-3026 Immediate values: 참조된 컴파운드를 다시 쓰면 즉시 값들이 삭제됩니다.

BIFROST-3154 속도 특성이 없을 때 볼륨으로부터 방출되는 파티클의 위치가 잘못 설정됩니다.

속도 데이터 없이 볼륨에서 파티클을 방출시키면 잘못된 결과가 나타납니다.

BIFROST-1581 Influences: 높이를 가진 지오메트리를 이용할 때 방사 필드에 예기치 않은 결과 발생합니다.

BIFROST-3028 각 면마다 음영이 있는 입력 메쉬에는 머티리얼을 할당할 수 없고, 가장 먼저 찾아낸 컴파운드가 사용되지 않습니다.

현재, 각 면마다 음영이 있는 입력 메쉬는 지원되지 않습니다.

BIFROST-2882 Input: 입력 노드에서 Normals 옵션을 변경하면 Maya 장면과의 연결이 끊어집니다.

Input by Path 노드 Parameter Editor에서 Normals per point와 Normals per face-vertex 옵션을 선택하거나 선택 해제하면 경로가 재평가되어 Maya 객체와의 연결이 끊길 수도 있습니다.

이 문제를 해결하려면, 해당 메쉬를 그래프로 다시 끌어다 놓거나 Parameter Editor에서 새로운 경로를 입력하는 방식으로 입력 노드와의 연결을 재설정하면 됩니다.

BIFROST-2722 Input: 가져온 그래프에서 경로별 입력이 끊어집니다.

해결방법: Maya 지오메트리를 입력으로 이용하는 그래프는 게시하지 마십시오.

그 대신, create_mesh_cube나 create_mesh_plane처럼 사용자가 나중에 자신이 원하는 것으로 직접 바꿀 수 있는 자리 표시자 지오메트리를 만드십시오.

BIFROST-928 Input: 메쉬를 그래프의 더 심층 레벨로 끌어다 놓을 때 쓸모 없는 입력 노드가 나타납니다.

BIFROST-3237 Inputs: 한 그룹에 속해 있는 네임스페이스가 있는 메쉬를 네임스페이스 없이 끌어다 놓으면 올바른 경로가 생성되지 않고 기본적으로 그 메쉬를 그래프로 가져올 수도 없습니다.

해결방법: 입력 노드를 통해 메쉬에 올바른 경로를 입력합니다. 여기서는 "긴" Maya 이름도 사용 가능합니다.

BIFROST-3164 Inputs: 머티리얼 참조의 경우, 머티리얼 이름 변경으로 인한 업데이트 문제가 발생합니다.

BIFROST-2889 Installers: SDK: 설치 관리자에 BifrostFluids SDK가 없습니다.

BIFROST-3198 Interaction: 와치포인트가 들어있는 그래프의 경우 새로운 Maya 장면을 생성할 때 이따금씩 충돌이 발생합니다.

BIFROST-2799 Interaction: 작업공간을 전환할 때 Bifrost Graph Editor가 깜빡입니다.

Graph Editor가 도킹된 상태에서 Ctrl+스페이스바를 사용할 때에도 Bifrost Graph Editor가 깜빡일 수 있습니다. (BIFROST-2861)

BIFROST-3070 Interaction: Shelf 아이콘을 클릭할 때 BifrostGraph 플러그인이 자동으로 로드됩니다.

Bifrost Graph Editor 또는 Shelf에 있는 Bifrost Browser 버튼을 클릭할 때 이 오류가 발생하는 경우:

// Error: line 1: Cannot find procedure "openBifrostGraphEditorFromSelection".

bifrostGraph 플러그인을 로드해야 합니다. 그러기 위해서 Windows > Settings/Preferences > Plugin Manager를 차례로 클릭합니다.

BIFROST-3021 Interaction: 옵션이 해제되어 있는데도 생성 중인 노드가 연결됩니다.

해결방법: Connect Nodes on Creation 옵션이 해제되어 있는 상태에서도 Tab 생성 노드가 노드들을 자동으로 연결할 때에는 Bifrost Graph Editor를 닫았다가 다시 여십시오.

BIFROST-3151 Interaction: 유형이 Any인 값 포트를 마우스 오른쪽 버튼으로 클릭할 때 충돌이 발생합니다. (예: set_property 노드에서)

유형이 Any인 포트를 마우스 오른쪽 버튼으로 클릭해 값 노드를 생성하려고 하면 충돌이 발생합니다.

해결방법: Value 노드를 하나 만든 다음, 이것을 set_property에 연결합니다.

BIFROST-2708 Interaction: 탭을 복제하고 장면을 열 때 충돌이 발생합니다.

컴파운드 안으로 들어가 탭을 마우스 오른쪽 버튼으로 클릭해 Duplicate Tab을 선택한 다음, 새 장면을 열면 충돌이 발생할 수 있습니다.

BIFROST-574 Interaction: 포커스 문제 – Parameter Editor에서 값을 편집할 때 컴파운드로 마우스를 가져가서 Enter를 누르면 그 컴파운드 안으로 들어가게 됩니다.

BIFROST-1802 Interaction: 노드나 컴파운드의 이름을 특수 문자로 바꾸면 이름 필드가 사라져 해당 노드의 이름을 더 이상 변경할 수 없습니다.

노드나 컴파운드의 이름을 바꿀 때, ! @ # \$ % ^ & () + = \/ > < . , " : ; ' ~ `와 같은 특수 문자나 ASCII 이외의 문자(예를 들어, 다른 언어의 문자)를 사용해서는 안 됩니다. 이런 문자를 사용할 경우 예기치 못한 반응이 나타날 수 있습니다.

BIFROST-3065 Interaction: 메뉴에서 생성된 입력 또는 출력이 스크린 밖의 안 좋은 위치나 기존 입력 위에 나타납니다.

그 대신, Tab을 이용해 노드를 만드십시오.

BIFROST-2194 Interaction: 컴파운드를 만들고 안으로 들어가면 출력 노드가 잘못된 위치에 나타납니다.

BIFROST-2428 Interface: Bifrost UI를 이용할 때 DPI 값이 커집니다

BIFROST-3062 Malformed geometry: face_vertex 데이터 입력과 construct_mesh의 연결을 끊으면 Maya와 충돌합니다.

BIFROST-3115 Maya 2018: bifrostshellnode 플러그인을 로드하지 않거나 설치하지 않을 경우, Attribute Editor가 비정상적으로 나타나고 오류가 발생합니다.

이런 오류는 Maya 2018, 2018.1 Update, 2018.2 Update에서 발생할 수 있습니다.

이 문제를 해결하려면, bifrostshellnode 플러그인을 설치하고 로드해야 합니다.

BIFROST-3166 Maya Plug-in: "Delete unused nodes"가 머티리얼 참조를 삭제합니다.

BIFROST-2334 Maya Plug-in: Bifrost 플러그인이 완벽히 언로드하지 않아 충돌이 발생할 수 있습니다.

동일 Maya 세션 안에서 Bifrost 플러그인을 언로드했다가 다시 로드하면 불안정성이나 기타 문제가 발생할 수 있습니다.

BIFROST-2444 Maya Plug-in: 뷰포트에서 bifShapes를 선택할 수 없습니다.

BIFROST-2997 Maya Plug-in: 포트 유형을 array에서 single로 바꿀 때 Attribute Editor가 올바로 업데이트되지 않습니다.

BIFROST-3138 Maya Plug-in: vnn 플러그인을 처음 로드할 때 Script Editor에서 Maya 2019.2 Update macOS 및 Linux: VNN / libaic 오류가 발생합니다.

이 오류는 무시해도 좋습니다.

BIFROST-1869 Maya Plug-in: 시뮬레이션 도중 ESC가 중단되기 전에 Maya가 별도의 프레임을 지연시킵니다.

BIFROST-3171 Maya Plug-in: 그래프를 포함하는 장면을 여러 번 참조할 때 한 개의 bifrost 그래프만 작동합니다.

BIFROST-2330 Maya Plug-in: 장면을 로드해도 mayapy에서 vnn 플러그인이 로드되지 않습니다.

Bifrost는 mayapy에서도 작동하지만 VNN 플러그인은 자동으로 로드되지 않기 때문에 VNN Python 명령이 효과가 없을 것입니다.

해결방법: mayapy에서 VNN 명령을 사용하려면, 먼저 loadPlugin 명령을 이용해 플러그인을 로드하십시오.

BIFROST-3073 Maya Plugin-in: bifrostGraph 플러그인이 없을 때 bifrostGraph 노드를 포함하는 장면을 로드하면 충돌이 발생합니다.

해결방법: Bifrost 노드를 포함하는 장면을 열기 전에 먼저 bifrostGraph 플러그인이 설치되어 있는지 확인합니다.

BIFROST-3243 BIFROST_LIB_CONFIG_FILES가 원격 네트워크 위치를 가리키는 경우 Maya가 매우 느리게 실행됩니다.

BIFROST-3152 Merge geometry는 첫 번째 객체를 이용해 병합 반응을 설정하나 그 대신 비어 있지 않은 첫 번째 객체의 스키마를 이용해야 하는 경우

따라서, 첫 번째 객체가 비어 있다면, merge geometry는 작동하지 않을 것입니다.

해결방법: 첫 번째 객체가 비어 있고 유형이 올바른지 확인하십시오.

BIFROST-985 Missing overloads: 최소/최대 노드는 호환되는 모든 유형에서 작동해야 합니다.

BIFROST-1655 Node library: 입력과 출력의 유형이 같을 경우, Conversion 노드가 올바르게 작동하지 않습니다.

BIFROST-3023 Node library: 폴더 이름의 프레임 토큰(예: #)이 write_OpenVDB와 같은 쓰기 파일 노드용으로 대체되지 않습니다.

이것은 폴더 이름이 아니라 파일 이름의 토큰과 함께 올바로 작동합니다.

BIFROST-3013 Publishing: 최근에 다시 쓴 컴파운드를 참조하는 컴파운드를 참조하면 충돌이 발생합니다.

A 컴파운드를 다시 게시한(다시 쓴) 경우, 이 컴파운드를 최근에 B 컴파운드에서 참조했다면, B 컴파운드를 참조할 때 충돌이 발생합니다.

해결방법: 다른 컴파운드에서 참조한 컴파운드를 다시 게시한(다시 쓴) 후 Maya를 다시 시작하십시오.

BIFROST-2729 Simulation: 느린 시뮬레이션으로 경계 성능

시뮬레이션 경계에 여전히 성능 개선이 필요합니다.

BIFROST-3195 Translation: 파일 로드 및 Maya 배치에서 'Type specified for new attribute's data type is unknown'과 같은 경고가 나타날 수 있습니다.

이런 경고는 무시해도 무방합니다.

BIFROST-2230 Value node: 구성원 출력에서 자동 루핑이 작동하지 않습니다.

BIFROST-1937 Value node: 전개된 구조(벡터, 행렬)의 구성원 연결을 끌어올 수 없습니다.

BIFROST-3108 Viewport Updating: 오류가 있는 그래프가 뷰포트 도면을 지우지 못합니다.

그래프가 오류 상태인 경우 뷰포트가 자동으로 업데이트되지 않아, 마지막으로 유효했던 상태의 도면이 계속 나타날 수 있습니다.

해결방법: 장면을 저장한 다음 다시 여십시오.

BIFROST-3059 Viewport Updating: 가닥의 색상을 설정할 때 뷰포트에서 자동으로 업데이트되지 않습니다.

BIFROST-2604 Viewport: 메쉬의 면-정점, 면 구성요소, 불변 사용자 데이터가 뷰포트에서 작동하지

않습니다.

BIFROST-2429 Viewport: Float4 컬러가 뷰포트 내 메쉬와 가닥에서 Z-Fighting을 유발합니다.

해결방법: 다른 Transparency Algorithm (Depth Peeling 또는 Alpha Cut)을 선택하십시오.

BIFROST-2638 Viewport: 인스턴스 뷰포트 렌더링 문제

노멀 없이 메쉬를 인스턴스화하면 검정색으로 그려집니다.

그래도 메쉬들은 정확히 그려지지만 인스턴스화하면 검정색으로 그려집니다.

인스턴스에서 머리티얼을 재지정하면 bif 모양이 자동으로 업데이트되지 않을 수도 있습니다. 장면을 저장한 후 다시 열어 뷰포트를 업데이트하십시오.

BIFROST-1982 Viewport: 뷰포트 내 다수의 볼륨들이 경우에 따라 투명도 순서대로 렌더링되지 않습니다.

BIFROST-2349 Viewport: Point Clouds 뷰포트 렌더링 문제

선택 항목 강조 표시 기능을 지원하지 않습니다.

Maya 2018에서는 스크린 정렬된 육면체와 구형을 그릴 수 없습니다. Maya 2019와 2020에서는 그릴 수 있어도 스크린 정렬이 불가능합니다.

Maya 2018에서는 포인트 숫자 값이 항상 0으로 그려집니다.

BIFROST-2706 Viewport: 뷰포트에서 사후 효과를 설정한 경우에는 포인트, 가닥, 메쉬가 올바로 그려지지 않을 수도 있습니다.

BIFROST-2380 Viewport: 가닥 뷰포트 렌더링 문제

육면체 그리기는 지원되지 않습니다.

"Use default materials"가 활성화되어 있으면, 리본이 와이어로 그려집니다.

Maya 2018에서는 'Ribbons with two sided lighting'이 활성화되어 있어도 기대한 대로 그려지지 않을 수도 있습니다.

스크린 정렬 가닥은 스큐잉(skewing) 때문에 너비가 잘못 그려집니다.

그래프에 standard_hair shader가 적용된 상태에서는 낱 가닥은 아예 그릴 수 없습니다.

그래프에 standard_hair shader가 적용된 경우, 가닥 어레이는 리본이 아닌 와이어로 그려집니다.

BIFROST-2067 Viewport: 볼륨 뷰포트 렌더링 문제

뷰포트에서 볼륨을 렌더링할 때에는 변형을 무시해도 좋습니다. Arnold에서는 올바로 렌더링됩니다.

볼륨 어레이를 렌더링할 때, 어레이의 첫 번째 요소만 그려집니다. Arnold에서는 나머지도 올바로 렌더링됩니다.

음영 모드에서는 선택 항목 강조 표시가 지원되지 않습니다.

BIFROST-2353 Viewport: OpenGL - Legacy 뷰포트에서 볼륨이 아예 나타나지 않습니다.

BIFROST-2573 Viewport: 뷰포트에서 사후 효과를 설정한 경우에는 볼륨이 올바르게 그려지지 않을 수도 있습니다.

뿐만 아니라, Viewport 2.0에서 'Use Default Material'을 활성화했다면 볼륨과 기타 bifShape들이 올바르게 그려지지 않습니다.

BIFROST-3054 Viewport: 쉐이더 특성에 따라 set_geo_property_reference가 작동되지 않는 경우도 있습니다.

BIFROST-1593 Watchpoint: 포트 유형 변경 후 와치포인트를 제거할 수 없습니다.

BIFROST-3162 bifcmd: BIFROST_LIB_CONFIG_FILES 환경 변수에서 config 파일을 로드할 수 없습니다.

ARNOLD-101 Bifrost Graph에서 2 GB 이상의 bob 파일을 읽을 때 렌더링이 되지 않습니다.

Bifrost Graph에서 2GB 이상의 .bob 파일을 읽을 때 Arnold를 이용한 렌더링에 실패할 수도 있습니다.

ARNOLD-93 선택한 bifShape들을 따로 분리할 때 Arnold Render View 충돌이 발생합니다.

ARNOLD-99 Arnold에서 Bifrost 볼륨을 인스턴스화할 수 없습니다.

ARNOLD-96 배치 및 시퀀스 렌더에서는 모션 블러를 포함하는 라이브 파티클 시뮬레이션에 실패합니다.

모션 블러를 포함하는 라이브 시뮬레이션의 렌더링은 지원되지 않습니다.

해결방법: 시뮬레이션을 .bob 파일 또는 .pdc로 캐싱한 다음 파일을 다시 불러와 렌더링합니다. 또는, file_cache 노드를 이용해 시뮬레이션을 읽고 쓸 수도 있습니다.

ARNOLD-67 Arnold에서 가닥을 렌더링하려면 반드시 set_strand_shape를 설정해야 합니다.

ARNOLD-92 On Windows 7: 볼륨을 킥 렌더링할 때 dll 오류를 알리는 대화상자가 나타나는 경우

이 대화상자를 무시하고 닫으면 렌더링에 성공할 것입니다.

ARNOLD-87 Arnold에서 방향이 있는 가닥이 렌더링되지 않습니다.

"camera facing"이 비활성화되어 있고 노멀들이 꼬여 있으며 노멀들의 접선을 중심으로 회전하는 가닥은 뷰포트에서는 올바로 나타나지만 Arnold에서는 카메라를 향하도록 렌더링됩니다.

"set_Arnold_strands_settings"에서 "oriented"로 설정된 가닥은 렌더링되지 않습니다.

ARNOLD-97 일부 볼륨이 렌더링되지 않거나 상당 부분이 잘립니다.

Arnold로 Bifrost 볼륨을 직접 렌더링할 때 잘림 현상이 나타날 수 있습니다.

해결방법: 최종 렌더링을 위해 OpenVDB로 캐싱합니다.

ARNOLD-84 Arnold IPR 세션 도중 bif들을 조작할 때 Script Editor에 'Warning: inputSurfa'라는 메시지가 나타납니다.

ARNOLD-91 Arnold에서 볼륨 렌더링 시간이 OpenVDB보다 최대 2.5배 더 길어집니다.

볼륨을 OpenVDB 파일로 캐싱하고 최종 렌더링을 위해 Arnold 스탠드 인을 이용합니다.

ARNOLD-45 bifshape를 포함하는 장면을 렌더링할 때 새로운 Arnold 신규 세션을 각각 시작할 때마다 ~6초 지연됩니다. (Arnold에서 Bifrost를 로딩하기 때문)

상위 페이지: Bifrost Extension 릴리즈 노트

Bifrost Extension에서 작업하기

Bifrost는 그래프를 만들어 효과를 연출할 수 있는 노드 기반 환경입니다.

Bifrost Extension을 이용하면 공개 시각 프로그래밍 환경에서 절차식 효과를 위한 재사용 가능한 그래프를 만들 수 있습니다. 또, 기존 그래프를 필요에 맞게 수정하거나, basic_combustion_graph와 같은 템플릿으로 시작하거나, 아예 처음부터 새로운 효과를 만들 수 있습니다.

전용 솔버도 있고 파티클, 공기(연기와 불에 필요), 모래, 눈, 기타 소재를 위한 MPM(Material Point Method) 등의 시뮬레이션을 위한 기타 노드도 있습니다. 뿐만 아니라, 인스턴싱과 같은 기타 작업을 위한 노드 세트와 수학과 기타 일반 연산을 위한 기초 노드도 있습니다.

그래프 생성 및 수정

Bifrost Graph Editor에서 그래프를 확인, 구축 및 수정할 수 있습니다.



1. 탭 및 이동 경로

장면에 있는 여러 가지 그래프가 별도의 탭에 표시됩니다.

- 탭을 클릭하면 그래프가 주요 편집 영역에 표시됩니다.
- 탭을 마우스 오른쪽 버튼으로 클릭하여 그래프를 복제합니다. 이렇게 하면 한 개의 그래프를 여러 가지 각도로 볼 수 있습니다. 탭의 명칭을 임의로 변경할 수도 있지만 탭의 기본 명칭은 항상 호스트 장면에 있는 그래프의 명칭을 토대로 설정됩니다.
- 탭을 드래그하여 그래프의 순서를 변경합니다.
- + 키를 클릭하여 기존 그래프를 열거나 장면에서 그래프를 새로 생성합니다.

주요 영역의 컴파운드를 더블 클릭하여 들어가면, "이동 경로 흔적"에서 입력한 노드가 표시됩니다. 노드의 명칭을 클릭하여 다시 돌아가거나, 혹은 좌측에 있는 화살표키를 이용합니다.

🖋 참고:

참조 컴파운드로 들어가 노드를 확인할 수도 있지만, 불러오기하여 편집 가능한 상태가 되어야만 내용을 수정할 수 있습니다.

2. 주요 편집 영역

그래프는 주요 편집 영역에 표시됩니다. 노드의 포트 사이에서 있는 연결을 따라 좌측에서 우측으로 흐르며, 이곳에서 데이터가 수정됩니다.

그래프 탐색하기:

• 마우스 중앙 버튼을 클릭하여 팬으로 드래그합니다. 혹은 Alt 키를 누른 상태에서 마우스 중앙 버튼을 클릭하여 드래그합니다.

- Alt 키를 누른 상태에서 마우스 오른쪽 버튼을 누르고 우측 또는 좌측으로 드래그하여 화면을 확대/축소할 수 있습니다. 혹은 마우스 휠을 사용할 수도 있습니다.
- F 키를 눌러 선택한 노드의 프레임을 설정합니다. 혹은 A 키를 눌러 모든 노드의 프레임을 설정합니다.
- 노드 선택하기:
 - 사각형을 클릭하고 드래그하여 내부에 있는 모든 노드를 marquee하여 선택합니다.
 - Shift 키를 누른 상태에서 클릭하여 노드 선택을 전환합니다.
 - Ctrl 키를 누른 상태에서 클릭하여 노드 선택을 해제합니다.
 - Ctrl 키와 Shift 키를 누른 상태에서 클릭하여 노드를 추가로 선택합니다.

노드, 포트, 연결을 마우스 오른쪽 버튼으로 클릭하거나 혹은 배경을 클릭하여 추가 옵션을 확인합니다.

▶ 키를 누르면 선택한 노드가 자동으로 정렬되며, 선택한 노드가 없다면 모든 노드가 자동으로 정렬됩니다.

3. 파라미터 편집기 및 정보 탭

파라미터 편집기는 선택한 노드에 대한 입력값을 표시합니다. 포트가 연결에 의해 설정되지 않는다면 여기에 값을 입력할 수 있습니다.

파라미터 편집기 상단에 있는 아이콘 옆에는 두 개의 스트링, 즉 노드의 명칭과 그 유형이 표시됩니다. 노드의 기본 명칭은 그 유형에 따라 설정되지만 그래프에서 명칭을 더블 클릭하여 변경할 수 있습니다.

Info를 클릭하면 선택한 노드와 그 포트에 대한 정보가 표시됩니다.

팁:

이 탭들을 분리하여 float하거나, Bifrost Graph Editor의 다른 곳에 도킹할 수 있습니다.

4. 정보 패널

Info Panel 팝업 창에는 노드, 포트, 또는 포인터 하단의 연결에 대한 정보가 표시됩니다. 이를 토대로 그래프 문제를 진단할 수 있습니다.

이 섹션의 페이지들

- 그래프의 구조
- 데이터 유형 정보
- 포트 유형과 아이콘에 대한 설명
- 노드 추가하기
- value 노드 추가하기
- 포트 연결하기 및 연결 끊기
- 곡선을 이용해 효과 조절하기
- 어레이 작업
- 컴파운드 생성 및 편집
- 프로그래밍 루프 작업
- 행렬과 변환 작업

BIFROST

- 기하학적 객체에서 특성 확인 및 설정
- 맞춤형 시뮬레이션 만들기
- 파일에서 나온 입력 및 출력 데이터
- 그래프 컴파일과 실행 작업 일시중지
- 그래프 진단 및 디버깅
- 배경을 이용해 노드에 주석을 달고 구조화하기
- 효과를 그래프로 게시하기

그래프의 구조

아래 그림은 샘플 파티클 그래프의 최상위 레벨을 보여줍니다. Bifrost 그래프라면 모두 이와 비슷한 요소들을 갖추고 있습니다.



데이터 플로우

Bifrost는 순수한 데이터 플로우 그래프입니다. 데이터는 노드에 있는 포트들 간의 연결을 따라 왼쪽에서 오른쪽으로 이동합니다.

입력은 호스트 장면으로부터 장면 지오메트리나 숫자 매개변수와 같은 input 노드를 거쳐 들어올 수 있습니다. 또는 그래프 자체 내에서 입력을 생성할 수도 있는데, 예를 들면 value 노드를 이용하거나 create_mesh_sphere와 같은 단순 지오메트리 생성 노드를 이용할 수 있습니다.

그래프의 컴퓨팅 결과는 output 노드와의 연결을 통해 해당 장면으로 전송됩니다. 대부분의 그래프는 Bifrost의 기하학적 객체를 출력하는데, 이런 출력이 장면에서 bif 객체를 생성합니다. 이런 객체들은 뷰포트에서 확인하거나 Arnold와 같이 호환되는 렌더러로 렌더링할 수 있습니다.

뿐만 아니라, Bifrost 그래프는 다른 렌더러에서 사용할 수 있도록 지오메트리를 Alembic이나 VDB와 같은 표준 형식으로 출력할 수도 있습니다. 파일의 입/출력 데이터를 참고하십시오.

노드

노드는 자체 입력에서 연산을 수행하고 그 결과를 출력합니다. 입력 포트는 왼쪽에, 출력 포트는 오른쪽에 있습니다.

add나 multiply와 같은 기초 기능을 위한 노드도 있습니다. 물론, simulate_particles처럼 많은 연산을 수행해 효과를 만들어내는 상위 레벨 노드도 있고 유용한 유틸리티 기능을 수행하는 중간 레벨 노드도 있습니다.

하이 레벨과 중간 레벨 노드들은 일반적으로 *컴파운드*입니다. 컴파운드는 다른 노드들의 부분 그래프를 포함하고 있는 노드이며, 다른 컴파운드를 포함하는 경우도 있습니다. 컴파운드를 더블클릭해 안으로 들어가면 해당 콘텐츠를 볼 수 있습니다. 또, 자신만의 컴파운드를 만들어 그래프를 구성하거나, 재사용할 수 있도록 그래프를 게시할 수도 있습니다. 자세한 내용은 컴파운드 만들기와 편집하기를 참고하십시오. 그래프에 추가되어 게시된 컴파운드는 기본적으로 *참조*됩니다. 즉, 이런 컴파운드는 디스크에서 노드 정의를 가리키는 셈입니다. 참조된 컴파운드의 내부 부분 그래프는 로컬 그래프로 가져오기 전에는 수정이 불가합니다. 자세한 내용은 참조된 컴파운드 가져오기를 참고하십시오.



- 노드 이름. 기본 이름은 노드의 유형에 따라 결정되지만, 노드를 더블클릭해서 새로운 이름을 입력할 수도 있습니다.
- 2. 마름모 아이콘. 이것은 노드가 참조된 컴파운드임을 나타냅니다.
- 3. 삼각형을 클릭해 노드를 펼치거나 접을 수 있습니다. 또는, Ctrl+1, Ctrl+2, Ctrl+3 핫키나 Display 메뉴에 있는 명령을 이용할 수도 있습니다.
- 4. 포트 그룹. +나 -를 클릭해 펼치거나 접을 수 있습니다.
- 5. 입력 포트. 다른 노드의 출력을 연결하거나 **Parameter Editor**에서 값을 입력합니다. 마우스 오른쪽 버튼을 클릭해 다른 옵션을 볼 수 있습니다.
- 6. 노드 메뉴. 노드의 메인 바디를 마우스 오른쪽 버튼으로 클릭해도 됩니다.
- 7. 출력 포트. 다른 노드의 입력에 연결됩니다. 마우스 오른쪽 버튼을 클릭해 다른 옵션을 볼 수 있습니다.
- 8. 노드 아이콘. 일부 노드는 자신의 기능과 관련된 특수 아이콘을 갖고 있습니다. 여기에 나와있는 아이콘은 기본적인 컴파운드용 아이콘입니다.

데이터 유형 정보

한 값의 *유형*은 정수, 10진수, 벡터 등과 같은 데이터의 종류를 가리킵니다. 이는 각 유형의 어레이일 수도 있고, 다수의 어레이들로 구성된 어레이일 수도 있습니다.

유형은 연결과 포트의 색상으로 표시됩니다. 각 포트의 색상이 서로 다르더라도 연결이 가능합니다.

- 입력되는 유형을 입력 포트의 유형으로 프로모션하는 것도 가능합니다. 예를 들어, float 값을 double로 프로모션하거나 uint 값을 float로 프로모션할 수 있습니다.
- 입력 포트의 유형은 auto일 수 있습니다. 유형이 auto인 포트는 여러 가지 유형의 연결을 수락할 수 있습니다.

색상	유형
	10진수 값. 단정도(부동) 또는 배정도 부동 소수점 값일 수 있습니다.
	2D, 3D 또는 4D 벡터
	행렬
	다양한 폭의 부호 붙임 정수나 무부호 정수
	문자열
	부울값. 이 색상은 rotation_order와 같은 이넘(Enum)에도 사용됩니다.
	객체. 객체는 메쉬, 볼륨, 포인트 클라우드와 같은 기하학적 객체일 수도 있고 특정 노드와 함께 사용하도록 만들어진 사용자 지정 데이터 구조일 수도 있습니다. 이런 객체 유형의 한 예로 any가 있는데, 이 유형은 런타임 시 유형 검사를 수행하기 위한 랩퍼(wrapper)로 이용할 수 있습니다.
	Auto (미해결 상태). Auto는 그래프를 컴파일 및 실행하기 전에 반드시 다른 특정 유형으로 바꿔야 하는 자리 표시자입니다.

auto 유형 포트 소개

auto 유형의 포트는 수신되는 연결에 따라 새로운 유형을 취할 수 있기 때문에 특별합니다. auto 유형은 자리 표시자 역할을 하는데, 포트 연결이나 기본값을 근거로 실제 구체적인 유형을 추론해야 합니다. auto 유형의 포트는 수직 "halo"와 추론한 유형의 색상을 갖습니다.



예를 들어, 기본적으로, length 노드는 float3이 되는 입력 포트와 float가 되는 출력 포트를 1개씩 갖습니다. 이들은 auto 포트이지만 double2를 입력으로 연결할 수 있고, 이 때 출력은 double이 됩니다.

auto 포트는 다수의 유형을 지원하지만, 모든 유형을 지원하지는 않습니다. 지원되는 특정 유형은 각 노드에 따라 달라집니다. 예를 들어, 형렬을 length에 대한 입력으로 이용할 수는 없는데, 행렬은 길이를 갖고 있지 않기 때문입니다.

상수 값과 auto 포트

연결을 포함한 값을 적용하는 대신에 Parameter Editor를 이용해서 auto 포트에 상수 값을 입력하는 경우, 이 포트는 현재 추론해 알게 된 유형뿐만 아니라 이 상수값까지 기억하게 됩니다. 이 경우 이 포트는 더 이상 auto 포트 역할을 하지 않으며 halo를 갖지 않습니다.

이 포트를 auto 유형으로 복원하려면, 이 포트를 마우스 오른쪽 버튼으로 클릭해 Clear Immediate Value를 선택합니다.

포트 유형과 아이콘에 대한 설명

특정 아이콘과 포트 모양은 특별한 의미를 갖습니다.

아이콘	포트 유형
	값들의 어레이. 어레이 작업을 참고하십시오.
	팬인(Fan-in). 이것으로 여러 값과 값들의 어레이를 연결해 한 어레이로 취급되도록 할 수 있습니다. 원하는 컴파운드의 팬인(fan-in) 포트를 만드는 방법은 어레이용 팬인(fan-in) 포트 만들기를 참고하십시오.
#	반복 카운터. 프로그래밍 루프 작업을 참고하십시오.
<	반복 제한. 프로그래밍 루프 작업을 참고하십시오.
•	반복 목표. 프로그래밍 루프 작업을 참고하십시오.
9	State 포트. 프로그래밍 루프 작업을 참고하십시오.
?	루핑 조건. 프로그래밍 루프 작업을 참고하십시오.
0	피드백 포트. 이것을 이용해서 컴파운드의 결과를 캐싱해 시뮬레이션에서와 같이 다음 실행을 위한 입력으로 이용할 수 있습니다. 사용자 시뮬레이션 만들기를 참고하십시오.

노드 추가하기

- 1. Tab 키를 눌러 Tab 메뉴를 엽니다.
- 2. 다음 중 한 가지 작업을 수행합니다.
 - 입력을 시작해 아래 나타나는 해당 노드들의 목록을 확인합니다. 노드 이름 내 문자 위치와 관계없이 1-4자의 연속된 문자 시퀀스를 입력할 수 있습니다. 예를 들어, "gepopo"는 get_point_position과 대응하며 "gpp"는 get_point_position 및 get_particle_solver_properties와 대응합니다. 포인터를 이용해 추가하려는 노드를 선택하거나 화살표 키를 이용해 노드를 강조표시한 다음 Enter를 누릅니다.
 - 포인터나 화살표 키를 이용해 우측 네임스페이스 목록에서 원하는 노드를 찾아 선택합니다.
 - 상단의 최근 사용 목록에서 노드를 선택합니다.

팁:

연결을 먼저 선택해서 연결된 두 노드 사이에 새로운 노드를 삽입할 수 있습니다.

참조 컴파운드 안에서는 노드를 추가하거나 변경을 가할 수 없습니다. 참조 컴파운드를 수정하려면, 먼저 로컬에서 해당 컴파운드를 그래프로 가져와야 합니다.

노드를 추가한 후 이름을 변경하려면 이름을 더블 클릭해 새로운 이름을 입력하면 됩니다. 노드의 유형이 아닌 목적을 나타내는 이름을 부여하면 그래프의 가독성이 좋아집니다.

또, Ctrl+C와 Ctrl+V를 이용해서 선택한 노드 또는 노드 그룹을 복사해 붙여넣을 수 있습니다. 심지어 서로 다른 세션에서 실행되고 있는 Graph Editor 인스턴스들 사이에서도 노드를 복사해 붙여넣을 수 있습니다.

value 노드 추가하기

value 노드를 이용해 데이터 유형과 값을 설정할 수 있습니다. 이 노드는 다음을 포함해 많은 경우에 유용합니다.

- 계산식에 상수 값을 반영.
- 한 값에 특정 데이터 유형을 강제로 지정. 가능하다면, 자체 입력에 연결되는 것이면 무엇이든 자체 유형으로 프로모션(그렇지 않으면 오류 발생).
- 벡터와 행렬의 구성요소에 액세스.

여느 노드와 마찬가지로 Tab을 눌러 "value"를 입력함으로써 그래프에 value 노드를 추가할 수 있습니다. 또는, 연결되지 않은 아무 포트나 마우스 오른쪽 버튼을 클릭해 **Create Value Node**를 선택할 수도 있습니다. Parameter Editor에서 노드의 값을 설정하거나, 데이터 유형을 먼저 변경할 수 있습니다.

Array Size 1	
Value 0	

value 노드의 데이터 유형 변경하기

Parameter Editor에 value 노드의 현재 유형이 표시됩니다. 이 유형을 변경하는 방법은 다음과 같습니다.

- 1. 자세히 보기 버튼 (...)을 클릭합니다.
- 2. 새 유형을 선택합니다.

유형이 어레이인 경우, Parameter Editor에서 해당 어레이의 크기와 모든 요소들의 기본 값을 설정할 수 있습니다.

유형이 벡터나 행렬인 경우, +와 – 아이콘을 이용해서 구성요소들의 포트에 액세스할 수 있습니다.

ŀ	value · · · ·	
	▼ :	
	output -	
	c0 —	
	x	
	у	
	- c1+	
	c2 +	
	+ value	

포트 연결하기 및 연결 끊기

데이터는 노드의 우측에 있는 출력 포트와의 연결을 따라 다른 노드의 왼쪽에 있는 입력 포트로 흘러갑니다. 한 출력을 원하는 입력 수만큼 연결할 수 있습니다.

포트 연결하기

한 노드의 출력 포트를 클릭해 또 다른 노드의 입력 포트로 드래그하거나, 혹은 그 반대로 수행합니다.



포트 연결 끊기

- 1. 포트 주변의 연결부에 포인터를 가져가 연결을 끊습니다. 연결은 흰색으로 강조 표시됩니다.
- 2. 연결을 클릭해 포트로부터 멀리 드래그합니다.



또, Alt+Shift를 누른 상태에서 드래그함으로써 다수의 연결을 한 번에 끊을 수도 있습니다.



다른 방법으로는, 클릭해 드래그함으로써 연결을 1개 이상 선택한 다음, Delete 또는 Backspace를 누릅니다.

곡선을 이용해 효과 조절하기

evaluate_fcurve 노드는 편집 가능한 함수 곡선을 기반으로 입력 ⊠를 출력 ፶로 리매핑합니다. 이 기능은 사용자 프로파일을 이용해 폴오프(falloff) 등의 효과를 조절하는 데 유용합니다.

함수 곡선의 기본 범위는 [0, 1]입니다. 하지만, 이 범위 밖의 가로, 세로 위치에 포인트를 정의할 수 있습니다. 또는, 곡선 편집기의 배경을 마우스 오른쪽 버튼으로 클릭해 사전 설정 곡선을 시작점으로 선택할 수 있습니다. 그런 다음 곡선을 원하는 대로 편집합니다.



- 포인트를 클릭해 선택하거나 박스를 클릭해 드래그함으로써 다수의 포인트를 한 번에 선택할 수 있습니다. Shift를 누른 채 클릭하는 방식으로 선택할 포인트를 변경할 수 있습니다. 선택한 포인트를 드래그해 위치와 값을 변경합니다. Bezier 포인트의 경우, 접선의 핸들을 드래그해 기울기를 조절할 수 있습니다. 다음과 같은 기타 옵션을 설정하기 위해 포인트를 마우스 오른쪽 버튼으로 클릭합니다.
 - Break Tangents로 왼쪽과 오른쪽 접선 핸들을 개별적으로 조절할 수 있습니다.
 - Unify Tangents로 모든 핸들을 한 번에 잠글 수 있습니다.
- 2. 곡선을 클릭해 포인트를 추가한 다음, 원하는 위치로 드래그합니다.
- 3. Alt 키를 이용해 곡선 편집기를 탐색합니다.
 - Alt+가운데 버튼 클릭으로 드래그해 초점을 옮길 수 있습니다.
 - Alt+오른쪽 버튼 클릭으로 드래그해 확대/축소할 수 있습니다.
- 4. 우측 상단에 있는 버튼을 이용해 다음과 같은 작업을 수행할 수 있습니다.
 - 곡선 위 모든 포인트들을 한 프레임에 넣습니다.

- 크기 조절이 가능한 별도의 창에서 곡선을 엽니다.
- 옵션 메뉴를 엽니다. 이것은 곡선 편집기 배경을 마우스 오른쪽 버튼으로 클릭할 때 표시되는 메뉴와 동일합니다.
- 5. 곡선 아래 있는 컨트롤들을 이용해 다음과 같은 작업을 수행할 수 있습니다.
 - 왼쪽에서 오른쪽 순으로 포인트를 선택합니다.
 - 선택한 포인트를 편집합니다.
 - 값 또는 다중 포인트들의 보간을 설정합니다.

어레이 작업

어레이는 인덱싱한 값들의 목록입니다. 예를 들어, 메쉬의 point_position 특성은 float3 값들의 어레이로 정점 개수로 인덱싱한 것입니다.

어레이에서 인덱스를 지정함으로써 개별 값을 얻거나 설정할 수 있습니다. Bifrost에서의 어레이는 *0 기준 인덱싱*을 이용합니다. 즉, 크기가 5인 어레이를 예로 들면, 첫 번째 값의 인덱스는 1이 아닌 0이며 마지막 값의 인덱스는 4입니다.

어레이 포트

포트의 유형이 어레이인 경우, 그 모양이 더 넓어서 노드에 닿기 때문에 측면이 실크 모자와 비슷한 형태를 띕니다.



어레이 포트 중에는 *팬인(fan-in)* 연결이 가능한 것도 있습니다. 즉, 단일 요소나 어레이의 다중 연결을 연결할 수 있습니다. 연결된 모든 입력은 단일 입력 어레이로 자동으로 연결되므로 build_array 노드를 이용하는 것과 비슷합니다.

- 연결을 메인 포트 아이콘에 갖다 놓는 방식으로 새로운 입력을 추가할 수 있습니다.
- 포트를 확장해 각각의 연결을 확인할 수 있으며, 이를 통해 특정 입력의 연결을 해제할 수 있습니다.



어레이 작업을 하기 위한 노드

Core::Array 네임스페이스에는 어레이 작업을 위한 노드가 다수 존재합니다. 뿐만 아니라, for-each 및 iterate 컴파운드를 이용해서 한 어레이의 값들을 반복 실행할 수 있습니다.

자동 루핑

대부분의 경우, 값들을 처리하기 위해 어레이를 반복 실행할 필요는 없습니다. 대부분의 노드들이 필요할 때 어레이의 모든 요소들을 알아서 계속 처리할 것이기 때문입니다.

예:

• 두 단일 값을 입력으로 add 노드에 연결하는 경우, 출력으로 단일 값을 얻게 됩니다. 이 경우에는 자동 루핑이 필요하지 않습니다.

- 단일 값 1개와 어레이 1개를 add 노드에 연결하는 경우, 이 단일 값은 어레이의 모든 값에 자동으로 더해집니다. 이 경우 합계를 포함하는 또 다른 어레이가 출력됩니다.
- 어레이 2개를 add 노드에 연결하는 경우, 그에 해당하는 어레이의 값들이 모두 더해지며 이 어레이가 출력됩니다. 어레이들의 크기가 서로 다른 경우, 출력 어레이는 더 작은 입력의 크기로 줄어듭니다.
컴파운드 생성 및 편집

컴파운드는 다른 노드들로 구성된 부분 그래프를 포함하는 노드로서, 다른 컴파운드를 포함할 수도 있습니다. 컴파운드로 대형 그래프를 체계화 및 단순화해 가독성을 높일 수 있습니다. 또, 재사용하도록 개시할 수도 있습니다.

이 섹션의 페이지들

- 컴파운드 생성하기
- 참조된 컴파운드 가져오기
- 포트 그룹 생성하기
- 어레이를 위한 팬인(fan-in) 포트 생성하기
- 컴파운드 게시하기
- 추가 컴파운드 위치 설정하기

컴파운드 생성하기

- 1. 한 그래프 안에서, 또는 다른 컴파운드 안에서 노드를 1개 이상 선택합니다.
- 2. Ctrl+G를 누릅니다.

선택된 노드가 새로운 컴파운드에 포함되고 기존의 모든 연결이 포트가 됩니다.

- 컴파운드의 이름을 더블 클릭해 이름을 변경합니다.
- 포트를 마우스 오른쪽 버튼으로 클릭해 이름을 변경하거나 옮길 수 있으며, 데이터 유형이나 기타 특성을 설정할 수도 있습니다.
- 컴파운드를 더블 클릭해 안으로 들어가 내부의 부분 그래프를 편집합니다.

컴파운드 전개하기

컴파운드를 전개하는 것은 만드는 것과 정반대입니다. 컴파운드를 전개하면 한 레벨 안의 모든 내용이 풀어집니다. 하지만, 모든 연결은 그대로 유지됩니다.

- 1. 컴파운드(포트가 아님)를 마우스 오른쪽 버튼으로 클릭합니다.
- 2. **Explode**를 선택합니다.

🖋 참고:

컴파운드가 참조될 경우, 자동으로 불려온 다음 전개됩니다.

참조된 컴파운드 가져오기

Tab 메뉴를 이용해 한 그래프에 컴파운드를 추가하면, 해당 컴파운드가 노드 정의 파일에서 참조됩니다. 참조된 컴파운드는 좌측 상단에 마름모 아이콘으로 표시됩니다.



Parameter Editor에서 참조된 컴파운드의 입력에 값을 설정할 수도 있고 연결을 이용해서 입력 값들을 조종할 수도 있습니다.

또, 더블클릭하여 컴파운드 안에서 내부의 부분 그래프를 확인할 수도 있습니다. 하지만, 컴파운드를 먼저 가져오지 않는 한, 내부 노드의 값을 변경하는 것을 포함해 어떤 식으로도 내부 부분 그래프를 변경할 수 없습니다.

- 1. 참조된 컴파운드(포트가 아님)를 마우스 오른쪽 버튼으로 클릭합니다.
- 2. Make Editable을 선택합니다.

✔ 참고:

또 다른 참조 컴파운드에 중첩되어 있는 참조 컴파운드를 가져오려면, 먼저, 최상위 컴파운드부터 시작해 참조된 상위 컴파운드들을 모두 가져와야 합니다.

이렇게 가져온 컴파운드는 더 이상 참조되지 않습니다. 원래의 노드 정의가 수정되어 다시 게시되더라도 이 컴파운드는 업데이트되지 않습니다.

포트 그룹 생성하기

포트 그룹을 이용해서 포트를 축소 가능한 헤딩으로 조직화할 수 있습니다. 헤딩은 그래프 내 노드와 Parameter Editor에 모두 표시됩니다.

volume_scope		volume			
	× :	pr	obe_geometry		
	out_scope_geo 📘		Volume Sampling		
	volume		sampler	kCubicC0	•
	Probe_geometry = Volume Sampling		divisions	50	
	··· sampler divisions	▶	Flowline Display		
	+ Flowline Display	►	Color		
	+ Color + Point Display	▶	Point Display		
	- Numeric Display		Numeric Display		
	··· numeric_display		numeric_display		
	···· numeric_property – Culling		numeric_property	voxel_velocity	
· –	··· cull_by_property		Culling		
	culling_property	-	Multi-Res Display		
	cull_by_property_threshold		display_tile_tree	v	
	+ Multi-Res Display		visible_levels	1	
			frame_thickness	0.01	
			tile_transparency	0.009	

- 1. 컴파운드 밖에서, 포트를 마우스 오른쪽 버튼으로 클릭해 Add to Group > New를 차례로 선택합니다.
- 2. 이 그룹의 이름을 입력합니다.
- 다른 포트들을 마우스 오른쪽 버튼으로 클릭해 이미 만들어놓은 그룹에 추가하거나 혹은 별도의 그룹을 만들 수 있습니다.
 - 중첩 포트 그룹을 만들려면, 한 그룹에 속한 포트를 마우스 오른쪽 버튼으로 클릭해 해당 포트를 새 그룹에 추가하면 됩니다.

한 포트가 한 그룹에 속해 있을 때, 상황별 메뉴에 있는 **Move Port** 명령어는 해당 그룹 내 포트의 위치에 영향을 미칩니다.

그룹에서 포트를 제거하려면, 해당 포트를 마우스 오른쪽 버튼으로 클릭한 다음 Remove from Group을 선택합니다.

어레이를 위한 팬인(fan-in) 포트 생성하기

팬인(fan-in) 포트는 단일 값과 어레이들의 다중 입력 연결을 지원합니다. 이런 포트는 컴파운드가 먼저 어레이를 확실하게 구성할 필요 없이 다수의 항목을 한 어레이로 처리하기를 원할 때 유용합니다. 연결된 모든 입력은 단일 입력 어레이로 자동으로 연결되므로 build_array 노드를 이용하는 것과 비슷합니다.



- 포트의 데이터 유형이 명시적 유형의 어레이인지 확인합니다(auto가 아님). 포트 아이콘에 수직 "halo"가 있거나 노드에 닿을 정도로 넓지 않고 사각형이라면, 포트를 마우스 오른쪽 버튼으로 클릭하고 Set Value Type을 선택해 변경합니다.
- 2. 컴파운드 밖에서, 포트를 마우스 오른쪽 버튼으로 클릭해 Fan-In을 설정하거나 해제할 수 있습니다.



컴파운드 게시하기

컴파운드를 게시하면 이를 다른 그래프에서 손쉽게 재사용하고 다른 사용자들과 공유할 수 있습니다. 일단 컴파운드가 게시되면 설치된 여느 노드와 마찬가지로, Tab 메뉴를 이용해 해당 컴파운드의 인스턴스를 원하는 그래프에 추가할 수 있습니다.

게시된 컴파운드는 JSON 형식의 노드 정의 파일로 저장됩니다.

- 1. 컴파운드 밖에서, 컴파운드(포트가 아님)를 마우스 오른쪽 버튼으로 클릭해 Publish compound를 선택합니다.
- 2. 정보를 입력하고 Publish를 클릭합니다.

Save file:	C:/Users/username/Autodesk/Bifros	st/Compounds/ 🔻	Browse		
Compound namespace:	User::Compounds		•		
Compound name:	my_compound				
	Overloadable				
The compound will be w	rritten to a new file.	Publish	Cancel		

파일 저장하기

노드 정의 파일의 위치와 이름을 선택합니다.

드롭다운 박스에 포함되어 있는 위치 중 한 곳에 보관되어 있는 컴파운드들만 그래프에 추가할 수 있습니다. 기본 사용자 위치 외에, 별도의 컴파운드 위치를 설정할 수도 있습니다.

컴파운드 네임스페이스

모든 노드가 네임스페이스를 필요로 합니다. Tab 메뉴에는 노드들이 네임스페이스별로 분류되어 있습니다. 이 네임스페이스를 이용해 이름은 같지만 네임스페이스가 다른 노드들을 구분할 수 있습니다.

Compound name

여기에 입력하는 이름은 해당 컴파운드 인스턴스들의 기본 이름으로 사용됩니다. 이 이름은 Parameter Editor에서 컴파운드의 유형으로도 표시되는데, 컴파운드 인스턴스의 이름이 바뀌더라도 관계없습니다.

Overloadable

오버로드(overload)는 이름이 같지만 네임스페이스가 다른 노드를 말하나, 데이터 유형이 다른 경우에 설정할 수 있습니다.

대개는 게시된 컴파운드에 대해 overloadable을 설정할 필요가 없습니다. 이것은 auto 유형의 포트가 유형이 서로 다른 연결을 수락할 수 있기 때문입니다.

하지만, 오버로드가 유용한 경우도 있습니다. 예를 들어, 여러 부분 그래프들을 구현해 여러 유형에 대해 동일한 함수를 계산하도록 해야 할 때 오버로드 세트를 만들 수 있습니다. overloadable로 설정된 컴파운드는 auto 유형의 포트를 가질 수 없습니다. 뿐만 아니라, 다른 데이터 유형에 대해 지정하는 오버로드 노드 정의의 경우, 포트의 개수와 이름, 그리고 순서가 반드시 같아야 하며, 데이터 유형은 달라도 무방합니다.

추가 컴파운드 위치 설정하기

기본적으로, 컴파운드는 Bifrost 사용자 위치로 게시됩니다. 하지만, 필요에 따라 디렉토리 위치를 추가로 설정할 수도 있습니다. 이렇게 하면, 게시할 때 원하는 위치를 선택할 수 있습니다. Tab 메뉴를 이용해 이들 위치에 있는 모든 컴파운드를 그래프에 추가할 수 있습니다.

- 1. 컴파운드들을 게시하고자 하는 폴더의 상위 디렉토리에 JSON 구성 파일을 만듭니다.
- 2. 컴파운드 폴더의 상대 경로를 구성 파일의 jsonLibs 섹션에 추가합니다.
- 3. 구성 파일의 전체 경로(파일 이름과 확장자 포함)를 BIFROST_LIB_CONFIG_FILES 환경 변수에 추가합니다.
- 4. Bifrost 플러그인을 다시 시작합니다.

샘플 구성 파일

구성 파일은 아래의 예시처럼 단순할 수 있습니다. 단, 컴파운드의 추가 폴더만 정의하는 경우에 한합니다. 경로들은 구성 파일의 자체 위치를 기준으로 합니다.

아래 예시에서는 해당 위치에서 두 하위 폴더 즉, compounds와 test를 정의합니다. 여기서 libraryName과 libraryVersion 항목을 사용해 컴파운드 파일을 로드할 때 발생하는 정보와 오류를 기록합니다.

```
{
  "AminoConfigurations": [
    {
      "libraryName": "My_Compounds",
      "libraryVersion": "3.1.4",
      "jsonLibs": [
        {
          "path": "compounds",
          "files": [ ]
        },
        {
          "path": "tests",
          "files": [ ]
        }
      1
    }
  ]
}
```

프로그래밍 루프 작업

루핑 컴파운드에는 세 가지 유형이 있습니다.

- iterate 루프는 복수의 요소들을 순차 처리합니다.
- for_each 루프는 복수의 요소들을 병렬로 처리합니다. 대개는 어레이를 연결할 때 대부분의 노드가 자동 루핑을 수행하기 때문에 for_each 루프를 이용할 필요가 없습니다.
- do_while 루프는 조건이 충족될 때까지 부분 그래프를 반복합니다. 단, 조건이 거짓인 경우에도 부분 그래프는 항상 한 번만 실행됩니다.

루핑 컴파운드 상의 특수 포트 유형

루핑 컴파운드 상의 포트들은 특별한 기능을 수행할 수 있습니다. 컴파운드 내 입력 또는 출력 노드에 있는 포트를 마우스 오른쪽 버튼으로 클릭해 이 기능을 설정하거나 해제할 수 있습니다.

반복 카운터

반복 카운터는 모든 루프에서 이용할 수 있습니다. 따라서 컴파운드 내 계산에서 현재 반복의 인덱스를 이용할 수 있습니다.

그래프에 루핑 컴파운드를 추가하면, <u>current_index</u>라는 이름의 기본 반복 카운터가 생성됩니다. 반복 카운터는 추가로 만들 수도 있습니다. 각 카운터는 매 반복 작업마다 독립적으로 값이 증가합니다.

반복 인덱스는 기본적으로 0에서 시작합니다. 컴파운드 밖에 있는 포트를 연결하거나 Parameter Editor에서 값을 설정함으로써 인덱스를 0이 아닌 값으로 초기화할 수 있습니다. 하지만, 인덱스를 0이 아닌 값으로 초기화하더라도 루프가 실행되는 횟수는 아무런 영향을 받지 않습니다.

< 반복 제한

반복 제한은 모든 루프에서 이용할 수 있습니다. 이 기능은 루프가 실행되는 횟수를 제한합니다.

그래프에 루핑 컴파운드를 추가하면, max_iterations라는 이름의 기본 반복 제한이 생성됩니다. 반복 제한은 추가로 설정할 수도 있습니다.

루프가 실행되는 실제 횟수는 다음 중 더 낮은 쪽으로 정해집니다.

- 반복 제한 포트의 값
- 반복 대상 포트에 연결되는 어레이 입력의 크기

诸 반복 대상

반복 대상은 for_each와 iterate 루프에서 이용할 수 있습니다. 이들 포트는 컴파운드 밖에서 어레이 형태로 나타나지만 내부의 단일 값입니다. 컴파운드는 어레이의 각 요소마다 한 번씩 평가를 수행합니다.

단, 더 작은 크기의 어레이가 다른 반복 대상에 연결되거나 반복 제한이 더 작은 값을 가질 때는 반복 횟수가 줄어들 수 있습니다.

🕑 State 포트

State 포트는 iterate와 do_while 루프에서 이용할 수 있으며, 이를 통해 누적 값을 한 반복 작업에서 다음 반복 작업으로 전달할 수 있습니다. 이 포트는 서로 대응하는 한 쌍의 입력 포트와 출력 포트로 이루어져 있어 출력 값이 루프의 다음 반복 작업의 입력으로 적용됩니다. State 포트를 만들 때, 대응하는 포트를 반드시 지정해야 합니다. 입력 값 또는 연결은 첫 번째 반복 작업의 초기 값으로 취급됩니다.

이 섹션의 페이지들

- iterate 루프 만들기
- for_each 루프 만들기
- do_while 루프 만들기

iterate 루프 만들기

iterate 루프는 복수의 요소들을 순차 처리합니다. 예를 들어, 반복 대상 입력 포트에 연결된 값들로 구성된 어레이의 곱을 계산하는 데 iterate 루프를 이용할 수 있습니다.

- 1. 그래프에 iterate 노드를 추가합니다.
- 2. 선택적으로, New Port 아이콘으로 연결을 끌어와 원하는 포트를 만들 수 있습니다. 포트를 마우스 오른쪽 버튼으로 클릭해 이름이나 기타 사항을 변경합니다.
- 3. 컴파운드로 들어가 루프에서 실행하려는 계산에 대한 부분 그래프를 만듭니다.
 - 컴파운드가 한 어레이의 모든 항목들에 대해 작업을 반복하도록 설정하려면, 입력 어레이 포트를 마우스 오른쪽 버튼으로 클릭한 다음, Set Port Iteration Target > true를 차례로 선택합니다.
 N번째 루프 반복이라면, 입력 반복 대상의 값은 연결된 어레이의 N번째 항목이 됩니다.
 - 입력 반복 목표를 1개 이상 설정했다면, max_iterations 포트를 삭제하는 것이 안전합니다. 그러면, 해당 어레이의 크기를 연결하지 않더라도 모든 항목이 처리됩니다.
 - 또, 출력 포트를 반복 대상으로 설정할 수도 있습니다. 값은 N번째 항목이 N번째 반복의 결과인 어레이입니다.
 - 한 반복 작업에서 다음 반복 작업으로 부분 합과 같은 중간 값을 전달하려면, "state" 포트 1쌍을 이용합니다. State 포트는 서로 대응하는 한 쌍의 입출력 포트로 구성되므로 출력 값이 루프의 다음 반복 작업의 입력으로 전달됩니다. State 포트 1쌍을 만들려면, 컴파운드의 input 또는 output 노드의 포트를 마우스 오른쪽 버튼으로 클릭한 다음, Set Port State 메뉴를 펼쳐 해당하는 포트를 선택합니다. 선택적으로, 루프 컴파운드 밖에서 값을 연결하거나 설정함으로써 첫 번째 반복 작업을 위해 입력 값을 초기화할 수 있습니다.

State 포트나 반복 대상이 아닌 출력은 모두 "auxiliary"입니다. 이 출력 값은 단순히 루프의 마지막 반복 작업에서 계산된 결과입니다.

상위 페이지: 프로그래밍 루프 작업

for_each 루프 만들기

for_each 루프는 복수의 요소들을 병렬로 처리합니다.

대부분의 경우, *자동 루핑*을 이용해 어레이를 다른 어레이나 단일 값과 결합할 수 있기 때문에 for_each 루프가 필요하지 않습니다. 예:

- 단일 값을 어레이에 추가하면 해당 값이 어레이의 각 요소에 더해집니다.
- 한 어레이를 또 다른 어레이에 추가하면 각 어레이의 해당 항목들이 더해집니다. 어레이마다 크기가 서로 다를 경우, 결과는 더 작은 쪽으로 줄어듭니다.

하지만 for_each 루프가 필요한 경우도 있는데 가령, 계산 과정에서 반복 인덱스를 이용해야 하는 경우를 수 있습니다.

- 1. 그래프에 for_each 노드를 추가합니다.
- 2. 선택적으로, New Port 아이콘으로 연결을 끌어와 원하는 포트를 만들 수 있습니다. 포트를 마우스 오른쪽 버튼으로 클릭해 이름이나 기타 사항을 변경합니다.
- 3. 컴파운드로 들어가 루프에서 실행하려는 계산에 대한 부분 그래프를 만듭니다.
 - 컴파운드가 한 어레이의 모든 항목들에 대해 작업을 반복하도록 설정하려면, 입력 어레이 포트를 마우스 오른쪽 버튼으로 클릭한 다음, Set Port Iteration Target > true를 차례로 선택합니다.
 N번째 루프 반복이라면, 입력 반복 대상의 값은 연결된 어레이의 N번째 항목이 됩니다.
 - 입력 반복 대상을 1개 이상 설정했다면, max_iterations 포트를 삭제하는 것이 안전합니다. 그러면, 해당 어레이의 크기를 연결하지 않더라도 모든 항목이 처리됩니다.

모든 출력 포트가 반복 대상입니다. 값은 N번째 항목이 N번째 반복의 결과인 어레이입니다.

상위 페이지: 프로그래밍 루프 작업

do_while 루프 만들기

do_while 루프는 조건을 충족하는 한 계산을 반복합니다. 예를 들어, 오차가 허용 오차범위를 벗어나지 않는 한, 값의 근사치를 반복해서 계산합니다.

- 1. 그래프에 do_while 노드를 추가합니다.
- 2. 선택적으로, New Port 아이콘으로 연결을 끌어와 원하는 포트를 만들 수 있습니다. 포트를 마우스 오른쪽 버튼으로 클릭해 이름이나 기타 사항을 변경합니다.
- 3. 컴파운드로 들어가 루프에서 실행할 계산에 대한 부분 그래프를 만듭니다.
 - output 노드의 looping_condition 포트는 기본적으로 false로 설정됩니다. 계산(루프 반복을 추가로 수행할 것인지를 결정)의 출력을 연결합니다. do_while 루프는 항상 적어도 한 번 이상 평가를 수행하며 looping_condition을 이용해서 다시 평가할 것인지를 결정합니다.
 - 모든 유형의 do_while 루프는 적어도 한 쌍 이상의 "state" 포트를 필요로 합니다. State 포트는 서로 대응하는 한 쌍의 입출력 포트로 구성되므로 출력 값이 루프의 다음 반복 작업의 입력으로 전달됩니다. State 포트 1쌍을 만들려면, 컴파운드의 input 또는 output 노드의 포트를 마우스 오른쪽 버튼으로 클릭하고 Set Port State 메뉴를 펼쳐 해당하는 포트를 선택합니다. 선택적으로, 루프 컴파운드 밖에서 값을 연결하거나 설정함으로써 첫 번째 반복 작업을 위해 입력 값을 초기화할 수 있습니다.
 - 실제 계산 과정에서 이용하지 않더라도 max_iterations 입력 포트를 삭제해서는 안 됩니다. 이 포트는 looping_condition의 입력이 항상 참인 경우에 무한 루프를 방지합니다. 컴파운드 밖에서 max_iterations의 값을 설정하거나 연결할 수 있습니다.

State 포트가 아닌 출력은 모두 "auxiliary"입니다. 이 출력 값은 단순히 루프의 마지막 반복 작업에서 계산된 결과입니다.

상위 페이지: 프로그래밍 루프 작업

행렬과 변환 작업

matrix_multiply 노드를 이용해 행렬을 곱하거나 행렬과 벡터를 곱할 수 있습니다.

🖋 참고:

행렬들의 차수가 서로 일치하는지 확인해야 합니다. 그렇지 않을 경우 예상치 못한 결과가 나올 수 있습니다. 예를 들어, 4×4 행렬과 3×3 행렬을 곱하는 경우, 3×3 행렬이 4×3 행렬로 자동 프로모션되므로 4×3 행렬이 출력됩니다.

Bifrost는 열 벡터를 이용하므로, 행렬과 벡터를 곱하면 행렬이 앞에 오고 벡터가 뒤에 오게 됩니다(사후 곱셈). p가 객체의 기준 프레임으로부터의 위치이고, L이 객체의 로컬 변환 행렬, W가 객체의 상위 월드 변환 행렬이라면, 월드 좌표에서의 위치는 다음과 같이 표현할 수 있습니다.



행 백터를 사용하는 소스에서 공식을 조정하는 경우에는 역순으로 곱해야 합니다. 또한 행렬의 순서가 아직 바뀌지 않은 경우 바꿔야 합니다.

기하학적 객체에서 특성 확인 및 설정

메쉬나 볼륨과 같은 기하학적 객체에서 point_position이나 voxel_fog_density와 같은 특성을 확인하고 설정할 수 있습니다. 또, 자신만의 특성을 만들어 사용자 효과를 제어할 수도 있습니다.

특성 설명

일반적으로 특성은 객체 상의 키값 쌍입니다. 키는 이름(문자열)이며 사실상 무엇이든 값이 될 수 있습니다. 대부분의 경우, 값은 또 다른 객체가 될 수도 있습니다.

특히, 기하학적 객체는 geo 특성을 가질 수 있습니다. point_position과 voxel_fog_density 둘 다 geo 특성의 예입니다. geo 특성값은 그 자체로 자신만의 특성을 갖고 있는 객체이며, 다음과 같은 것들이 있습니다.

- 특성의 대상이 되는(즉, 특성과 연결된) 기하학적 구성요소. 예를 들어, point_position, point_normal, point_size는 모두 point_component 를 대상으로 하는 반면, voxel_fog_density와 voxel_temperature 는 voxel_tile_tree 를 대상으로 합니다.
- 특성의 기본값.
- 특성의 데이터, 예를 들어, float3 포인트 위치들의 어레이.

특성을 확인하고 설정하기 위한 노드

특성을 확인하고 설정할 수 있는 노드가 몇 가지 있습니다. 거의 대부분, get_geo_property와 set_geo_property_data를 이용해야 합니다. 예를 들어 메쉬를 변형하려면 먼저 get_geo_property를 이용해 point_position 데이터를 구한 다음 값들을 수정하고 마지막으로 set_geo_property_data를 이용해 값들을 객체에 저장합니다.

get_geo_property

get_geo_property를 이용해 geo 특성값들의 어레이를 반환합니다. 또, 대상 구성요소의 기본값과 이름도 반환합니다.

get_geo_property_check</mark>라고 하는 변형된 노드도 있는데, 이것은 추가로 Boolean 값을 반환해 해당 특성을 찾았는지 여부를 알려줍니다.

set_geo_property_data

set_geo_property_data를 이용해 기존 geo 특성의 데이터 어레이를 업데이트합니다. 이것은 set_geo_property보다 편리한데, 이는 기본값이나 대상 구성요소를 지정할 필요가 없기 때문입니다.

set_geo_property

set_geo_property 를 이용해 geo 특성을 생성하고 초기 데이터값을 설정합니다. 원하는 용도에 맞게 사용자 geo 특성을 생성하고 수정할 수 있습니다. geo 특성이 생성되면, get_geo_property와 set_geo_property_data 를 이용해 그 특성을 확인하고 설정할 수 있습니다.

get_property, set_property

get_property와 set_property를 이용해 원하는 특성을 확인하고 설정합니다. geo 특성 이외에 다른 특성도 가능합니다. 값의 유형이 올바른지 꼼꼼히 확인합니다.

이 섹션의 페이지들

• 여러 위치에서 geo 특성 얻기

여러 위치에서 geo 특성 얻기

한 메쉬 객체 상의 여러 위치에서 geo 특성 값들을 얻을 수 있습니다. 예를 들어, 한 메쉬의 정점에 컬러 값이 저장되어 있다면, 메쉬의 어느 곳에서든 그 값을 보고 보간된 컬러 값을 얻을 수 있습니다.

- 1. 위치 객체를 만듭니다.
 - get_closest_locations 를 이용해 한 지오메트리에서 여러분이 지정하는 위치와 가장 가까운 위치를 파악합니다.
 - generate_sample_locations 를 이용해 지오메트리 상에 임의의 위치를 만듭니다.
- 최종 위치와 지오메트리를 연결해 sample_property 노드를 조회합니다. 찾아볼 특성 이름을 지정하고 그 특성을 찾지 못하거나 위치가 유효하지 않은 경우(예를 들어, get_closest_locations)를 이용할 때 위치가 컷오프 거리를 벗어난 경우)에 사용할 기본 값을 지정합니다.

상위 페이지: 기하학적 객체에서 특성 확인 및 설정

맞춤형 시뮬레이션 만들기

컴파운드의 피드백 포트를 이용해 한 부분 그래프의 최종 평가 결과를 캐싱하고 이를 다음 평가를 위한 입력으로 삼을 수 있습니다. 이것이 바로 시뮬레이션의 작동 방식이며 한 프레임의 결과가 그 다음 프레임의 기초가 됩니다.

피드백 포트는 한 컴파운드에서 서로 대응하는 입력 포트와 출력 포트 1쌍으로 구성되어 다음에 해당 컴파운드나 전체 그래프를 평가할 때 출력값이 입력으로 적용됩니다.

이 작동 방식의 예를 보려면, <u>simulation_example</u> 컴파운드 안을 들여다보십시오. 이것은 일반 시뮬레이션의 템플릿 역할을 하며 이를 편집 가능한 상태로 만들거나 요건에 맞게 수정할 수 있습니다.

한 컴파운드에 피드백 포트 1쌍을 만드는 방법:

- 컴파운드에 유형이 동일하여 서로 대응하는 입력 포트와 출력 포트가 있는지 확인합니다. 단, auto 포트는 지원되지 않습니다. 필요하다면, 포트를 마우스 오른쪽 버튼으로 클릭해서 Set Value Type을 선택하고 명확한 유형으로 설정합니다.
- 컴파운드 안에서, 입력 포트와 출력 포트 중 한 쪽을 마우스 오른쪽 버튼으로 클릭해 Set Feedback
 Port를 선택하고 그에 대응하는 포트를 선택합니다.

최상의 결과를 위해, 사용자 시뮬레이션은 다음과 같아야 합니다.

- 첫 프레임에서 시뮬레이션을 초기 상태로 재설정합니다.
- 시뮬레이션이 진행되어야 하는지 여부(예를 들어, 현재 프레임이 마지막으로 캐싱된 프레임보다 1 큰 경우)를 결정합니다.
- 다른 경우에 일어나는 상황, 예를 들어 최종 결과를 통과해야 하는지 아니면 현재 프레임이 마지막으로 캐싱된 프레임보다 1 크기 않을 때에는 재설정해야 하는지를 결정합니다.

파일에서 나온 입력 및 출력 데이터

지오메트리를 호스트 장면으로 입력 및 출력시키는 대안으로, 파일을 입력하거나 출력할 수 있습니다. 이것은 예를 들어 팜(farm)을 캐싱하거나 렌더링할 때 유용합니다. Bifrost의 네이티브 객체 형식 외에도, 일부 표준 형식을 위한 read_* 노드와 write_* 노드가 있습니다.

이용할 형식은 지오메트리의 유형에 따라 달라집니다.

- Bifrost 바이너리 객체 형식(.bob)은 Bifrost에서 모든 유형의 지오메트리 유형에 이용할 수 있습니다. 하지만, 이 형식은 Bifrost 그래프로만 읽을 수 있습니다.
- 메쉬 지오메트리에는 Alembic (.abc)를 이용합니다. 메쉬는 파일당 1개씩만 지원됩니다.
- 볼륨 또는 포인트에는 OpenVDB (.vdb)를 이용합니다.
- 볼륨에는 Field3d (.f3d)를 이용합니다.
- 포인트에는 PDC (.pdc)를 이용합니다.

팁:

file_cache 노드로 시도해볼 수도 있습니다. 이것은 확장자에 따라 .bob, .abc, .vdb를 읽고 쓸 수 있는 실험용 노드입니다.

참고:

- 그래프 최상위 레벨의 output에는 파일의 출력이 적어도 1개 이상 직접 또는 간접적으로 연결되어야 합니다.
- time 노드의 출력을 frame 입력에 연결할 수 있습니다. 경우에 따라, to_int 를 이용해서 프레임 값을 float에서 int 로 변환하거나, 또는 one_over 를 이용해서 frame_step을 프레임 속도로 변환해야 할 수도 있습니다.
- 파일 이름에 프레임 번호를 포함시키려면, 토큰 #을 이용합니다. 프레임 패딩을 위해 다수의 문자를 이용할 수 있는데, 이를 테면, 0001, 0002 등을 위해 ####을 이용할 수 있습니다. 물론, @도 이용할 수 있습니다.
- 포함시킬 특성을 지정할 때는 명확히 지정하거나 모든 항목에 대해 📲 입력할 수도 있습니다.
- 디렉터리가 있는지 확인합니다.

그래프 컴파일과 실행 작업 일시중지

경우에 따라 큰 그래프로 작업할 때, 각 사항을 변경한 후에 그래프가 다시 컴파일되거나 실행될 때까지 기다리고 싶지 않을 때가 있습니다. 이 경우, Ctrl+.(마침표)를 누르거나 Edit > Pause/Resume Graph Execution을 이용해 그래프 컴파일을 일시중지했다가 다시 시작하면 됩니다.

그래프가 일시중지되면, Bifrost Graph Editor 우측 상단에 표시가 생깁니다.



그래프 진단 및 디버깅

복잡한 그래프 작업을 하다 보면, 진단과 디버깅이 필요한 문제가 발생할 수 있습니다. 이런 작업에 도움이 될 수 있는 몇 가지 기본 도구를 소개합니다.

• 진단 색상은 그래프에서 문제가 있는 위치를 알려줍니다. Info Bar 안의 메시지에서 자세한 정보를 확인할 수 있습니다.

진단 노드 색상

문제가 있는 노드는 서로 다른 색상으로 강조표시됩니다. 강조표시된 노드가 컴파운드라면, 안으로 들어가서 문제가 있는 내부 노드를 살펴볼 수 있습니다.



- 오류가 있는 노드는 빨간색으로 강조표시됩니다. 오류가 있다는 것은 해당 그래프를 컴파일하거나 실행할 수 없음을 의미합니다.
- 경고가 있는 노드는 주황색으로 강조표시됩니다. 경고는 문제가 있음을 의미하지만, 그래프를 컴파일하거나 실행하는 것은 가능한 상태입니다.

노드 위로 포인터를 가져가면 Info Bar에서 전체 메시지를 확인할 수 있습니다.

연결에 와치포인트 설정하기

와치포인트를 이용하면 연결을 따라 흐르는 값들의 정보를 확인할 수 있습니다.



와치포인트를 추가하려면, 연결을 마우스 오른쪽 버튼으로 클릭하고 Set Watchpoint를 선택합니다.

연결을 따라 눈 모양 아이콘을 드래그해 와치포인트를 옮길 수 있습니다. 눈 모양 아이콘을 마우스 오른쪽 버튼으로 클릭해 와치포인트를 축소 또는 제거합니다.

와치포인트를 추가한 후에는 해당 그래프가 다시 실행되기 전까지는 비어 있는 상태가 됩니다. 입력을 변경하거나 시뮬레이션에서 프레임을 전진하여 그래프를 실행할 수 있습니다.

와치포인트는 데이터가 연결을 따라 흐를 때만 업데이트됩니다. 그래프 출력으로 이어지지 않은 연결이 있거나, if 노드 때문에 평가되지 않았다면 와치포인트는 업데이트되지 않습니다.

와치포인트에 나타나는 정보는 데이터 유형에 따라 달라집니다.

- 단일 값인 경우, 와치포인트는 마지막 실행 값과 더불어, 현재 세션에서 나타난 최소값과 최대값을 보여줍니다.
- 어레이인 경우, 와치포인트는 어레이 크기, 그리고 마지막 실행의 최소값과 최대값을 보여줍니다.
- 객체인 경우, 와치포인트는 지정되어 있는 특성의 정보를 보여줍니다.

객체 데이터를 텍스트 파일로 덤핑하기

와치포인트가 제공하는 객체 정보가 충분하지 않을 경우, 해당 객체의 정보를 텍스트 파일로 출력할 수 있습니다.

- 1. 그래프에 dump_object 노드를 추가합니다.
- 2. 텍스트 파일의 folder 및 filename 매개변수를 설정합니다. 확장자 .txt가 자동으로 붙습니다.
- 3. sample_size는 어레이에서 포함시키려는 항목들의 최대 개수로 설정합니다. 그러면, 객체가 너무 많아 파일이 비대해지는 것을 막을 수 있습니다.
- 4. 텍스트로 출력하려는 객체를 object 입력에 연결합니다.
- 5. out_object 출력을 그래프의 출력으로 연결합니다. 보통은 상류 노드에 있는 원래의 연결을 간단히 대체할 수 있습니다.



dump_object의 출력 중 적어도 한 개가 그래프 출력에 연결되어야 하는데, 그렇지 않을 경우 노드가 실행되지 않습니다.

배경을 이용해 노드에 주석을 달고 구조화하기

배경(Backdrop)은 텍스트와 컬러가 적용된 배경으로, 그래프에서 노드에 주석을 달고 노드를 구성하는 데 이용할 수 있습니다. 배경을 드래그해 그 안에 있는 모든 노드들을 한 번에 옮길 수 있습니다.



배경 생성 방법:

- 1. 선택사항으로, 애초에 배경에 포함시킬 노드를 선택합니다.
- 2. 다음 중 한 가지 작업을 수행합니다.
 - 배경을 마우스 오른쪽 버튼으로 클릭해 Create Backdrop을 선택합니다.
 - 메뉴에서 Create > Backdrop을 차례로 선택합니다.
 - Tab 메뉴를 이용해 backdrop을 추가합니다.

배경이 완성되면, 다음과 같이 할 수 있습니다.

- 우측 하단에 있는 삼각형 모양을 드래그해 배경의 크기를 조절합니다.
- 삼각형을 더블클릭해 컬러를 변경합니다.
- 배경 안에서 더블클릭해 안으로 들어가 텍스트를 편집합니다. 화살표 키를 이용해 삽입점을 옮기고 Shift+화살표 키를 눌러 텍스트를 선택합니다. 작업을 마쳤으면 배경 바깥을 더블클릭합니다.
- 노드를 선택해 드래그하여 위치를 조정하거나 배경 안팎으로 옮깁니다.

효과를 그래프로 게시하기

효과 전체를 **Bifrost Browser**에 나타나는 그래프로 게시할 수 있습니다. 게시된 그래프는 게시된 컴파운드와 다릅니다. 게시된 컴파운드는 그래프 안에 포함시킬 수 있는 재사용 가능한 부분 그래프이지만, 게시된 그래프는 그래프의 최상위 레벨 및 호스트 장면과의 연결을 포함합니다.

게시된 그래프는 썸네일 이미지나 장면의 예와 같은 지원되는 파일을 포함할 수도 있습니다.

팁:

그래프에 메쉬와 같은 장면 지오메트리 입력이 있고 그 그래프를 가져올 장면에 이름이 동일한 객체가 있다면, 자동으로 연결됩니다.

그래프를 게시하는 방법:

- 1. 그래프 최상위 레벨에서, Create > Publish Graph를 차례로 선택합니다.
- 2. 원하는 정보를 입력합니다.
 - Graph name: Bifrost Browser에 나타날 그래프의 이름입니다. 이것은 그래프를 가져올 호스트 장면에서 Bifrost 그래프 노드의 기본 이름으로도 사용됩니다.
 - Graph namespace: 모든 그래프가 네임스페이스를 필요로 합니다.
 - Browser category: Bifrost Browser에서 그래프가 나타날 카테고리입니다. 카테고리가 없는 그래프는 목록에 포함되지 않습니다.
 - Author: 작성자의 이름이며, 선택 사항입니다. 이것은 Bifrost Browser에서 정보 아이콘을 클릭할 때 나타납니다.
 - Description: 설명 입력란이며 선택 사항입니다. 이 역시 정보 아이콘을 클릭하면 나타납니다.
- 3. 지원되는 파일을 선택합니다(선택 사항). 이 파일들은 공유하기 쉽도록 게시 위치로 복사됩니다.
 - Thumbnail: Bifrost Browser에 나타날 이미지입니다.
 - Documentation: 그래프에 대한 추가 정보를 제공할 때 이용할 수 있는 Markdown 파일입니다. 그래프 최상위 레벨에서 노드를 하나도 선택하지 않았을 때 나타납니다. GitHub-flavor Markdown 형식을 대부분 지원합니다.
 - Example Scene: Bifrost Browser에서 열어볼 수 있는 장면입니다.
- 4. 게시 옵션 설정하기:
 - Include connected materials: 그래프에서 별도 장면으로 참조한 장면 머리티얼이 있으면 모두 저장하고 그래프를 가져올 때 이들을 자동으로 다시 연결합니다.
 - Save sidecar files: Adjacent to compound 옵션은 모든 파일을 공통의 하위 폴더에 저장하므로 보다 손쉽게 압축하고 공유할 수 있습니다. In Autodesk-default folders는 그래프와 다른 파일들을 별도의 폴더에 넣습니다.
- 5. Save file 이름과 디렉터리를 선택합니다. 드롭다운 박스 목록에 있는 위치 중 한 곳에 보관되어 있는 컴파운드들만 Bifrost Browser에서 이용할 수 있습니다. 기본 사용자 위치 외에, 별도의 컴파운드 위치를 설정할 수도 있습니다. 기본 사용자 위치는 다음과 같습니다.
 - Windows: \Users\username\Autodesk\Bifrost\Compounds

- macOS: /Users/username/Autodesk/Bifrost/Compounds
- Linux: /home/username/Autodesk/Bifrost/Compounds

동적 효과 시뮬레이션

기본적인 파티클 시스템, 공기역학, 연소, 모래나 눈 등의 낟알 재질과 같은 동적 효과를 주기 위해 특수 솔버를 이용할 수 있습니다. 콜라이더뿐만 아니라 바람과 같은 효과에서도 모든 시뮬레이션을 비슷한 방식으로 제어할 수 있습니다.

이 섹션의 페이지들

- 파티클 시뮬레이션 연출
- 공기 및 연소 시뮬레이션 연출
- MPM 시뮬레이션 연출
- 추가 방출 소스 설정
- 방출 특성의 변화
- 콜라이더(collider) 추가
- influence 적용

파티클 시뮬레이션 연출

파티클 시스템은 동적으로 시뮬레이션된 *포인트 객체*입니다. 포인트 객체는 포인트가 있는 지오메트리의 한 유형으로 대단히 제한적입니다. 반면, 파티클 시스템은 보통 point_velocity와 같은 다른 특성을 갖습니다. 파티클은 스파크, 폭죽, 마법 가루 등과 같은 것들을 시뮬레이션하는 데 이용됩니다.

포인트 객체는 *인스턴스*의 기반이기도 합니다. 인스턴스는 포인트의 포지션에 위치한 메쉬와 같은 지오메트리의 복제물입니다. 인스턴스는 지형에 나무와 잔디, 바위 등을 구현하는 등 다양한 용도를 갖고 있습니다.

이 섹션의 페이지들

• 파티클 시뮬레이션 설정

상위 페이지: 동적 효과 시뮬레이션

파티클 시뮬레이션 설정

여기서는 기본 파티클 시뮬레이션을 설정하고 이를 여러 가지 효과에 맞게 조정하는 방법에 대해 간략히 설명합니다.

- 1. 그래프의 최상위 레벨에서, <u>basic_particle_graph</u> 노드를 추가합니다. 이것은 단순히 기본 그래프를 포함하는 컴파운드이므로 입력이나 출력이 없습니다.
- 마우스 오른쪽 버튼을 클릭해 Explode를 선택합니다. 그러면 컴파운드의 내용이 펼쳐집니다. 이것은 simulate_particles 노드에 source_particles와 particle_solver_settings 노드가 연결된 상태로 구성되어 있습니다.



- 3. 지오메트리에서 파티클을 방출하려면, 메쉬나 기타 지원되는 기하학적 객체 1개 이상을 source_particles의 geometry 포트에 연결합니다.
- 4. <u>simulate_particles</u>의 <u>particles</u> 출력을 그래프의 <u>output</u> 노드에 새 포트로 연결합니다. 장면에 <u>bif</u>라는 이름의 Bifrost 객체가 생성됩니다.

- 5. 이때, 장면을 재생하여 소스 지오메트리에서 방출되는 파티클을 확인할 수 있습니다. 만족스러운 효과를 위해서는 추가적으로 편집해야 합니다.
 - 추가 방출 소스 설정
 - 방출의 특성 변화
 - 콜라이더 추가
 - influence 적용

상위 페이지: 파티클 시뮬레이션 연출

공기 및 연소 시뮬레이션 연출

공기 또는 공기역학 시뮬레이션은 기체의 움직임을 묘사합니다. 이를 통해, 연기나 분무, 증기 효과 등을 연출할 수 있습니다. 연소 시뮬레이션은 연료의 연소 현상을 묘사하며, 이를 이용해 불이나 폭발 효과 등을 연출할 수 있습니다.

이 섹션의 페이지들

- 공기 또는 연소 시뮬레이션 설정
- 공기 및 연소 시뮬레이션에 음영 적용
- 공기 및 연소 시뮬레이션의 상세도 향상
- 공기와 연소 시뮬레이션을 위한 적응성 설정
- 연소 시 불꽃과 연기 제어

상위 페이지: 동적 효과 시뮬레이션

공기 또는 연소 시뮬레이션 설정

여기서는 연소 시뮬레이션을 설정하고 이를 여러 가지 효과에 맞게 조정하는 방법에 대해 간략히 설명합니다.

- 1. 그래프의 최상위 레벨에서, basic_aero_graph 또는 basic_combustion_graph 노드를 추가합니다. 이것은 단순히 기본 그래프를 포함하는 컴파운드이므로 입력이나 출력이 없습니다.
- 2. 마우스 오른쪽 버튼을 클릭해 **Explode**를 선택합니다. 그러면 컴파운드의 내용이 펼쳐집니다. 이것은 아래의 컴파운드들을 포함하는 부분 그래프로 구성되어 있습니다.
 - source_air는 방출된 기체의 특성을 정의합니다.
 - source_fuel은 연소할 연료의 특성을 정의합니다(연소만 해당).
 - collider는 공기 흐름의 방해물을 만들 때 사용됩니다.
 - aero_solver_settings는 시뮬레이션의 여러 가지 측면을 제어합니다.
 - combustion_settings는 연료의 연소 방식을 제어합니다(연소만 해당).
 - simulate_aero는 시뮬레이션을 컴퓨팅합니다.
- 3. 메쉬나 볼륨 1개 이상을 source_air 노드의 geometry 포트에 연결함으로써 방출하는 객체를 정의합니다.
- 4. 선택사항으로, 메쉬나 볼륨 1개 이상을 collider 노드의 geometry 포트에 연결해 장애물을 정의합니다.
- 5. <u>simulate_node</u>의 <u>out_aero_body</u> 출력을 그래프의 <u>output</u> 노드에 새 포트로 연결합니다. 장면에 <u>bif</u>라는 이름의 Bifrost 객체가 생성됩니다.
- 6. <a>aero_solver_setting 노드에서 boost_detail_with_points가 작동중일 때만 <a>out_aero_point_body 객체에 데이터가 생성됩니다. 공기 및 연소 시뮬레이션의 상세도 향상을 참조합니다.

이때, 장면을 재생하여 소스 지오메트리에서 방출되는 모습을 확인할 수 있습니다.

하지만, 시뮬레이션을 올바로 미리보기 하려면, 공기 및 연소 시뮬레이션에 음영(shading) 적용을 참조합니다. 여기서 원하는 효과를 완벽히 구현하려면 추가적인 조정이 필요합니다.

공기 및 연소 그래프 수정

노드에서 단순히 매개변수만 조율하는 것 외에도, 효과를 수정할 수 있는 몇 가지 방법이 있습니다.

- 적응형 시뮬레이션에서는 선택한 영역에서 상세도를 낮춤으로써 컴퓨팅 시간과 메모리 사용량을 줄일
 수 있습니다. 공기와 연소 시뮬레이션을 위한 적응성 설정을 참조하십시오.
- 연소 그래프인 경우, <u>source_air</u>와 <u>simulate_aero</u> 노드 사이에 다수의 <u>source_fuel</u> 노드를 데이지 체인 방식으로 연결해 여러 가지 연료의 혼합물을 만들 수 있습니다.

모든 시뮬레이션 그래프에 적용되는 수정 항목도 있습니다.

- 추가 방출 소스 설정
- 방출의 특성 변화
- 콜라이더 추가
- influence 적용

공기 및 연소 시뮬레이션에 음영 적용

기본 장면 재질은 공기와 연소 시뮬레이션을 렌더링하기에 적합하지 않습니다. 볼륨 쉐이더(volume shader)를 적용하면 이 문제를 해결할 수 있습니다.

- 1. 장면에서, 지원되는 렌더러용 볼륨 쉐이더(예: Arnold용 aiStandardVolume 쉐이더)를 이용해 재질을 만듭니다.
- 2. 이 재질을 Bifrost 출력 객체(기본 이름은 "bif")에 적용합니다.
- 3. 쉐이더 설정에서, 밀도용 채널을 voxel_fog_density 로 설정합니다.
- 4. 다음 중 한 가지를 수행합니다.
 - 블랙바디 방사가 없는 단순한 공기 시뮬레이션인 경우, 방출을 해제합니다.
 - 연소일 경우, 블랙바디 방출용 채널을 voxel_temperature로 설정합니다. 채널의 곱수를 약 1.0으로 설정하고, 방출 강도를 원하는 대로 조절합니다. aiStandardVolume 쉐이더에서는 이런 매개변수로 Blackbody Kelvin과 Blackbody Intensity가 있습니다.

공기 및 연소 시뮬레이션의 상세도 향상

몇 가지 방법을 이용해 공기 및 연소 시뮬레이션의 상세도를 높일 수 있습니다.

디테일 사이즈 줄이기

source_air 노드에 설정된 fluid_detail_size는 표현할 수 있는 기류의 최소 디테일 사이즈를 결정합니다. 이 값을 낮추면 시뮬레이션의 해상도가 높아집니다.

디테일 사이즈에 영향을 미치는 또 다른 source_air 설정으로 geo_volume_mode 가 있습니다. 이것을 Absolute로 설정하면 디테일 사이즈는 월드 단위가 되어 이미터가 클수록 복셀 수가 많아집니다. 이것을 기본값인 Relative로 설정하면, 이미터가 크든 작든 복셀 수는 대략 같아집니다.

source_air의 geo_detail_size 및 collider 노드의 detail_size를 낮춰 방출과 충돌의 정확성을 높이는 방법도 있습니다.

🖋 참고:

다른 모드에서도 마찬가지지만 특히 Absolute 모드에서는 디테일 사이즈를 너무 많이 낮추지 않도록 주의합니다. 복셀이 너무 많아지면 메모리 사용량과 컴퓨팅 시간이 크게 늘어날 수 있기 때문입니다.

선명화(sharpening) 적용

시뮬레이션이 너무 매끄러운 경우, aero_solver_settings 노드의 apply_sharpening을 적용하면 비교적 저렴한 방법으로 상세도를 높일 수 있습니다.

sharpening_amount를 낮게 유지합니다. sharpening_radius를 높이면 선명도가 높아지지만 값이 너무 크면 컴퓨팅 시간이 길어집니다.

포인트로 상세도 높이기

aero_solver_settings 노드의 boost_detail_with_points 옵션은 시뮬레이션에 파티클을 추가합니다. 이 옵션을 적용하면 확산을 줄고 결과의 상세도가 높아지지만 메모리 사용량과 컴퓨팅 시간이 늘어납니다.

이 옵션이 작동할 때, 디버깅을 목적으로 simulate_aero 노드에서 out_aero_point_body 를 호스트 장면으로 출력해 포인트를 시각화할 수 있습니다.

공기와 연소 시뮬레이션을 위한 적응성 설정

적응형 시뮬레이션에서는 여러 영역의 상세도를 낮춤으로써 컴퓨팅 시간과 메모리 사용량을 줄일 수 있습니다. 특정 기준에 따라 상세도를 자동으로 낮추도록 할 수도 있고 메쉬 지오메트리를 이용해 상세도가 높은 영역과 낮은 영역을 명확히 지정할 수도 있습니다.

- 1. 공기 또는 연소 그래프 최상위 레벨에 aero_adaptivity 노드를 추가합니다.
- 2. 이 노드의 출력을 simulate_aero 노드의 adaptivity 포트에 연결합니다.
- 3. 다음 중 한 가지를 수행합니다.
 - enable_auto_adaptivity를 활성화하고 coarsen 기준을 선택합니다.
 - enable_bounded_adaptivity 를 활성화하고 메쉬 객체들을 연결해 각 해상도의 월드 축 정렬 바운딩 박스를 정의합니다.

연소 시 불꽃과 연기 제어

연소 시뮬레이션에서는 기본적으로 인화 가스에 의해 폭발성 화구가 생성됩니다. 이 설정을 조정하여 캠프파이어나 촛불 등 다양한 상황을 연출할 수 있습니다.

스케일이 중요

방출 소스와 기타 객체들의 사이즈는 불꽃의 형상에 큰 영향을 미칩니다. 장면이 모델링되는 스케일에 따라, aero_solver_settings 노드에서 scene_units_in_meters 를 조절해야 합니다. 예를 들어, 1 그리드 유닛이 장면에서 1cm라면 이 값을 0.01로 설정합니다.

불꽃의 높이 제어

연소 시뮬레이션을 화구보다는 연소용 고체 연료에 가깝게 연출하려면, 불꽃의 사이즈와 높이를 조절하면 됩니다. 여기에는 몇 가지 방법이 있는데, 다음과 같은 기법들을 잘 조합해 이용합니다.

- combustion_settings 노드에서 expansion_scale 을 낮춥니다. 이렇게 하면 연소의 폭발성이 줄어듭니다.
- combustion_settings 노드에서 radiative_cooling을 높입니다. 차가운 기체는 빠르게 상승하지 않지만, 이 경우 그을음이 많아지고 연기가 무거워집니다. 또, temperature_diffusion을 높이는 방법도 있는데, 이 경우 결과가 없어지는 경향이 있습니다.
- aero_solver_settings 노드에서 buoyancy의 크기를 줄입니다. 그러면 뜨거운 기체가 비교적 천천히 상승하게 됩니다.
- source_fuel 노드에서 burn_rate 를 낮춥니다. 그러면, 발생하는 열의 양이 줄어듭니다. 불꽃이 지속되도록 유지하려면 burn_rate와 radiative_cooling 의 균혀을 유지해야할 수도 있습니다. ignition_temperature 를 낮추는 것도 효과적일 수 있습니다.

연기 제어

다음은 연기 양을 조절하는 데 도움이 되는 몇 가지 설정입니다.

- combustion_settings 노드의 soot_formation_rate 는 연기처럼 보이는 그을음의 양에 직접적인 영향을 미칩니다. 이것을 0으로 설정하거나 emit_soot를 해제하게 되면 그을음이 voxel_fog_density 채널에 저장되어 시뮬레이션이 눈에 보이지 않게 되므로, source_air 노드에서 fog_density를 0보다 큰 값으로 설정해 시뮬레이션이 다시 보이게 만들어야 합니다.
- combustion_settings 노드에서 oxidize_soot 를 설정해 연기의 일부를 제거합니다. 이것은 불꽃의 사이즈와 열의 양에도 영향을 줍니다. 원하는 불꽃 높이를 구현하려면 soot_oxidation_rate와 oxygen_diffusion의 균형을 유지해야할 수도 있습니다.

불꽃으로 연료를 점화

성냥으로 불을 점화하는 것처럼, 한 가지 연료 소스를 이용해 또 다른 연료 소스를 점화할 수 있습니다.

이를 위해서는 source_air와 source_fuel 노드, 두 개의 체인이 필요합니다. 하나는 이미 연소 중인 소스를 위한 것이고 나머지 하나는 나중에 점화될 소스를 위한 것입니다. 이 둘을 simulate_aero 노드의 sources 입력에 각각 연결해야 합니다.

두 번째 연료 소스가 즉시 점화되는 것을 방지하려면, source_air 노드의 temperature를 source_fuel 노드의 ignition_temperature 보다 낮게 설정합니다.
장면에서 첫 번째 소스가 두 번째 소스에 아주 가깝게 접근하거나 아예 닿도록 애니메이트합니다. combustion_settings 노드의 두 매개변수는 두 번째 소스가 점화되는 방식을 제어합니다.

- radiative_heating은 주변 소스에 따른 연료의 가열 속도를 제어합니다.
- flame_propagation_speed는 불이 점화된 후 확산되는 속도를 제어합니다.

상위 페이지: 공기 및 연소 시뮬레이션 연출

MPM 시뮬레이션 연출

MPM(Material Point Method)은 광범위한 물질을 시뮬레이션하는 데 이용할 수 있는 수치 기반 메소드입니다. 눈 또는 젖은 모래와 같은 낟알 재질을 시뮬레이션할 수 있는 노드를 포함하고 있습니다.

simulate_mpm 노드에서 이용되는 솔버는 액체, 천, 섬유 가닥 등 기타 재질을 시뮬레이션할 수 있습니다. 하지만, 이러한 시뮬레이션을 설정하는 데 활용할 노드는 별도로 지원하지 않기 때문에 모험심을 발휘해 자유롭게 실험해보기 바랍니다.

이 섹션의 페이지들

• 낟알 시뮬레이션 설정

낟알 시뮬레이션 설정

낟알 시뮬레이션 중에는 고체 재질의 알갱이가 흘러갈 수 있는 시나리오가 있습니다. 기본 재질은 모래와 눈, 이 두 가지입니다.

- 모래는 좀 더 단순하고 빠른 메소드를 이용합니다. 젖은 모래나 마른 모래를 비롯해 다른 종류의 재질도 시뮬레이션할 수 있습니다.
- 눈은 재질이 서로 뭉쳐져 더 단단해질 수 있는 메소드를 이용합니다. 눈 이외의 재질에도 사용할 수 있습니다.

낟알 시뮬레이션을 설정하려면

- 1. 그래프의 최상위 레벨에 simulate_mpm 노드를 추가합니다.
- 2. mpm_solver_settings 노드를 추가하고 이 노드의 출력을 simulate_mpm의 settings 포트에 연결합니다.
- 3. 다음 노드 중 하나를 추가하고 그 노드의 출력을 simulate_mpm의 sources 입력에 연결합니다.
 - source_mpm_sand
 - source_mpm_snow
- 이미터 지오메트리를 소스 노드의 geometry 입력에 연결합니다. 소스 노드는 기본적으로 프레임 1에 일회성 방출을 연출하도록 설정되는데, 이것은 눈더미나 모래 사장과 같은 요소에 적합합니다. 일회성이 아닌 연속 방출을 원한다면, 소스 노드에서 use_end_frame을 비활성화합니다.
- 이 과정을 반복해 이미터를 추가로 생성합니다. 다수의 지오메트리가 동일한 설정을 따르도록 하려면 이들을 동일한 소스 노드에 연결하면 됩니다. 여러 가지 설정을 이용하려면, 소스 노드를 추가로 이용해 이들을 simulate_mpm의 sources 입력에 연결합니다.
- 6. collider 노드를 추가하고, 이 노드의 출력을 simulate_mpm의 colliders 입력에 연결한 후, 장애물 역할을 할 지오메트리를 geometry 입력에 연결합니다. 소스에서와 마찬가지로, collider 노드에 다수의 지오메트리를 연결할 수 있고 simulate_mpm 노드에도 다수의 colliders 노드를 연결할 수 있습니다. 예를 들어 발자국을 연출하려면, 눈이나 모래의 낙하를 방지해줄 지반용 콜라이더와 도보 중인 사람들을 위한 콜라이더가 필요합니다.
- 7. 선택사항으로, 원하는 효과에 적합할 경우 바람과 같은 효과를 추가합니다.
- 8. simulate_mpm 노드의 granular_points 출력을 그래프의 output 노드에 연결합니다.
- 9. 장면을 재생하여 설정을 조절합니다.

습기 및 기타 재질의 성질을 조절

다음은 소스 노드의 매개변수를 이용해 모래와 눈의 여러 조건을 에뮬레이션하는 데 유용한 도움말입니다.

- cohesion은 알갱이들의 응집력을 제어하므로, 젖은 모래나 눈을 연출하려면 높은 값을 적용합니다.
- 눈이나 모래에 습기가 너무 많아서 액체처럼 작용하기 시작하면 friction을 낮춥니다.
- 눈의 경우, <u>initial_firmness</u>는 눈이 뭉치는 방식을 제어합니다. 밀도가 낮고 가벼운 눈을 연출하려면 낮은 값을 적용합니다.

상위 페이지: MPM 시뮬레이션 연출

추가 방출 소스 설정

시뮬레이션 그래프의 최상위 레벨에 있는 소스 노드의 geo 포트에 기하학적 객체를 추가로 연결할 수 있습니다. 연결된 이미터 지오메트리는 모두 동일한 설정을 따릅니다.

rate, size, speed 등 설정을 달리하여 방출하려면, 다음과 같이 합니다.

- 1. 그래프에 source_particles 노드를 하나 더 추가합니다.
- 2. 이미터 지오메트리를 자체 geometry 포트에 연결합니다.
- source_particles 노드의 particle_source 출력을 simulate_particles 노드의 sources 입력에 연결합니다.

sources 포트에 source_particles 노드를 원하는 개수만큼 연결할 수 있습니다.

방출 특성의 변화

소스에서 방출할 때, vary_source_property 노드를 이용해 특성을 변화시킬 수 있습니다. 예를 들어, 기본 파티클의 사이즈, 연기 밀도, 모래 점도 등을 바꿀 수 있습니다.

- 1. 이미터 지오메트리를 vary_source_property 노드의 geometry 입력에 연결합니다.
- 2. vary_source_property 노드의 out_geometry 출력을 시뮬레이션 소스 노드의 geometry 입력에 연결합니다.
 - 이미터 지오메트리가 이미 연결되어 있는 경우에는 fan-in 포트를 확장해 기존의 연결을 대체하거나 연결을 해제해야 합니다. 그렇지 않으면, 지오메트리가 두 번 연결되어 한 번은 변화하고 한 번은 변화하지 않게 됩니다.
- 3. 변화시킬 property 를 지정합니다. 이용 가능한 특성의 이름은 시뮬레이션의 유형에 따라 다릅니다. 이에 대한 목록은 Info 탭을 참조합니다.
- 4. 값을 변화시킬 방법을 선택합니다.
 - 값을 무작위화하려면, randomize_values 를 활성화하고 min, max, bias, animated 를 원하는 대로 설정합니다. 지오메트리 상에서 값을 변화시키기 좋은 대상은 밀도 큰 메쉬입니다. direction과 같은 벡터 값을 변화시키려면, float3 value 노드로 min 또는 max를 조절합니다.
 - 지오메트리에 설정된 색상을 기준으로 값을 설정하려면, use_color_set 를 활성화하고 color_set_name 을 호스트 장면의 지오메트리상의 이름과 똑같이 지정합니다. 싱글 채널 컬러 세트를 이용해 float 값을 변화시키고 RGB 컬러 세트를 이용해 float3 값을 변화시킵니다.
 - randomize_values와 use_color_set가 모두 활성화된 경우, 컬러 세트가 무작위 값들의 곱수 역할을 합니다.
 - 선택적으로, multiplier를 이용해 값을 높이거나 낮춥니다.

특성을 추가로 변화시키려면, 이미터 지오메트리와 소스 노드 사이에 vary_source_property 노드를 추가로 데이지 체인 방식으로 연결합니다.

콜라이더(collider) 추가

콜라이더는 시뮬레이션에서 움직임을 막는 객체입니다.

- 1. 아직 없다면, 시뮬레이션 그래프 최상위 레벨에 collider 노드를 추가합니다.
- 2. collider 노드의 collider 출력을 simulate 노드의 colliders 입력에 연결합니다.
- 3. 메쉬 또는 볼륨 객체 1개 이상을 collider 노드의 geometry 입력에 연결합니다.
 - 볼륨 객체일 경우, collider 노드의 method를 반드시 Volume으로 설정해야 합니다.

collider 노드에 지오메트리를 원하는 개수만큼 연결할 수 있습니다. 또, simulate 노드에 collider 노드를 원하는 개수만큼 연결할 수 있으며, 이때 설정이 각각 달라야 합니다.

충돌 메소드 선택

collider 노드의 method 매개변수는 콜라이더 지오메트리를 메쉬와 볼륨 중 어느 쪽으로 취급할 것인지를 정합니다.

- Volume은 모든 지오메트리를 볼륨으로 취급합니다. 메쉬 입력은 자동으로 변환됩니다. 충돌을 더 정확히 구현하려면 detail_size를 줄입니다.
- Mesh_for_Points 는 메쉬를 메쉬로 취급합니다. 이 메소드는 메쉬 서피스와 충돌 위치 사이의 간극을 방지합니다.

볼륨 객체인 경우, method는 반드시 Volume이어야 합니다. 또, 볼륨 객체는 레벨 세트를 포함해야 합니다(voxel_signed_distance).

파티클의 충돌 반응 제어

파티클은 콜라이더에 부딪힐 때 튀어 오르거나 들러붙을 수 있습니다. 이러한 현상은 여러 인수를 조합하여 제어할 수 있습니다.

- 탄성(Bounciness)은 충돌에 반영될 충돌 서피스에 대한 속도 노멀의 비를 제어합니다. 예를 들면, 파티클이 땅에 부딪힌 다음 튀어 오르는 높이에 영향을 줍니다. 최종 값은 파티클과 콜라이더에 설정된 bounciness의 조합입니다. 두 값이 모두 1.0인 경우 파티클이 완벽히 튀어 오르고, 두 값이 모두 0.0인 경우에는 전혀 튀어 오르지 않습니다.
- collider 노드에 설정된 friction 매개변수는 콜라이더의 서피스를 따라 발생하는 움직임을 제어합니다. 이 값이 1.0이면 파티클이 땅에 부딪힌 후 굴러가지 않으며, 값이 0.0이면 파티클이 속도 저하 없이 굴러가게 됩니다.
- collider 노드의 roughness 매개변수는 파티클이 튀어 오르는 각도를 변화시킵니다. 이것은 거친 표면에 부딪히는 모습을 에뮬레이션합니다.

빠르게 움직이는 파티클이 콜라이더로 떨어지면, particle_solver_settings 노드의 substeps을 높여봅니다.

influence 적용

Influence는 시뮬레이션의 움직임에 변화를 일으키는데, 바람이나 난류와 같은 효과가 여기에 포함됩니다.

- 1. **Simulation > Influence** 네임스페이스에서 노드를 시뮬레이션 그래프(예: wind_influence)에 추가합니다.
- 2. out_influence 출력을 그래프 simulate 노드의 influence_fields 입력에 연결합니다.

influence를 influence_fields 포트에 원하는 개수만큼 연결해 복잡한 효과를 연출할 수 있습니다.

또, 한 influence의 out_influence 출력을 다른 influence의 influence 입력에 연결하는 방식으로 여러 개의 influence를 데이지 체인 방식으로 연결할 수 있습니다. 이로써 여러 효과를 한꺼번에 감출 수도 있습니다.

이 섹션의 페이지들

- influence 감추기
- influence 조절

influence 감추기

마스크를 이용해 influence가 적용되는 영역을 제어할 수 있습니다.

- 1. 감추고자 하는 influence로 mask_influence 노드를 데이지 체인 방식으로 연결합니다. mask_influence 노드는 이 체인 내 어디서든 나타날 수 있습니다. 그래프 simulate 노드의 influence_fields 입력에 체인의 끝을 반드시 연결해야 합니다.
- 2. 지오메트리를 mask_influence 노드의 geometry 입력에 연결해 감출 영역을 정의합니다.



상위 페이지: 효과(influence) 적용

influence 조절

modulate_influence 노드는 또 다른 특성에 따라 influence 의 강도를 변화시킬 수 있습니다. 예를 들어, point_size를 기준으로 파티클이 wind_influence의 영향을 더 많이 받거나 더 적게 받도록 설정할 수 있습니다.

- 감추고자 하는 influence로 modulate_influence 노드를 데이지 체인 방식으로 연결합니다.
 modulate_influence 노드는 이 체인 내 어디서든 나타날 수 있습니다. 그래프 simulate 노드의 influence_fields 입력에 체인의 끝을 반드시 연결해야 합니다.
- 2. 사용하려는 property 를 지정하여 influence의 효과를 조절합니다. 특성 이름을 정확히 지정해야 하는데, 이를 테면, size가 아니라 point_size로 지정합니다.
- 3. 다른 매개변수들도 원하는 대로 설정합니다.
 - min은 적용되는 influence 효과 강도의 최소 비, 그리고 max는 최대 비입니다.
 - invert는 조절을 반전시켜 지정 특성 값이 클수록 influence 효과의 강도가 낮아집니다.

상위 페이지: 효과(influence) 적용

기하학적 객체의 인스턴스 생성하기

인스턴스는 파티클 시스템의 포인트처럼, 포인트 위치에 존재하는 기하학적 객체의 복사본입니다. 인스턴스를 이용하면 마스터 지오메트리와 포인트만 있으면 되므로 장면이 더 가벼워집니다. 인스턴스를 이용해서 숲에 있는 나무, 더미 위로 떨어지는 객체 등 다양한 요소를 연출할 수 있습니다.

인스턴스는 포인트를 *인스턴스 모양*과 연결해야 작동하는데, 인스턴스 모양은 지오메트리와 뷰포트에 표시하기 위해 이용하는 프리뷰 지오메트리(선택사항)와 같은 추가 정보로 이루어집니다.

인스턴스를 만들 수 있는 노드에는 몇 가지가 있습니다.

- 대부분의 경우, set_instance_geometry를 선호합니다. 이것은 옵션이 내장되어 있는 하이 레벨 노드로서 여러 마스터 모양 사이에서 전환하는 데 이용합니다. 또 인스턴스 모양을 먼저 만들지 않고도 기하학적 객체들을 직접 연결할 수 있습니다.
- 또는, set_instance_shape 를 이용할 수 있습니다. 이것은 좀 더 기초적인 노드로서, geo_instance 나 다른 노드를 이용해 지오메트리에서 인스턴스 모양을 만들어야 합니다.

어느 경우든, <u>selector_instance</u> 노드를 이용해 복잡한 방식으로 각종 마스터 모양 사이에서 상호 전환할 수 있습니다.

이 섹션의 페이지 내용

- set_instance_geometry로 인스턴스 생성하기
- set_instance_shape로 인스턴스 생성하기
- 다수의 인스턴스 지오메트리 중에서 선택하기
- 인스턴스 변형 조절

set_instance_geometry로 인스턴스 생성하기

set_instance_geometry는 포인트 객체에서 지오메트리 인스턴스를 만들 수 있는 하이 레벨 노드입니다. 저장된 포인트별 long 값이나 long 값들의 입력 어레이를 기준으로 몇 가지 마스터 지오메트리 사이에서 상호 전환할 수 있는 옵션들이 내장되어 있습니다.

- 1. set_instance_geometry 노드를 그래프에 추가합니다.
- 2. 포인트 객체(예: 파티클 시스템의 출력)를 points 입력에 연결합니다.
- 3. 메쉬와 같은 기하학적 객체 1개 이상을 instance_geometries 포트에 연결합니다. 지오메트리 외에도 다음 항목들의 출력과 같은 인스턴스 모양을 연결할 수 있습니다.
 - geo_instance: 각 인스턴스 마스터 모양마다 서로 다른 프리뷰 지오메트리를 이용하고 싶은 경우.
 - render-archive-instance: 렌더 아카이브 파일을 디스크로 인스턴스화하려는 경우.
 - selector_instance: 모든 마스터 지오메트리 사이에서 상호 전환하는 것이 좀 더 복잡한 경우.
- 4. 각 포인트에 지오메트리를 지정할 방법을 설정합니다. 다음과 같이 두 가지 방법이 있습니다.
 - 사용자 지정 long 특성의 이름을 instance_id_property 입력으로 지정합니다. 이것은 업스트림에서 set_geo_property와 같은 노드를 이용해서 포인트 객체의 point_component를 대상으로 하는 특성을 지정했을 때를 전제로 합니다. 이 특성은 인스턴스화할 지오메트리를 지정해야 하는데, 이때 0은 instance_geometries에 연결되는 첫 번째 지오메트리이고, 1은 두 번째인 식입니다.
 - long 어레이를 instance_id_values 포트에 연결하고 overwrite_instance_id를 활성화합니다. 어레이가 포인트 개수보다 더 길다면, 포인트 개수를 넘어서는 값들은 무시됩니다. 반대로, 어레이가 더 짧다면, 나머지 포인트들에 대해서는 첫 번째 연결된 지오메트리가 이용됩니다.
- 5. 원하는 경우, 1개 이상의 단순 지오메트리를 preview_geometries 포트에 연결합니다. 이러한 지오메트리는 뷰포트에서 미리 보기할 스탠드-인(stand-in)으로 작용합니다. 인스턴스 모양 중에는 이미 프리뷰 지오메트리가 정의된 것도 있을 수 있는데, 가령 지오메트리를 직접 연결하는 대신 geo_instance 노드를 instance_geometries 포트에 연결하는 경우를 들 수 있습니다. 또, 여기에 지오메트리를 1개만 연결하고 이를 이용해 이미 자체 프리뷰 지오메트리가 정의된 인스턴스를 포함해 모든 인스턴스를 미리 보기하는 경우, 다른 작업을 하는 도중에 뷰를 작게 표시할 수 있습니다.
- 6. 최종 instances 를 장면에 출력합니다.

set_instance_shape로 인스턴스 생성하기

set_instance_shape 는 한 포인트 객체의 모든 포인트에 인스턴스 모양을 지정하는 기본 노드입니다. set_instance_geometry 와는 달리, 지오메트리를 직접 연결할 수 없습니다. 그 대신, geo_instance의 출력을 연결해 단일 지오메트리를 인스턴스화하거나 selector_instance의 출력을 연결해 몇 가지 지오메트리 중에서 원하는 것을 선택해야 합니다.

- 1. set_instance_shape 노드를 그래프에 추가합니다.
- 2. 포인트 객체(예: 파티클 시스템의 출력)를 points 입력에 연결합니다.
- 3. 다음 중 한 가지 작업을 수행합니다.
 - 단일 모양을 인스턴스화하려면, geo_instance 노드를 추가합니다. 인스턴스화하려는 지오메트리를 geometry 입력에 연결합니다. 선택사항으로, 뷰포트에서 미리 보기할 스탠드-인(stand-in)으로 작용할 메쉬 1개를 preview_geometry 입력에 연결합니다. 이것의 출력을 set_instance_shape 노드의 instance_shape 입력에 연결합니다.
 - 디스크의 렌더 아카이브 파일이나 시퀀스를 인스턴스화하려면, render-archive-instance 노드를 추가하고 이것의 출력을 set_instance_shape 노드의 instance_shape 입력에 연결합니다.
 - 다수의 인스턴스 모양 중에서 원하는 것을 선택하려면, selector_instance 노드를 추가하고 이것의 출력을 set_instance_shape 노드의 instance_shape 입력에 연결합니다.
- 4. 최종 instances 를 장면에 출력합니다.

다수의 인스턴스 지오메트리 중에서 선택하기

selector_instance 노드는 다수의 모양 중에서 하나를 선택하므로, 예를 들어, 여러 종류의 나무들로 숲을 만들 수 있습니다. 또, selector_instance 노드 몇 개를 데이지 체인 방식으로 연결해, 예를 들어, 한 가지 기준에 따라 소나무와 잎이 무성한 나무 중 하나를 선택한 다음 다른 기준에 따라 여러 예시 중 하나를 선택할 수 있습니다. selector_instance의 출력을 set_instance_geometry 노드나 set_instance_shape 노드로 연결할 수 있습니다.

- 1. 그래프에 selector_instance 노드를 추가합니다.
- 2. 메쉬와 같은 기하학적 객체 1개 이상을 instance_geometries 포트에 연결합니다. 지오메트리 외에도 다음 항목들의 출력과 같은 인스턴스 모양을 연결할 수 있습니다.
 - geo_instance: 각 인스턴스 마스터 모양마다 서로 다른 프리뷰 지오메트리를 이용하고 싶은 경우.
 - render-archive-instance: 렌더 아카이브 파일을 디스크로 인스턴스화하려는 경우.
 - selector_instance: 모든 마스터 지오메트리 사이에서 상호 전환하는 것이 좀 더 복잡한 경우.
- 3. 사용자 지정 long 특성의 이름을 instance_id_property 입력으로 지정합니다. 이것은 업스트림에서 set_geo_property와 같은 노드를 이용해서 포인트 객체의 point_component를 대상으로 하는 특성을 지정했을 때를 전제로 합니다. 이 특성은 인스턴스화할 지오메트리를 지정해야 하는데, 이때 0은 instance_geometries에 연결되는 첫 번째 지오메트리이고, 1은 두 번째인 식입니다.
- 4. 원한다면, 단순 지오메트리 1개를 preview_geometry 포트에 연결합니다. 이 지오메트리는 뷰포트에서 미리 보기할 스탠드-인(stand-in)으로 작용합니다. 이것은 모든 인스턴스에 적용되는데, 이미 프리뷰 지오메트리가 정해진 인스턴스라도 상관 없으며, 예를 들어, 지오메트리를 직접 연결하는 대신 geo_instance 노드를 instance_geometries 포트에 연결한 경우를 들 수 있습니다.
- 5. selector_instance 를 set_instance_geometry 노드나 set_instance_shape 노드에 연결합니다.

인스턴스 변형 조절

포인트 객체의 point_position 특성은 인스턴스의 위치를 좌우합니다. 그 밖에, 인스턴싱된 지오메트리의 변형에 영향을 줄 수 있는 몇 가지 다른 특성이 있습니다.

이런 특성들이 아직 없다면 set_geo_property 를 이용해 만들 수 있습니다. 이미 있다면, set_geo_property_data 를 이용해서 각 특성의 값을 변경할 수 있습니다. 이런 특성들은 포인트 객체의 point_component 를 대상으로 해야 합니다.

- point_size: 모든 축에 있는 모양들을 균등하게 스케일링하는 float.
- point_scale: 균등하지 않게 스케일링하는 float3.
- point_orientation: 4원수로 방향을 나타내는 float4.

뿐만 아니라, geo_instance나 render_archive_instance와 같은 노드의 transform 매개변수를 설정할 수 있습니다. 이것은 마스터 지오메트리가 월드 센터로부터 오프셋될 때 지오메트리의 모든 인스턴스를 보정하면서 이와 동시에 특정 인스턴스의 크기와 방향을 개별적으로 제어하고 싶을 때 유용합니다.

재질 적용하기

Bifrost에서 만들었거나 수정한 기하학적 객체에 재질을 적용하는 방법에는 몇 가지가 있습니다. 재질을 지정하는 기본 노드는 assign_material입니다.

Bifrost는 어느 렌더링이든 스스로 수행하지 못하며, 재질은 특별한 이름의 특성을 가지는 지오메트리의 하위 객체로서 이런 특성들은 뷰포트에서 읽거나 뷰포트에 표시하는 데 이용할 수 있고 또는 Arnold와 같이 호환되는 렌더러로 렌더링할 수도 있습니다.

🖋 참고:

각 객체마다 1개의 재질만 지원되며, 면들과 같은 집합체에는 지원되지 않습니다.

재질의 유형

재질에는 기본적으로 두 가지 유형이 있습니다.

- 참조된 재질은 호스트 장면에 존재하는 재질의 참조입니다. 장면에서 가져온 객체는 현재 적용된 재질에 참조를 자동으로 포함시킵니다. 뿐만 아니라, 지오메트리를 가져오는 것과 같은 방식으로 장면의 재질을 그래프로 가져올 수 있습니다.
- 재질 정의는 색상, 투명도 등과 같은 특성들의 모음입니다. 이러한 특성들은 전형적인 쉐이더에서 찾아볼 수 있는 설정을 그대로 따릅니다. 재질 정의를 만들 수 있는 노드는 두 가지입니다. standard_surface_mat 과 constant_surface_mat 입니다.

🖊 참고:

볼륨일 경우에는 재질 정의가 지원되지 않습니다.

재질의 우선순위

재질은 우선순위에 따라 적용됩니다.

- 호스트 장면의 기본 재질은 우선순위가 가장 낮습니다. 이 재질은 지오메트리를 출력할 때 기본적으로 자동 적용되지만, 입력 지오메트리 상의 재질 참조나 그래프에서 명확히 지정한 재질 등 다른 재질이 1개라도 있다면 이 재질은 사용되지 않습니다.
- Bifrost 지오메트리에서 정의하는 재질은 중간 우선순위를 갖습니다. 이것은 호스트 장면에 있는 입력 지오메트리의 자동 재질 참조뿐만 아니라 assign_material을 이용해 지정한 재질까지 포함합니다.
- 호스트 장면에서 그래프의 출력으로 명확히 지정된 재질(기본 장면 재질 외)은 우선순위가 가장 높습니다. 이런 재질들은 그 밖의 모든 것들에 우선합니다.

이 섹션의 페이지 내용

• geo 특성으로 재질 특성 제어

geo 특성으로 재질 특성 제어

기하학적 객체들의 특성을 이용해 재질 정의 특성을 제어할 수 있습니다. 예를 들어 진단을 목적으로 재질의 base_ color 를 파티클 시스템의 point_velocity로 설정하거나, 사용자 지정 geo 특성을 이용해 재질 특성을 조정해 원하는 미적 결과를 얻을 수 있습니다.



🖋 참고:

이 절차는 재질 정의에서만 효과가 있습니다. 참조된 재질은 지원되지 않습니다.

- 1. standard_surface_mat 나 constant_surface_material과 같은 재질 정의를 그래프에 추가합니다.
- 2. 이것의 출력을 material_definition port of a set_geo_property_reference 노드에 연결합니다.
- 3. material_property 를 base_color와 같은 제어할 재질 특성의 이름으로 설정합니다.
- 4. geo_property를 재질 특성을 제어하는 데 이용할 기하학적 객체의 특성 이름으로 설정합니다.
- 5. set_geo_property_reference의 출력을 assign_material 노드의 재질 입력 중 하나에 연결합니다.
- 6. 기하학적 객체를 assign_material 노드의 geometry 포트에 연결합니다. 결과를 장면에 출력하거나 그래프에서 원하는 만큼 더 수정합니다.



재질 속성을 뷰포트에서 직접 제어

set_geo_property_reference 를 이용하는 대신, 쉐이더 속성과 이름이 같은 geo 특성을 만들 수 있습니다. 이것은 뷰포트에만 영향을 주므로 디버깅이나 기타 목적의 시각화 도구로 효과적입니다.

예를 들어 set_geo_property 노드를 1개 이상 이용해서 point_component를 대상으로 하는 특성을 만들고, 이름을 color, transparency, incandescence 등으로 지정한 다음, 이런 특성을 이용해서 장면에서 Blinn이나 Lambert와 같은 쉐이더의 해당 속성을 제어할 수 있습니다. 쉐이더 속성의 이름을 매치하려면, 소문자를 사용하고 빈 칸을 없앱니다(예: ambientcolor).

float3과 float4 유형의 속성만 지원됩니다.

상위 페이지: 재질 적용하기

Bifrost 명령행 도구

Bifrost Extension에는 bifcmd가 포함되어 있는데, 이것은 Maya와 상관 없이 명령행에서 컴파운드를 실행할 수 있는 도구입니다.

이 섹션의 페이지 내용

- Windows 전용 환경 설정
- bifcmd 이용하기

Windows 전용 환경 설정

bifcmd는 실행 시 Bifrost Extension에 포함되어 있는 타사 라이브러리를 이용합니다.

C:\Program Files\Autodesk\Bifrost\<maya_version>\<bifrost_version>\bifrost\thirdparty\bin을 PATH 변수에 추가합니다.

중요 정보:

Maya 또는 Bifrost Extension 버전이 2개 이상 설치되어 있다면, 명령 창에서 bifcmd를 실행할 경로를 설정합니다. 글로벌 환경 변수에서 이를 설정하면 Maya나 Bifrost Extension의 다른 버전들과 충돌할 수 있습니다.

PATH=%PATH%;C:\Program Files\Autodesk\Bifrost\<maya_version>\<bifrost_version>\bifrost\thirdparty\bin 설정합니다.

상위 페이지: Bifrost 명령행 도구

bifcmd 실행하기

bifcmd를 이용하면 Maya와 상관 없이 명령행에서 컴파운드를 실행할 수 있습니다.

bifcmd는 컴파운드의 json 파일 형식을 취하며 입력 포트에서 값들을 변경할 수 있습니다. 컴파운드가 시뮬레이션인 경우, 실행할 프레임 개수도 제한할 수 있습니다.

이 명령은 파일에 컴파운드의 출력을 작성하지 않습니다. 작업자가 출력을 저장하기를 원할 때 이 명령 자체가 파일에 작성됩니다.

bifcmd --help는 전체 명령행 인수 목록을 출력합니다.

bifcmd는 Bifrost 설치 장소에 위치합니다. 이 설치 디렉터리를 찾아들어가 호출하거나 전체 경로를 이용해 호출할 수 있습니다.

bifcmd의 기본 경로:

Linux	/usr/autodesk/bifrost/ <maya_version>/<bifrost_version>/bifrost/bin</bifrost_version></maya_version>
macOS	Applications/Autodesk/bifrost/ <maya_version>/<bifrost_version>/bifrost/bin</bifrost_version></maya_version>
Windows	C:\Program Files\Autodesk\Bifrost\ <maya_version>\<bifrost_version>\bifrost\bin</bifrost_version></maya_version>

bifcmd의 디렉터리를 PATH 변수에 추가하면 디렉터리를 변경하거나 전체 경로를 이용하지 않고도 이 명령을 호출할 수 있습니다.

예를 들어, Bifrost Extension 2.0.1.0 for Maya 2019를 이용하고 있다면, PATH 변수에 아래 경로를 추가합니다.

Linux	/usr/autodesk/bifrost/maya2019/2.0.1.0/bifrost/bin
macOS	/Applications/Autodesk/bifrost/maya2019/2.0.1.0/bifrost/bin
Windows	C:\Program Files\Autodesk\Bifrost\Maya2019\2.0.1.0\bifrost\bin

중요 정보:

Maya 또는 Bifrost Extension 버전이 2개 이상 설치되어 있다면, 명령 또는 터미널 창에서 bifcmd를 실행할 PATH 변수를 수정합니다. .bash_properties 파일 또는 쉘(shell) 구성 파일에서 이것을 글로벌 환경 변수로 설정하면 설치되어 있는 다른 Maya나 Bifrost Extension 버전들과 충돌할 수도 있습니다.

상위 페이지: Bifrost 명령행 도구

참조

이 섹션의 페이지들

- bif
- file
- fluids
- geometries
- mpm
- particles
- simulation

bif

이 섹션의 페이지들

- add_points
- cumulative_sum_array
- delete_points
- get_array_indices
- get_closest_locations
- get_geo_component_count
- get_geo_property_check
- get_geo_property
- get_point_count
- get_point_normal
- get_point_position
- get_point_velocity
- get_property
- get_sub_object
- geo_instance
- render_archive_instance
- selector_instance
- set_instance_geometry
- set_instance_shape
- is_empty
- assign_material
- constant_surface_mat
- set_geo_property_reference
- merge_geometry
- remove_from_array
- sample_property
- set_geo_property_data
- set_geo_property
- set_motion_blur_mode
- set_point_normal
- set_point_position
- set_point_shape
- set_point_velocity
- set_property
- standard_surface_mat

상위 페이지: 참조

add_points

파티클 시스템과 같은 포인트 객체에 포인트를 추가합니다. point_component 를 대상으로 하는 특성은 모두 새로운 포인트의 기본 값으로 설정되지만, new_collection 을 이용해서 이런 값들을 다른 값으로 초기화할 수도 있습니다.

Inputs

geometry

포인트 객체.

positions

포인트를 추가할 위치들의 어레이

new_collection

새로운 포인트를 추가할 포인트 컬렉션의 이름. 이것은 선택사항입니다. 컬렉션이 이미 존재하는 경우에는 대체됩니다.

컬렉션에 사용할 이름을 부여하면, new_collection을 이용해서 point_size와 같은 새로운 포인트의 특성 값을 설정할 수 있습니다.

- get_geo_property와 여기에 입력한 컬렉션 이름을 이용해 출력 지오메트리에서 특성 값을 가져옵니다.
 Boolean value 노드를 type 포트로 연결합니다.
- 2. get_geo_property의 data 출력을 if 노드의 condition 포트로 연결합니다.
- 3. 또 다른 get_geo_property 노드를 이용해 원래의 포인트 객체의 기존 특성 값들을 불러오고 data 값들을 if 노드의 false_case 포트로 연결합니다.
- 4. 부분 그래프를 만들어 새로운 포인트에 사용할 값들을 계산하고 출력을 if 노드의 true_case 포트로 연결합니다.
- 5. set_geo_property_data 노드를 이용해서 포인트 객체에 결과 값을 설정합니다.
 - add_points의 출력을 set_geo_property_data의 geometry 포트로 연결합니다.
 - property를 기존 특성의 이름으로 설정합니다.
 - if 노드의 출력을 data 포트로 연결합니다.
- 6. 초기화하고자 하는 다른 특성들에 대해서도 동일한 작업을 반복합니다.

Output

out_geometry

포인트가 추가된 새로운 포인트 객체

cumulative_sum_array

입력 어레이의 부분 합 어레이를 반환합니다. 출력 어레이의 첫 번째 요소는 입력 어레이의 첫 번째 요소이고, 두 번째 요소는 첫 두 값의 합이며, 세 번째 요소는 첫 세 값들의 합, 이런 식으로 이어집니다.

Input

array

숫자 값들의 어레이

Output

cumulative

부분 합들의 어레이

delete_points

포인트 객체(예: 파티클 시스템)에서 포인트를 제거합니다.

Inputs

geometry

포인트를 삭제하려는 포인트 객체

point_indices

삭제할 포인트 인덱스(long)들로 구성된 어레이

Output

out_geometry 삭제된 포인트가 없는 새로운 포인트 객체

get_array_indices

입력 어레이의 인덱스들을 포함한 어레이를 반환합니다. 예를 들어, 입력 어레이에 5개의 요소가 있다면, 출력은 [0, 1, 2, 3, 4]가 됩니다.

Inputs

array

어레이

index_type

반환된 인덱스들의 유형. 기본적으로, 출력은 long 어레이이지만 또 다른 유형을 연결해서 변경할 수 있습니다. 이것은 길이와 관계없이 부호 붙임 정수와 무부호 정수 유형을 지원합니다.

Output

indices

인덱스들의 어레이

get_closest_locations

입력 지오메트리 상에서 입력 위치에 가장 가까운 위치들을 반환합니다. 그런 다음, sample_property를 이용해 이들 위치에 있는 지오메트리의 특성 보간 값들을 불러올 수 있습니다.

Inputs

geometry

가장 가까운 위치들을 찾을 지오메트리. 오직 메쉬만 지원되는데, 다른 지오메트리 유형들은 유용한 결과를 제공하지 않기 때문입니다.

positions

가장 가까운 위치들을 찾을 위치

use_cutoff_distance

지정된 cutoff_distance 내에서만 검색합니다.

cutoff_distance

검색 범위가 될 거리

Outputs

locations

위치들을 포함하는 객체. 이를 sample_property 노드의 locations 포트로 연결할 수 있습니다.

found

검색 결과에 해당하는 부울 값들의 어레이. 한 요소가 false일 경우, 해당 위치는 유효하지 않으며 이를 샘플 지오메트리에 이용하면 기본 값들을 반환하게 됩니다. 이는 차단 거리 내에 위치들이 존재하지 않거나 해당 지오메트리가 지원되지 않아서 일 수 있습니다.

get_geo_component_count

메쉬 내 포인트 개수처럼 기하학적 객체 내 구성요소들의 개수를 반환합니다.

Inputs

geometry

기하학적 객체

component

구성요소 이름으로는 point_component, face_component, face_vertex_component, voxel_component 등이 있습니다.

Output

count

구성요소 개수

get_geo_property_check

포인트의 point_size 값처럼 기하학적 객체 내 구성요소들의 특성 값들로 구성된 어레이를 반환합니다. 이것이 get_geo_property와 다른 것은 그 특성이 객체 상에 존재하는지 여부까지 반환하기 때문입니다.

Inputs

geometry

기하학적 객체

property

특성의 이름

type

특성의 유형. 특성을 찾지 못한 경우, 이것은 반환된 어레이의 기본 값이기도 합니다.

Outputs

data

값들로 구성된 어레이.

default

특성의 기본 값

target

특성의 대상이 되는 기하학적 구성요소를 말하며, point_component, face_component, face_vertex_component, voxel_component 등이 있습니다.

found_property

특성을 찾았는지 여부

get_geo_property

포인트 point_size 값처럼 기하학적 객체의 구성요소들의 특성 값들로 구성된 어레이를 반환합니다.

Inputs

geometry

기하학적 객체.

property

특성의 이름.

type

특성의 유형. 이것은 특성을 찾지 못한 경우에 반환된 어레이의 기본 값이기도 합니다.

Outputs

data

값들로 구성된 어레이.

default

특성의 기본 값.

target

특성이 대상으로 하는 기하학적 구성요소로, point_component, face_component, face_vertex_component, voxel_component 등이 있습니다.

get_point_count

한 기하학적 객체 내 포인트 개수를 불러옵니다. 이것은 포인트를 포함하고 있는 모든 객체에 효과가 있는데, 여기에는 파티클 시스템과 같은 포인트 객체뿐만 아니라 메쉬, 인스턴스 등도 포함됩니다.

Input

geometry

기하학적 객체

Output

count

지오메트리 내 포인트의 개수

get_point_normal

한 메쉬에 대한 포인트 노멀들로 구성된 어레이를 반환합니다.

Input

geometry

메쉬 객체

Output

point_normal

포인트 노멀들로 구성된 어레이

get_point_position

기하학적 객체에 대한 포인트 노멀들로 구성된 어레이를 반환합니다. 이것은 포인트를 포함하고 있는 모든 객체에 효과가 있는데, 여기에는 파티클 시스템과 같은 포인트 객체뿐만 아니라 메쉬, 인스턴스 등도 포함됩니다.

Input

geometry

기하학적 객체

Output

point_position

포인트 위치들로 구성된 어레이

get_point_velocity

기하학적 객체에 대한 포인트 속도들로 구성된 어레이를 반환합니다. 이것은 포인트를 포함하고 있는 모든 객체에 효과가 있는데, 여기에는 파티클 시스템과 같은 포인트 객체뿐만 아니라 메쉬, 인스턴스 등도 포함됩니다.

Input

geometry

기하학적 객체

Output

point_velocity

포인트 속도들로 구성된 어레이

get_property

객체에서 특성을 확인합니다. 특성은 키-값 쌍으로서, 키는 이름(문자열)이고 사실상 무엇이든 값이 될 수 있습니다. 대부분의 경우, 값은 또 다른 객체가 될 수 있습니다.

보통, get_property와는 다른 노드를 사용하게 될 것입니다. 예를 들어, 기하학적 객체에서 point_size와 같은 특성의 값을 구하려면, 그 대신 get_geo_property를 이용해야 합니다. 일부 공통 특성의 경우, get_point_position과 같은 특정 노드가 있을 수도 있습니다.

Inputs

object

기하학적 객체.

default_and_type

특성의 값과 유형이 같은 value 노드를 연결합니다. 이것은 해당 특성이 존재하지 않거나 유형이 다를 경우에 출력으로 반환되는 값이기도 합니다.

key

특성의 이름.

Output

value

특성의 값, 혹은 실패할 경우 default_and_type입니다.

get_sub_object

객체의 하위 객체 즉, 데이터 유형이 또 다른 객체인 키-값 쌍의 값을 반환합니다.

Inputs

object

그 하위 객체를 포함하고 있는 객체.

name

하위 객체의 이름(키).

Output

sub_object

이름이 있는 키의 값인 객체. 키가 존재하지 않거나 값이 객체가 아닌 경우, 반환되는 객체는 비어 있습니다.
geo_instance

기하학적 객체로부터 인스턴스 모양을 만듭니다. 인스턴스 모양은 set_instance_shape, set_instance_geometry, selector_instance 중 하나로 연결할 수 있습니다.

Inputs

geometry

기하학적 객체.

preview_geometry

미리보기에 이용할 기하학적 객체(선택 사항). 한 객체가 연결되면, 그 객체는 뷰포트에서만 볼 수 있고 렌더링에는 이용할 수 없습니다.

name

선택적 식별자.

transform

인스턴싱된 지오메트리에 적용할 변환 행렬.

Output

geo_instance

인스턴스를 만들 때 이용할 수 있는 인스턴스 모양.

render_archive_instance

디스크에 있는 렌더 아카이브 파일을 참조하는 인스턴스 모양을 만듭니다. 인스턴스 모양은 set_instance_shape, set_instance_geometry, selector_instance 중 하나로 연결할 수 있습니다.

Inputs

archive_file

렌더 아카이브의 경로와 파일 이름. 시퀀스의 프레임 번호를 표시할 때는 ####을 사용합니다.

frame_property

시퀀스에서 인스턴스화할 프레임을 지정하는 포인트 객체의 특성 이름. 그래프가 매 프레임마다 각 포인트의 이 특성을 업데이트하는지 확인해야 합니다.

preview_geometry

미리보기에 이용할 기하학적 객체(선택 사항). 한 객체가 연결되면, 그 객체는 뷰포트에서만 볼 수 있고 렌더링에는 이용할 수 없습니다.

name

선택적 식별자.

transform

인스턴싱된 지오메트리에 적용할 변환 행렬.

Output

render_archive_instance

인스턴스를 만들 때 이용할 수 있는 인스턴스 모양.

selector_instance

각 포인트의 특성 또는 ID 어레이를 기준으로 지오메트리나 인스턴스 모양 중 인스턴스화할 1개를 선택합니다. 출력은 set_instance_geometry 또는 set_instance_shape 노드에 연결할 수 있습니다.

또, <u>selector_instance</u> 노드 몇 개를 데이지 체인 방식으로 연결할 수 있는데 예를 들어, 한 가지 기준에 따라 소나무와 잎이 무성한 나무 중 하나를 선택한 다음 다른 기준에 따라 각 유형의 특정 버전들을 선택할 수 있습니다.

Inputs

instance_shapes

인스턴스화할 한 개 이상의 객체 또는 객체들의 어레이. 이것은 정상적인 기하학적 객체일 수도 있고 selector_instance, geo_instance 또는 render_archive_instance와 같은 노드들의 출력인 인스턴스 모양일 수도 있습니다.

selector_property

각 포인트마다 인스턴스화할 인스턴스 모양을 지정하는 포인트 객체의 long 특성의 이름으로, 0은 instance_shapes 어레이의 첫 번째 요소이고, 1은 두 번째 요소인 식입니다.

preview_geometry

렌더링은 하지 않고 뷰포트에서 미리보기할 때 이용할 기하학적 객체(선택 사항). 여기서 한 객체가 연결되는 경우, geo_instance 또는 이와 유사한 노드에 의해 instance_shapes에 연결되는 객체들에 이미 설정해 놓은 미리보기 지오메트리가 있다면 그 모든 지오메트리가 무효화됩니다.

wrap_out_of_range

인스턴스 모양들의 어레이 크기를 초과하는 인스턴스 ID 값들을 래핑합니다.

name

선택적 식별자.

transform 인스턴싱된 지오메트리에 적용할 변환 행렬.

Output

selector_instance

인스턴스 모양을 선택하는 객체.

set_instance_geometry

파티클 시스템과 같은 포인트 객체의 위치에서 기하학적 객체를 인스턴스화합니다.

Inputs

points

포인트 객체.

instance_geometries

인스턴스화할 1개 이상의 객체 또는 객체들의 어레이. 이것은 정상적인 기하학적 객체일 수도 있고 selector_instance, geo_instance 또는 render_archive_instance와 같은 노드들의 출력인 인스턴스 모양일 수도 있습니다.

preview_geometries

렌더링은 하지 않고 뷰포트에서 미리보기할 때 이용할 선택적 기하학적 객체. 여기에 연결되어 있는 객체가 있다면 모두 instance_geometries에는 연결되나 geo_instance 또는 이와 유사한 노드에 의해 정의되는 자체 미리보기 지오메트리가 없는 객체들과 관련해 표시됩니다.

팁:

한 객체를 연결하는 경우, 이 객체는 앞서 정의된 미리보기 지오메트리를 모두 무효화하고 모든 인스턴스를 위해 표시됩니다.

instance_id_property

각 포인트마다 인스턴스화할 인스턴스 모양을 지정하는 포인트 객체의 long 특성의 이름으로, 0은 instance_geometries 어레이의 첫 번째 요소이고, 1은 두 번째 요소인 식입니다.

overwrite_instance_id

사실인 경우, instance_id_property를 무효화하고 그 대신 연결된 instance_id_values를 이용합니다.

instance_id_values

각 포인트에서 인스턴스화할 인스턴스 모양을 지정하는 long 값들의 어레이.

wrap_out_of_range

인스턴스 모양들의 어레이 크기를 초과하는 인스턴스 ID 값들을 래핑합니다.

Output

instances

인스턴스 객체.

set_instance_shape

파티클 시스템과 같은 포인트 객체의 위치에서 기하학적 객체를 인스턴스화합니다.

이 노드의 대안으로, 많은 경우에 더 편리할 수 있는 set_instance_geometry를 이용하는 것을 고려해보십시오.

Inputs

points

포인트 객체.

instance_shape

인스턴스 모양. 이것은 selector_instance, geo_instance, render_archive_instance와 같은 노드의 출력이어야 합니다.

cleanup_unused_components

구현 불가.

Output

instances 인스턴스 객체.

is_empty

객체가 비어 있는지 여부를 반환하는데, 예를 들면, 포인트가 1개도 없는 파티클 시스템이 있을 수 있습니다.

Input

object

객체

Output

empty

객체가 비어 있다면 true를 반환합니다.

assign_material

기하학적 객체에 재질을 지정합니다.

Bifrost는 어떠한 렌더링 작업도 수행하지 못합니다. 이 노드는 Arnold나 뷰포트의 하드웨어 쉐이딩처럼 나중에 호환되는 렌더러들이 읽고 이용할 수 있게 되는 기하학적 객체의 특성을 설정하는 기능만 수행합니다. Bifrost 그래프에 지정된 재질은 호스트 장면에 있는 출력 bif 객체에 지정된 재질로 재지정될 수 있습니다.

Inputs

geometry

기하학적 객체

set_surface

기존의 표면 재질을 덮어쓰기 합니다.

surface_material

표면 색상으로 사용할 재질을 연결합니다.

set_displacement

기존의 변위 재질을 덮어쓰기 합니다.

displacement_material

변위에 사용할 재질을 연결합니다.

Output

out_geometry 수정된 객체

constant_surface_mat

색상이 일정한 단순 재질을 만듭니다.

Inputs

name

선택적 식별자

color

표면 RGB 값.

transparency

투명 RGB 값.

emission

방출 RGB 값.

Output

material_definition

재질 객체. 이것을 assign_material의 surface_material 포트로 연결할 수 있습니다.

set_geo_property_reference

객체의 geo 특성을 이용해 머티리얼 매개변수를 조종합니다.

Inputs

material_definition

수정할 머티리얼 객체.

material_property

제어할 머티리얼 매개변수의 이름(예: color).

geo_property 이용할 기하학적 특성(예: point_color).

Output

out_material_definition 수정된 머티리얼.

merge_geometry

기하학적 객체들의 어레이를 하나로 합칩니다. 객체들은 모두 유형이 같아야 하는데, 예를 들면 모두 메쉬이거나 모두 포인트 객체여야 합니다. 어레이에 유형이 서로 다른 지오메트리가 포함되어 있는 경우, 어레이에서 첫 번째 객체와 유형이 같은 객체들만 합쳐집니다.

Inputs

geometry

기하학적 객체들의 어레이

point_collections

합쳐진 객체에 포함될 포인트 컬렉션들의 목록으로 범위가 정해져 있습니다. 원래의 지오메트리들이 여기에 포함된 이름의 포인트 컬렉션을 갖고 있는 경우, 모든 컬렉션들이 출력 지오메트리에서 이름이 동일한 컬렉션으로 합쳐집니다. 지오메트리 내에 일치하는 컬렉션이 없는 경우, 자체 포인트들에 해당하는 합쳐진 컬렉션의 요소들은 false가 됩니다.

other_collections

합쳐진 객체에 포함될 기타 컬렉션들의 목록으로 범위가 정해져 있습니다. 메쉬 상에서는 페이스(face) 컬렉션만 지원됩니다.

Output

merged 하나로 병합된 지오메트리

remove_from_array

인덱스가 지정되어 있는 요소들을 입력 어레이에서 삭제합니다.

Inputs

array

요소들을 제거할 어레이

indices_to_remove 제거할 인덱스들의 어레이

Output

out_array 수정된 어레이

sample_property

한 지오메트리 상의 모든 위치에서 한 특성 값을 반환합니다. 모든 값은 가장 가까운 기하학적 구성요소들로부터 보간될 수 있습니다.

Inputs

geometry

표본을 추출할 지오메트리

locations

지오메트리 상의 위치들을 포함하는 객체로, get_closest_locations의 출력을 예로 들 수 있습니다.

default

특성과 유형이 같은 value 노드를 연결합니다. 이 값은 실패할 경우에 기본적으로 반환되는데, 예를 들면, 특성이 존재하지 않거나 위치가 유효하지 않은 경우가 있을 수 있습니다.

property

특성의 이름

method

구성요소들 사이에서 값을 반환하는 방법:

- <u>interpolated</u>는 가장 가까운 구성요소들에서 특성의 평균 값을 반환하며 거리를 기준으로 가중치가 적용됩니다.
- from_closest는 가장 가까운 구성요소에서 특성 값을 반환합니다.

Outputs

sampled_data

모든 위치에서의 특성 값들을 포함하는 어레이

success

검색 결과에 해당하는 부울 값들의 어레이. 한 요소가 false인 경우, 그 특성이 존재하지 않거나 해당 위치가 유효하지 않은 것입니다. sampled_data 어레이의 해당 요소는 default 값입니다.

set_geo_property_data

기존 기하학적 특성의 데이터 값을 설정합니다. 예를 들어, 입력 객체에 point_size가 존재하는 경우, 이 노드를 각 포인트의 크기로 이용할 수 있습니다.

이 특성이 존재하지 않는다면, 먼저 set_geo_property를 이용해서 이 특성을 만들어야 합니다.

일부 공통 특성의 경우, set_point_position과 같은 특정 컴파운드를 이용할 수도 있습니다.

Inputs

geometry

기하학적 객체.

property

특성의 이름. 이 특성이 존재하지 않는다면, 객체는 수정되지 않습니다.

data

특성의 대상이 되는 각 구성요소의 값들로 구성된 어레이.

- 이 어레이가 구성요소 개수보다 작다면, 이 특성의 기본 값으로 확장됩니다.
- 반대로, 어레이가 더 크다면, 잘립니다.
- 어레이의 유형이 잘못되었다면, 객체는 수정되지 않습니다.

Output

out_geometry

수정된 객체.

set_geo_property

기하학적 객체에서 포인트와 같은 구성요소를 대상으로 하는 특성을 만들고 초기화합니다.

한 객체에 해당 특성이 이미 존재한다면, 그 대신 set_geo_property_data 를 이용하면 됩니다. 일부 공통 특성의 경우, set_point_position과 같은 특정 컴파운드를 이용할 수도 있습니다.

Inputs

geometry

기하학적 객체.

property

특성의 이름.

default

특성의 기본 값.

data

특성의 데이터로, 보통 각 구성요소의 값 어레이를 나타냄.

target

특성이 대상으로 하는 기하학적 구성요소로, point_component, face_component, face_vertex_component, voxel_component 등이 있습니다.

Output

out_geometry

수정된 지오메트리.

set_motion_blur_mode

호환되는 렌더러에서 모션 블러의 렌더링 방식을 제어하기 위해 이용할 수 있는 특성을 기하학적 객체에 저장합니다.

Inputs

geometry

기하학적 객체.

motion_blur_mode

- Auto: 객체의 point_velocity 값들이 존재한다면 이 값들을 이용합니다. 그렇지 않으면, 서브프레임에서 평가합니다.
- Velocity: point_velocity 값들을 이용합니다.
- Sampled: 서브프레임에서 평가합니다.

Output

out_geometry

수정된 객체.

set_point_normal

point_normal 특성을 어레이 값들로 설정합니다.

Inputs

geometry 포인트를 포함하는 기하학적 객체

normals 노멀을 가리키는 float3 값들의 어레이

Output

out_geometry 수정된 기하학적 객체

set_point_position

point_position 특성을 어레이 값들로 설정합니다.

Inputs

geometry 포인트를 포함하는 기하학적 객체

positions 위치를 가리키는 float3 값들의 어레이

Output

out_geometry 수정된 기하학적 객체

set_point_shape

파티클 시스템에서와 같이 포인트를 나타내고 렌더링하는 데 사용하는 모양을 설정합니다.

Inputs

points

포인트 객체

shape

사용할 모양

🖋 참고:

Numeric은 효과가 없습니다.

screen_aligned

모양들이 카메라를 향하도록 설정합니다.

numeric_property

효과가 없습니다.

default_size 모양의 크기. 이 값은 포인트 구성요소에 point_size 특성이 있을 때 재지정됩니다.

Output

out_points

수정된 객체

set_point_velocity

point_velocity 특성을 어레이 값들로 설정합니다.

이 노드는 파티클을 실제로 이동시키지는 않습니다. 파티클을 옮기려면, 시뮬레이션을 통해서 point_velocity를 기준으로 point_position을 업데이트해야 합니다.

Inputs

geometry

포인트를 포함하는 기하학적 객체

velocities

속도를 가리키는 float3 값들의 어레이. 값의 단위는 m/s입니다.

Output

out_geometry

수정된 기하학적 객체

set_property

한 객체의 키-값 쌍을 설정합니다. 대부분의 경우, 이와 다른 노드를 이용하게 됩니다.

- <u>set_geo_property_data</u>를 이용해 파티클 시스템의 <u>point_velocity</u>와 같은 기존의 구성요소별 특성들의 값을 설정합니다.
- set_geo_property 를 이용해 각 구성요소의 특성을 만들고 초기화합니다.
- 일부 공통 특성의 경우, set_point_position과 같은 특정 컴파운드를 이용할 수도 있습니다.

Inputs

object

기하학적 객체.

key

특성의 이름.

value

Output

out_object

수정된 객체.

standard_surface_mat

Arnold 표준 표면 재질과 특성 이름이 같은 재질 정의를 생성합니다.

Inputs

name

선택적 식별자

base_weight

기본 색상의 무게

base_color

기본 RGB 값

diffuse_roughness

표면의 거칠기

metalness

재질이 나타내는 금속성의 정도

specular_weight

스페큘러 하이라이트의 휘도

specular_color

스페큘러 하이라이트의 색상

roughness

스페큘러 하이라이트의 광도

IOR

굴절률

anisotropy 브러쉬 메탈의 경우처럼 광반사의 지향성 편향도

rotation

이방성 효과를 순환합니다.

transmission_weight

표면을 투과하는 빛의 비율

transmission_color

투과된 빛의 색조를 조절합니다.

depth

transmission_color가 도달하는 깊이

scatter

산란도

scatter_anisotropy

산란의 지향성 편향 또는 이방성. 양의 값들은 앞쪽으로 산란하고 음의 값들은 뒤쪽으로 산란합니다.

dispersion_abbe

재질의 Abbe 값

extra_roughness

굴절을 블러링합니다(Blur). 양의 값은 굴절을 스페큘러 리플렉션보다 더 많이 블러링하고 음의 값은 이보다 적게 블러링합니다.

transmit_AOVs

전송을 통해서 렌더러에게 AOV(Arbitrary Output Variable)들을 통과시키라고 전달합니다.

subsurface_weight

디퓨스(diffuse)와 서브서피스 스캐터링(subsurface scattering) 간의 혼합

subsurface_color

서브서피스 스캐터링의 색조를 조절합니다.

radius

빛이 표면 아래로 확산할 수 있는 최대 거리

scale

빛이 표면 아래서 반사되는 지점까지 이동하는 거리를 제어합니다.

subsurface_anisotropy

서브서피스 스캐터링의 방향으로, -1은 완전히 뒤로 산란하고, 1은 완전히 앞으로 산란하며, 0은 고르게 산란합니다.

type

산란의 유형:

- 0: Diffusion
- 1: Random walk
- 2: Random walk v2

coat_weight 재질 표면의 클리어 코트 층의 무게

coat_color 클리어 코트 층의 색상

coat_roughness 클리어 코트 층의 스페큘러 리플렉션(specular reflection)의 광도. 이 값이 낮을수록 반사가 더 선명해집니다.

coat_IOR 클리어 코트 층의 굴절률

BIFROST

coat_anisotropy

클리어 코트 층의 광반사의 지향성 편향도

coat_rotation

클리어 코트 층의 이방성 효과를 순환합니다.

sheen_weight

벨벳이나 새틴과 같은 표면들의 근사치를 계산하는 데 적용할 수 있는 광택 층의 무게

sheen_color

광택 층의 색상

sheen_roughness

광택 층의 거칠기

emission_weight

방출된 빛의 양

emission_color

방출된 빛의 색상

Output

standard_surface

재질 객체. 이것을 assign_materia의 surface_material 포트로 연결할 수 있습니다.

file

이 섹션의 페이지들

- read_Alembic 노드
- read_bif_particles 노드
- read_Field3d 노드
- read_OpenVDB_points 上드
- read_OpenVDB_volume 노드
- read_OpenVDB_points 노드
- read_PDC 노드
- write_Alembic 노드
- write_Field3d 노드
- write_OpenVDB_points 上드
- write_OpenVDB_volume 노드
- write_OpenVDB 上드
- write_PDC 노드

상위 페이지: 참조

read_Alembic 노드

이 노드는 Alembic파일에서 Bifrost 메쉬를 읽어들입니다.

Inputs

filename [in] 파일 이름

directory [in] 디렉터리.

frame [in] 프레임 번호

frame_rate [in] 프레임 속도

properties [in] 읽어야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

interpolation [in] 서브 프레임 위치에서 데이터를 보간하기 위한 방법

Outputs

geometry [out] 출력 Bifrost 메쉬

read_bif_particles 노드

이 노드는 Maya Bifrost Fluids (.bif) 파일에서 Bifrost 포인트 클라우드를 읽어들입니다.

Inputs

directory [in] .bif 파일을 포함하고 있는 디렉터리

frame [in] 프레임 번호

enable_bounds [in] 정의된 최대/최소 경계 내에 있는 유체의 타일만 로드합니다.

min_bound [in]

최소 경계 상자 범위

max_bound [in] 최대 경계 상자 범위

Outputs

points [out] 출력 Bifrost 포인트 클라우드

success [out]

파일을 성공적으로 읽었다면 True입니다.

read_Field3d 노드

이 노드는 Field3D파일에서 Bifrost 볼륨을 읽어들입니다.

Inputs

filename [in] 파일 이름

directory [in] 디렉터리

frame [in] 프레임 번호

field_name [in] 읽어야 하는 Field3D 필드들의 목록으로 공간 범위가 정해져 있습니다.

properties [in] 읽어야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

tile_tree_traits [in] 출력 Bifrost 볼륨의 타일 트리 구성

resample [in] True인 경우, 볼륨을 다시 샘플링해 잔여 변환을 제거합니다.

Outputs

residual_translation [out] 볼륨 월드 스페이스 좌표를 저장된 Field3D 변환에 맞춰야 하는 오프셋

volume [out] 출력 Bifrost 볼륨

read_OpenVDB_points 노드

이 노드는 OpenVDB 파일에서 Bifrost 포인트 클라우드를 읽어들입니다.

Inputs

filename [in]

파일 이름

directory [in]

디렉터리

frame [in]

프레임 번호

properties [in]

읽어야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

points [out] 출력 Bifrost 포인트 클라우드

read_OpenVDB_volume 노드

이 노드는 OpenVDB 파일에서 Bifrost 볼륨을 읽어들입니다.

Inputs

filename [in]

파일 이름

directory [in]

디렉터리

frame [in] 프레임 번호. 이것은 전체 파일 이름에 포함됩니다.

properties [in] 읽어야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

tile_tree_traits [in]

출력 Bifrost 볼륨의 타일 트리 구성.

Outputs

volumes [out] 출력 Bifrost 볼륨

read_OpenVDB_points 노드

이 노드는 OpenVDB 파일에서 Bifrost 포인트 클라우드를 읽어들입니다.

Inputs

filename [in]

파일 이름

directory [in]

디렉터리

frame [in]

프레임 번호

properties [in]

읽어야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

points [out] 출력 Bifrost 포인트 클라우드

read_PDC 노드

이 노드는 Maya PDC 파일에서 Bifrost 포인트 클라우드를 읽어들입니다.

Inputs

filename [in]

파일 이름

directory [in]

디렉터리

frame [in] 프레임 번호.

Outputs

points [out]

출력 Bifrost 포인트

success [out] 파일을 성공적으로 읽었다면 True입니다.

write_Alembic 노드

이 노드는 Alembic파일에 Bifrost 메쉬를 씁니다.

Inputs

geometry [in]

쓰려는 Bifrost 메쉬

enable [in]

True인 경우, 파일 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다.

filename [in] 파일 이름

directory [in] 디렉터리

. . . .

frame [in]

프레임 번호

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

properties [in] 써야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

out_geometry [out] 입력 Bifrost 메쉬의 패스쓰루

success [out] 파일을 성공적으로 썼다면 True입니다.

file_path [out] 작성한 파일의 전체 경로

write_Field3d 노드

이 노드는 Field3D파일에 Bifrost 볼륨을 씁니다.

Inputs

volume [in] 쓰려는 Bifrost 볼륨

properties [in]

써야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

enable [in]

True인 경우, 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다.

filename [in] 파일 이름

directory [in]

디렉터리

frame [in] 프레임 번호

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

filter_property [in] 저장할 화소를 결정하는 데 이용하는 화소 특성

threshold [in] 크기가 threshold보다 더 큰 화소들이 저장됩니다.

file_format [in] Field3D와 함께 이용할 파일 형식. 잘 모르겠다면, Ogawa를 이용합니다.

field_name [in] 지정되면, Field3D 파일에 저장된 내부 파티션 이름을 덮어씁니다.

preview_name [in] 지정되면, 프리뷰 파티션에 있는 모든 특성들의 다운샘플링 버전을 주어진 이름으로 저장합니다.

preview_level [in] 프리뷰 필드의 해상도는 원래의 해상도를 2로 나눠 preview_level의 거듭제곱으로 계산됩니다.

Outputs

out_volume [out]

입력 볼륨의 패스쓰루

success [out] 파일을 성공적으로 썼다면 True입니다.

file_path [out] 작성한 파일의 전체 경로

write_OpenVDB_points 노드

이 노드는 OpenVDB 파일에 Bifrost 포인트 클라우드를 씁니다.

Inputs

points [in] 쓰려는 Bifrost 포인트 클라우드.

enable [in]

True인 경우, 파일 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다.

filename [in] 파일 이름

directory [in] 디렉터리

니팩니니

frame [in]

프레임 번호

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

properties [in] 써야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

out_points [out] 입력 포인트 클라우드의 패스쓰루.

success [out] 파일을 성공적으로 썼다면 True입니다.

file_path [out] 작성한 파일의 전체 경로

write_OpenVDB_volume 노드 이 노드는 OpenVDB 파일에 Bifrost 볼륨을 쑵니다.

Inputs

volumes [in] 쓰려는 Bifrost 볼륨

enable [in] True인 경우, 파일 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다.

filename [in] 파일 이름

directory [in] 디렉터리

frame [in] 프레임 번호. 이것은 전체 파일 이름에 포함됩니다.

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

properties [in] 써야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

filter_property [in] 저장할 화소를 결정하는 데 이용하는 화소 특성

Outputs

out_volumes [out] 입력 볼륨의 패스쓰루

success [out] 파일을 성공적으로 썼다면 True입니다.

file_path [out] 작성한 파일의 전체 경로
write_OpenVDB 노드

이 노드는 OpenVDB 파일에 Bifrost 포인트 클라우드와 볼륨을 씁니다.

Inputs

points [in] 쓰려는 Bifrost 포인트 클라우드

volumes [in] 쓰려는 Bifrost 볼륨

enable [in] True인 경우, 파일 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다

filename [in] 파일 이름

directory [in] 디렉터리

frame [in] 프레임 번호

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

properties [in] 써야 하는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

out_points [out] 입력 포인트 클라우드의 패스쓰루

out_volumes [out] 입력 볼륨의 패스쓰루

success [out] 파일을 성공적으로 썼다면 True입니다.

file_path [out] 작성한 파일의 전체 경로

상위 페이지: 파일

write_PDC 노드

이 노드는 Maya PDC 파일에 Bifrost 포인트 클라우드를 씁니다.

Inputs

points [in] 쓰려는 Bifrost 포인트

enable [in] True인 경우, 파일 쓰기가 가능합니다. 그렇지 않으면, 노드가 패스쓰루 역할을 합니다.

filename [in] 파일 이름

directory [in] 디렉터리

frame [in] 프레임 번호

overwrite [in] True인 경우, 기존 파일을 덮어씁니다.

Outputs

out_points [out] 입력 포인트의 패스쓰루.

success [out] 파일을 성공적으로 썼다면 True입니다.

상위 페이지: 파일

fluids

이 섹션의 페이지들

- add_acceleration 노드
- add_buoyancy 노드
- add_buoyant_acceleration 노드
- add_vorticity_confinement_acceleration 노드
- add_vorticity_confinement 上드
- aero_adaptivity
- aero_body 노드
- aero_point_body 노드
- aero_solver_settings
- basic_aero_graph
- basic_combustion_graph
- coarsen_refine 노드
- combust 노드
- combustion_settings
- compute_adaptivity 노드
- compute_mass_density 노드
- compute_mass_fractions 노드
- compute_time_step 노드
- condense_water_vapor 노드
- create_combustion_channels 노드
- create_improved_time_integration_channels 노드
- debug 노드
- diffuse 노드
- emit_points 노드
- emit 노드
- min_start_frame 노드
- set_resolution_field 노드
- simulate_aero
- solve_pressure 노드
- source_air
- source_fuel
- split_points 노드

- stash_acceleration 노드
- to_aero_transport_style 노드
- to_areo_transport_interpolation_type 노드
- to_int 노드
- transport 노드

상위 페이지: 참조

add_acceleration 노드

이 노드는 속도 필드에 가속을 추가하고 베타만큼 조정합니다.

Inputs

beta [in]

가속 스케일

improved_time_integration [in]

'improved time integration'이 활성화되어 있다면 True입니다.

body [inout]

입력/출력 바디 (Amino::Object)

add_buoyancy 노드

이 노드는 입력 바디의 속도 필드에 부력 가속을 추가하고 그 결과를 outBody에 넣습니다.

Inputs

aeroBody [in] 입력 에어로 바디 (Amino::Object)

time [in] 현재 시각

timeStep [in] 이 시간 단계의 크기(델타 타임)

coarsestDOFDepth [in] 타일 트리에서 부력이 추가되어야 하는 최저 해상도의 깊이

ambientTemperature [in] 연기 주변 온도(섭씨)

buoyancyVector [in] 부력 가속의 방향 벡터

buoyancyModel [in] 사용된 기초 부력 모델

mask [in] 부력에 곱해진 옵션 마스크

body [inout] 공기 바디 (Amino::Object)

add_buoyant_acceleration 노드

이 노드는 입력 바디의 부력 가속을 계산해 그 결과를 입력/출력 어레이에 추가합니다.

Inputs

body [in] 입력 바디 (Amino::Object)

time [in] 현재 시각

coarsest_dof_depth [in] 타일 트리에서 부력이 추가되어야 하는 최저 해상도의 깊이

ambient_temperature [in] 연기 주변 온도(섭씨)

buoyancy_vector [in] 부력 가속의 방향 벡터

buoyancy_model [in] 사용된 기초 부력 모델

mask [in] 부력에 곱해진 옵션 마스크

acceleration [inout] 산출된 가속도가 이 어레이에 추가됩니다.

add_vorticity_confinement_acceleration 노드

이 노드는 가속 어레이에 소용돌이 제한 가속을 추가합니다.

Inputs

body [in] 입력 바디 (Amino::Object)

vorticityConfinementStrength [in]

소용돌이 제한의 강도. [0;1]에서는 작은 값이 더 나은 결과를 만들어냅니다.

vorticityConfinementScale [in]

소용돌이 제한력의 공간 스케일. 0과 1 사이의 숫자(0은 가장 미세한 스케일을 1은 가장 미세하지 않은 스케일을 의미)

coarsestDOFDepth [in]

타일 트리에서 이동이 발생해야 하는 최저 해상도의 깊이

mask [in]

소용돌이 제한에 곱해진 옵션 마스크

acceleration [inout] 산출된 가속도가 어레이에 추가됩니다.

add_vorticity_confinement 노드

이 노드는 입력 바디의 속도 필드에 소용돌이 제한을 추가하고 그 결과를 outBody에 넣습니다.

Inputs

timeStep [in] 이 타임 스텝의 크기(델타 타임)

vorticityConfinementStrength [in]

소용돌이 제한의 강도. [0;1]에서는 작은 값이 더 나은 결과를 만들어냅니다.

vorticityConfinementScale [in]

소용돌이 제한력의 공간 스케일. 0과 1 사이의 숫자(0은 가장 미세한 스케일을, 1은 가장 미세하지 않은 스케일을 의미)

coarsestDOFDepth [in]

타일 트리에서 이동이 발생해야 하는 최저 해상도의 깊이

mask [in] 소용돌이 제한에 곱해진 옵션 마스크

body [inout] 공기 바디 (Amino::Object)

aero_adaptivity

공기 및 연소 시뮬레이션에서 상세도(공간 적응 해상도)가 서로 다른 영역들을 정의합니다.

솔버의 복셀 해상도는 공간에 따라 달라지며 상세도가 높을수록 카메라에 더 가까운 영역이나 상세도를 높일 것으로 기대되는 지역에 집중됩니다. 공간 적응 시뮬레이션은 대체로 공간 균일 시뮬레이션과 비슷하거나 더 높은 상세도를 실현하지만 컴퓨팅 비용은 더 적게 듭니다.

aero_adaptivity 노드는 시뮬레이션의 적응성을 두 가지 방식으로 제어할 수 있습니다.

- Auto-adaptivity는 지정된 기준에 따라 각 공간의 해상도를 낮추거나 높일 수 있습니다. 예를 들어, 복셀 포그(voxel fog) 밀도에서 시각적으로 상세도가 높은 근접 영역들의 해상도를 높일 수 있습니다.
- Resolution bounds는 공간 내 영역별로 해상도 수준을 지정합니다. 예를 들어, 카메라에서 멀리 떨어진 영역에 해상도를 낮게 지정할 수 있습니다.

적응성은 컴퓨팅에 일정량의 오버헤드를 더합니다. 이 오버헤드는 대체로 성능 개선 효과로 벌충됩니다. 하지만 그렇지 않은 경우도 있을 수 있는데, 예를 들면, 적응성이 실내에서는 떨어지도록 설정했지만 시뮬레이션이 매우 얇은 흐름이라면, 그 실내에서는 해상도를 낮출 수 있는 요소들이 그리 많지 않게 됩니다.

🖋 참고:

공간 적응성을 실현하는 또 다른 방법은 여러 fluid_detail_size를 포함하고 있는 소스를 이용하는 것입니다. 예를 들어, 카메라에서 멀리 있는 이미터(emitter)는 해상도가 낮을 수 있습니다(higher fluid_detail_size).

Auto Adaptivity`

enable_auto_adaptivity

솔버가 coarsen 설정에 따라 알아서 해상도를 조절하는 자동 적응성(auto-adaptivity) 모드를 활성화합니다.

coarsen

해상도를 높이거나 낮추는 기준을 선택합니다.

- interior: 복셀 포그 밀도가 interior_density_threshold 보다 큰 경우에 연기 기둥의 내부 해상도를 낮춥니다.
- <u>low_disturbance</u>: 속도 필드에서 disturbance가 vorticity, stretching, shearing의 조합으로 정의되는 low disturbance 영역에서는 해상도를 낮춥니다.
- 1ow_fog_detail: 복셀 포그 밀도가 충분히 잔잔해 더 낮은 해상도에서도 정확히 표현할 수 있는 포그(fog) 상세도가 낮은 영역에서 해상도를 낮춥니다.
- all: 모든 가용 기준을 동시에 반영해 해상도를 낮춥니다.

interior_density_threshold

자동 적응성 interior 옵션은 interior_density_threshold 를 이용해 복셀 포그 밀도의 영역을 분류하는데, 그러면, 이를 연기 기둥의 내부로 구분할 수 있습니다. 이 매개변수의 설정 가능한 값은 연기를 렌더링하는 방식에 따라 달라지는데, 렌더 설정이 카메라 뷰포인트로부터 연기 기둥을 얼마나 상세하게 감지할 수 있는지를 좌우하기 때문입니다.

fidelity

자동 적응성 알고리즘이 해상도를 최대한 낮출 수 있는 수준을 정의합니다. fidelity는 0과 1 사이의 값으로, 1은 해상도 낮추기가 실행되지 않음을 의미하고, 0은 기본적으로 최대 해상도 시뮬레이션에 비해 복셀의 단 30%만 유지되는 수준으로 그리드의 해상도를 낮출 수 있음을 의미합니다. 여기서 30% 설정은 aero_adaptivity 노드 안에 있는 fidelity_zero_accuracy를 조절해 변경할 수 있으나 aero_adaptivity 노드 자체에서는 노출되지 않습니다. fidelity 값이 낮을수록 시뮬레이션 속도가 빨라지지만, 최대 해상도 시뮬레이션에서의 결과와는 크게 다를 수도 있습니다. 적응 해상도 시뮬레이션과 최대 해상도 시뮬레이션이 최대한 비슷해지도록 하려면, adaptivity_smoothness 매개변수를 이용하면 됩니다.

adaptivity_smoothness

자동 적응성 알고리즘이 해상도에서 시각적으로 명확한 점프 형태의 인공물을 만들어내지 않도록 하기 위해 0과 1 사이의 값을 설정할 수 있습니다. 값이 클수록 해상도 낮추기 절차에서 시각적 인공물이 생길 가능성이 줄어듭니다. 하지만 값이 클수록 적응성이 더 적게 낮아지기 때문에 성능 면에서 손해를 볼 수도 있습니다. 이 매개변수의 설정은 연기를 렌더링하는 방식에 따라 달라집니다. 연기 기둥이 매우 투명하다면 보통 더 낮은 adaptivity_smoothness 값으로도 충분합니다.

Resolution Bounds

enable_resolution_bounds

해상도 바운드를 기준으로 적응성을 활성화하고, 메쉬를 이용해 해상도 바운드를 구성합니다.

해상도 레벨은 메쉬가 연결되는 포트에 의해 결정됩니다. 해상도는 각 해상도 레벨 사이에서 2 단위씩 올라갑니다. 입력 메쉬의 바운딩 박스에 정렬된 축은 실제 메쉬 자체가 아니라 해상도 바운드를 결정하는 데 이용됩니다.

더 낮은 해상도 바운드 안에 더 높은 해상도 바운드를 반드시 중첩시킬 필요는 없는데, 바운드를 임의로 배치할 수 있기 때문입니다. 더 높은 해상도 바운드 안에 더 낮은 해상도 바운드가 완전히 속해 있어도 이 낮은 해상도 바운드들은 아무 영향도 받지 않는데, 중첩 영역에서는 항상 가장 높은 해상도가 이용되기 때문입니다. 해상도 바운드는 이미터에서 fluid_detail_size 설정을 무효화합니다.

해상도 바운드를 나중에 변경할 수도 있는데, 예를 들면, 시뮬레이션에서 최대 해상도로 실행할 필요가 없는 프리 롤(pre-roll)이 필요한 경우가 있습니다. 이를 위해, if 노드를 이용해 현재 프레임 번호를 기준으로 포트의 입력 메쉬를 변경하면 됩니다.

base_resolution

연기와 연소 모습을 최대 해상도 수준에서 시뮬레이션해야 하는 공간의 영역을 지정하는 메쉬들의 어레이.

half_resolution

연기와 연소 모습을 기본 해상도의 절반 수준에서 시뮬레이션해야 하는 공간의 영역을 지정하는 메쉬들의 어레이.

quarter_resolution

연기와 연소 모습을 기본 해상도의 1/4 수준에서 시뮬레이션해야 하는 공간의 영역을 지정하는 메쉬들의 어레이.

eighth_resolution

연기와 연소 모습을 기본 해상도의 1/8 수준에서 시뮬레이션해야 하는 공간의 영역을 지정하는 메쉬들의 어레이.

Output

adaptivity

출력은 적응성 입력의 총합을 공기 솔버에 포함시킨 객체이며 이를 simulate_aero 노드의 adaptivity 포트에 연결해야 합니다.

aero_body 노드

이 노드는 연기 밀도와 온도 채널, 그리고 컨테이너인 복셀토폴로지를 포함한 빈 공기 바디를 생성합니다.

Inputs

source [in]

입력 소스들의 어레이. 비어 있지 않다면 detailSize를 좌우합니다.

detailSize [in]

연기를 최소 분해능으로 나타낼 수 있는 디테일 사이즈. 복셀 기반 솔버의 경우, 이것은 복셀 크기를 가리킵니다. 소스 입력이 비어 있지 않다면 무시해도 좋습니다.

ambientTemperature [in]

연기 주변 온도(섭씨)

create_mass_density_property [in]

True일 경우, 공기 객체에 질량 밀도 채널이 추가됩니다.

Outputs

outAeroBody [out]

출력 공기 바디 (Amino::Object).

aero_point_body 노드

이 노드는 포인트 속도, 연기 밀도, 온도 채널, 그리고 컨테이너를 포함하는 빈 에어로 바디를 생성합니다.

Inputs

ambientTemperature [in] 연기 주변 온도(섭씨).

Outputs

outAeroPointBody [out] 출력 공기 포인트 바디 (Amino::Object)

aero_solver_settings

공기와 연소 시뮬레이션을 위한 옵션을 설정합니다. 이 노드의 출력을 simulate_aero 노드의 settings 입력에 연결해야 합니다.

Inputs

combustion_settings

연소를 시뮬레이션할 때는 combustion_settings 객체를 combustion_settings 포트에 반드시 연결해야 합니다.

simulation_bounds

시뮬레이션 범위를 공간 내 한 영역으로 한정합니다. Bifrost 공기 솔버를 이용하면 바운드 없이 희소 및 공간 적응 시뮬레이션이 가능하지만 때로는 이것이 어려운 경우도 있습니다. 예를 들어, 전체 공간 중 한 특정 영역만 관련이 있다면 이 영역 외부까지 시뮬레이션할 필요는 없습니다. simulation_bounds 포트는 메쉬를 수용하며, 이 메쉬의 전역 축 정렬 바운딩 박스를 이용해 해당 영역을 결정합니다. 이 해당 영역 밖에서는 시뮬레이션 그리드의 해상도가 떨어지고 사실상 어떠한 시뮬레이션도 진행되지 않습니다.

General

enable_solver

공기 솔버를 작동시키거나 해제합니다.

use_master_start_frame

개별 이미터와 콜라이더(collider) 상의 프레임을 aero_solver_settings의 master_start_frame으로 재지정합니다.

master_start_frame

use_master_start_frame이 활성화되면, master_start_frame을 모든 이미터와 콜라이더의 글로벌 스타트 프레임으로 이용하게 됩니다.

store_mass_density

공기 객체의 voxel_mass_density 특성에 저장되는 유체의 질량 밀도를 계산합니다 . 후처리 또는 렌더링에 질량 밀도를 반영해야 하는 경우에는 이것을 설정하기만 하면 됩니다. 이 설정은 시뮬레이션 자체에는 영향을 주지 않습니다.

Aero Solver Globals

ambient_temperature

연기 주변의 공기 온도(섭씨)

style

시간의 흐름에 따라 유체 특성을 순방향으로 전달할 때 솔버가 이용하는 공간 보간 수단입니다. 이 보간 수단의 이름은 각각의 시각적 특성에 따라 정해지며 다음과 같이 정의합니다.

- smooth: 선형 보간
- fluffy: 복셀 간 CO 연속성을 가진 큐빅(cubic) 보간
- busy: 복셀 간 C1 연속성을 가진 큐빅(cubic) 보간(Catmull-Rom)
- wispy: 복셀 간 C1 연속성을 가진 단조 큐빅(cubic) 보간

simulation_speed

감지한 시뮬레이션의 시각 속도에 적용되는 스케일입니다. 1보다 큰 값은 시뮬레이션 속도를 높이고 1 미만인 값은 시뮬레이션 속도를 늦춥니다.

enable_buoyancy

부력을 활성화하거나 비활성화합니다.

buoyancy

부력 가속도의 방향과 크기를 결정합니다. 연기가 더 빨리 올라가게 하려면, 온도를 높이는 것보다 부력 벡터 크기를 높이는 것이 더 효과적입니다.

velocity_smoothness

대부분, 이 컨트롤은 필요하지 않기 때문에 그냥 0으로 설정해 놓으면 됩니다. 이 컨트롤이 필요한 상황이 되면, 더 높은 값으로 설정해 유체에 생기는 시각 라인 인공물의 개수를 줄일 수 있습니다. velocity_smoothness 는 겉보기 점도를 초래하지 않고 시각 라인 인공물을 제거하려고 하지만, 대부분의 경우, 시뮬레이션을 좀 더 점착성 있게 만듭니다. 따라서, velocity_smoothness 유효값 중 최저값을 권장하며 이 값으로 실험할 때에는 낮은 값(예: 0.1)으로 시작하는 것이 좋습니다. 중요한 사실은 시각 라인 인공물이 나타나는 경우에 겉보기 점도를 초래하지 않고 인공물을 제거할 수 있는 다른 유리한 방법도 있다는 것입니다.

- max_steps 를 낮춥니다.
- source_air 노드에서 fluid_detail_size 를 낮춥니다.
- style을 더 평활도 높은 인터폴레이터(interpolator)로 전환합니다. 평활도(낮음부터 높음까지)의 관점에서, style 모드의 순서는 다음과 같습니다. busy, fluffy, wispy, smooth.
- 장면에서 힘의 장 또는 소스가 시뮬레이션에 상당한 고주파 상세도를 주사하는 경우, 그것이 그리드에 주사되기 전에 이 기여도를 조금 평탄하게 하는 것(필터 폭이 1 복셀인 스무딩 필터 이용)도 도움이 될 수 있습니다.

kill_voxel_fog_threshold

공간 내 영역의 해상도를 낮출 수 있는 시점을 결정합니다. 특히, voxel_fog_density 값이 kill_voxel_fog_threshold보다 작을 경우, 솔버는 voxel_fog_density가 너무 낮아 시각화하기가 어렵다는 전제 하에서 결국 복셀의 해상도를 낮춰 컴퓨팅 속도를 높입니다. 권장 값은 연기의 렌더링 방식에 따라 달라집니다. 연기를 상당히 투명하게 렌더링하는 경우에는, 꽤 높은 kill_voxel_fog_threshold 값이 필요할 것입니다.

scene_units_in_meters

장면의 단위 크기(m). 이것은 장면 내 지오메트리(소스, 콜라이더 등)를 모델링한 스케일을 보상합니다. 이것이 중요한 이유는 많은 유체 현상이 스케일에 따라 크게 달라지기 때문입니다. 예를 들어, 장면 내 모든 지오메트리를 10으로 스케일링한 다음(실제 물리적 크기 기준) 원래의 스케일로 시뮬레이션 하고자 한다면, scene_units_in_meters = 0.1로 설정해야 합니다.

Accuracy

time_step_size

한 유체 구획이 추가 서브스텝을 트리거링하기 전에 한 타임 스텝에 이동할 수 있도록 하는 최대 복셀 수를 정의합니다. 이것을 CFL(Courant-Friedrichs-Lewy) 조건이라고도 합니다. 보통, 콜라이더를 포함한 상당히 세부적인 상호작용인 경우에는 5 정도 값이면 충분하나 1 또는 0.1까지 낮추어야 하는 상황도 있습니다. 경우에 따라, 콜라이더 상호작용의 품질을 높이려면, simulate-aero 아래 solve_aero_substep 컴파운드 내부에 중첩되는 <u>solve_pressure</u> 노드에서 IOP(Iterated Orthogonal Projection) 알고리즘의 최대 반복 횟수를 높여야 할수도 있습니다. 반면, <u>time_step_size</u>를 높이면 빠르게 이동하는 유체가 있을 때 이것이 보다 적은 서브스텝으로 이어질 것이므로 시뮬레이션 시간을 단축할 수 있습니다. 이 방법은 일반적으로 미사일 궤적이나 화쇄 효과와 같은 고속 시뮬레이션에 권장됩니다.

min_steps

매 프레임마다 서브스텝의 최소 min_steps 수를 거치도록 합니다.

max_steps

매 프레임마다 서브스텝의 최대 max_steps 수를 거치도록 합니다. 미사일 궤적과 같은 일부 시뮬레이션의 경우, 컴퓨팅 시간을 단축하기 위해 max_steps 를 1로 설정할 수 있습니다.

🖋 참고:

aero_solver_settings 안에는 aero_settings 노드에 대한 또 다른 매개변수로 max_transport_steps 가 있습니다. 이 항목은 이동 단계에서 서브스텝의 최대 수를 결정합니다. 경우에 따라 이 값을 낮춰서 성능을 높이거나 복셀 포그 밀도의 확산을 줄일 수 있습니다. 이 경우 aero_solver_settings 컴파운드를 먼저 불러와야 합니다.

Diffusion Handling

boost_detail_with_points

포인트를 이용해서 포그 밀도와 온도 특성을 추적함으로써 포인트들이 유체 내에서 이동할 때 시간의 흐름에 따른 확산을 방지합니다. 포인트들은 simulate_aero의 aero_points 포트에서도 출력됩니다.

point_emission_threshold

voxel_fog_density 임계값을 정의하는데, boost_detail_with_points가 활성화된 경우에 이 값보다 높으면 포인트들이 방출됩니다.

apply_sharpening

voxel_fog_density에 적용되는 샤프닝(sharpening) 필터를 활성화하고, 이를 통해 유체를 따라 이동할 때 시간에 따른 확산을 방지합니다. boost_detail_with_points에 비해 적은 비용으로 확산을 방지할 수 있습니다.

sharpening_amount

샤프닝(sharpening) 필터의 강도. 여기서는 큰 값이 voxel_fog_density를 왜곡할 수 있으므로 가급적 작은 값을 권장합니다.

sharpening_radius

샤프닝(sharpening) 필터의 반경. 반경이 클수록 결과가 더 선명해지지만 비용이 더 많이 든다는 단점이 있습니다.

Output

solver_settings

출력은 공기 관련 입력의 총합을 공기 솔버에 포함시킨 객체이며 이를 simulate_aero 노드의 settings 포트에 연결해야 합니다.

basic_aero_graph

basic_aero_graph는 피어오르는 연기 기둥을 만들어내는 단순 그래프를 설정합니다. 이 노드를 그래프의 최상위 레벨에 추가하고, 전개한 다음, 입력과 출력을 연결합니다.

basic_combustion_graph

basic_combustion_graph는 피어오르는 연기 기둥에 선행하는 폭발을 만들어내는 단순 그래프를 설정합니다. 이 노드를 그래프의 최상위 레벨에 추가하고, 전개한 다음, 입력과 출력을 연결합니다.

coarsen_refine 노드

이 노드는 입력 바디 컨테이너의 해상도를 높이거나 낮춰 흥미로운 특징을 모두 캡처하고 그 결과를 outBody에 넣습니다.

Inputs

source [in]

소스는 복셀화된 소스를 포함하는 Amino::Object 또는 BifrostExp::Objects 목록입니다. 소스가 서로 다른 시점의 객체들을 포함하고 있다면, 목록은 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 목록의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다.

refinementRegions [in]

공간 내 한 영역 그리고 해상도를 높여야 하는 깊이 레벨을 정의하는 계층형 바운딩 박스를 지정합니다. 목록의 인덱스가 깊이를 결정하는데, 첫 번째 요소가 해상도가 가장 높습니다.

resolutionFields [in]

공간 내 한 영역 그리고 해상도를 높여야 하는 깊이 레벨을 정의하는 바운딩 박스를 지정합니다. 0은 최고 해상도 레벨이며 값이 올라갈수록 디테일 사이즈가 제곱만큼 커집니다.

이는 레벨을 명시적으로 지정한다는 것과 어느 레벨에서든 다수의 바운드를 가질 수 있다는 점에서 refinementRegions와 다릅니다.

significanceChannelNames [in]

기본값(voxel_fog_density와 voxel_fuel_*)에 더해 중요성 채널로 사용될 추가 채널의 이름들로 구성된 어레이

frame [in]

현재 프레임

time [in]

현재 시각

timeStep [in] 이 타임 스텝의 크기(델타 타임)

frameStartTime [in]

프레임 시작 시간(서브프레임 보간에 유용)

frameEndTime [in]

프레임 종료 시간(서브프레임 보간에 유용)

cfl [in]

타임 스텝의 CFL 값

grading [in]

트리의 해상도를 결정합니다. 특히, 흥미로운 피처까지의 거리 또는 소스가 복셀 대각선 x 그레이딩의 1/2 미만인 경우 복셀의 해상도를 높입니다.

BIFROST

voxelPadding [in]

흥미로운 피처는 최소한 'voxelPadding'만큼 두꺼운 복셀 계층으로 채워집니다.

velocityPaddingFactor [in]

타일 트리가 속도 x timeStep의 방향과 크기로 확장되며 모든 값에 'velocityPaddingFactor'가 곱해집니다.

divergencePaddingFactor [in]

타일 트리가 발산으로 추정한 공칭 속도 x 'divergencePaddingFactor'만큼 방사형태로 바깥으로 확장됩니다.

significanceThreshold [in]

'significanceThreshold'보다 큰 값을 유효한 것으로 간주합니다.

body [inout] 출력 바디 (Amino::Object).

Outputs

outCoarsestDOFDepth [out]

타일 트리에서 DOF(Degree of Freedom)를 갖는 해상도가 가장 낮은 깊이.

outSimulationDomain [out]

월드 스페이스 좌표에 정의되는 이 바운딩 박스 안에서 컴퓨팅이 실행됩니다.

combust 노드

이 노드는 입력 매개변수와 객체에 의해 지정된 연소를 컴퓨팅 합니다.

Inputs

body [in] 입력 바디 (Amino::Object)

time [in] 현재 시각

timeStep [in] 이 시간 단계의 크기(델타 타임)

temperature [in] 온도 관련 매개변수

rateScale [in] rateScale 관련 매개변수

heatTransfer [in] 열 전달 관련 매개변수

soot [in] 그을음 관련 매개변수

flamePropagation [in] 불꽃 전파 관련 매개변수

coarsestDOFDepth [in] 타일 트리에서 이동이 발생해야 하는 최저 해상도의 깊이

body [inout] 공기 바디 (Amino::Object).

combustion_settings

연소 시뮬레이션 옵션을 설정합니다. 이 노드의 출력은 aero_solver_settings 노드의 combustion_settings 포트에 연결해야 합니다.

연소 현상(폭발, 불, 불꽃)은 공기 솔버로 시뮬레이션할 수 있습니다. 특정 현상을 연출하려면, 연소 물리학의 기초를 염두에 두어야 합니다.

- 온도가 특정 온도(발화점)를 넘으면, 연료와 산소가 반응해 수증기, 이산화탄소, 그을음 등과 같은 열과 부산물로 변합니다.
- 이렇게 생긴 열은 유체의 온도와 질량 밀도를 변화시켜 부피 팽창을 초래하고 폭발의 규모와 같은 요소를 좌우합니다.
- 열 전달은 복사, 전도(확산), 대류(이동)의 세 가지 기초 프로세스가 좌우합니다.

General

enable_combustion

연소 솔버를 활성화합니다.

combustion_smoothness

voxel_fog_density, voxel_fog_temperature 또는 블랙바디 복사 렌더에 계단현상(jaggies)이 있으면 이 값을 높입니다. 그러면 그을음이 더 고르게 발생하고 불꽃면의 온도가 더 평탄해집니다.

일반적으로 연소 시뮬레이션은 온도와 그을음 변화도를 키울 수 있습니다(특히 oxidize_soot가 활성화된 경우). 이것은 블랙바디 복사 렌더링에서 계단현상(jaggies)으로 나타날 수 있습니다.

트리큐빅 보간을 지원하는 고품질 렌더러에서 계단현상이 지속될 경우, 그을음과 온도에 대한 약간의 후처리가 필요할 수 있습니다.

- voxel_fog_density에 계단현상이 있을 경우, remap_property 노드를 이용해 0~1 범위로 다시 매핑합니다.
- voxel_temperature에서 강렬한 흰색에서 검정 배경으로 바뀌는 불꽃의 끝에 계단현상이 있거나 그을음 산화로 인해 매우 가늘어지는 불꽃 기둥에 계단현상이 있다면, 평활도(smoothness) 범위가 0~1인
 smooth_aero_temperature 노드를 이용해보십시오. 단, smooth_aero_temperature 를 적용한 다음에는 온도 스케일을 살짝 미조정해서 정확한 블랙바디 복사 형상을 복원해야 할 수도 있습니다.
- 또는, smooth_volume 노드를 이용해 일반적인 평활화를 voxel_fog_density 또는 voxel_temperature에 적용할 수도 있으나, 이 방법은 더 미세한 부분을 평활화할 수도 있고 현실적인 블랙바디 복사 렌더링을 위해 온도를 다시 매핑해야 할 수도 있기 때문에 최후의 방법으로 고려해야 합니다.

store_blue_flame

voxel_combustion_rate 특성의 출력을 활성화하는데, 이것은 스펙트럼선 방사에 해당하는 파란색 불꽃으로 렌더링됩니다.

store_water_vapor

연료가 있고 가시 뿌염 농도가 있는 지역에서 벗어나더라도 솔버가 수증기를 안정적으로 추적할 수 있게 합니다. 수증기를 렌더링하거나 후처리에 이용해야 하는 경우, store_water_vapor를 활성화해야 하지만 이 설정은 시뮬레이션 자체에 아무런 영향을 주지 않습니다. 수증기는 voxel_gas_vapor 특성에 저장되며 condense_water_vapor 노드를 이용해 렌더링에 대비할 수 있습니다.

Thermodynbamic Properties

expansion_scale

연소로 인한 온도 변화의 결과로 부피가 변하는 것을 스케일링합니다. expansion_scale 값이 클수록, 폭발은 더욱 격렬해집니다. expansion_scale을 높이면 속도가 크게 증가할 수 있는데, 그러면 서브스텝이 늘어나 시뮬레이션 속도가 떨어질 수 있습니다. 기본 설정으로 시작해 원하는 효과가 연출될 때까지 expansion_scale을 점차적으로 조절합니다.

radiative_heating

복사열의 양을 스케일링해 물질 열이 더 빨라지거나 느려지도록 만듭니다. 값이 커질수록 불꽃 속도가 증가하고 멀리 있는 연료가 점화될 수도 있습니다. 복사열은 멀리 떨어져 있는 물질 사이에 존재하는 공기는 가열하지 않고 물질만 가열할 수 있습니다.

radiative_cooling

복사 냉각에 적용되는 별도의 스케일링 인수로서 물질 냉각의 속도를 조절합니다. 값이 클수록 불꽃의 높이가 낮아집니다. 복사 냉각은 슈테판-볼츠만(Stefan-Boltzmann) 법칙을 따릅니다.

flame_propagation_speed

불꽃이 전파되는 속도를 스케일링합니다. 공간 내 특정 지점에서의 불꽃 전파 속도는 일반적으로 해당 지점에 존재하는 연료, 산소, 부산물의 조합에 의해 결정됩니다. Bifrost에서의 불꽃 전파 속도는 층류 불꽃 속도의 실제 측정값을 기준으로 하지만, flame_propagation_speed 매개변수로 이 속도를 변경할 수 있습니다.

temperature_diffusion

열이 인근 복셀들로 퍼져 나가는 것을 제어합니다. 단, 복사 열과는 달리, 확산에 의해 공기가 가열됩니다.

oxygen_diffusion

산소 확산량을 제어합니다. 산소 확산은 그을음 산화와 같이 공기 중의 산소가 그을음 덩어리를 공격해 다른 (보이지 않는) 부산물과 연기 없는 불꽃을 만들어내는 현상을 모델링하는 데 중요합니다. 산소 확산은 확산 불꽃을 시뮬레이션할 때도 중요한데, 이 경우에는 연료가 산호와 기혼합되지 않지만 연료와 산화제 사이에서 산소와 연료가 확산 프로세스에 의해 혼합되는 박판에서만 실제 연소가 일어납니다.

Soot Properties

emit_soot

연료가 탈 때 그을음이 자동으로 생기도록 합니다. 그을음은 voxel_fog_density 특성에 추가됩니다.

soot_model

그을음을 만들어내는 데 이용하는 모델을 결정합니다. 현재, 두 가지 옵션이 있습니다.

- Physical: 기본적으로 실제 그을음이 생기는 곳에서만 그을음을 만들어내는 물리적 검증을 거친 그을음 모델에 따라 그을음이 생성됩니다. 이것은 혼동을 유발할 수도 있는데, 모든 가용 연료와 산호가 반응해 부산물을 형성하는 화학량 조건에서는 그을음이 생기지 않기 때문입니다. 촛불처럼 규모가 작은 현상의 경우, 물리적 그을음 모델과 store_blue_flame 옵션을 함께 이용해서 그을음과 파란색 불꽃의 조합을 렌더링하는 것이 유리합니다.
- Simple: 이 모델은 반응 속도에 따라 그을음을 만들어내므로 이 경우에는 화학량 조건에서 그을음이 생성됩니다.

soot_formation_rate

그을음 형성 속도에 적용되는 스케일링 인수로서, 생생되는 그을음의 양을 조절합니다.

lower_formation_limit

Physical 그을음 모델에만 해당되며 그을음이 형성되는 위치를 좌우할 최저 한도를 스케일링합니다. 0으로 설정할 경우, 연료 질량 분율이 0보다 클 때마다 그을음이 형성됩니다. 이는 실제 관측 결과와는 다르겠지만 예술적 효과를 주기에 유용할 수 있습니다.

upper_formation_limit

그을음이 형성되는 위치를 결정할 최대 한도를 스케일링합니다. 이것은 Physical 그을음 모델에만 해당됩니다. 실제로는 연료 질량 분율이 특정 한도를 초과할 경우 그을음이 형성되지 않지만, upper_formation_limit = 이용해서 이 한도를 수정할 수 있습니다. 만일, 그을음이 너무 적게 생긴다면, upper_formation_limit = 2 이상으로 설정하면 됩니다.

oxidize_soot

그을음을 산화시킵니다. 그을음 산화는 공기 중의 산소가 그을음 덩어리를 공격해 다른 (보이지 않는) 부산물을 만들어내는 프로세스입니다. 이를 이용해 깨끗한 촛불처럼 연기 없는 불꽃을 연출할 수 있습니다. 이 옵션으로 산화 프로세스가 시작되도록 하려면 보통 oxygen_diffusion도 활성화해야 합니다. Bifrost에 적용된 산호 모델은 실제 그을음 산화 측정 결과를 기준으로 물리적 검증을 거쳤습니다.

soot_oxidation_rate

그을음 산화 속도를 스케일링해 산화되는 그을음의 양을 조절합니다. 예를 들어, 이것을 이용해서 불꽃 속 연기의 높이나 불꽃의 높이를 낮출 수 있습니다. 실제로, 원하는 불꽃 높이를 실현하기 위해서는 soot_oxidation_rate와 oxygen_diffusion을 1보다 큰 값으로 설정해야 할 수도 있습니다.

lower_oxidation_limit

그을음 산화가 일어나는 시점을 결정할 최저 온도를 스케일링합니다. 실험 결과를 토대로 할 때, 기본적으로 그을음 산화가 시작되는 최저 온도는 1300K입니다. 하지만, 예술적 효과를 위해 모든 온도에서 그을음이 산화되기를 바란다면 lower_oxidation_limit를 0으로 설정하면 됩니다.

upper_oxidation_limit

그을음 산화가 시작되는 시점의 최고 한도를 스케일링합니다. 특히, 연료 질량 분율이 특정 한도를 초과할 경우, 실제로는 산화가 일어나지 않지만 예술적 효과를 위해 upper_oxidation_limit 를 이용해서 이 한도를 변경할 수 있습니다.

Output

combustion_settings

출력은 연소 관련 입력의 총합을 공기 솔버에 포함시킨 객체이며 이를 aero_solver_settings 노드의 combustion_settings 포트에 연결해야 합니다.

compute_adaptivity 노드

이 노드는 해상도를 높이거나 낮추는 coarsen/refine 노드를 유도하는 적응성 채널을 컴퓨팅합니다.

Inputs

adaptivity_params [in]

입력 적응성 매개변수

significanceChannelNames [in]

기본값(voxel_fog_density와 voxel_fuel_*)에 더해 중요성 채널로 사용될 추가 채널의 이름들로 구성된 어레이

significance_threshold [in]

'significanceThreshold' 보다 큰 값을 유효한 것으로 간주합니다.

interior_distance_mapping [in]

공기-유체 인터페이스 거리를 유체 내 트리에서 원하는 깊이 레벨로 매핑하는 데 영향을 미치거나 내부 거리 충실도가 적응성에 미치는 영향을 제어하는 F-곡선.

strain_rate_mapping [in]

변형 속도 크기를 트리에서 원하는 깊이 레벨로 매핑하는 데 영향을 미치거나 변형 속도 충실도가 적응성에 미치는 영향을 제어하는 F-곡선.

vorticity_mapping [in]

소용돌이 크기를 트리에서 원하는 깊이 레벨로 매핑하는 데 영향을 미치거나 소용돌이 충실도가 적응성에 미치는 영향을 제어하는 F-곡선.

fog_density_mapping [in]

뿌염 농도 충실도가 적응성에 미치는 영향을 제어하는 F-곡선.

body [inout]

해상도 향상 채널을 포함한 입력/출력 바디 (Amino::Object)

compute_mass_density 노드

이 노드는 이상 기체 법칙에 따라 모든 가용 종의 질량 밀도와 온도를 컴퓨팅해 출력을 voxel_mass_density 특성에 넣습니다.

compute_mass_fractions 노드

이 노드는 연소 속도와 산소 분율을 기준으로 연료, 지소, 산소의 질량 분율을 계산합니다.

Inputs

burn_rate [in] 입력 연소 속도가 0은 연료 풍부 조건에, 1은 화학량론에, 2는 연료 희박에 각각 해당합니다.

oxygen_fraction [in] 산화제 내 산소의 분율 (1=순수 산소, 0.2=공기, 0=산소 없음).

fuel_type [in] 연료의 유형:

Outputs

out_fuel_mass_fraction [out] 출력 연료 질량 분율.

out_oxygen_mass_fraction [out] 출력 산소 질량 분율.

out_nitrogen_mass_fraction [out] 출력 질소 질량 분율.

compute_time_step 노드

이 노드는 서브 스텝 루프의 타임 스텝을 계산합니다.

Inputs

body [in]

복셀 속도 필드를 포함하는 것으로 추정되는 입력 바디 (Amino::Object)

collision [in]

'collision'은 복셀화된 콜라이더를 포함하는 Amino::Object 또는 BifrostExp::Objects 목록입니다. 콜리즌(collision)이 서로 다른 시점의 객체들을 포함하고 있다면, 목록은 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 목록의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다.

cfl [in]

CFL 값 즉, 한 서브 스텝으로 전송할 수 있는 최대 복셀 수

minSteps [in] 서브스텝 루프가 거치는 최소 스텝 수

maxSteps [in] 서브스텝 루프가 거치는 최대 스텝 수

time [in]

현재 시각

startTime [in]

프레임 시작 시간

endTime [in]

프레임 종료 시간

frame [in]

현재 프레임

startFrame [in]

시작 프레임

Outputs

timeStep [out] 이 시간 단계의 크기(델타 타임)

done [out]

true인 경우, endTime에 도달한 것입니다.

condense_water_vapor 노드

이 노드는 온도 특성 그리고 사용자 지정 온도 간격(실제로는 섭씨 0도~100도) 내 상대 습도를 기준으로 수증기 특성을 응결시킵니다. 입력 객체의 수증기 특성이 응결된 경우에는 출력 객체의 수증기 특성이 0 이외의 값이 됩니다. 예술적 효과를 위해, 최저 온도와 최고 온도 한 쌍이 노출되어 응결이 일어난 온도 범위를 미조정할 수 있게 됩니다.

Inputs

min_temperature [in]

수증기가 응결하는 최저 온도. 이 온도 밑에서는 모든 수증기가 응결됩니다.

max_temperature [in]

수증기가 응결하는 최고 온도. 이 온도 위에서는 수증기가 전혀 응결되지 않습니다.

body [inout]

공기 바디 (Amino::Object)

create_combustion_channels 노드

이 노드는 연소에 필요한 채널을 입력 객체에 추가합니다.

Inputs

create_combustion_rate_channel [in]

true라면, 솔버가 "combustion_rate"라는 이름의 채널을 만들고 이 채널이 연소 속도를 추적합니다.

body [inout]

공기 바디 (Amino::Object).

create_improved_time_integration_channels 노드

이 노드는 개선된 시간 적분에 필요한 채널을 입력 객체에 추가합니다.

Inputs

body [inout]

바디 (Amino::Object).

debug 노드

이 노드는 솔버가 디버그 정보를 출력할 때 이용하는 각종 디버그 매개변수를 사용자가 설정할 수 있도록 허용합니다.

Inputs

body [in] 입력 패스쓰루 바디 (Amino::Object)

enable [in] 디버깅 모드와 슬라이스 이미지 렌더링을 활성화하려면 true로 설정

visualizeTiles [in] 렌더링한 슬라이스에서 타일을 시각화 하려면 true로 설정

visualizeCorners [in]

렌더링한 타일에서 코너를 시각화 하려면 true로 설정

useLinearSampler [in] 슬라이스를 렌더링할 때 선형 샘플러를 이용하려면 true로 설정

enableDebugPrint [in] 렌더링한 슬라이스와 (tsX, tsY) 지점에서 값을 출력하려면 true로 설정

tsX [in] 디버그 정보가 stdout로 프린팅 될 X 타일 좌표

tsY [in] 디버그 정보가 stdout로 프린팅 될 Y 타일 좌표

frame [in] 시뮬레이션의 입력 프레임

startFrame [in] 디버깅 모드가 시작되는 프레임

verboseLevel [in] 솔버에서 이용하는 verbose 레벨

imageScale [in] 이미지 스케일: 출력 슬라이스 이미지의 각 복셀이 이 값만큼 스케일링됩니다.

slice [in] 슬라이스 좌표. 슬라이스는 XY 평면과 병렬을 이룹니다.

path [in]

슬라이스 이미지가 출력될 경로

Outputs

outBody [out]

출력 패스쓰루 바디 (Amino::Object)

diffuse 노드

매개변수

Inputs

maxiterations [in] 최대 반복 횟수

relativeResidual [in] 이 상대 잔차(현재의 오차 기준을 최초 오차 기준으로 나눈 값)에 도달할 때 솔버가 멈춥니다.

diffusionConstant [in] 확산 상수(반드시 0보다 커야 함). 확산 상수가 클수록 확산이 커집니다.

timeStep [in] 타임 스텝

coarsestDOFDepth [in] 타일 트리에서 확산이 발생해야 하는 최저 해상도의 깊이

attribute [in] 확산되고 있는 채널/속성의 이름

body [inout] 공기 바디 (Amino::Object)
emit_points 노드

이 노드는 포인트를 bif Amino::Object로 방출합니다.

Inputs

aeroBody [in]

파티클을 효율적으로 방출하는 데 사용되는 타일 트리를 포함하는 입력 에어로 바디 (Amino::Object).

sources [in]

소스는 복셀화된 소스를 포함하는 Amino::Object 또는 BifrostExp::Objects 어레이입니다. 소스가 서로 다른 시점의 객체들을 포함하고 있다면, 어레이는 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 어레이의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다. 소스는 다음과 같은 이름의 특성들을 포함할 수 있고 이런 특성들은 노드의 입력 매개변수들에 우선합니다.

- "amount" (float): 한 타임 스텝 중에 방출하는 연기의 양(방출 속도).
 - "voxel_temperature" (float): 온도 방출 속도.
- "continuous" (float): 0보다 크면 방출이 계속되고 그렇지 않으면 시작 프레임에서만 발생합니다.
- "emitVelocity" (float3): 소싱된 복셀의 속도에 더해지는 등가속.
- "density" (float): 소스에 의해 방출되는 물질의 물리적 밀도(질량/부피). 그을음인 경우, 밀도는 1800kg/m^3입니다.
- "inheritGeoVelocity" (float): 소싱된 복셀들이 소스 자체의 속도를 상속하는 방식을 결정하는 인수. 값이 0이면 상속이 이뤄지지 않음을 의미합니다.
- "fluidDetailSize" (float): 소스가 방출할 해상도 레벨을 결정합니다.

sourcingAcceleration [in]

소스 BIF에서 복셀화된 가속 채널의 이름으로 소싱된 복셀의 속도에 더해집니다. 비어 있다면, 소스 내 복셀들의 가속도가 0으로 설정됩니다.

time [in]

현재 시각

timeStep [in] 이 타임 스텝의 크기(델타 타임)

velocityTimeStep [in]

속도에 반영되는 이 타임 스텝의 크기(델타 타임)

frameStartTime [in]

프레임 시작 시간(서브프레임 보간에 유용)

frameEndTime [in]

프레임 종료 시간(서브프레임 보간에 유용)

BIFROST

coarsestDOFDepth [in]

타일 트리에서 방출이 발생해야 하는 최저 해상도의 깊이

pointDensity [in]

단위 부피당 포인트 개수

densitySignificanceThreshold [in]

소스의 밀도가 이 임계값을 초과하면 포인트가 복셀로 방출됩니다.

bif [inout]

방출되는 포인트를 포함하는 입력/출력 Amino::Object.

emit 노드

이 노드는 공기 바디로 발산해 결과를 outAeroBody에 넣습니다.

Inputs

source [in]

소스는 복셀화된 소스를 포함하는 Amino::Object 또는 BifrostExp::Objects 어레이입니다. 소스가 서로 다른 시점의 객체들을 포함하고 있다면, 어레이는 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 어레이의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다. 소스는 다음과 같은 이름의 특성들을 포함할 수 있고 이런 특성들은 노드의 입력 매개변수들에 우선합니다.

- "amount" (float): 한 타임 스텝 중에 방출하는 연기의 양(방출 속도).
 - "voxel_temperature" (float): 온도 방출 속도.
- "continuous" (float): 0보다 크면 방출이 계속되고 그렇지 않으면 시작 프레임에서만 발생합니다.
- "emitVelocity" (float3): 소싱된 복셀의 속도에 더해지는 등가속.
- "density" (float): 소스에 의해 방출되는 물질의 물리적 밀도(질량/부피). 그을음인 경우, 밀도는 1800kg/m^3입니다.
- "inheritGeoVelocity" (float): 소싱된 복셀들이 소스 자체의 속도를 상속하는 방식을 결정하는 인수. 값이 0이면 상속이 이뤄지지 않음을 의미합니다.
- "fluidDetailSize" (float): 소스가 방출할 해상도 레벨을 결정합니다.

sourcingAcceleration [in]

소스 BIF에서 복셀화된 가속 채널의 이름으로 소싱된 복셀의 속도에 더해집니다. 비어 있다면, 소스 내 복셀들의 가속도가 0으로 설정됩니다.

time [in]

현재 시각

timeStep [in] 이 타임 스텝의 크기(델타 타임)

velocityTimeStep [in]

속도에 반영되는 이 타임 스텝의 크기(델타 타임)

frameStartTime [in]

프레임 시작 시간(서브프레임 보간에 유용)

frameEndTime [in]

프레임 종료 시간(서브프레임 보간에 유용)

coarsestDOFDepth [in]

타일 트리에서 방출이 발생해야 하는 최저 해상도의 깊이

simulationDomain [in]

월드 스페이스 좌표에 정의되는 이 바운딩 박스 안에서 방출이 일어나게 됩니다.

aeroBody [inout]

공기 바디 (Amino::Object).

min_start_frame 노드

이 노드는 입력 소스의 최소 startFrame 특성을 찾아냅니다.

Inputs

sources [in] 소스는 복셀화된 소스를 포함하는 Amino::Object 또는 BifrostExp::Objects 어레이입니다.

Outputs

out_start_frame [out] 입력 소스 중 최소 startFrame

set_resolution_field 노드

이 노드는 ResolutionField 구조를 생성합니다.

Inputs

min [in] 바운딩 박스에 정렬된 필드 축의 최소 값(float3)

max [in]

바운딩 박스에 정렬된 필드 축의 최대 값(float3)

level [in]

필드의 해상도 레벨(int). 0은 최대 해상도, 1은 1/2 해상도, 2는 1/4 해상도, 3은 1/8 해상도 등입니다.

Outputs

field [out] 출력 필드 (Fluids::ResolutioField)

simulate_aero

simulate_aero 컴파운드는 공기 및 연소 시뮬레이션을 위한 코어 솔버를 포함하고 있습니다.

사용 방법:

- aero_solver_settings 노드를 settings 입력에 연결하고, 선택 사항으로, combustion_settings를 aero_solver_settings 노드에 데이지 체인 방식으로 연결합니다.
- source, collider, influence 노드를 해당하는 simulate_aero 입력에 각각 연결합니다.
- 공간 적응 시뮬레이션의 경우, aero_adaptivity 노드를 adaptivity 포트에 연결합니다.

Inputs

sources

그리드로 방출하려는 위치와 대상을 정의하는 source_air와 source_fuel 노드에서 출력되는 객체들의 어레이.

collisions

연기가 따라서 흐르고 이리저리 흩어지게 하는 솔리드 객체에 해당하는 collider 객체들의 어레이.

influence_fields

힘과 미적 컨트롤을 정의하는 influence 필드들의 어레이.

settings

aero_solver_settings 노드의 출력 객체에 연결되어야 하며, 기본적인 공기 및 연소 설정을 구성합니다.

adaptivity

aero_adaptivity 노드의 출력에 연결되어야 합니다.

Output

aero_volume 연기 및 연소 특성의 복셀 부피

aero_points

```
aero_solver_settings에서 boost_detail_with_points가 활성화된 경우에만 정의할 수 있는 포인트 객체
```

solve_pressure 노드

이 노드는 입력 바디 내 속도 필드의 압력 투영을 실행해 결과를 outBody에 넣습니다.

Inputs

collision [in]

'collision'은 복셀화된 콜라이더를 포함하는 Amino::Object 또는 BifrostExp::Objects 목록입니다. 콜리즌(collision)이 서로 다른 시점의 객체들을 포함하고 있다면, 목록은 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 목록의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다.

time [in]

현재 시각

timeStep [in] 이 타임 스텝의 크기(델타 타임)

frameStartTime [in] 프레임 시작 시간(서브프레임 보간에 유용)

frameEndTime [in] 프레임 종료 시간(서브프레임 보간에 유용)

ambientRho [in]

이산(discretization)에 사용되는 주변의 물리적 밀도(단위 부피당 질량). 섭씨 20도인 공기일 경우, 이 값은 1.2041입니다.

relativeLinearSolverResidual [in]

이 상대 잔차(현재의 오차 기준을 최초 오차 기준으로 나눈 값)에 도달할 때 솔버가 멈춥니다. 결과의 정확도를 낮추더라도 더 빠른 결과를 원한다면 값을 높입니다. 적정 값의 범위는 1e-5 ~ 1e-2입니다.

maxLinearSolverIterations [in]

선형 솔버가 거치는 최대 반복 횟수. 결과의 정확도를 낮추더라도 더 빠른 결과를 원한다면 값을 높입니다. 멀티그리드 솔버일 경우, 적정 값의 범위는 1 ~ 10입니다.

numPreSmoothingIterations [in]

각 사이클 중 멀티그리드 솔버가 거치는 프리 스무딩(pre-smoothing) 반복 횟수

numPostSmoothingIterations [in]

각 사이클 중 멀티그리드 솔버가 거치는 포스트 스무딩(post-smoothing) 반복 횟수

maxIterationsIOP [in]

IOP(Iterated Orthogonal Projection) 방식에서 솔리드 경계 조건을 강제 적용하기 위해 거치는 최대 반복 횟수

BIFROST

errorThresholdIOP [in]

IOP(Iterated Orthogonal Projection) 방식에서 솔리드 경계 조건을 강제 적용하기 위해 적용하는 최소 오차 임계값(멈춤 기준)

velocityFilterType [in] 속도 필터의 유형

numVelocityFilterIterations [in]

속도 필터가 적용되는 횟수. 속도 필터는 라인 인공물을 제거하지만 오버헤드를 더합니다.

velocityFilterScale [in]

컴퓨팅 된 속도 필터의 무게에 곱해지는 스케일로 필터의 영향력을 높이거나 낮춥니다. 예를 들어, 인공물을 제한하고 동시에 속도-확산을 방지하려면 작은 스케일만으로도 충분할 수 있습니다.

time [in]

현재 시각

coarsestDOFDepth [in]

타일 트리에서 이동이 발생해야 하는 최저 해상도의 깊이

simulationDomain [in]

월드 스페이스 좌표에 정의되는 이 바운딩 박스 안에서 압력 솔버 컴퓨팅이 실행됩니다.

pressureSolveType [in]

압력 솔버 내에서 이용하는 이산(discretization)의 유형.

body [inout]

공기 바디 (Amino::Object).

source_air

시뮬레이션 그리드로 특성을 방출하는 소스의 매개변수들을 지정합니다.

Inputs

geometry

소스 지오메트리를 정의하는 입력 메쉬, 포인트, 볼륨들의 어레이

General

enable_air_source

소스를 활성화합니다.

start_frame

소스의 방출이 시작되는 프레임

use_end_frame

활성화해서 특정 프레임에서 방출이 멈추도록 할 수 있습니다. 그렇지 않을 경우 시뮬레이션 내내 방출이 계속됩니다.

end_frame

소스의 방출이 중단되는 프레임

Resolution

resolution_mode

볼륨 해상도를 좌우하는 메소드

- Absolute: 이미터 복셀의 크기는 월드 스페이스 단위의 fluid_detail_size 값입니다.
- Relative: fluid_detail_size는 소스 축에 정렬된 바운딩 박스의 최대 치수에 비례하도록 설정합니다.
 따라서 크기가 20 유닛인 큐브는 Relative 모드에서 지정된 fluid_detail_size 값보다 20배나 더 큰 월드 스페이스 디테일 사이즈가 됩니다. 소스의 경우, 시작 프레임의 소스 메쉬 바운드를 이용해 전체 애니메이션의 fluid_detail_size와 geo_detail_size를 계산하는데, 소스 사이즈를 애니메이트할 때에도 마찬가지입니다.

geo_detail_size

복셀화된 메쉬에서 최소 분해능으로 나타낼 수 있는 디테일 사이즈. 해상도가 낮아 성능을 높이려면 이 값을 fluid_detail_size보다 높게 설정하면 됩니다.

geo_volume_mode

입력 메쉬를 복셀화하는 방식을 정의합니다.

- Solid: 이 모드는 다각형 메쉬를 솔리드로 복셀화하는데, 메쉬가 워터타이트(watertight)일 때 가장 효과적입니다.
- Shell: 이 모드는 메쉬를 쉘로 복셀화하며 메쉬가 워터타이트일 필요는 없습니다.

geo_volume_offset

복셀화된 볼륨을 안으로 또는 밖으로 상쇄할 정도

fluid_detail_size

연기 또는 이 소스에서 방출되는 기타 특성에서 최소 분해능으로 나타낼 수 있는 디테일 사이즈. 임의의 fluid_detail_size 를 갖는 소스가 다수 존재하는 경우, 특정 소스의 fluid_detail_size 는 정확하지 않을 수도 있습니다. 좀 더 구체적으로 말해서, 시뮬레이션에서 최고 해상도 레벨은 모든 입력 소스의 fluid_detail_size에 의해 결정됩니다. 나머지 소스의 경우, 소스의 해상도 레벨은 적응형 그리드 구조에서 가장 가까운 이산 레벨로 매핑됩니다. 즉, fluid_detail_size 의 차이가 작을 경우에는 가시적인 변화가 없을 수도 있습니다.

Air Properties

fog_density

fog_density_mode로 지정한 모드를 이용해 voxel_fog_density 특성으로 방출되는 연기의 시각 밀도입니다.

fog_density_mode

fog_density가 복셀로 방출되는 방식

- set: 소스를 특성으로 설정합니다(입력에서 에지가 선명하다면 출력에서도 마찬가지임).
- rate: 소스를 시간 종속 속도로 한 특성에 혼합 및 추가합니다.
- add: 소스를 특성에 혼합 및 추가합니다.
- subtract: 소스를 특성에 혼합하고 뺍니다.
- over: 소스를 특성에 혼합합니다.
- min: 소스와 특성의 최소값을 기준으로 혼합합니다.
- max: 소스와 특성의 최대값을 기준으로 혼합합니다.
- multiply: 소스와 특성의 결과물을 기준으로 혼합합니다.

temperature

temperature_mode에서 지정한 모드를 이용해 voxel_fog_temperature 특성으로 방출되는 연기의 온도(섭씨). 연소 작업을 수행할 때, 온도가 물리적으로 비현실적인 수준으로 매우 급격히 상승하지 않는 한 set temperature_mode를 이용할 것을 권장합니다.

temperature_mode

fog_density가 복셀로 방출되는 방식. 옵션은 fog_density_mode와 동일합니다.

initial_speed

방출된 특성의 초기 속도(속도 크기)

initial_speed_direction

방출된 특성의 초기 속도-방향

speed_mode

initial_speed와 initial_speed_direction이 복셀로 방출되는 방식. 옵션은 fog_density_mode와 동일합니다.

inherit_velocity

소스에 의해 방출되는 특성의 속도에 더해지는 소스 지오메트리 속도의 비.

trail_smoothness

소스 속도가 클 때 방출되는 연기의 평활도. 방출되는 연기 트레일에 노이즈나 분명한 홀(hole)이 너무 많은 경우, trail_smoothness 를 1씩 높여 그러한 인공물을 줄일 수 있습니다.

source_air 안에 있는 iterate 컴파운드 내에 중첩된 두 추가 컨트롤을 이용하면 속도가 크거나 inherit_velocity 값이 매우 큰 양의 값 또는 음의 값일 때에도 방출을 평활화할 수 있습니다.

- source_time_shift는 지정된 타임 스텝의 비에 따라 제 시간에 맞춰 방출을 앞으로 또는 뒤로 옮깁니다.
 inherit_velocity 값이 매우 큰 양의 값 또는 음의 값일 때 이것을 이용하면 빠르게 이동하는 이미터의 앞에 또는 뒤에 방출이 나타나는 것을 방지할 수 있습니다. 이 값은 지속적으로 바뀔 수 있습니다. 값이 0이면 현재의 타임 스텝에서 이동하는 범위 전역에 방출하고, -1이면 직전 타임 스텝에서 이동한 범위 전역에 방출하고, 1이면 직전 타임 스텝에서 이동한 범위 전역에 방출합니다.
- source_time_stretch는 소스가 이동하거나 소스로 방출되는 공간의 영역을 확장하기 위한 인수입니다.
 inherit_velocity가 매우 큰 음의 값일 때 연기에 간극이 있거나 이미터에 뒤쳐지는 경우, 이 값을 2 내지 3으로 설정해봅니다.

Output

air_source

출력은 소스 관련 입력의 총합을 공기 솔버에 포함시키는 객체이며 이를 simulate_aero 노드의 sources 포트나 source_fuel 노드의 air_source 포트에 연결해야 합니다.

source_fuel

시뮬레이션 그리드로 특성을 방출하는 연료 소스의 매개변수들을 지정합니다.

Inputs

air_source

source_air 또는 또 다른 source_fuel 노드의 출력을 연결합니다. 특히, 몇몇 source_fuel 노드의 경우 데이지 체인 방식으로 연결해 좀 더 고급화된 연료와 산화제의 혼합물을 만들 수 있습니다. 예를 들어, 실제 VFX용으로 흔히 사용되는 MAPP 가스는 프로핀, 프로파디엔, 프로판으로 구성되며, 경우에 따라 프로필렌과 부탄까지 함유하기도 합니다.

General

enable_fuel_source

이 연료 소스를 활성화합니다.

start_frame

연료 소스의 방출이 시작되는 프레임

use_end_frame

특정 프레임에서 방출을 중단합니다. 그렇지 않으면, 시뮬레이션 내내 방출이 계속됩니다.

end_frame

소스의 방출이 중단되는 프레임

Fuel Properties

fuel_type

드롭다운 메뉴에서 선택할 수 있는 연료의 유형. 각 source_fuel 노드마다 연료를 한 가지 유형만 지정할 수 있지만, source_fuel 노드 몇 개를 데이지 체인 방식으로 연결할 수도 있습니다. 이로써, 실제 풋티지(footage)를 연료 유형에 대응시킬 수 있습니다. 그렇지 않으면, 이를 기본값으로 놔두고 다른 매개변수들을 조절해 원하는 미적 결과를 얻을 수도 있습니다.

ignition_temperature

연료가 점화되는 온도(섭씨). 발화점 이하에서는 연소가 일어나지 않습니다. 각 연료 유형마다 실제 발화점이 있지만 여기서는 예술적 효과를 위해 재정의할 수 있습니다.

팁:

폭발 강도가 클수록 발화점을 낮춰야 합니다.

oxygen_percentage

산화제 내 산소 함유율. 예를 들어, oxygen_percentage가 0.4로 설정되면, 이 산화제는 산소 40%와 질소 60%로 이루어진다는 것을 의미합니다. 보통, 연료가 충분한 경우에 oxygen_percentage를 높이면 폭발이 더 격렬해져 불꽃 전파 속도가 빨라지고 온도가 상승합니다.

기본적으로, 주변 환경에는 공기(거의 대부분이 질소와 산소)가 있어 연료로 확산되는 산소가 충분하다면 oxygen_percentage가 0이더라도 연료가 연소됩니다.

burn_rate

연료와 산화제 혼합물의 화학량을 정의하는 0~2 범위의 값. 값이 0이면 연기를 동반한 불꽃이 발생하는데 더 구체적으로 말하면 연료만 방출됩니다. 1 미만인 값은 연료 풍부 조건에 해당해 연소 프로세스 후에도 연료 과잉 상태가 됩니다. 값을 0에서 1로 높이면 연기를 동반한 불꽃에서 좀 더 깨끗하지만 격렬한 연소로 시뮬레이션이 점차적으로 바뀝니다. 값이 1이라는 것은 화학량 조건이 완벽하다는 의미로, 산소나 연료의 과잉 없이 모든 가용 산호와 연료가 반응해 부산물이 생성됩니다. 1보다 큰 값은 연료 희박 조건에 해당해 연소 후 산소 과잉 상태가 됩니다.

Output

fuel_source

출력은 연료 소스 관련 입력의 총합을 공기 솔버에 포함시키는 객체이며 이를 simulate_aero 노드의 sources 포트나 source_fuel 노드의 air_source 포트에 연결해야 합니다.

split_points 노드

이 노드는 포인트들을 "분할해" 불분명성을 완화하는데, 이런 불분명성은 공간 분포가 고르지 않아 생길 수 있습니다. 관측 결과, 포인트들의 밀도 분포를 고르게 함으로써 불분명성을 없앨 수 있습니다. 따라서, 주어진 복셀에서 포인트 평균을 지나치게 벗어나는 포인트를 감지할 때마다 새로운 포인트들을 샘플링할 수 있습니다.

Inputs

aero_body [in] 파티클을 효율적으로 방출하는 데 이용하는 타일 트리를 포함하는 입력 공기 바디 (Amino::Object).

time [in] 현재 시각

time_step [in] 이 타임 스텝의 크기(델타 타임)

point_density [in] 단위 부피당 포인트 개수

density_significance_threshold [in] 복셀 밀도가 이 임계값을 초과하면 포인트가 복셀로 방출됩니다.

violation_factor [in] 한 포인트가 기여할 수 있는 평균에서 벗어난 최대 인수

coarsest_depth [in] 타일 트리에서 분할이 발생해야 하는 최저 해상도의 깊이

enable [in] true로 설정하면 분할이 일어나고 그렇지 않으면 이 노드는 패스쓰루로 작동합니다.

points [inout] 방출되는 포인트를 포함하는 입력/출력 Amino::Object.

stash_acceleration 노드

이 노드는 속도 차를 기준으로 가속도를 계산해 그 결과를 저장합니다.

Inputs

time_step [in] 타임 스텝

improved_time_integration [in]

'improved time integration'이 활성화되어 있다면 True입니다.

body [inout]

입력/출력 바디 (Amino::Object).

stash_velocity 노드

이 노드는 속도 필드를 업데이트합니다.

Inputs

improved_time_integration [in]

'improved time integration'이 활성화되어 있다면 True입니다.

body [inout]

입력/출력 바디 (Amino::Object).

to_aero_transport_style 노드

이 노드는 TransportStyle 유형 값을 TransportInterpolationType으로 변환합니다.

Inputs

fromValue [in] 변환될 값

Outputs

converted [out] 변환된 값

to_areo_transport_interpolation_type 노드

이 노드는 TransportInterpolationType 유형 값을 TransportStyle로 변환합니다.

Inputs

fromValue [in] 변환될 값

Outputs

converted [out] 변환된 값

to_int 노드

이 노드는 SourcePropertyMode 유형 값을 int로 변환합니다.

Inputs

from [in]

변환될 값

Outputs

int [out]

변환된 값

transport 노드

이 노드는 바디 복셀 토폴로지의 속도 필드에 있는 입력 바디의 채널들을 이동시킵니다.

Inputs

channelNames [in]

옮겨지는 채널들의 이름

collision [in]

'collision'은 복셀화된 콜라이더를 포함하는 Amino::Object 또는 BifrostExp::Objects 목록입니다. 콜리즌(collision)이 서로 다른 시점의 객체들을 포함하고 있다면, 목록은 사전식 순서(시점, 객체)를 따르므로 서로 다른 시점에 있는 동일 객체를 목록의 인덱스와 주어진 객체의 타임 스탬프를 보고 관련성을 파악할 수 있습니다.

time [in]

현재 시각

timeStep [in] 이 타임 스텝의 크기(델타 타임).

frameStartTime [in] 프레임 시작 시간(서브프레임 보간에 유용)

frameEndTime [in]

프레임 종료 시간(서브프레임 보간에 유용)

cflTransport [in]

이동의 CFL 값. 이것은 한 속성이 각 스텝마다 복셀의 최대 'cflTransport' 값만큼 이동할 수 있음을 의미합니다.

minTransportSteps [in] 이동이 거치는 최소 스텝 수

maxTransportSteps [in] 이동이 거치는 최대 스텝 수

transportInterpolationMode [in]

이동하는 모든 필드를 보간하는 데 이용하는 보간 메소드

transportVelocityInterpolationMode [in]

이동 중 속도 필드를 보간하는 데 이용하는 보간 메소드

transportTraceMode [in]

트레이스를 분석하는 데 사용하는 ODE 메소드

coarsestDOFDepth [in]

타일 트리에서 이동이 발생해야 하는 최저 해상도의 깊이

enableTransportInsideColliders [in]

활성화되면 콜라이더 내에서 연기 밀도와 다른 채널들이 이동하고, 그렇지 않으면 밀도가 방출되거나 콜라이더로 유출되지 않습니다.

body [inout]

공기 바디 (Amino::Object)

geometries

이 섹션의 페이지들

- analyze_volume 노드
- Arnold_ray_bitmask
- attract_repulse_influence
- buoyancy_influence
- change_range
- compute_on_frame
- construct_points
- contour_dual_marching_cubes node
- convert_to_volume 노드
- create_empty_volume 노드
- create_mesh_cube
- create_mesh_plane
- create_mesh_sphere
- decrement
- dissipation_influence
- drag_influence
- find_all_in_array
- find_orthogonal_vectors
- get_volume_tile_info 노드
- get_voxel_count 노드
- get_voxel_size 노드
- gravity_influence
- half_of
- hash_object 上드
- hash_points 노드
- hash_volume 노드
- increment
- kill_influence
- level_set_to_fog_density 上드
- mask_influence
- mesh_to_level_set 노드
- mesh_to_volume 노드
- modifier_influence
- modulate_influence
- offset_level_set 노드

BIFROST

- parse_string 노드
- point_scope
- points_to_level_set 노드
- points_to_volume 노드
- project_vector
- radial_influence
- remap_property 上드
- resample_volume 노드
- resize_level_set 노드
- rotation_between_vectors
- set_Arnold_geo_settings
- set_Arnold_mesh_settings
- set_Arnold_points_settings
- set_Arnold_strands_settings
- set_Arnold_volume_settings
- set_voxel_size 노드
- sharpen_voxel_property 노드
- smooth_mesh 노드
- smooth_voxel_property 노드
- smooth_voxel_temperature 노드
- splat_points_into_volume 노드
- string_to_array
- turbulence_influence
- twice_of
- update_mesh_normal
- update_voxel_position 노드
- vdb_influence
- volume_bounds 노드
- volume_influence
- volume_scope
- volume_to_mesh 노드
- vorticity_influence
- wind_influence

상위 페이지: 참조

analyze_volume 노드

이 노드는 입력 볼륨에 대한 다양한 특성을 분석해서 새로운 결과 특성이 추가된 볼륨을 출력합니다.

Inputs

volume [inout] 추가된 결과 채널들을 포함하는 볼륨 (Amino::Object)

properties [in] 분석할 특성들의 이름

mode [in] 입력 볼륨에서 수행할 분석의 유형

mask_property [in] 마스크 특성의 이름. 이 거리 특성이 음의 값이면 오퍼레이터만 적용됩니다.

output_mode [in] 결과를 출력할 메소드

Arnold_ray_bitmask

Arnold 전용 지오메트리 설정을 설정하는 set_Arnold_* 노드와 함께 사용됩니다. Arnold에서는 일부 가시성 설정이 각 레이(ray) 유형마다 설정되는데, 이를 테면, autobump, sidedness, visibility 등이 있습니다. 이런 설정은 비트마스크로 인코딩됩니다.

이 노드는 마스크와 같은 구성을 허용합니다. 출력은 set_Arnold_*_settings 노드와 함께 사용되어 각 레이(ray) 유형별로 해당하는 설정들을 설정할 수도 있습니다.

이런 설정들은 "by_ray_type"으로 표시되는데, 이를 테면, set_Arnold_mesh_settings 매개변수, autobump_visibility_by_ray_type 가 있습니다.

set_Arnold_geo_settings에서, "가시성(visibility)"은 공통적으로 사용되며 비트마스크의 구성이 해당 노드에 적용됩니다.

attract_repulse_influence

attract_repulse influence는 입력 지오메트리 쪽 또는 반대편으로 힘을 적용합니다. 지오메트리를 복셀화하고 거리 필드의 변화를 이용해 미는 힘의 방향을 제시합니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

detail_size

거리 필드(또는 레벨 설정)를 만들어 작용력에 활용하려고 할 때 입력 지오메트리를 얼마나 정밀하게 복셀화할 것인지를 결정합니다. 값이 작을수록 입력 지오메트리를 더 상세하게 캡처하지만 속도가 떨어지고 메모리 사용률이 증가하는 단점이 있습니다. 이 사이즈는 입력 지오메트리의 바운딩 박스에 비례하므로, 값이 0.1이면 복셀을 바운딩 박스의 1/10 크기로 만듭니다.

attraction

적용되는 미는 힘의 크기를 제어합니다. 양의 값이면 힘이 지오메트리 쪽으로 향하고 음의 값이면 반대편으로 작용합니다.

min_distance

메쉬까지의 거리가 min_distance 미만이면 최대 크기의 힘이 적용됩니다.

max_distance

지오메트리까지의 거리가 max_distance를 초과하면 아무런 효과가 나타나지 않습니다.

dropoff

거리가 min_distance와 max_distance의 사이에 있을 때 힘이 얼마나 빠르게 약해지는지를 결정합니다. 값이 0.0이면 max_distance 이내의 모든 거리가 최대 효과를 내며 이 거리를 초과하면 아무 효과가 없습니다. 값이 1.0이면 최대 효과를 내는 min_distance에서 아무 효과가 없는 max_distance 까지 직선으로 떨어질 것입니다. 값이 2.0이면 2제곱만큼 약해집니다. 값이 더 커지면 힘이 거리에 따라 더 빠르게 약해집니다.

drag

Influence 속도 쪽으로 끌어가는 힘을 적용합니다. Influence 필드에서 끌어가는 힘은 모든 작용력 효과들의 속도의 합을 바람으로 취급하고 그 움직임 쪽으로 끌어갑니다(단순히 0의 속도 쪽으로 끌어가는 것이 아님).

geometry

이것은 힘이 향하거나 멀어지는 지오메트리입니다. 현재, 복셀화는 항상 채우기 유형의 복셀화이므로 현재로서는 폐쇄형 객체를 이용해야 합니다(예: 평면이 아닌 박스나 구).

Output

out_influence 시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

buoyancy_influence

부력 효과는 부력 방향을 따라 온도에 비례해 가속도를 적용합니다. 이것은 현재 simulate_aero 노드에서만 작동하고 simulate_particles에서는 지원되지 않습니다.

이 노드를 이용하려면, 이 노드의 출력을 simulate_aero의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

buoyancy

부력 벡터. 이것은 상대 온도에 따라 적용되는 미는 힘의 방향과 크기를 제시합니다. 기본값은 위(up)이고 뜨거운 가스가 중력의 반대 방향으로 상승하는 방식을 시뮬레이션 합니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체.

change_range

1개 이상의 values를 한 범위에서 또 다른 범위로 재매핑합니다.

입력 value 가 from_start와 from_end 사이에서 갖는 비율과 동일한 to_start와 to_end 사이의 간격 상의 비율을 갖는 result를 반환합니다.

Clamp 옵션은 범위를 벗어나는 값들을 제한합니다.

예:

value: 0.25
from_start: 0
from_end: 1
to_start: 100
to_end: 200
result: 125

부동 유형, 벡터 유형, 정수 유형과 함께 작동합니다(물론, 부동 소수점 유형으로 변환됨).

compute_on_frame

특정 프레임에서만 값을 산출하고 다른 곳에서는 하지 않는 실험용 노드.

프레임 컴퓨팅(Compute on Frame)은 주어진 프레임에서 재설정되는 시뮬레이션 워크플로우와 함께 작동하도록 되어 있으며, 파일 로드나 처리 등과 같은 그래프 내 일부 작업이 해당 프레임 번호에서만 수행되도록 설정할 수 있습니다.

value_to_pull은 현재 프레임이 frame_to_pull_on과 일치할 때만 컴퓨팅됩니다. cached_value는 지정된 프레임을 포함해 모든 프레임에서 저장되어 있는 값을 출력합니다.

construct_points

points 지오메트리를 구성하고 그래프에서 원본 데이터를 가지고 points를 생성합니다. 최소 객체를 생성하며 point_size, point_orientation, 기타 선택적 특성을 설정하지는 않습니다.

Points는 매우 단순한 지오메트리 유형으로, 스키마에서는 point_component와 point_position만 필요합니다.

point_position 포트는 point_position 특성을 직접 설정하는데, 어레이 내 각 요소는 한 포인트에 해당합니다.

contour_dual_marching_cubes node

이 노드는 Schaefer와 Warren의 2004 알고리즘 "Dual Marching Cubes: Primal Contouring of Dual Grids" 즉, 듀얼 그리드에 있는 Marching Cubes를 이용해 설정된 레벨의 윤곽을 형성합니다.

Inputs

volume [in] 윤곽을 형성하기 위해 설정된 레벨을 포함하는 볼륨

iso_value [in] 윤곽을 형성하기 위한 부호 붙임 거리 함수의 iso 값

detail_size [in] 글로벌 디테일 사이즈

detail_size_property [in] 가변 공간 디테일 사이즈 특성의 이름

Outputs

mesh [out] 윤곽 형성으로 생성되는 메쉬

convert_to_volume <u>누드</u>

이 컴파운드는 입력 지오메트리를 레벨 세트(부호 붙임 거리 필드) 및/또는 뿌염 농도로 복셀화합니다. 또, 입력 지오메트리에 저장된 특성도 복셀화할 수 있습니다.

Inputs

geometry

복셀화하려는 포인트 또는 메쉬 지오메트리

detail_size

출력 볼륨에서 복셀의 월드 스페이스 사이즈를 지정합니다.

enable_bandwidth

활성화하면, 입력 지오메트리 주변의 bandwidth 복셀-스페이스 유닛 대역에서 고해상도 복셀을 보장합니다. 이것은 예를 들어, 레벨 세트 서피스에 큰 변위를 적용할 계획이어서 더 낮은 해상도의 복셀로 대체되는 서페이스 때문에 인공물을 경험하고 있는 경우에 유용합니다.

bandwidth

입력 지오메트리 주변에서 보장할 수 있는 월드 스페이스 대역폭

volume_mode

볼륨 모드는 입력 지오메트리가 볼륨으로 변환되는 방식을 결정합니다. 단, 이 모드는 입력 지오메트리가 메쉬인 경우에만 지원됩니다.

- Solid 모드에서는 입력 메쉬가 안이 꽉 차게 복셀화됩니다.
- Shell 모드에서는 입력 메쉬가 안이 텅 비어 있으며 두꺼운 서페이스로 복셀화됩니다.

offset

출력 레벨 세트에 적용되는 월드 스페이스 오프셋

store level set

true인 경우, 입력 지오메트리에서 레벨 세트 볼륨을 계산해 그 결과를 저장합니다.

store_fog_density

true인 경우, 입력 지오메트리에서 뿌염 농도 볼륨을 계산해 그 결과를 저장합니다.

fog_density

뿌염 농도 볼륨에 적용되는 곱수

properties

복셀화하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

resolution_mode

해상도 모드는 detail_size의 단위를 결정합니다.

- Absolute 모드인 경우, detail_size가 월드 스페이스 단위로 지정됩니다.
- Relative 모드인 경우, detail_size가 입력 지오메트리의 바운딩 박스에 비례하도록 지정됩니다.

Outputs

volume 출력 Bifrost 볼륨

create_empty_volume 노드

빈 볼륨을 만듭니다.

Inputs

detail_size [in] 월드 스페이스(float)에 있는 볼륨 Object의 복셀 크기

Outputs

volume [out] 빈 볼륨 (Amino::Object)
create_mesh_cube

노멀을 이용해 메쉬 큐브를 생성합니다.

Inputs

length, width, height 큐브의 치수.

position 메쉬가 생성되는 위치.

up_axis 메쉬의 방향을 결정할 벡터.

base_to_pivot 활성화되면, 메쉬가 position의 중심이 아니라 그 위에 생성됩니다.

create_mesh_plane

노멀을 이용해 메쉬 플레인(plane)을 생성합니다.

Inputs

length, width

구의 반경.

length_segments, width_segments 메쉬 내 서브디비전의 개수.

position 메쉬가 생성되는 위치.

up_axis 메쉬의 방향을 결정할 벡터.

create_mesh_sphere

노멀을 이용해 메쉬 스피어(sphere)를 생성합니다.

Inputs

radius

구의 반경.

axis_segments, height_segments

메쉬에서 축(세로 디비전)을 따라 최대 높이(위도 디비전)까지 있는 서브디비전의 개수

position 메쉬가 생성되는 위치.

up_axis 메쉬의 방향을 결정할 벡터.

base_to_pivot 활성화되면, 메쉬가 position의 중심이 아니라 그 위에 생성됩니다.

decrement

value에서 1(또는 다른 정수)을 뺍니다. by를 이용해 뺄 값을 지정할 수 있습니다.

부호 없는 정수 유형에서의 빼기에는 특별한 처리 방법이 없습니다. 그렇지 않으면, 이것은 모든 정수 유형, 부동 소수점 유형, 그리고 일반적이지는 않지만 벡터까지도 줄일 수 있습니다.

dissipation_influence

소실(dissipation) 효과는 지정된 감쇠율에 따라 정의된 특성을 배경 값 쪽으로 옮깁니다.

이 노드를 이용하려면, 이 노드의 출력을 simulate_aero, simulate_particles, simulate_mpm 등과 같은 simulate 노드의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

rate

이것은 특성이 배경 값을 향해 감쇠하는 속도를 좌우합니다. 이 값이 0일 때에는 아무런 효과가 없습니다. 이 속도는 기하 감쇠율이므로 이 속도가 무한대인 경우에만 배경 값이 정확하게 도달합니다.

property

이것은 시뮬레이션 객체에서 수정하려는 지오메트리 특성입니다.

background_value

이것은 감쇠 효과가 향하는 목표 값입니다. voxel_density와 같은 공기 특성일 경우, 보통 0.0이 적절한 값이므로 시간이 갈수록 밀도가 점차 약해집니다. voxel_temperature의 경우에는 주변 배경 온도 값을 이용할 수도 있는데 이 값은 20.0입니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

drag_influence

이 drag_influence는 공기 저항 효과를 시뮬레이션 하는 데 이용할 수 있습니다. 기본적으로, drag_influence는 시뮬레이션 속도를 저하시킵니다. 여기에 다른 influence 노드가 연결되어 있어 가속을 더한다면, 이것은 움직임을 0의 속도가 아닌 그 가속도 쪽으로 옮기게 됩니다. 그러면, 물과 같은 매체가 흐르는 상황에서 큰 저항력을 시뮬레이션할 수 있습니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

drag

이것은 속도가 influence 속도 쪽으로 끌려가는 속도인데, 다른 influence 노드가 연결되어 있지 않다면 influence 속도는 0이 됩니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

file_cache

시뮬레이션을 파일로 작성함으로써 시뮬레이션을 캐싱하고 읽을 수 있는 실험용 노드.

이 노드는 그래프에서 시뮬레이션 업스트림(우측방향)에 위치하도록 되어 있습니다. 이것은 시뮬레이션을 디스크로 캐싱하거나 디스크에서 시뮬레이션의 캐시를 읽을 수 있습니다. Lazy mode는 파일이 디스크에 존재하는지 여부에 따라 "읽기"에서 "쓰기"로 동적 전환할 수 있습니다.

이것은 .bob(Bifrost Object) 파일, Alembic, OpenVDB 파일을 작성할 수 있습니다. 파일 유형은 filename 매개변수의 확장자로 제어하는데, 확장자가 ".abc"인 파일은 Alembic으로, ".vdb"인 파일은 OpenVDB로, ".bob"(또는 기타 모든 확장자)인 파일은 Bifrost 객체 파일로 각각 작성됩니다.

사용 방법

사용 방법:

- 1. filename에 시뮬레이션 캐시의 파일 경로를 설정합니다. 홈 디렉터리에 해당하는 \$HOME과 같은 환경 변수를 포함할 수도 있습니다.
- 2. 시뮬레이션의 출력을 objects 로 연결하고 out_objects 를 출력으로 이용합니다.
- 3. 시뮬레이션의 시작 프레임으로 이동합니다.
- 4. 모드를 lazy mode로 설정합니다.

스크러빙과 재시작이 가능한 시뮬레이션

모드가 lazy mode로 설정되어 있고 업스트림 시뮬레이션이 File cache까지인 경우, 이 노드를 스크러빙과 재시작이 가능한 시뮬레이션으로 이용할 수 있습니다.

시뮬레이션의 일부를 재생하고 일시중지합니다. 시작 프레임으로 되돌아갑니다. 그러면, 시뮬레이션이 디스크에서 읽히고 업스트림 시뮬레이션은 평가되지 않을 것입니다. 다시 재생하면, 현재 시간이 캐싱되지 않은 첫 프레임에 도달할 때까지 이동하며, 이 지점에서 시뮬레이션이 재시작되므로 시뮬레이션을 중단하지 않고도 이미 캐싱된 프레임을 스크러빙할 수 있습니다.

단, 시뮬레이션은 그래프에서 원하는 매개변수를 변경하여 손쉽게 중단시킬 수 있는데, 이 경우, 컴파일이 다시 시작되고 피드백 포트 상태가 사라지며 모든 시뮬레이션이 재설정됩니다. 이 경우 시뮬레이션을 더 이상 재시작할 수 없습니다.

Mode

- Lazy mode는 디스크에 프레임이 존재하는지 여부에 따라 읽기와 쓰기 사이에서 상호 전환됩니다. 이 모드는 스크러빙이 불가하고 재시작이 가능한 캐시로 이용할 수 있습니다.
- Write mode는 업스트림 프레임을 끌어오고 프레임을 작성하며 지오메트리를 통과시킵니다.
- Read mode는 업스트림 프레임을 끌어오지도 또 프레임을 읽지도 않으며, 지오메트리를 출력합니다.
- Passthrough mode 는 항상 업스트림 프레임을 끌어오고, 어느 것도 작성하지 않으며, 지오메트리를 통과시킵니다.

Alembic 및 OpenVDB 설정

이것에 대한 자세한 내용은 write_Alembic 및 write_OpenVDB에 대한 설명서를 참고하십시오. 이런 노드들은 내부적으로 사용됩니다. File cache는 이런 노드들과 동일한 기능 및 제한사항을 공유합니다.

프레임 범위

active_frame_range 가 활성화되면, 노드가 기본적으로 지정된 프레임 범위 밖에서 passthrough 반응을 보입니다. inverted 가 활성화되면, 지정된 프레임 범위 내에서의 passthrough 반응으로 전환됩니다.

find_all_in_array

array에서 value와 동일한 모든 indices를 반환합니다. equal과 견줄 수 있는 모든 유형의 어레이에서 작동합니다.

이것은 한 어레이에서 특정 값과 정확히 일치하는 값들을 모두 찾아내는 데 이용할 수 있는데, 예를 들면 다음과 같습니다.

Array: [0, 1, 2, 0, 1, 2] Value: 1 Invert: False Indices: [1, 4]

Array: [0, 1, 2, 0, 1, 2] Value: 1 Invert: True Indices: [0, 2, 3, 5]

또, 이것은 Boolean 어레이에서 True 값들을 찾는 데에도 이용할 수 있습니다. 이를 통해 쿼리, 예를 들면, "3 미만"과 같은 쿼리에서 True를 반환하는 모든 값들을 간접적으로 찾을 수 있습니다.

```
Original array: [0, 1, 2, 3, 4, 5]
Apply less than 3 to the array: [True, True, True, False, False, False]
Value: True
Invert: False
Indices: [0, 1, 2]
```

Inputs

array

비교할 값들의 어레이.

value

어레이에서 찾는 값. 이것은 equal 비교이므로 정확하게 동일한 값들만 포함됩니다.

invert

포트 유형: bool.

True인 경우, 이 어레이에서 value와 같지 않은 모든 값들을 찾습니다.

Outputs

Indices (OUt)

포트 유형: array<long>.

쿼리와 일치하는 인덱스들의 어레이.

find_orthogonal_vectors

입력 vector와 직각이고 상호 간에 직각인 임의의 벡터 한 쌍을 반환합니다.

normalize_outputs를 활성화해 반환된 벡터의 길이가 1.0이 되도록 합니다.

get_volume_tile_info 노드

이 노드는 입력 볼륨으로부터 복셀 타일 정보를 가져옵니다.

Inputs

volume [in] 입력 볼륨 (Amino::Object)

Outputs

voxel_level [out] 타일의 깊이 레벨

voxel_count [out] 이 타일 안에서 가장 높은 해상도 레벨의 복셀 카운트

voxel_min_bound [out] 타일의 최소 월드 스페이스 바운드

voxel_max_bound [out] 타일의 최대 월드 스페이스 바운드

max_level [out] 타일 트리의 최대 깊이 레벨. 0은 최저 해상도 레벨입니다.

traits [out] 볼륨 내 타일 트리의 traits 구성

get_voxel_count 노드

이 노드는 볼륨으로부터 복셀 카운트를 가져옵니다.

Inputs

volume [in] 입력 볼륨 (Amino::Object)

Outputs

voxel_count [out] 복셀 카운트

get_voxel_size 노드

이 노드는 볼륨으로부터 복셀 사이즈를 가져옵니다.

Inputs

volume [in] 입력 볼륨 (Amino::Object)

Outputs

voxel_size [out] 복셀 사이즈

gravity_influence

중력 효과는 한 방향을 따라 단순 가속을 적용합니다.

이 노드를 이용하려면, 이 노드의 출력을 simulate_aero, simulate_particles, simulate_mpm 등과 같은 simulate 노드의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 영향력의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

magnitude

이것은 Meters/Second/Second 단위로 발생하는 가속의 강도입니다. 기본값 9.8은 지구의 해발 중력 값입니다.

direction

이것은 필드 방향에 대한 벡터입니다. 기본값 0, -1, 0은 아래 방향입니다. 자체적으로 정규화 되므로 크기가 아닌 방향에만 영향을 줍니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

half_of value를 2로 나눠 half_value를 반환합니다. 이것은 모든 정수 유형, 부동 소수점 유형, 그리고 일반적이지는 않지만 벡터까지도 될 수 있습니다. 정수 유형은 정수 나눗셈이 적용됩니다. 7 / 2 = 3

7 / 2 = 3-7 / 2 = -3

hash_object 노드

이 노드는Bifrost 객체를 해시(hash)합니다.

Inputs

object 해시(hash)할 객체

properties 해시(hash)하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

hash 특성들의 결합 해시(hash)

hash_points 노드

이 노드는 포인트 지오메트리(포인트, 메쉬, 스플라인)를 해시(hash)합니다.

Inputs

geometry [in] 해시(hash)할 포인트 지오메트리

properties [in]

해시(hash)하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

hash [out]

포인트 특성들의 결합 해시(hash)

hash_volume 노드

이 노드는 볼륨을 해시(hash)합니다.

Inputs

volume [in] 해시(hash)할 볼륨

properties [in] 해시(hash)하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

hash [out] 볼륨 특성들의 결합 해시(hash)

increment

value에 1(또는 다른 정수)을 더합니다. 财를 이용해 더할 값을 지정할 수 있습니다. 모든 정수 유형, 부동 소수점 유형, 그리고 일반적이지는 않지만 벡터까지도 늘릴 수 있습니다.

kill_influence

제거(kill) 효과는 정해진 속도로 파티클이나 밀도를 제거합니다. 이를 mask_influence와 결합해 특정 영역 안에서만 제거할 수도 있고 modulate_influence와 결합해 시뮬레이션 특성에 따라 제거할 수도 있습니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

rate

이것은 파티클들이 임의로 사라지거나 공기 voxel_density가 약해지는 속도를 결정합니다.

absolute

이것을 활성화하면, 제거가 즉시 실행되므로 속도가 무한대인 경우와 같아지게 됩니다. 보통, absolute kill은 마스크 또는 변조 효과와 결합하여 이용합니다. 그렇지 않을 경우, 전체 시뮬레이션이 즉시 제거됩니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

level_set_to_fog_density 노드

이 노드는 입력 레벨 세트 볼륨을 뿌염 볼륨으로 변환합니다. 산출된 뿌염 볼륨이 기존의 레벨 세트 볼륨에 추가됩니다. 입력 볼륨에 뿌염 농도 볼륨이 이미 포함되어 있는 경우에는 이전의 뿌염 농도 볼륨이 대체됩니다.

Inputs

level_set [in] 레벨 세트 볼륨

radius [in]

서피스 주변에서 '뿌염 없음'에서 '완전 포화 뿌염'으로 증가하는 반경(복셀 단위)

Outputs

volume [out]

산출된 뿌염 볼륨이 추가된 입력 볼륨

mask_influence

마스크(mask) 효과를 이용하면 지오메트리까지의 거리를 통해 다른 influence 노드들의 효과를 조정할 수 있습니다. 이것은 influence 포트를 통해 다른 influence 노드들과 결합할 때에만 효과가 있습니다. 또, 일부 지오메트리를 geometry 포트에 연결해야 합니다.

이 노드를 이용하려면, 이 노드의 출력을 wind_influence와 같은 influence 노드의 influence 입력에 연결합니다. 또, 이용하려는 지오메트리를 geometry 포트에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, modulate_influence 노드를 연결해 시뮬레이션 특성에 따라 효과를 추가적으로 조정할 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

shape

이것은 지오메트리 거리에 사용하는 메소드를 정의합니다.

- Mesh: 입력 객체까지의 실제 거리를 이용합니다. 이 메소드는 지정된 detail_size를 이용해 입력 지오메트리를 볼륨 거리 필드로 변환하는데, detail_size 값이 작을 때에는 속도가 저하될 수도 있습니다. 복셀화는 채우기(fill) 메소드를 이용하므로, 구와 같은 폐쇄형 모양이 평면과 같은 개방형 메쉬보다 훨씬 더 효과적입니다.
- Bounding Box: 이것은 단순히 입력 지오메트리의 축 정렬 바운딩 박스로 마스킹합니다. 현재 이 메소드를 이용할 경우 최소/최대 거리와 dropoff는 효과가 없습니다.
- Bounding Sphere: 이것은 입력 지오메트리를 위한 바운딩 스피어(sphere)로 마스킹합니다. 현재 이 메소드를 이용할 경우 최소/최대 거리와 dropoff는 효과가 없습니다.

detail_size

이것은 지오메트리를 거리 필드로 변환할 때 이용하려는 바운딩 박스 상대 복셀 사이즈를 정의합니다. 이것은 shape가 Mesh일 때만 사용됩니다.

min_distance

그 안에서는 효과 레벨이 최대인 지오메트리까지의 거리. 이것은 <u>shape</u>가 <u>Mesh</u>일 때만 효과가 있습니다. 구와 같은 폐쇄형 메쉬가 평면과 같은 개방형 메쉬보다 훨씬 더 효과적입니다. 지오메트리 안에서는 거리가 음의 값이 되므로 음의 값인 <u>min_distance</u>를 이용해서 한 메쉬 내 최대 효과 지점을 좀 더 상쇄할 수 있습니다.

max_distance

그 이후부터는 아무 효과가 없는 지오메트리까지의 거리. 이것은 shape가 Mesh일 때만 효과가 있습니다.

dropoff

거리가 min_distance와 max_distance의 사이에 있을 때 효과가 얼마나 빠르게 약해지는지를 결정합니다. 값이 0.0이면 max_distance 이내의 모든 거리가 최대 효과를 내며 이 거리를 초과하면 아무런 효과가 나타나지

않습니다. 값이 1.0이면 최대 효과를 내는 min_distance에서 아무 효과가 없는 max_distance까지 직선으로 떨어집니다. 값이 2.0이면 2제곱만큼 약해집니다. 값이 더 커지면 힘이 거리에 따라 더 빠르게 약해집니다.

invert

이것을 활성화하면 마스크를 거꾸로 합니다. 따라서, 이것이 활성화되면 지오메트리 안에서는 아무 효과가 없고 밖에서는 최대 효과를 내게 됩니다.

geometry

이것은 마스크에 사용하는 지오메트리를 정의합니다. geometry에 연결되어 있지 않으면 이 노드는 아무런 효과가 없습니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

mesh_to_level_set 노드

이 노드는 입력 메쉬 지오메트리를 레벨 세트(부호 붙임 거리 필드)로 복셀화합니다. 또, 메쉬의 점, 선, 면에 저장된 특성도 복셀화합니다.

Inputs

mesh [in] 복셀화할 메쉬 지오메트리

detail_size [in]

볼륨에서 복셀의 (월드 스페이스) 사이즈를 지정합니다.

volume_mode [in]

솔리드 모드는 안이 꽉 차도록 메쉬를 복셀화합니다. 쉘(Shell) 모드는 메쉬를 안이 텅 비어 있고 두꺼운 서페이스로 복셀화합니다.

properties [in]

복셀화하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

level_set [out] 입력 메쉬에 해당하는 레벨 세트

mesh_to_volume 노드

이 컴파운드는 입력 지오메트리를 레벨 세트(부호 붙임 거리 필드) 및/또는 뿌염 농도로 복셀화합니다. 또, 입력 메쉬에 저장된 특성도 복셀화할 수 있습니다.

Inputs

geometry

복셀화할 메쉬 지오메트리

detail_size

출력 볼륨에서 복셀의 월드 스페이스 사이즈를 지정합니다.

enable_bandwidth

활성화하면, 입력 지오메트리 주변의 bandwidth 복셀 스페이스 유닛 대역에서 고해상도 복셀을 보장합니다. 이것은 예를 들어, 레벨 세트 서피스에 큰 변위를 적용할 계획이어서 더 낮은 해상도의 복셀로 대체되는 서피스 때문에 인공물을 경험하고 있는 경우에 유용합니다.

bandwidth

입력 지오메트리 주변에서 보장할 수 있는 월드 스페이스 대역폭

volume_mode

볼륨 모드는 입력 지오메트리가 볼륨으로 변환되는 방식을 결정합니다. 단, 이 모드는 입력 지오메트리가 메쉬일 경우에만 지원됩니다.

- Solid 모드에서는 입력 메쉬가 안이 꽉 차게 복셀화됩니다.
- Shell 모드에서는 입력 메쉬가 안이 텅 빈 두꺼운 서피스로 복셀화됩니다.

offset

출력 레벨 세트에 적용되는 월드 스페이스 오프셋

store_level_set

true인 경우, 입력 지오메트리에서 레벨 세트 볼륨을 계산해 그 결과를 저장합니다.

store_fog_density

true인 경우, 입력 지오메트리에서 뿌염 농도 볼륨을 계산해 그 결과를 저장합니다.

fog_density

뿌염 농도 볼륨에 적용되는 곱수

properties

복셀화하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

resolution_mode

해상도 모드는 detail_size의 단위를 결정합니다.

- Absolute 모드인 경우, detail_size 는 월드 스페이스 단위로 지정됩니다.
- Relative 모드인 경우, detail_size는 입력 메쉬의 바운딩 박스에 비례하도록 지정됩니다.

Outputs

volume 출력 Bifrost 볼륨

modifier influence

이 수정자 효과를 이용하면 voxel_density, point_velocity, point_size와 같은 시뮬레이션 특성의 값을 바꾸고, 더하고, 곱하고, 고정시킬 수 있습니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 simulate 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

property

수정하려는 시뮬레이션 특성의 이름

operation

지정 값을 지정된 시뮬레이션 특성에 반영하려는 수학 연산:

- Replace: 현재 값을 지정 value 로 바꿉니다.
- Multiply: 현재 값에 value 를 곱합니다.
- Add: 현재 값에 value를 더합니다.

value

이것은 바꾸거나 곱하거나 더하려는 값입니다. 보통, 특성과 유형이 같지만 float3 특성을 곱할 수 있습니다.

rate

이것은 연산이 얼마나 빠르게 적용될 것인지를 가리킵니다. 예를 들어, rate가 상당히 크지만 않다면 replace는 시뮬레이션을 즉시 value로 설정하지 않습니다.

enable_clamp

이것이 활성화되면, 정의된 특성의 값들이 지정된 min과 max 값 범위 내로 고정됩니다. clamp는 rate를 이용하지 않지만 rate가 0인 경우에도 값들을 즉시 고정시킵니다. 수학 연산 없이 고정시키기만 하려면 rate를 0으로 만들면 됩니다. 현재, clamp는 float과 float3 특성에만 효과가 있습니다. float3 특성의 경우, 최소/최대 값을 벡터 길이로 간주합니다. 예를 들면, 최대값이 0.2인 clamp를 이용해서 최대 point_velocity를 0.2로 제한할 수 있습니다. 또는, 최저 속도를 0이 아닌 값으로 설정해 시뮬레이션이 계속 실행되도록 할 수도 있습니다.

min

고정시키려는 최소값(enable_clamp를 활성화한 경우에만 효과가 있음).

max

고정시키려는 최대값(enable_clamp를 활성화한 경우에만 효과가 있음).

Output

out_influence 시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

modulate influence

조절(modulate) 효과를 이용하면 시뮬레이션 특성을 조작해 다른 influence 노드에 의해 적용되는 효과의 정도를 스케일링할 수 있습니다. 이 노드는 mask_influence와 비슷하나 지오메트리까지의 거리 대신 시뮬레이션 특성을 이용해 마스크 값을 생성합니다. 이것은 대체로 매우 유용합니다. 예를 들면, 공기 특성 voxel_fog_density를 이용해서 gravity_influence를 조절할 수 있습니다. 이것은 공기 부력과 비슷한 효과를 내지만 온도가 아닌 밀도를 이용합니다. 만일, 이 조절 효과 없이 중력을 적용한다면, 그 안에 밀도를 가진 것들뿐만 아니라 모든 복셀(빈 공기 포함)에 가속도를 적용하므로 뿌염이 공기에 비례하여 움직이지 않을 것입니다.

이 노드를 이용하려면, 이 노드의 출력을 wind_influence와 같은 influence 노드의 influence 입력에 연결합니다. 이 노드는 다른 influence 노드와 결합하지 않는 한 아무런 효과가 없습니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 nfluence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다. 또, 이것은 이 포트에 연결되는 다른 모든 influence도 조절합니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

property

influence에 이용하려는 시뮬레이션 특성의 이름. 현재 이 특성은 float 또는 float3입니다. point_velocity와 같은 float3 특성의 경우, 특성의 크기를 이용해 influence 효과를 스케일링할 수 있습니다. 예를 들면, 이것을 drag_influence와 결합해 point_velocity가 클 경우에만 파트클에 더 큰 효과를 적용할 수 있습니다.

min

이것은 시뮬레이션 특성의 최소값으로, 그 아래에 연결되어 있는 모든 influence 노드들은 아무런 효과도 내지 않습니다. 특성 값이 max 값이 될 때까지 효과가 단계적으로 커지며 최대값에서는 연결된 모든 influence가 최대 효과를 나타냅니다.

max

이것은 시뮬레이션 특성의 최대값으로, 그 위에 연결되어 있는 모든 influence 노드들은 최대 효과를 냅니다.

invert

이것은 이 influence의 효과를 반전시킵니다. 활성화되면, min 아래의 시뮬레이션 특성 값들은 최대 효과를 내고 max 위의 값들은 아무런 효과도 내지 않습니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

offset_level_set 노드

이 노드는 레벨 세트에 균등 오프셋을 적용합니다. 필요할 때 새로운 타일을 추가해 레벨 세트를 동일 해상도로 유지합니다.

Inputs

level_set [in] 레벨 세트 볼륨 (Amino::Object)

offset [in] 입력 레벨 세트에 적용되는 월드 스페이스 오프셋. (float)

runtime_services [in] 로깅에 이용되는 Amino 런타임 서비스

Outputs

out_level_set [out] 오프셋 레벨 세트 (Amino::Object)

parse_string 노드

이 노드는 입력 표현식에 따라 문자열을 분석합니다.

Inputs

expression [in]

수락하려는 문자열들의 목록으로 공간 범위가 정해져 있습니다.

string [in]

분석할 문자열

Outputs

success [out]

string이 expression과 일치하면 true입니다.

point_scope

포인트 범위는 포인트, 가닥(strands), 메쉬, 인스턴스 등 포인트 기반 지오메트리에 대한 시각적 진단 정보를 제공합니다.

이것은 수치로 된 드로잉 벡터와 드로잉 값을 포함해 포인트와 포인트 특성을 여러 방식으로 얻기 위해 이용할 수 있습니다.

사용 방법

이것을 사용하려면, 1개의 지오메트리 또는 지오메트리 어레이를 point_geometry 플러그로 연결하고 이 출력을 다른 출력에 연결합니다.

수치 표시

수치 표시는 한 특성의 콘텐츠를 수치로 보여주는 points 객체를 생성합니다.

벡터 표시

벡터 표시는 point_normal이나 point_velocity처럼 벡터 데이터를 시각적으로 보여주는 strands 객체를 생성합니다.

points_to_level_set 노드

이 노드는 포인트 지오메트리(포인트, 메쉬, 스플라인)를 레벨 세트(부호 붙임 거리 필드)로 복셀화합니다. 또, 포인트에 저장되어 있는 특성도 복셀화합니다.

Inputs

point [in] 복셀화할 포인트 지오메트리

detail_size [in]

볼륨에서 복셀의 (월드 스페이스) 사이즈를 지정합니다.

radius [in]

point_radius 채널이 없을 때 포인트에 할당되는 (월드 스페이스) 반경.

properties [in]

복셀화하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

level_set [out] 입력 포인트에 해당하는 레벨 세트

points_to_volume 노드

이 컴파운드는 입력 포인트 지오메트리를 레벨 세트(부호 붙임 거리 필드) 및/또는 뿌염 농도 볼륨으로 복셀화합니다. 또, 입력 포인트에 저장된 특성도 복셀화할 수 있습니다.

Inputs

geometry

복셀화할 포인트 지오메트리

detail_size

출력 볼륨에서 복셀의 월드 스페이스 사이즈를 지정합니다.

enable_bandwidth

활성화하면, 입력 지오메트리 주변의 bandwidth 복셀 스페이스 유닛 대역에서 고해상도 복셀을 보장합니다. 이것은 예를 들어, 레벨 세트 서피스에 큰 변위를 적용할 계획이어서 더 낮은 해상도의 복셀로 대체되는 서피스 때문에 인공물을 경험하고 있는 경우에 유용합니다.

bandwidth

입력 지오메트리 주변에서 보장할 수 있는 월드 스페이스 대역폭

offset

출력 레벨 세트에 적용되는 월드 스페이스 오프셋

store_level_set

true인 경우, 입력 포인트에서 레벨 세트 볼륨을 계산해 그 결과를 저장합니다.

store_fog_density

true인 경우, 입력 포인트에서 뿌염 농도 볼륨을 계산해 그 결과를 저장합니다.

fog_density

뿌염 농도 볼륨에 적용되는 곱수

properties

복셀화하려는 특성들의 목록으로 공간 범위가 정해져 있습니다.

resolution_mode

해상도 모드는 detail_size의 단위를 결정합니다.

- Absolute 모드의 경우, detail_size는 월드 스페이스 단위로 지정됩니다.
- Relative 모드의 경우, detail_size는 입력 포인트의 바운딩 박스에 비례하도록 지정됩니다.

Outputs

volume

출력 Bifrost 볼륨

project_vector

한 벡터를 또 다른 벡터로 투영해서 병렬 및 직각을 이루는 두 구성요소를 모두 반환합니다.

입력

first

투영할 벡터.

second

투영될 벡터.

출력

parallel_vector

first를 second로 투영. 이것은 second에 병렬하는 first의 구성요소입니다.

orthogonal_vector

first에서 parallel_vector를 뺀 나머지입니다. 이것은 second에 대한 first의 '거부'라고도 하며 second에 직교하는 first의 구성요소입니다. 3D 벡터일 경우, 이것은 second에 일반적인 플레인으로 first를 투영한 것과도 같습니다.

radial_influence

방사(radial) 효과는 지정된 포인트 쪽으로 또는 반대 쪽으로 가속도를 적용합니다. 또, 이것은 해당 포인트까지의 거리에 따른 효과 감소를 포함하며, 여기에 연결되어 있는 다른 influence 노드들에도 영향을 미칩니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

dropoff

이것은 해당 포인트까지의 거리를 기준으로 효과의 정도를 약화시킵니다. 이 값이 0.0이면 모든 곳에서 최대 효과를 냅니다. 값이 1이면 center까지의 거리에 따라 효과가 선형적으로 약해지며, 값이 클수록 효과가 더 빠르게 약해집니다. 단, 이 하락 효과는 1 내지 2 유닛 거리 이내에서만큼 강하지는 않습니다. 적용되는 공식은 (1.0/pow(거리+1.0,dropoff)입니다. 크기를 0으로 설정해 가속도가 적용되지 않도록 함으로써 이것을 다른 influence들에 대한 포인트 거리 마스크로 이용할 수도 있습니다.

center

이것은 방사 가속의 원점입니다. 효과 감소는 이 원점까지의 거리에 따라 달라지며 이 원점 쪽으로 또는 반대 쪽으로 가속이 적용됩니다.

magnitude

이것은 적용되는 가속의 강도입니다. 양의 값은 힘이 중심에서 멀어지게 하고 음의 값은 힘이 중심 쪽으로 향하게 합니다.

drag

이것은 현재의 시뮬레이션 속도를 떨어뜨려 시간이 지남에 따라 속도 증가 효과가 줄어듭니다. drag 값이 크면 지속적으로 가속을 더하는 대신 시뮬레이션 속도를 방사 가속으로 완전히 교체합니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체
remap_property 노드

이 노드는 이름이 'property'인 복셀 특성에서 [min,max] 범위 내 모든 값들을 새로운 [new_min, new_max] 범위로 다시 매핑합니다. 최소값과 최대값은 특성에서 확인할 수 있고 사용자가 지정할 수 없습니다.

Inputs

geometry [in] 특성을 포함하는 볼륨 (Amino::Object)

property [in] 특성의 이름

new_min [in] 특성 값들이 매핑될 범위의 새로운 최소값

new_max [in]

특성 값들이 매핑될 범위의 새로운 최대값

Outputs

out_geometry [out] 특성을 포함하는 출력 볼륨 (Amino::Object)

resample_volume 노드

이 노드는 입력 볼륨의 특성들을 이름이 동일한 resampled_volume 특성들로 다시 샘플링합니다.

Inputs

source_volume [in] 소스 특성을 포함하는 볼륨 (Amino::Object)

target_volume [in] 입력 볼륨을 샘플링할 목표 볼륨 (Amino::Object)

mode [in] 리샘플링 모드 (Geometry::Volume::ResampleMode)

properties [in] 다시 샘플링할 특성들의 이름

transform [in] 아핀(Affine) 변환 (float4x4)

voxel_offset [in] 복셀 오프셋만큼 이동 [무손실] (int3)

voxel_size [in] 복셀 사이즈를 지정된 복셀 사이즈(float)로 스케일링합니다.

voxel_scale [in] 복셀 사이즈를 이 스케일(float)만큼 스케일링합니다.

rebuild_level_set [in] 활성화되면, 어느 레벨 세트든 단위 기울기를 갖도록 재조정합니다. 이것이 필요한 이유는 다수의 변환 과정에서 이 특성이 파괴될 수 있기 때문입니다. (bool)

Outputs

resampled_volume [out] 다시 샘플링된 특성들을 포함하는 볼륨 (Amino::Object)

resize_level_set 노드

이 노드는 레벨 세트 주변의 "협대역" 크기를 조정해 레벨 세트 주변의 '대역폭' 복셀-스페이스 유닛 대역 안에서 고해상도 복셀을 보장합니다. 단, '대역폭'의 대역폭을 보장하는 데 필요한 것보다 더 많은 타일이 있을 경우, 초과분은 제거됩니다.

Inputs

level_set [in] 레벨 세트 볼륨 (Amino::Object)

bandwidth [in] 입력 레벨 세트 주변에서 보장할 수 있는 월드 스페이스 대역폭. (float)

runtime_services [in] 로깅에 이용되는 Amino 런타임 서비스

Outputs

out_level_set [out] 새로운 레벨 세트 (Amino::Object)

rotation_between_vectors

from_vector를 to_vector와 일치시키는 최단 호를 따르는 회전에 해당하는 4원수를 반환합니다.

set_Arnold_geo_settings

Arnold 전용 지오메트리 설정을 설정하며 이것은 Arnold에서 Arnold-네이티브 설정으로 적용됩니다. 이 설정은 모든 Arnold 지오메트리 유형이 공유합니다. 이 설정은 뷰포트에는 영향을 주지 않으며 Arnold 렌더링에만 영향을 줍니다.

이 설정은 내부 Arnold 기능에 매우 가깝게 매핑되므로 Arnold Node Reference를 본 문서와 연계해 이용할 수도 있습니다. 이때, 관련 문서는 Arnold 모양 유형, 특히, polymesh, curves, points, volume, implicit에 관한 것입니다. https://docs.arnoldrenderer.com/display/NodeRef/shapes

레이(Ray) 유형별 가시성

Arnold에서는 가시성을 레이(ray) 유형별로 제어합니다. 이러한 레이(ray) 유형별 플레그는 비트마스크로 인코딩되며 Arnold의 네이티브 형식으로 설정됩니다.

참고: https://docs.arnoldrenderer.com/display/A5AFHUG/Visibility

다른 가시성 설정(예: 측면성 - double_sidedness_per_ray_type으로 노출)도 레이(ray) 유형별 비트마스크로 제어할 수 있으나, 경우에 따라 이 노드가 레이(ray) 유형별 부울(boolean) 값만 노출시킬 수도 있습니다. 다른 상황에서도 그러한 UI가 필요하다면, Arnold_ray_bitmask 노드를 이용해서 이러한 레이(ray) 유형별 사항을 설정하면 됩니다.

불투명

지오메트리에서 불투명 플래그를 설정합니다.

무광

지오메트리에서 무광 플래그를 설정합니다.

볼륨

볼륨 뿐만 아니라 다른 많은 지오메트리를 볼륨으로 렌더링할 수 있습니다.

set_Arnold_volume_settings도 참고하십시오.

기타

흔히 사용되지 않는 다른 항목들을 설정합니다.

set_Arnold_mesh_settings

Arnold 전용 mesh(Arnold에서: polymesh) 설정을 설정하며 Arnold에서 Arnold-네이티브 설정으로 적용됩니다. 이 설정은 뷰포트에는 영향을 주지 않으며 Arnold 렌더링에만 영향을 줍니다.

이 설정은 내부 Arnold 기능에 매우 가깝게 매핑되므로 Arnold Node Reference를 본 문서와 연계해 이용할 수도 있습니다. 이때, 관련 문서는 Arnold polymesh 유형 즉, Bifrost mesh 유형에 관한 것입니다. https://docs.arnoldrenderer.com/display/NodeRef/polymesh

사용 방법

이것은 Arnold 지오메트리에 공통적인 항목들을 설정하는 set_Arnold_geo_settings와 연계하여 이용할 수도 있습니다.

서브디비전

Arnold 서브디비전과 관련된 각종 항목들을 설정합니다. 3D 응용프로그램에서 공통적으로 이용하는 기본 서브디비전의 경우, subdiv_type을 Catmull-Clark 로 설정하고 레벨과 관련해서는 subdiv_iterations 를 이용합니다.

Arnold 서브디비전에 대한 자세한 내용은 Arnold Node Reference 설명서에서 polymesh 부분을 참고하십시오.

디스플레이스먼트

Arnold 디스플레이스먼트와 관련된 각종 항목들을 설정합니다. 이것은 디스플레이스먼트 쉐이더를 할당하는 것이 아니라 디스플레이스먼트 항목만 설정하므로 할당을 위해서는 assign_material을 이용하시기 바랍니다.

Arnold 메쉬 디스플레이스먼트에 대한 자세한 내용은 Arnold Node Reference 설명서에서 polymesh 부분을 참고하십시오.

노멀

smoothing 노멀의 유연한 보간을 비활성화하는 데 이용합니다.

set_Arnold_points_settings

Arnold 전용 points (Arnold에서: points) 설정을 설정하며 Arnold에서 Arnold-네이티브 설정으로 적용됩니다. 이 설정은 뷰포트에는 영향을 주지 않으며 Arnold 렌더링에만 영향을 줍니다.

이 설정은 내부 Arnold 기능에 매우 가깝게 매핑되므로 Arnold Node Reference를 본 문서와 연계해 이용할 수도 있습니다. 이때, 관련 문서는 Arnold points 유형 즉, Bifrost points 유형에 관한 것입니다. https://docs.arnoldrenderer.com/display/NodeRef/polymesh

사용 방법

이것은 Arnold 지오메트리에 공통적인 항목들을 설정하는 set_Arnold_geo_settings와 연계해서 이용할 수도 있습니다.

Arnold 포인트

points_mode_override

Arnold 포인트에서 mode는 포인트 객체가 디스크, 스피어, 쿼드 중 무엇으로 렌더링될 것인지를 가리킵니다. 이것은 Bifrost points의 작동 방식과 다릅니다.

기본값은 Bifrost 설정의 기본 해석을 이용함을 의미합니다. Arnold 포인트가 지원하지 않는 지오메트리의 경우, 이런 포인트는 그 대신 지오메트리의 Arnold ginstance가 됩니다.

min_pixel_width

스크린 상에서의 한 포인트의 최소 크기를 설정합니다. Arnold에서 min_pixel_width의 반응에 대해서는 Arnold points 노드 참조를 참고하십시오.

set_Arnold_strands_settings

Arnold 전용 strands (Arnold에서: curves) 설정을 설정하며 Arnold에서 Arnold-네이티브 설정으로 적용됩니다. 이 설정은 뷰포트에는 영향을 주지 않으며 Arnold 렌더링에만 영향을 줍니다.

이 설정은 내부 Arnold 기능에 매우 가깝게 매핑되므로 Arnold Node Reference를 본 문서와 연계해서 이용할 수도 있습니다. 이때, 관련 문서는 Arnold curves 유형 즉, Bifrost strands 유형에 관한 것입니다. https://docs.arnoldrenderer.com/display/A5NodeRef/curves

사용 방법

이것은 Arnold 지오메트리에 공통적인 항목들을 설정하는 set_Arnold_geo_settings와 연계해서 이용할 수도 있습니다.

변환

match_ends 는 Arnold 커브의 엔드-포인트가 strands 와 일치할 것인지 여부를 설정합니다. 엔드 포인트가 커브 모드와 일치하려면, 엔드 포인트를 연장해야 합니다. 이것이 비활성화되면, 정점이 있는 그대로 사용됩니다.

Arnold 커브

mode_override

Arnold에서 mode는 가닥이 카메라를 향하는 리본, 방향이 정해진 리본, 두꺼운 가닥 중 무엇으로 렌더링될 것인지를 가리킵니다.

Bifrost에도 비슷한 설정이 있으며 기본값은 Bifrost 설정의 기본 Arnold 해석을 이용함을 의미합니다. 나머지 설정 항목들은 Arnold 항목들을 직접 설정합니다.

basis_override

Arnold에서 basis는 커브가 컨트롤 포인트에서 형성되는 방식을 설명합니다.

Bifrost(선형만 해당) strands의 기본 변환은 Catmul-Rom으로 변환되는 것이며 이것은 엔드 포인트(match_ends)와 일치하게 됩니다. 이 기본 해석이 적용되거나 기본 항목이 직접 설정될 수도 있습니다.

Override_min_pixel_width

Bifrost 가닥은 현재 min_pixel_width를 가질 수 없으나 향후에 변경될 수 있으며, 이 경우 오버라이드로 노출됩니다.

Arnold에서 min_pixel_width의 반응에 대해서는 Arnold curves 노드 참조를 참고하십시오.

set_Arnold_volume_settings

Arnold 전용 volume (Arnold에서: volume과 implicit) 설정을 설정하며 Arnold에서 Arnold-네이티브 설정으로 적용됩니다. 이 설정은 뷰포트에는 영향을 주지 않으며 Arnold 렌더링에만 영향을 줍니다.

이 설정은 내부 Arnold 기능에 매우 가깝게 매핑되므로 Arnold Node Reference를 본 문서와 연계해서 이용할 수도 있습니다. 이때, 관련 문서는 Arnold volume 또는 implicit 유형 즉, Bifrost mesh 유형에 관한 것입니다. https://docs.arnoldrenderer.com/display/NodeRef/volume

사용 방법

이것은 Arnold 지오메트리에 공통적인 항목들을 설정하는 set_Arnold_geo_settings와 연계해서 이용할 수도 있습니다.

Bifrost volume 유형은 두 가지 Arnold 유형 즉, volume과 implicit에 해당합니다.

Volume과 Implicit의 비교

Volume은 구름, 안개, 불, 먼지와 같은 볼륨을 나타내는 데 이용합니다.

Implicit는 레벨 세트와 같이 볼륨에 의해 정의되는 서피스를 렌더링하는 데 이용합니다.

Arnold 볼륨

Arnold로 변환되는 Bifrost 볼륨의 기본 모드는 Arnold 볼륨입니다.

step_size는 볼륨의 최소 형상을 캡처할 수 있을 정도로 충분히 작아야 하는데, 0으로 설정하면 자동이고 복셀 크기를 이용합니다. step_scale은 스텝 크기를 스케일링합니다.

step_scale = 20 그리고 step_size = 0인 경우, 스텝 크기가 복셀 크기보다 20배가량 크다는 것을 의미합니다. 이것은 다수의 추가 샘플 없이도 볼륨의 렌더링을 부드럽게 하는 데 이용할 수 있습니다.

Arnold Implicit

as_implicit이 활성화되면, 볼륨이 볼륨 대신 서피스로 렌더링됩니다. 여기서 볼륨 대신 Arnold implicit 지오메트리 유형이 사용됩니다.

이때, 레벨 세트를 이용해야 하는데, 레벨 세트는 voxel_signed_distance 와 같은 특성을 이용해 지정합니다. 이것은 Arnold implicit에서 field_channel로 매핑됩니다.

solver 매개변수는 서피스 문제를 해결할 때 사용할 솔버를 설정합니다. levelset 솔버는 적합한 부호 붙임 거리 필드를 이용해야 하고 속도가 더 빠른 반면, uniform 솔버는 어느 데이터로도 서피스를 렌더링할 수 있으며 속도가 더 느립니다.

threshold는 볼륨에서 서피스를 렌더링할 지점을 정의합니다. 내부 값이 음수이고 외부 값이 양수인 부호 붙임 거리 필드의 기본값은 @이나, 밀도 필드에는 사용자 밀도 값을 입력해야 합니다.

set_voxel_size 노드

이 노드는 볼륨에 복셀 사이즈를 설정합니다.

Inputs

voxel_size [in] 새로운 복셀 사이즈

volume [inout] 입력/출력 볼륨 (Amino::Object)

sharpen_voxel_property 노드

이 노드는 이름이 'property'인 복셀 특성을 주어진 표준 편차와 필터 폭으로 정의된 가우스 필터로 평활화합니다.

Inputs

property [in]

특성의 이름

sharpening [in]

샤프닝(sharpening)의 정도

voxel_sharpening_radius [in]

샤프닝(sharpening) 필터의 복셀 반경

use_time_step [in]

true인 경우, 샤프닝 수준을 time_step만큼 스케일링합니다.

time_step [in] 타임 스텝

num_resolution_levels [in] 평활화하려는 해상도 레벨 수. 1인 경우 최고 레벨만 평활화합니다.

volume [inout] 특성을 포함하는 볼륨 (Amino::Object)

smooth_mesh 노드

이 노드는Bifrost 메쉬를 평활화(smooth)합니다.

Inputs

mesh [in]

입력 메쉬

strength [in]

메쉬에 적용할 평활화의 정도. 값이 클수록 평활화의 강도가 커집니다.

Outputs

out_mesh [out]

평활화된 메쉬

smooth_voxel_property 노드

이 노드는 이름이 'property'인 복셀 특성을 주어진 표준 편차와 필터 폭으로 정의된 가우스 필터로 평활화합니다.

Inputs

property [in] 특성의 이름

voxel_standard_deviation [in] 복셀의 가우스 표준 편차

voxel_filter_width [in] 복셀의 가우스 필터 폭

num_resolution_levels [in] 평활화하려는 해상도 레벨 수. 1인 경우 최고 레벨만 평활화합니다.

volume [inout] 특성을 포함하는 볼륨 (Amino::Object)

smooth_voxel_temperature 노드

이 노드는 가시 voxel_fog_density의 에지 주변의 스무스 폴오프(smooth falloff)로 온도 특성을 재매핑합니다. 이로써, 좀 더 매끄러운 블랙바디 방사 렌더링이 만들어집니다.

Inputs

temperature_property [in]

온도 특성의 이름

fog_density_property [in] 뿌염 농도 특성의 이름(평활화에 필요)

smooth [in]

0과 1 사이의 평활도 매개변수로, 재매핑된 온도의 에지에서 평활도의 수준을 좌우합니다.

volume [inout] 볼륨 (Amino::Object)

splat_points_into_volume 노드

이 노드는 포인트들을 입력/출력 볼륨의 특성들로 스플랫(splat)합니다. 포인트에 반경 채널이 있을 경우, 포인트를 스플랫(splat)할 때 이 연결된 반경을 이용하게 됩니다. 그렇지 않으면, 입력 반경 포트의 값을 사용하게 됩니다. 현재 더 낮은 해상도 레벨로 스플랫(splat)할 때는 포인트에 반경 채널이 있더라도 입력 반경 포트가 사용됩니다.

Inputs

volume [inout] 포인트들을 스플랫(splat)할 목표 볼륨 (Amino::Object)

create_properties [in] true인 경우, 이미 존재하는 경우가 아니라면 포인트 특성에 해당하는 새로운 복셀 특성들이 생성됩니다(bool).

properties [in] 스플랫(splat)할 포인트 특성들의 이름

points [in] 스플랫(splat)할 포인트 (Amino::Object)

kernel [in] 포인트들을 스플랫(splat)할 때 이용할 커널(SplatKernelType)

radius [in] 포인트들을 스플랫(splat)할 때 이용할 커널의 복셀 반경(float)

add_to_weights [in]

각 포인트의 무게에 추가되는 정규화. 이 값이 클수록 스플랫(splat)되는 값이 해당 복셀에 기여하는 모든 포인트들의 평균에 더 가까워집니다. 즉, 이 값이 클수록 포인트와 복셀 간의 거리가 줄어듭니다(float).

add_to_denominator [in]

각 포인트의 무게에 추가되는 정규화이지만 분모에만 적용됩니다. 이것은 스플래팅(splatting) 커널 서포트 에지에 하드 에지가 생기는 것을 방지하는 데 유용합니다(float).

smoothing [in]

스플래팅(splatting) 후 적용되는 평활화의 정도. 이것은 포인트 스플랫(splat) 때문에 때때로 생길 수 있는 하드 에지와 불분명성을 완화하는 데 유용합니다.

coarsest_depth [in]

타일 트리에서 스플래팅이 발생해야 하는 최저 해상도의 깊이(int)

string_to_array

comma_separated_string을 문자열, long 정수 또는 floats의 어레이로 변환합니다. 이로써, 그래프에 문자열, 부동 또는 (long) 정수들의 어레이를 빠르게 작성할 수 있습니다.

문자열은 쉼표로 구분하며 부분 문자열의 각 끝에 있는 빈 칸은 생략합니다.

<pre>comma_separated_string:</pre>	"2, 4, 6"
<pre>string_array:</pre>	["2", "4", "6"]
float_array:	[2.0, 4.0, 6.0]
long_int_array:	[2, 4, 6]

부분 문자열의 시작 또는 끝에 있는 빈 칸은 생략합니다.

<pre>comma_separated_string:</pre>	"One	string,	two string "
string_array:	["0ne	string",	"two string"]

정수 또는 부동으로 변환할 수 없는 값은 부동 또는 정수 어레이에서 0이 됩니다.

Inputs

comma_separated_string

포트 유형: 문자열.

어레이로 처리되는 문자열.

Outputs

string_array (OUt)

포트 유형: array<string>.

이것은 시작과 끝에 빈 칸이 생략된 채로 분리된 문자열입니다.

float_array (OUt)

포트 유형: array<float>.

이것은 부동으로 변환되는 분할값입니다. 변환할 수 없는 값들은 0.0이 됩니다.

long_int_array (OUt)

포트 유형: array<long>.

이것은 부동으로 변환되는 분할값입니다. 변환할 수 없는 값들은 0이 됩니다.

turbulence influence

터뷸런스(turbulence) 효과는 컬(curl) 노이즈 가속을 적용하는데, 이 기능은 난기류와 같은 효과를 연출하는 데 유용합니다. 컬 노이즈의 장점은 최종 속도가 기류처럼 비압축성이라는 점인데, 이 때문에 파티클 연기 시뮬레이션이 더욱 자연스러워집니다. 이때, drag 값이 클수록 시간이 흐름에 따라 속도가 증가하는 대신 파티클들이 컬 노이즈 흐름을 더욱 가까이 따라가게 됩니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

turbulence

시뮬레이션 속도에 적용되는 가속의 크기

frequency

이것은 노이즈의 사이즈를 제어합니다. 값이 클수록 소용돌이 개수가 줄어듭니다. 시뮬레이션에 따라 대형 소용돌이와 소형 소용돌이가 모두 필요한 경우도 있는데, 이 경우 주파수가 서로 다른 turbulence_influence를 두 개 이상 연결하면 됩니다. 주파수 값이 낮을수록 보통 더 큰 터뷸런스 값이 필요합니다.

change_rate

이것은 시간이 흐름에 따라 컬 노이즈가 얼마나 빠르게 발달하는지를 제어합니다. 이 값이 0이면, 터뷸런스의 소용돌이가 고정되며 시간이 흘러도 변하지 않습니다.

drag

이것은 현재의 시뮬레이션 속도를 떨어뜨려 시간이 갈수록 속도 증가 효과가 줄어듭니다. drag 값이 크면 지속적으로 가속을 더하는 대신 시뮬레이션 속도를 터뷸런스 가속으로 완전히 교체합니다. 이것은 사나운 바람이 밀어내는 연기 파티클들을 시뮬레이션할 때 유용합니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

twice_of

value에 2를 곱해 double_value를 반환합니다.

이것은 모든 정수 유형, 부동 소수점 유형, 그리고 일반적이지는 않지만 벡터까지도 배가시킬 수 있습니다. 정수 오버플로우나 언더플로우에 대한 특별한 처리 방법은 없습니다.

update_mesh_normal

mesh에서 point_normal 특성을 다시 계산합니다. 이것은 해당 정점의 인접 면의 노멀들의 평균으로 계산됩니다.

Inputs

mesh

입력 메쉬.

remove_face_vertex_normal

face_vertex_normal 특성의 삭제 여부. 이 옵션은 face_vertex_normals가 있을 때 지원되며, 뷰포트 렌더링에서는 point_normal을 이용하지 않습니다.

Outputs

out_mesh (OUt)

업데이트된 메쉬.

point_normal (OUt)

point_normal 특성의 콘텐츠, 포인트별 메쉬 노멀값.

update_voxel_position 노드

이 노드는 볼륨에 'voxelpos'라는 이름의 새로운 float3 복셀 채널을 만들고 이를 voxel_offset 매개변수에서 제공하는 오프셋을 이용해 복셀의 월드스페이스 포지션으로 초기화합니다.

Inputs

voxel_offset [in] 복셀 포지션 생성에 이용되는 오프셋

volume [inout]

추가된 채널을 포함하는 볼륨 (Amino::Object)

vdb_influence

VDB 효과는 VDB 파일에서 얻은 속도를 시뮬레이션 속도에 적용합니다. drag 값이 크고 force_scale 이 1.0인 경우, 시뮬레이션은 정확히 VDB 속도를 따릅니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>influence_fields</u> 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

force_scale

이것은 가속으로 적용하려는 vdb_file의 속도의 곱수입니다.

filename

VDB 파일의 기본 파일이름

directory

VDB 파일이 들어있는 폴더의 파일 경로

force_attribute

이것은 속도로 이용하려는 VDB의 특성의 이름입니다. float3 특성입니다.

start_frame VDB 파일에서 가져와 이용하려는 프레임 범위의 시작

end_frame VDB 파일에서 가져와 이용하려는 프레임 범위의 끝

frame_offset

이것은 적용하려는 현재 프레임으로부터의 오프셋입니다. 따라서, frame_offset이 100이었다면, VDB 파일에서 가져온 프레임 1은 프레임 101에서 실행됩니다.

이것은 VDB 필드의 사이즈를 스케일링합니다.

position_offset

position_scale

이것은 VDB 필드의 포지션을 변환합니다.

drag

이것은 현재의 시뮬레이션 속도를 떨어뜨려 시간이 갈수록 속도 증가 효과가 줄어듭니다. drag 값이 크면 지속적으로 가속을 더하는 대신 시뮬레이션 속도를 VDB 가속으로 완전히 교체합니다. 파티클들이 VDB 흐름을 정확히 따르도록 하려면 큰 drag 및 mak 값을 이용합니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

volume_bounds 노드

이 노드는 한 볼륨의 최소 및 최대 바운드를 계산합니다. 이때 레벨 세트나 뿌염 농도 볼륨을 고려하며, 이 두 가지가 모두 존재하는 경우에는 두 바운드 수의 합이 반환됩니다.

Inputs

volume [in] 입력 볼륨 (Amino::Object)

Outputs

min_bound [out] 입력 볼륨의 최소 바운드 (float3)

max_bound [out] 입력 볼륨의 최대 바운드 (float3)

volume_influence

볼륨 효과는 볼륨 객체의 voxel_velocity 특성을 가속으로 적용합니다. 이를 이용해 공기 시뮬레이션에서의 속도로 파티클들을 밀어낼 수 있습니다. 이때, force_scale을 1.0으로 설정하고 drag 값이 큰 drag_influence의 influence 포트에 추가로 연결하는 것이 좋습니다. 이로써 파티클들이 공기 시뮬레이션의 흐름을 정확히 따르게 됩니다.

이 노드를 이용하려면, 볼륨 객체를 volume 포트에 연결한 다음, out_influence를 simulate_aero, simulate_particles, simulate_mpm과 같은 simulate 노드의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 영향력의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

volume

이것은 simulate_aero 노드의 출력처럼 voxel_velocity 특성을 가진 볼륨 객체입니다.

force_scale

이것은 가속으로 적용하려는 볼륨 voxel_velocity의 곱수입니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

volume_scope

볼륨 범위는 볼륨을 보간해 시각적 진단 정보를 제공합니다.

이는 볼륨 데이터를 속도의 플로우라인, 타일 트리 구조, 볼륨의 숫자 값 등 다양한 방식으로 시각화하는 데 이용할 수 있습니다.

사용 방법

사용하려면, 단일 볼륨이나 볼륨 어레이(예, Aero 시뮬레이션의 출력)를 볼륨 플러그에 연결하고 이 출력을 다른 출력에 연결합니다. probe_geometry 를 연결해 샘플을 어디서 수집할 것인지를 결정합니다. 이 출력은 볼륨을 시각적으로 보간하는 진단 지오메트리들의 어레이입니다.

프로브 지오메트리 및 볼륨 샘플링

볼륨에서 보간할 포지션은 여기서 포인트 기반 지오메트리(메쉬, 포인트, 가닥, 인스턴스) 형태로 제공될 수 있습니다. 지오메트리 포인트의 포지션(point_position)이 사용됩니다.

프로브 지오메트리가 제공되지 않은 경우, 샘플링 포인트들의 3D 그리드가 자동으로 구성됩니다. 이것은 divisions로 제어합니다.

Sampler 설정은 볼륨의 샘플링 방식을 제어합니다.

컬링(Culling)

컬링은 샘플을 제거합니다.

특성별 컬링은 일부 특성에서 0에 가까운 샘플이 있으면 모두 제거하는 것을 의미합니다. 예를 들어, 안개 밀도가 0인 경우 모든 값들을 제거하는 데 이 컬링을 이용할 수 있습니다.

culling_bounds_geometry를 이용해 지오메트리를 제공하여 컬링할 수도 있습니다.

컬링 작업에 특성이나 바운딩 박스를 이용할 수 있습니다.

플로우라인 표시

플로우라인은 샘플링 포지션에서 시작하는 strands이며, 속도 필드를 통과하는 흐름을 추적합니다. 이것은 볼륨을 통과해 흐르는 파티클의 경로를 시뮬레이션하는 것으로 생각할 수도 있습니다. 이 기능은 프로브 지오메트리, 샘플링, 컬링, 컬러와 함께 이용할 수 있습니다. 플로우라인은 voxel_velocity 또는 기타 벡터 특성을 따라 생성될 수 있습니다.

플로우라인이 한 흐름 속 파티클의 경로를 시뮬레이션할 때, 그 파티클이 흐름 속에 머무는 시간을 지정함으로써 길이를 제어할 수 있습니다. flowline_frame_length_flowlines 매개변수는 샘플링된 속도에서 한 프레임과 같아지도록 흐름 거리를 설정합니다. 그렇지 않으면, flowline_length_in_seconds로 시간(초 단위)을 설정합니다. 흐름 속 시간이 길수록 가닥의 길이가 길어집니다.

flowline_samples_per_second는 시간에 따른 샘플 밀도를 제어하며 만들어진 커브를 부드럽게 하는 매개변수 역할을 합니다. flowline_samples_per_second가 100인 경우, 0.05초 길이의 플로우라인은 샘플 5개로 구성되고, 1초 길이의 플로우라인은 샘플 100개로 이루어집니다. 샘플이 많을수록 더 부드러워지는 대신 컴퓨팅 시간이 길어집니다.

컬러

상한값, 중간값, 하한값에 대해 세 가지 컬러를 지정할 수 있습니다. 각종 진단 양식에는 이러한 컬러들로 다시 스케일링되는 값들이 입력됩니다. 플로우라인 상의 속도인 경우, 이 값은 속도의 분포를 나타냅니다.

포인트 디스플레이 및 숫자 디스플레이

샘플링된 로케이션에서 주어진 특성을 샘플링하고 플로우라인에서처럼 컬러를 사용해 표시 가능한 포인트 지오메트리를 생성합니다. 이 기능은 프로브 지오메트리, 샘플링, 컬링, 컬러와 함께 이용할 수 있습니다.

숫자 디스플레이는 샘플링된 로케이션에서 값들을 수치로 보여줍니다.

Multi-res-display

display_tile_tree 매개변수는 멀티-해상도 볼륨 계층구조의 시각화를 활성화합니다. 가장 깊은 n 레벨(visible_levels)이 표시됩니다. 이 시각화는 박스와 strands 외곽선으로 구성됩니다. 이 기능은 컬링 및 컬러와 함께 이용할 수 있습니다.

고해상도 타일은 high_color(기본값은 빨간색)로 그려집니다. 이어서, 중간 해상도 타일은 mid_color(기본값은 녹색)로, 저해상도 타일은 low_color(기본값은 파란색)로 각각 바뀝니다. 화면에 나타나는 박스는 복셀이 아니라 타일인 점을 유념하십시오. 각 타일 안에는 약 5x5x5 실복셀의 미니 그리드가 있는데, 이것은 볼륨의 설정에 따라 달라집니다.

이 기능은 컬링 지오메트리 및 컬러와 함께 이용할 수 있습니다.

volume_to_mesh 노드

이 노드는 Bifrost 레벨 세트를 Bifrost 메쉬로 변환합니다.

Inputs

volume

윤곽을 형성하기 위해 설정된 레벨을 포함하는 볼륨

iso_value

윤곽을 형성하기 위한 부호 붙임 거리 함수의 iso 값

detail_size

글로벌 (월드 스페이스) 디테일 사이즈. 이 값이 입력 볼륨의 디테일 사이즈보다 클 경우, 삼각형이 적은 개수로 더 크게 생성됩니다.

detail_size_property

가변 공간 (월드 스페이스) 디테일 사이즈 특성의 이름. 입력 볼륨에 이 이름의 float 복셀 특성이 있다면, mesher가 이 특성을 이용하여 생성되는 삼각형들의 사이즈를 공간적으로 다양하게 만듭니다.

smoothing

메쉬에 적용할 평활화의 정도. 값이 클수록 평활화의 강도가 커집니다.

properties

메쉬로 샘플링 하려는 복셀 특성들의 목록으로 공간 범위가 정해져 있습니다.

Outputs

mesh

출력 메쉬

vorticity_influence

이것은 시뮬레이션의 속도에 소용돌이 제한을 적용합니다. 이것은 시뮬레이션에서 약해질 수 있는 작은 소용돌이를 일으키는 데 효과적입니다. 또, 시뮬레이션을 더욱 격렬하게 그리고 점성이 약해지도록 만듭니다. 현재 이 노드는 simulate_aero에서만 효과가 있습니다.

이 노드를 이용하려면, 이 노드의 출력을 simulate_aero의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, modulate_influence 노드를 이용하면 voxel_temperature가 높을 때만 소용돌이가 커지도록 만들 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

vorticity

시뮬레이션에 적용되는 추가 가속의 강도. 최상의 값은 일반적으로 시뮬레이션의 detail_size에 따라 달라집니다. 디테일 사이즈가 더 작을 경우, 소용돌이 값을 낮추어 시뮬레이션에 노이즈가 커지는 것을 방지해야 합니다.

scale

이것은 공기 시뮬레이션에서 적응성이 더 큰 복셀인 경우에 소용돌이 값이 커지는 정도를 말합니다. 보통 이렇게 더 큰 복셀들은 밀도가 거의 없거나 아예 없는 영역에 위치하게 됩니다. 기본 스케일인 0은 사이즈와 관계없이 모든 복셀에 미는 힘을 똑같이 적용합니다. 값이 1.0이면, 미는 힘은 타일 트리의 relate 복셀 사이즈에 정확히 비례합니다. 그 결과, 소용돌이가 더 커지게 됩니다. 이 특성의 범위는 0.0 ~ 1.0입니다.

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

wind_influence

바람 효과는 한 방향으로 흐르는 단순 바람을 시뮬레이션 합니다.

이 노드를 이용하려면, 이 노드의 출력을 simulate_aero, simulate_particles, simulate_mpm 등과 같은 simulate 노드의 influence_fields 입력에 연결합니다.

Inputs

influence

선택적으로, influence 필드 out_influence를 이 포트에 연결할 수도 있습니다. 예를 들어, mask_influence 노드를 연결해 이 influence의 효과를 감출 수 있습니다. Influence 필드를 원하는 만큼 데이지 체인 방식으로 연결해 복합 필드를 구성할 수도 있습니다.

enable

해제하면 효과가 비활성화됩니다. 그러면, 이 노드는 패스쓰루 노드가 됩니다. 이 노드를 비활성화하더라도 Influence 입력 포트에 연결된 다른 모든 influence 필드는 계속해서 활성화된 상태를 유지합니다.

wind_speed

이것은 drag 값이 클 때 생기는 흐름의 속도입니다. drag 값이 작을 때에는 시간이 갈수록 파티클들이 이 속도보다 좀 더 빠르게 가속될 수 있습니다.

wind_direction

이것은 바람의 전반적인 방향 벡터입니다. 이 방향은 자체적으로 정규화 되므로 벡터의 길이는 중요하지 않습니다.

drag

이것은 바람 속도 쪽으로 drag를 적용합니다. Influence 필드에서 끌어가는 힘은 모든 작용력 효과들의 속도의 합을 바람으로 취급하고 그 움직임 쪽으로 끌어갑니다(단순히 0의 속도 쪽으로 끌어가는 것이 아님).

Output

out_influence

시뮬레이션 노드나 다른 influence 노드에 연결할 수 있는 출력 influence 객체

이 섹션의 페이지들

- make_mpm_cloth
- make_mpm_fiber
- mpm_solver_settings
- simulate_mpm
- source_mpm_fluid
- source_mpm_sand
- source_mpm_snow
- split_particles_by_material

상위 페이지: 참조

make_mpm_cloth

simulate_mpm이 천을 생성해야 하는 위치를 지정합니다.

이 컴파운드의 cloth_source 출력을 simulate_mpm의 sources 입력에 연결해야 합니다.

Inputs

geometry

천이 생성되는 위치를 정의하는 데 이용할 메쉬. 이 메쉬는 면이 없고 정점도 3개 미만이어야 합니다. 각이 작은 면은 피하는 것이 좋습니다.

General

enable

이 소스를 이용해야 하는지를 결정합니다. 설정하지 않으면 이 소스는 무시되며 천을 생성하지 않습니다.

start_frame

천 시뮬레이션이 시작되는 프레임. mpm_solver_settings 컴파운드에서 use_master_start_frame을 선택하면 이 옵션이 재지정됩니다.

Cloth Properties

mass_density

입방미터당 질량의 양(kg). 값이 클수록 천이 더 무거워지고 더 가벼운 재질을 마음대로 다룰 수 있게 됩니다.

viscosity

주변 포인트들의 움직임에 대한 평균값을 얼만큼 낼 수 있는지를 의미합니다. 이것은 동점도(제곱미터/초)입니다.

area_preservation

천이 자체 면의 면적 대 면의 모양을 유지하는 정도. 이 값은 단위가 없고 범위는 0부터 1까지입니다. 1은 모양이 전혀 유지되지 않음을 의미하고, 0은 면적이 전혀 유지되지 않음을 의미합니다. 대부분의 솔리드의 경우, 0.5가 적절하며 이 값에서는 모양과 면적이 균등하게 유지됩니다. 예를 들어, 천 조각 하나의 두 반대 면을 밀 때, 보통 서로 다른 이 두 면은 바깥으로 휘어 전체 면적을 유지하지만, <u>area_preservation</u>이 0일 때는 천이 전혀 늘어나지 않고 모양이 변할 때 뒤쪽으로만 밀립니다. 이 매개변수는 공학에서 흔히 사용되는 포아송 비(Poisson Ratio)와 관련이 있습니다.

vibration_speed

진동이 천을 통과하는 속도(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. 이것은 천 안에서의 음속이며 기압파의 자연 속도입니다. 일반적으로 값이 클수록 컴퓨팅 시간이 길어지므로 가장 작은 값을 사용하는 것이 결과가 좋습니다. 천이 눈에 보일 정도로 튕기지 않는다면, 진동이 한 프레임 내에서 천을 충분히 통과할 수 있을 만큼 진동 속도가 커야 합니다.

collision_max_speed

충돌력의 크기(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. vibration_speed와 마찬가지로, 값이 클수록 컴퓨팅 시간이 길어집니다. 상호 침투에 문제가 있다면 이 값을 높이십시오.

thickness

천의 두께(m)로, mass_density 및 면의 면적과 함께 이용해서 포인트당 질량을 설정합니다.

initial_speed

포인트들의 초기 속도. 여기에 direction이 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다. 값의 단위는 m/s입니다.

direction

포인트들의 초기 속도 방향. 여기에 initial_speed가 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다.

inherit_velocity

geometry 입력의 속도 비로, 포인트들의 초기 속도에 더해집니다.

reference_point_position

천의 휴면 상태를 정의합니다. 포지션들의 어레이를 연결해 천이 늘어나지도 않고 압축되지도 않은 상태를 정의합니다. 천이 처음에 쭉 펼쳐져 있기를 원한다면 이 기능을 이용합니다.

Interior Volume

preserve_interior_volume

천이 마치 안에 공기가 갇혀 있는 것처럼 반응합니다. 메쉬 geometry는 폐쇄형이어야 하며(구멍이 없음) 모든 면이 한 방향을 향하고 있어야 이 옵션이 작동합니다.

target_volume_ratio

reference_point_position(또는 start_frame에서의 geometry point_position)의 공기량과의 비로서 천 안에 들어 있는 공기의 양을 제어합니다. 천은 목표량에 맞추기 위해 밖으로 펼쳐지거나 안으로 줄어듭니다. 이것은 preserve_interior_volume이 활성화 되어야만 효과를 냅니다.

volume_stiffness

내부 볼륨 제약의 강연도. 값이 작을수록 힘이 작아지고 전반적으로 탄력 효과를 갖습니다.

Failure

failure_post_process

천 오류 즉, 에지가 failure_stretch보다 더 많이 펼쳐지는 현상을 처리하는 방법에 대한 옵션.

- no_failure: 천에 오류가 생기지 않습니다.
- no_post_processing: 에지에 오류가 생겨 더 이상 참조 길이로 돌아가려 하지 않지만 지오메트리는 변하지 않습니다.
- delete_failed_faces: 메쉬가 삼각형 모양이 되며 오류가 발생한 에지에 인접한 삼각형들은 삭제됩니다.
- subdivide_and_split: 전체 메쉬가 삼각형 모양이 되고 중심에 정점을 삽입함으로써 각 삼각형이 사변형 4개로 나뉘어집니다. 원래의 메쉬에서 오류가 발생한 에지에서는 그에 해당하는 사변형들이 분리되어 나옵니다. 메쉬의 모든 면이 한 방향을 향해야 하며, 어느 에지에서도 면 3개 이상이 만날 수 없어야 이 옵션이 작동합니다.

failure_stretch

참조 에지 길이에 대한 현재 에지 길이의 비로, 에지에 오류를 초래합니다. 에지 오류 효과에 대해서는 failure_post_process를 참조하십시오. 입력 지오메트리에 각 에지의 신축(stretch) 오류를 설정하는 데 이용할 수 있는 특성 face_vertex_failure_stretch 가 있다면 이것에 우선합니다. 해당 에지에 있는 첫 번째 정점이라면 그 면의 정점은 에지에 연결됩니다.

Constraints

collide_constrained

구속된 포인트들의 충돌 발생 여부를 결정합니다.

geo_constraint_property

입력 geometry의 포인트 지오메트리 특성의 이름으로, 입력 point_position을 따르도록 구속될 포인트들을 정의합니다. 값이 0이면 포인트가 구속되지 않음을 의미하고, 1이면 완전 구속됨을 의미합니다. 분수값들은 약한 구속을 의미하고 마치 0의 휴면 길이 스프링에 연결된 것처럼 반응합니다.

상위 페이지: mpm

make_mpm_fiber

simulate_mpm이 섬유를 생성해야 하는 위치를 지정합니다.

이 컴파운드의 fiber_source 출력을 simulate_mpm의 sources 입력에 연결해야 합니다.

Inputs

geometry

섬유가 생성되는 위치를 정의하는 데 이용할 가닥. 이 메쉬는 면이 없고 정점도 3개 미만이어야 합니다. 각이 작은 면은 피하는 것이 좋습니다.

General

enable

이 소스를 이용해야 하는지를 결정합니다. 설정하지 않으면 이 소스는 무시되며 섬유를 생성하지 않습니다.

start_frame

섬유 시뮬레이션이 시작되는 프레임. mpm_solver_settings 컴파운드에서 use_master_start_frame을 선택하면 이 옵션이 재지정됩니다.

Fiber Properties

mass_density

입방미터당 질량의 양(kg). 값이 클수록 섬유가 더 무거워지고 더 가벼운 재질을 마음대로 다룰 수 있게 됩니다.

viscosity

근처 포인트들의 움직임에 대한 평균을 얼마만큼 낼 수 있는지를 의미합니다. 이것은 동점도(제곱미터/초)입니다.

vibration_speed

진동이 섬유를 통과하는 속도(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. 이것은 섬유 안에서의 음속이며 기압파의 자연 속도입니다. 보통 값이 클수록 컴퓨팅 시간이 길어지므로 가장 작은 값을 사용하는 것이 결과가 좋습니다. 섬유가 눈에 보일 정도로 튕기지 않는다면, 진동이 한 프레임 내에서 천을 충분히 통과할 수 있을 만큼 진동 속도가 커야 합니다.

collision_max_speed

충돌력의 크기(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. vibration_speed와 마찬가지로, 값이 클수록 컴퓨팅 시간이 길어집니다. 상호 침투에 문제가 있다면 이 값을 높입니다.

thickness

섬유의 두께(m)로 mass_density 및 세그먼트의 길이와 함께 이용해서 포인트당 질량을 설정합니다.

initial_speed

포인트들의 초기 속도. 여기에 direction이 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다. 값의 단위는 m/s입니다.

direction

포인트들의 초기 속도 방향. 여기에 initial_speed가 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다.

inherit_velocity

geometry 입력의 속도 비로, 포인트들의 초기 속도에 더해집니다.

reference_point_position

섬유의 휴면 상태를 정의합니다. 포지션들의 어레이를 연결해 섬유가 늘어나지도 않고 압축되지도 않은 상태를 정의합니다. 섬유가 처음에 쭉 펼쳐져 있기를 원한다면 이 기능을 이용합니다.

Failure

allow_failure

에지가 failure_stretch 보다 더 많이 늘어날 때 오류가 생기도록 할 것인지의 여부

failure_stretch

참조 에지 길이에 대한 현재 에지 길이의 비로, 에지에 오류를 초래합니다.

Constraints

collide_constrained

구속된 포인트들의 충돌 발생 여부를 결정합니다.

geo_constraint_property

입력 geometry의 포인트 지오메트리 특성 이름으로, 입력 point_position을 따르도록 구속될 포인트들을 정의합니다. 값이 0이면 포인트가 구속되지 않음을 의미하고, 1이면 완전 구속됨을 의미합니다. 분수값들은 약한 구속을 의미하고 마치 0의 휴면 길이 스프링에 연결된 것처럼 반응합니다.

상위 페이지: mpm

mpm_solver_settings

이 컴파운드의 settings 출력을 simulate_mpm의 settings 입력에 연결해야 합니다.

General

enable_solver

시뮬레이션의 실행 여부. 이 기능을 통해 시뮬레이션이 필요하지 않을 때 모든 시뮬레이션을 정지시킬 수 있습니다.

use_master_start_frame

master_start_frame을 이용하면 simulate_mpm의 colliders와 sources의 start_frame 특성을 무효화할 수 있습니다. 그리고, 모든 것이 동시에 시작되어야 하는 단순한 상황에서 시뮬레이션이 시작되는 시점을 손쉽게 변경할 수 있습니다.

master_start_frame

use_master_start_frame 옵션을 선택했을 때 이용하게 될 프레임. 이 프레임은 simulate_mpm의 colliders와 sources의 start_frame 특성을 무효화합니다.

MPM Solver Globals

detail_size

simulate_mpm에서 volume에 이용할 복셀 사이즈(m). 값이 작을수록 시뮬레이션의 정확도는 향상되지만 런타임이 길어지고 메모리 사용량이 늘어납니다. 이 값은 항상 0보다 커야 합니다. 시뮬레이션 속 파티클들은 influence 영역을 갖는데, 이 영역은 detail_size로 스케일링되며 detail_size가 작을수록 파티클의 크기도 작아집니다.

enable_gravity

시뮬레이션에 이 미만의 gravity 벡터가 적용되어야 하는지 여부.

gravity

시뮬레이션의 gravity 벡터(m/s2). 파티클들이 중력 방향의 힘을 받게 됩니다.

scene_units_in_meters

장면의 단위 크기(m). 이것은 장면 내 지오메트리(소스, 콜라이더 등)를 모델링한 스케일을 보상합니다. 이것이 중요한 이유는 많은 유체 현상이 스케일에 따라 크게 달라지기 때문입니다. 예를 들어, 장면 내 모든 지오메트리를 10으로 스케일링한 다음(실제 물리적 크기에 비례) 원래의 스케일로 시뮬레이션 하고자 한다면, scene_units_in_meters]를 0.1로 설정해야 합니다.

style

특정 형상을 더 좋게 만들기 위해서 simulate_mpm이 이용하는 메소드를 변경합니다.

- smooth : 더욱 매끄러운 움직임을 연출하며 가까운 파티클들은 유사 속도를 갖게 됩니다.
- chunky: 더 다양한 속도를 만들어내 크랙은 더 선명해지고 전반적으로 노이즈가 증가합니다.
- automatic_from_sources 는 sources의 실제 재질에 따라 smooth와 chunky 중 하나를 선택합니다.
time_step_size

기본 타임 스텝의 스케일링 인수이며 <u>simulate_mpm</u> 컴파운드에서 각종 기준에 따라 결정됩니다. time_step_size가 클수록 전반적인 시뮬레이션 속도가 향상됩니다. 하지만, 값이 너무 클 경우에는 시뮬레이션에서 파티클들이 너무 빠르게 움직이는 등 안정성 문제가 발생합니다.

min_steps

각 프레임의 최소 타임 스텝 수

max_steps

각 프레임의 최대 타임 스텝 수

Point Display

shape

출력 granular_points의 디스플레이 모양을 설정합니다. 옵션에 대한 자세한 내용은 set_point_shape를 참조하십시오.

face_camera

출력 granular_points 디스플레이 지오메트리가 카메라 쪽으로 향해야 하는지 여부. 자세한 내용은 set_point_shape를 참조합니다.

numeric_property

shape가 Numeric으로 설정되어 있을 때, 출력 granular_points에 나타나는 포인트 특성. 자세한 내용은 set_point_shape를 참조합니다.

Advanced

max_voxel_movement

시뮬레이션이 불안정해지고 있는 것으로 간주하고 해당 프레임이 더 작은 타임 스텝으로 재시도되기 전에 한 파티클이 이동할 수 있는 복셀의 최대 복셀 거리(detail_size 참조). 시뮬레이션이 max_voxel_movement에 반복해서 닿는 경우, 그 대신 time_step_size 를 변경해볼 수 있습니다. 이 경우 재시도할 필요가 없어 시간이 절약됩니다..

max_retries

파티클들이 한 타임 스텝에서 maximum_voxel_movement 보다 많이 이동해 솔버가 그 프레임을 재시도하는 최대 횟수

label_sources

granular_points 출력에 숫자 특성 point_source_id를 추가로 할 것인지를 가리키며, 해당 포인트의 위치에서 소스 지오메트리를 식별합니다.

label_point_id

granular_points 출력에 숫자 특성 point_source_id 를 추가로 저장할 것인지를 가리키며, 일부 포인트들이 제거되더라도 전체 시뮬레이션 과정에 걸쳐 각 포인트마다 고유합니다.

lag_colliders

입력 colliders를 한 프레임 뒤쳐지도록 할 것인지의 여부. 뒤쳐지면, 직전 프레임의 지오메트리를 이용해 이번 프레임의 시뮬레이션과 충돌시킵니다. 이 옵션을 이용하면 simulate_mpm이 콜라이더에서 벌어지는 상황을 확인할 수 있어, 끝 지점에 다다르기 위해 빠르게 움직이는 콜라이더가 해당 프레임 중에 지정된 종료 포지션을 지나쳐 다시 돌아가는 오버슈팅을 방지할 수 있습니다. 지오메트리가 정확한 point_velocity를 갖는 경우에는 이 옵션을 이용하지 않고도 오버슈팅을 방지할 수 있습니다.

simulation_bounds

시뮬레이션을 연결된 지오메트리의 월드 스페이스 바운딩 박스로 국한시킵니다(지오메트리가 있는 경우).

상위 페이지: mpm

simulate_mpm

Bifrost의 MPM(Material Point Method)용 메인 컴파운드. 이것을 이용해서 모래, 눈, 유체, 천, 섬유 등 다양한 실재 재질들을 시뮬레이션할 수 있습니다.

이 컴파운드를 이용하려면, sources(각종 source_mpm 및 make_mpm 노드 중 선택)를 1개 이상 연결하고 mpm_solver_settings 노드를 settings 포트에 연결합니다.

MPM은 파티클에서 각종 물리적 특성들을 추적하는 방식으로 작동하는데, 이런 특성들은 복셀화되어 한 순간 해당 재질의 볼륨 표현을 얻게 됩니다. 볼륨의 순간적인 움직임이 컴퓨팅 되고 colliders에 따라 충돌한 다음 이를 통해 파티클을 움직이고 이들의 특성을 업데이트합니다. 이 프로세스는 한 프레임에서 여러 번 반복되는데, 이때, 각 반복을 타임 스텝이라고 부릅니다. 움직임은 복셀화된 표현을 통해 컴퓨팅 되므로 개별 파티클의 움직임은 볼륨에 의해 나타낼 수 있는 만큼으로 제한됩니다. 즉, 파티클은 개별 속도에 따라 움직이는 것이 아니라 볼륨의 평균 속도에 따라 움직입니다.

Inputs

sources

source_mpm_snow, source_mpm_sand, source_mpm_fluid, make_mpm_cloth, make_mpm_fibers의 출력 1개 이상. source_mpm 노드는 파티클들을 배치하고 granular_particles 출력을 만들어내고, make_mpm_cloth는 cloth_mesh 출력을 만들어내며, make_mpm_fibers는 fiber_strands 출력을 만들어냅니다.

colliders

collider 노드의 출력 0개 이상으로, 파티클 흐름을 방해하는 작용을 합니다.

influence_fields

influence 노드의 출력 0개 이상으로, 힘을 더하고, 특성을 변경하며, 파티클을 없애는 등의 작용을 합니다.

settings

mpm_solver_settings 노드의 출력 설정.

Outputs

granular_particles

source_mpm 노드 패밀리에서 생성되어 배치된 파티클. 각 입력 재질마다 어레이에 1개의 객체가 있습니다. 즉, 모든 모래 파티클들이 한 객체로 모이고, 모든 눈(snow) 파티클들이 또 다른 한 객체로 모입니다. split_particles_by_material을 이용해 이 출력을 각 재질 별 출력 파티클들로 나눕니다.

volume

시뮬레이션 내 모든 재질의 복셀화된 볼륨. 이것은 voxel_mass_density와 voxel_velocity 특성을 포함하고 있습니다. 일반적으로, 다른 시뮬레이션에 영향을 주는 데 유용합니다. 서피싱의 경우, 파티클을 가지고 레벨 세트를 만들고 또 이 레벨 세트를 가지고 메쉬를 만들 것을 권장합니다.

cloth_mesh

각 입력 make_mpm_cloth당 cloth_mesh 1개씩.

fiber_strands

각 입력 make_mpm_fiber 당 cloth_mesh 1개씩.

상위 페이지: mpm

source_mpm_fluid

simulate_mpm이 유체 파티클을 만들거나 얻어야 하는 위치를 지정합니다. 이것은 유체의 연속 스트림이나 유체의 한 영역을 생성하는 데 이용할 수 있습니다.

이 컴파운드의 fluid_source 출력을 simulate_mpm의 sources 입력에 연결해야 합니다.

Inputs

geometry

파티클이 생성되는 위치를 정의하는 포인트, 메쉬 또는 볼륨

General

enable

이 소스의 이용 여부를 결정합니다. 설정하지 않으면 이 소스는 무시되며 파티클을 생성하지 않습니다.

start_frame

파티클이 방출되기 시작하는 프레임. mpm_solver_settings 컴파운드에서 use_master_start_frame을 선택하면 이 옵션이 재지정됩니다.

use_end_frame

선택하면, 파티클 방출이 end_frame 뒤로도 계속됩니다.

end_frame

파티클 방출이 끝나는 프레임. start_frame과 같을 경우, 파티클들이 순간적으로 방출됩니다.

Particle Settings

distribution

파티클을 입력 geometry 중 어디에서 얻을 것인지를 지정합니다.

- Surface: 바깥 레이어 또는 서피스에서만 방출됩니다.
- Volume: 내부 또는 폐쇄형 볼륨에서 방출됩니다.

particles_per_voxel

한 복셀에 배치되어야 하는 평균 파티클 개수로, 여기서 복셀의 사이즈는 mpm_solver_settings의 detail_size에서 주어집니다. 값이 클수록 시뮬레이션 되는 파티클의 개수가 증가해 디테일이 추가되지만 움직임은 복셀화된 볼륨을 통해 컴퓨팅되기 때문에 이 값을 높이더라도 전체 움직임의 해상도는 그대로 유지됩니다. 또, 파티클 개수와 상관이 없는 이유로 detail_size를 줄여야 하는 경우, 예를 들어, 얇은 콜라이더 주변의 움직임을 자세히 표현하고 싶지만 파티클 개수를 늘리고 싶지는 않을 때는 이 값을 낮추는 것이 유용합니다.

particle_display_scale

파티클을 더 크게 또는 더 작게 만듭니다. 이것은 출력 point_size의 곱수로 작용하지만 시뮬레이션에는 영향을 주지 않습니다. 파티클의 유효 시뮬레이션 사이즈를 변경하려면, mpm_solver_settings의 detail_size 특성을 이용합니다.

Fluid Properties

mass_density

입방미터당 질량의 양(kg). 값이 클수록 유체가 더 무거워지고 더 가벼운 재질을 마음대로 다룰 수 있게 됩니다.

viscosity

근처 포인트들의 움직임에 대한 평균을 얼마만큼 낼 수 있는지를 의미합니다. 이것은 동점도(제곱미터/초)입니다.

surface_tension

유체가 전체 서피스 면적을 얼마나 축소시키려고 할 것인지를 지정합니다(kg/s2). 이것은 유체가 물방울이 되도록 만듭니다.

vibration_speed

진동이 유체를 통과하는 속도(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. 이것은 유체 안에서의 음속이며 기압파의 자연 속도입니다. 보통, 값이 클수록 컴퓨팅 시간이 길어지므로 가장 작은 값을 사용하는 것이 결과가 좋습니다. 유체가 눈에 보일 정도로 튕기지 않는다면, 진동이 한 프레임 내에서 유체를 충분히 통과할 수 있을 만큼 진동 속도가 커야 합니다.

initial_speed

파티클의 초기 속도. 여기에 direction 이 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다. 값의 단위는 m/s입니다.

direction

파티클의 초기 속도 방향. 이것은 initial_speed로 스케일링 되어 inherit_velocity 특성의 모든 속도에 더해집니다.

inherit_velocity

geometry 입력의 속도 비로, 파티클의 초기 속도에 더해집니다.

viscoelasticity

유체에게 유체가 변형된 방식에 대한 기억을 제공합니다. 이 유체는 이전 모양으로 되돌아가려 합니다. 값이 클수록 효과가 강해집니다. 값의 단위는 m2/s2입니다.

relaxation_time

viscoelasticity의 타임스케일로, 유체가 자신이 어떻게 변형되었는지를 얼마나 빨리 잊는지를 가리킵니다. 이 값이 0이면, 재질이 자신의 변형 사실을 금새 잊어버리고 더 점성 있는 반응을 나타냅니다. 값이 클수록 반응의 탄력성도 증가합니다. 값의 단위는 초(s)입니다.

set_particle_properties

set_property 노드를 연결해 시뮬레이션 되는 파티클에서 추가 특성을 정의하고 초기화합니다. 이런 특성들은 시뮬레이션 도중 influence_fields = 이용해 조정할 수 있습니다.

point_orientation은 솔버에 의해 업데이트되는데, 단, 그것이 float4에 저장된 4원수인 경우에 한 합니다.

상위 페이지: mpm

source_mpm_sand

simulate_mpm이 모래 파티클을 만들거나 얻어야 하는 위치를 지정합니다. 이것은 모래의 연속 스트림이나 모래의 한 영역을 생성하는 데 이용할 수 있습니다.

이름과는 달리, source_mpm_sand는 모래 뿐만 아니라 다른 낟알 재질들에도 이용할 수 있습니다. 모래는 source_mpm_snow보다 더 단순한 모델을 따릅니다. 가장 큰 차이점은 source_mpm_snow에서는 눈을 뭉치고 부피를 줄이고 더 단단하게 만들 수 있다는 것입니다.

이 컴파운드의 sand_source 출력을 simulate_mpm의 sources 입력에 연결해야 합니다.

Inputs

geometry

파티클이 생성되는 위치를 정의하는 포인트, 메쉬 또는 볼륨

General

enable

이 소스의 이용 여부를 결정합니다. 설정하지 않으면 이 소스는 무시되며 파티클을 생성하지 않습니다.

start_frame

파티클이 방출되기 시작하는 프레임. mpm_solver_settings 컴파운드에서 use_master_start_frame을 선택하면 이 옵션이 재지정됩니다.

use_end_frame

선택하면, 파티클 방출이 end_frame 뒤로도 계속됩니다.

end_frame

파티클 방출이 끝나는 프레임. start_frame과 같을 경우, 파티클들이 순간적으로 방출됩니다.

Particle Settings

distribution

파티클을 입력 geometry 중 어디에서 얻을 것인지를 지정합니다.

- Surface: 바깥 레이어 또는 서피스에서만 방출됩니다.
- Volume: 내부 또는 폐쇄형 볼륨에서 방출됩니다.

particles_per_voxel

한 복셀에 배치되어야 하는 평균 파티클 개수로, 여기서, 복셀의 사이즈는 mpm_solver_settings의 detail_size에서 주어집니다. 값이 클수록 시뮬레이션 되는 파티클의 개수가 증가해 디테일이 추가되지만 움직임은 복셀화된 볼륨을 통해 컴퓨팅되기 때문에 이 값을 높이더라도 전체 움직임의 해상도는 그대로 유지됩니다. 또, 파티클 개수와 상관이 없는 이유로 detail_size를 줄여야 하는 경우, 예를 들어 얇은 콜라이더 주변의 움직임을 더욱 자세히 표현하고 싶지만 파티클 개수를 늘리고 싶지는 않을 때는 이 값을 낮추는 것이 유용합니다.

particle_display_scale

파티클을 더 크게 또는 더 작게 만듭니다. 이것은 출력 point_size의 곱수로 시뮬레이션에는 영향을 주지 않습니다. 파티클의 유효 시뮬레이션 사이즈를 변경하려면, mpm_solver_settings의 detail_size 특성을 이용합니다.

Sand Properties

mass_density

입방미터당 질량의 양(kg). 값이 클수록 모래가 더 무거워지고 더 가벼운 재질을 마음대로 다룰 수 있게 됩니다.

viscosity

근처 포인트들의 움직임에 대한 평균을 얼마만큼 낼 수 있는지를 의미합니다. 이것은 동점도(제곱미터/초)입니다.

volume_preservation

모래가 부피 대 모양을 유지하는 수준. 이 값은 단위가 없고 범위는 0부터 1까지입니다. 1은 모양이 전혀 유지되지 않음을(모래가 액체처럼 반응) 의미하고, 0은 부피가 전혀 유지되지 않음을 의미합니다. 대부분의 솔리드의 경우, 0.5가 적절하며 이 값에서는 모양과 부피가 모두 똑같이 유지됩니다. 예를 들어 재질 한 덩어리의 두 반대 면을 밀 때, 보통 서로 다른 이 네 면은 바깥으로 휘어 전체 부피를 유지하지만, volume_preservation = 0일 때는 재질이 전혀 늘어나지 않고 모양이 변할 때 뒤로만 밀립니다. 이 매개변수는 공학에서 흔히 사용되는 포아송 비(Poisson Ratio)와 관련이 있습니다.

vibration_speed

진동이 모래를 통과하는 속도(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. 이것은 모래 안에서의 음속이며 기압파의 자연 속도입니다. 보통 값이 클수록 컴퓨팅 시간이 길어지므로 가장 작은 값을 사용하는 것이 결과가 좋습니다. 모래가 눈에 보일 정도로 튕기지 않는다면, 진동이 한 프레임 내에서 모래를 통과할 수 있을 정도로 충분히 진동 속도가 커야 합니다.

friction

모래가 미끌어질 때 자신에 대한 저항력으로 압축 정도에 대한 비입니다. 값이 클수록 주어진 압축 정도에서 미끄러짐에 대한 저항력이 더 커집니다. 이것을 내부 마찰 계수라고도 합니다. 음의 값이어서는 안 됩니다. 실제 값은 보통 0부터 1까지이나 더 큰 값도 지원됩니다.

cohesion

모래가 서로 붙거나 응집하려는 경향성(m2/s2). 0은 기본적으로 마른 모래이며, 값이 클수록 모래가 더 많이 뭉칩니다. 이것은 모래가 팽창에 얼마나 저항할 수 있느냐를 나타냅니다. 이것은 (눈의 cohesion x 현재의 firmness)와 같습니다.

initial_speed

파티클의 초기 속도. 여기에 direction이 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다. 값의 단위는 m/s입니다.

direction

파티클의 초기 속도 방향. 이것은 initial_speed 로 스케일링 되고 inherit_velocity 특성의 모든 속도에 더해집니다.

inherit_velocity

geometry 입력의 속도 비로, 파티클의 초기 속도에 더해집니다.

set_particle_properties

set_property 노드를 연결해 시뮬레이션 되는 파티클에서 추가 특성을 정의하고 초기화합니다. 이런 특성들은 시뮬레이션 도중 influence_fields 를 이용해 조정할 수 있습니다.

point_orientation은 솔버에 의해 업데이트되는데, 단, 그것이 float4에 저장된 4원수인 경우에 한 합니다.

상위 페이지: mpm

source_mpm_snow

simulate_mpm으로 눈이 내리는 것을 연출할 때 눈 파티클이 만들어지거나 나오는 위치를 지정합니다. 이것은 눈의 연속 스트림이나 눈의 한 영역을 생성하는 데 이용할 수 있습니다. 그 이름과는 달리, source_mpm_snow는 눈 뿐만 아니라 다른 낟알 재질들에도 이용할 수 있습니다. 이것은 source_mpm_sand보다 좀 더 복잡한 모델을 따르는데, 가장 큰 차이점은 source_mpm_snow로는 눈을 뭉치고 부피를 줄이고 더 단단하게 만들 수 있다는 것입니다.

이 컴파운드의 snow_source 출력을 simulate_mpm의 sources 입력에 연결해야 합니다.

Inputs

geometry

파티클이 생성되는 위치를 정의하는 데 이용할 포인트, 메쉬 또는 볼륨

General

enable

이 소스의 이용 여부를 결정합니다. 설정하지 않으면 이 소스는 무시되며 파티클을 생성하지 않습니다.

start_frame

파티클이 방출되기 시작하는 프레임. mpm_solver_settings 컴파운드에서 use_master_start_frame을 선택하면 이 옵션이 재지정됩니다.

use_end_frame

선택하면, 파티클 방출이 end_frame 뒤로도 계속됩니다.

end_frame

파티클 방출이 끝나는 프레임. start_frame과 같을 경우, 파티클들이 순간적으로 방출됩니다.

Particle Settings

distribution

파티클을 입력 geometry 중 어디에서 얻을 것인지를 지정합니다.

- Surface: 바깥 레이어 또는 서피스에서만 방출됩니다.
- Volume: 내부 또는 폐쇄형 볼륨에서 방출됩니다.

particles_per_voxel

한 복셀에 배치되어야 하는 평균 파티클 개수로, 여기서 복셀의 사이즈는 mpm_solver_settings의 detail_size에서 주어집니다. 값이 클수록 시뮬레이션 되는 파티클의 개수가 증가해 디테일이 추가되지만 움직임은 복셀화된 볼륨을 통해 컴퓨팅되기 때문에 이 값을 높이더라도 전체 움직임의 해상도는 그대로 유지될 것입니다. 또, 파티클 개수와 상관이 없는 이유로 detail_size를 줄여야 하는 경우, 예를 들어, 얇은 콜라이더 주변의 움직임을 더욱 자세히 표현하고 싶지만 파티클 개수를 늘리고 싶지는 않을 때는 이 값을 낮추는 것이 유용합니다.

particle_display_scale

파티클을 더 크게 또는 더 작게 만듭니다. 이것은 출력 point_size의 곱수로 작용하지만 시뮬레이션에는 영향을 주지 않습니다. 파티클의 유효 시뮬레이션 사이즈를 변경하려면, mpm_solver_settings의 detail_size 특성을 이용합니다.

Snow Properties

mass_density

입방미터당 질량의 양(kg). 값이 클수록 눈이 더 무거워지고 더 가벼운 재질을 마음대로 다룰 수 있게 됩니다.

viscosity

근처 포인트들의 움직임에 대한 평균을 얼마만큼 낼 수 있는지를 의미합니다. 이것은 동점도(제곱미터/초)입니다.

volume_preservation

눈이 부피 대 모양을 유지하는 수준. 이 값은 단위가 없고 범위는 0부터 1까지입니다. 1은 모양이 전혀 유지되지 않음을(눈이 액체처럼 반응) 의미하고, 0은 부피가 전혀 유지되지 않음을 의미합니다. 대부분의 솔리드의 경우, 0.5가 적절하며 이 값에서는 모양과 부피가 모두 똑같이 유지됩니다. 예를 들어, 재질 한 덩어리의 두 반대 면을 밀 때, 보통 서로 다른 이 네 면은 바깥으로 휘어 전체 부피를 유지하지만, volume_preservation = 0일 때는 재질이 전혀 늘어나지 않고 모양이 변할 때 뒤로만 밀립니다. 이 매개변수는 공학에서 흔히 사용되는 포아송 비(Poisson Ratio)와 관련이 있습니다.

vibration_speed

진동이 눈을 통과하는 속도(m/s)로, 이 값은 양수입니다. 값이 클수록 강연도가 커집니다. 이것은 눈 안에서의 음속이며 기압파의 자연 속도입니다. 보통 값이 클수록 컴퓨팅 시간이 길어지므로 가장 작은 값을 이용하는 것이 결과가 좋습니다. 눈이 눈에 보일 정도로 튕기지 않는다면, 진동이 한 프레임 내에서 눈을 충분히 통과할 수 있을 만큼 진동 속도가 커야 합니다.

friction

모래가 미끌어질 때 자신에 대한 저항력으로 압축 정도에 대한 비입니다. 값이 클수록 주어진 압축 정도에서 미끄러짐에 대한 저항력이 더 커집니다. 이것을 내부 마찰 계수라고도 합니다. 음의 값이어서는 안 됩니다. 실제 값은 보통 0부터 1까지이나 더 큰 값도 지원됩니다.

cohesion

모래가 서로 뭉치거나 응집하려는 경향성. 값의 단위는 없습니다. 0은 기본적으로 마른 눈이며, 값이 클수록 모래가 더 많이 뭉칩니다. 여기에 현재 point_firmness가 곱해져 눈이 흩어지는 것(팽창에 항복)에 대한 저항성이 정해집니다. 괄호 안의 값은 source_mpm_sand의 cohesion 입니다.

initial_firmness

눈이 처음에 뭉치는 것(압축에 항복)에 대한 저항성을 가리킵니다(m2/s2). 항복이란 눈이 모양과 부피의 변화에 저항할 수 있는 힘의 한계에 도달한 것으로, 이렇게 되면 자신에게 가해지는 힘에 따라 추가적인 저항 없이 변형되거나 흘러갑니다. point_firmness 값은 시뮬레이션의 과정을 따라 진전하는데, hardening과 눈이 뭉치거나 흩어지는 양에 따라 달라집니다. 이것은 양 또는 음의 값으로 설정할 수 있습니다. 음의 값은 눈이 압축에 저항하기 전에 더 많이 뭉쳐져야 하는 것을 의미합니다.

hardening

뭉침(압축에 항복)과 흩어짐(팽창에 항복)이 경도에 미치는 영향을 제어합니다. 값의 단위는 없습니다. 값이 클수록 눈을 뭉칠 때 더 빠르게 단단해지고 흩어질 때 더 빠르게 약해집니다. 즉, 값이 클수록 더 쉽게 흩어지는 눈이 만들어지고 값이 작을수록 연성이 커집니다. 재질이 겪고 있는 압축의 정도와 재질이 저항할 수 있는 압축의 정도(단단함)의 차를 계산하고 여기에 경화(hardening)를 곱하여 경도(firmness)에 더합니다.

음의 값은 특수한 경우로 취급하며 의미가 전환되어 뭉침이 경도를 높이지 않고 낮춥니다. 경도(firmness)가 0이 되면, 눈이 완전히 흩어져 마른 눈으로 분리되는 것으로 간주하며, hardening이 절대 값으로 설정되고

cohesion이 0으로 설정됩니다. 이런 성질을 이용해 눈에 위크 레이어(weak layer)를 만들 수 있는데 이 레이어가 붕괴되면 원격 촉발 눈사태가 발생합니다. 이를 위해서는 시뮬레이션을 중력 영점에서 시작해 천천히 높여 메인 시뮬레이션에서 눈이 안정되도록 함으로써 위크 레이어가 너무 일찍 붕괴되지 않도록 설정합니다.

initial_speed

파티클의 초기 속도. 여기에 direction이 곱해지고 inherit_velocity 특성의 모든 속도에 더해집니다. 값의 단위는 m/s입니다.

direction

파티클의 초기 속도 방향. 이것은 initial_speed 로 스케일링 되어 inherit_velocity 특성의 모든 속도에 더해집니다.

inherit_velocity

geometry 입력의 속도 비로, 파티클의 초기 속도에 더해집니다.

set_particle_properties

set_property 노드를 연결해 시뮬레이션 되는 파티클에서 추가 특성을 정의하고 초기화합니다. 이런 특성들은 시뮬레이션 도중 influence_fields를 이용해 조정할 수 있습니다.

point_orientation은 솔버에 의해 업데이트되는데, 단, 그것이 float4에 저장된 4원수인 경우에 한 합니다.

상위 페이지: mpm

split_particles_by_material

granular_particles 출력을 각 재질 당 1개씩 개별 포인트 클라우드로 분리합니다.

상위 페이지: mpm

particles

이 섹션의 페이지들

- basic_particles_graph
- get_normalized_age
- get_particle_solver_properties
- get_particle_solver_settings
- get_particles
- particle_solver_settings
- property_kill_points
- set_orientation_value
- set_particle_property_from_age
- set_particles
- set_spin_value
- simulate_particles
- source_particles

상위 페이지: 참조

basic_particles_graph

basic_particles_graph는 단순 기본 그래프를 설정하는데, 중력의 영향을 받고 1초 후에 사라지는 파티클이 이 그래프에서 방출됩니다.

get_normalized_age

이것을 이용하면 파티클이 생성될 때 0.0살이고 사라질 때 1.0살인 파티클 시스템의 나이 값들로 구성된 출력 어레이를 얻을 수 있습니다. 이것은 음영 특성을 설정하거나 2차 방출을 다룰 때와 같은 상황에 유용합니다.

Inputs

particles

이것은 simulate_particles 컴파운드 또는 캐싱된 파티클들의 출력 파티클입니다. point_age와 point_age_limit의 포인트 지오메트리 특성을 가진 객체라면 모두 작동합니다.

Output

normalized_age

이것은 파티클들의 정규화 된 나이 값들(0.0~1.0 범위)로 구성된 float 어레이입니다.

get_particle_solver_properties

이것은 <u>simulate_particles</u> 컴파운드 편집에 사용됩니다. 이것을 이용하면 파티클 솔버 객체에서 모든 특성에 손쉽게 액세스할 수 있습니다.

simulate_particles 노드를 편집 가능하게 만든 경우, 다소 깊이 위치해 있어 찾아 들어가야 하는 사용자 작업용 컴파운드가 다음과 같이 생성됩니다. simulate_particles: solve_particles: advance_one_frame: advance_one_timestep: custom_effects. 여기서 파티클 시스템을 직접 수정할 수 있는 컴파운드를 구축할 수 있습니다.

Inputs

solver_object

이것은 simulate_particles 컴파운드 안에 있는 내부 solver_object에 연결해야 합니다.

Output

particles

이것은 모든 파티클들을 포함하고 있는 파티클 시스템 객체로, simulate_particles 컴파운드에서 최종적으로 출력됩니다.

solver_settings

이것은 <u>simulate_particles</u>로 전달되는 파티클 솔버 설정입니다. 이것을 <u>get_particle_solver_settings</u> 컴파운드에 연결해 개별 설정을 불러올 수 있습니다. 사용자 특성을 솔버로 손쉽게 전달하려면, <u>particle_solver_settings</u>와 <u>simulate_particles</u> 사이에 set_property 콜(call)을 추가하면 됩니다. 그런 다음, 솔버 설정에서 get_property를 해제하고 사용자 특성을 불러올 수 있습니다. 이로써 포트들이 끝까지 작동하는 상황을 방지할 수 있습니다.

sources

이것은 simulate_particles 로 전달되는 소스들의 어레이입니다.

colliders

이것은 simulate_particles 로 전달되는 콜라이더들의 어레이입니다.

fields

이것은 simulate_particles 로 전달되는 influence 필드들의 어레이입니다.

rurrent_time

이것은 현재 서브스텝 내에서 현재의 솔브(solve) 타임(초)입니다.

step_length 서브스텝의 길이(초)

frame_length 한 프레임의 길이(초)

frame_end_time 현재 프레임의 처리가 종료되는 시간 after_start_frame

simulate가 시작 프레임 시간 이후인 경우 이것은 true입니다.

get_particle_solver_settings

파티클 솔버 설정 객체의 개별 특성에 액세스하도록 허용합니다.

Inputs

solver_settings

이것은 particle_solver_settings 객체에 연결해야 합니다.

Output

gravity_enabled

particle_solver_settings의 중력 특성

gravity_vector

particle_solver_settings의 중력 벡터 특성

simulation_speed

particle_solver_settings의 시뮬레이션 속도 특성

start_frame

이 컴파운드를 simulate_particles 안에서 이용할 때는 시뮬레이션에서 이용할 작동 시작 프레임 특성이며, 이용하지 않을 때는 master_start_frame 값입니다.

substeps

particle_solver_settings의 sub_steps 특성

get_particles

simulate_particles 컴파운드 안에서 편집할 때, 이것을 이용해 파티클 솔버 객체의 파티클 객체에 액세스할 수 있습니다.

Inputs

solver_object

이것은 simulate_particles 컴파운드 안에 있는 내부 solver_object에 연결해야 합니다.

Output

particles

이것은 모든 파티클을 포함하고 있는 파티클 지오메트리인 동시에 simulate_particles 컴파운드의 출력이기도 합니다.

particle_solver_settings

사용자가 simulate_particles 컴파운드에서 이용할 글로벌 매개변수를 지정하도록 허용합니다.

이것은 simulate_particles 컴파운드의 설정(settings) 포트에 연결해야 합니다. simulate_particles에 연결된 설정이 없는 경우에는 설정 기본값이 사용됩니다.

Inputs

simulation_bounds

시뮬레이션 범위를 공간 내 한 영역으로 제한하는 데 이용됩니다. 사용되는 바운드는 이 포트에 연결된 지오메트리의 축 정렬 바운딩 박스입니다. 아무것도 연결되지 않으면 바운드도 존재하지 않습니다. 파티클들은 이 바운딩 박스 밖으로 방출되지 않으며 이 박스 밖으로 나오는 파티클은 모두 사라집니다.

use_master_start_frame

개별 이미터와 콜라이더 상의 시작 프레임을 particle_solver_settings의 master_start_frame으로 재지정할 수 있습니다.

master_start_frame

use_master_start_frame이 활성화되면, 모든 이미터와 콜라이더의 글로벌 스타트 프레임으로 master_start_frame이 사용됩니다.

gravity

이것이 활성화되면, 시뮬레이션이 particle_solver_settings에서 gravity vector 를 이용해 모든 파티클에 적용할 고정 가속을 결정합니다.

gravity_vector

이것은 모든 파티클에 적용되는 고정 가속 벡터입니다. 기본값(0, -9.8, 0)은 장면의 단위가 미터(m)인 지구 중력의 시뮬레이션을 제공합니다.

sub_steps

이것은 각 프레임에서 산출된 시뮬레이션 스텝의 수입니다. 경우에 따라, 1보다 큰 값은 충돌 문제를 더 효과적으로 해결하기도 하며, influence 필드를 더 정확히 따르게 하거나, 빠르게 움직이는 이미터를 다루는 데 효과적입니다.

simulation_speed

감지한 시뮬레이션의 시각 속도에 적용되는 스케일입니다. 1보다 큰 값은 시뮬레이션 속도를 높이고 1 미만인 값은 시뮬레이션 속도를 낮춥니다.

shape

드로우(draw)에 사용할 모양:

- point: 단순한 1 픽셀 포인트
- disk: 꽉 채워진 원
- circle: 1줄 원
- cube: 속이 꽉 찬 정육면체
- sphere: 속이 꽉 찬 구

• <u>numeric</u>: 이것을 이용하면 진단을 목적으로 파티클 특성들을 수치로 나타낼 수 있습니다. 표시될 특성은 <u>numeric_property</u> 문자열로 지정합니다.

point와 numeric을 제외한 모든 모양은 파티클의 point_size 특성을 이용해서 월드스페이스 사이즈를 결정합니다.

numeric_property

이것은 모양을 numeric으로 설정했을 때 이용할 수 있는 파티클 특성입니다.

face_camera

이것이 비활성화되면, disk, circle, quad 모양은 파티클의 point_orientation(회전)을 이용하고, 그렇지 않으면, 이 세 모양은 회전하여 항상 카메라 쪽을 향하게 됩니다.

Output

solver_settings

출력은 모든 설정 특성을 포함하고 있는 객체이며 simulate_particles 노드의 settings 포트에 연결해야 합니다.

property_kill_points

이것을 이용하면 임계값과 함께 지정된 포인트 특성을 기준으로 포인트 객체에서 포인트들을 제거할 수 있습니다. 이것은 한 시스템에서 다른 시스템으로 파티클을 방출하고 있는 상황에 유용합니다. 예를 들어, 2차 파티클 시스템의 소스로 전달되기 전에 정해진 나이 미만의 모든 포인트들을 제거하여 오래 된 파티클들만 방출되도록 할 수 있습니다.

Inputs

object

이것은 포인트들을 제거할 포인트 지오메트리입니다.

property

이것은 제거할 포인트를 조정할 때 이용하는 특성의 이름입니다. 파티클 시스템의 point_age나 point_size처럼 float 어레이 geo 특성입니다.

threshold

지정된 특성이 이 임계값을 초과하면 포인트가 제거됩니다.

reverse

이것이 활성화되면, 지정된 특성이 임계값 미만일 때 포인트가 제거됩니다.

Output

solver_settings

출력은 모든 설정 특성을 포함하고 있는 객체로, 이것을 simulate_particles 노드의 settings 포트에 연결해야 합니다.

set_orientation_value

이것을 이용해 Euler 각(도)에서 4원수 로테이션 값을 설정할 수 있습니다. 이것은 vary_source_property를 이용할 때처럼, point_orientation을 설정할 때 유용합니다.

Inputs

degrees

Euler 로테이션(도)

Order

이것은 각 축을 따라 로테이션이 진행되는 순서를 지정합니다.

- XYZ
- YZX
- ZXY
- XZY
- YXZ
- ZYX

Output

quaternion

이것은 정해진 로테이션을 나타내는 4원수 벡터입니다.

set_particle_property_from_age

이것은 <u>simulate_particles</u> 컴파운드의 하류 단계에서 정규화된 파티클 나이를 기준으로 특성을 설정하는 데 이용할 수 있습니다. 예를 들어, 파티클 나이에 따라 point_size나 색상을 설정할 수 있습니다. 생성(birth)과 사라짐(death)에 정의된 값들은 파티클 나이를 기준으로 보간됩니다.

Inputs

particles

이것은 <u>simulate_particles</u> 컴파운드, 캐싱된 파티클, 또는 <u>point_age</u>와 <u>point_age_limit</u> 특성을 지닌 모든 객체에서 연결되어야 합니다.

property

설정할 특성. 필요 시 생성됩니다.

birth_value

파티클 나이가 0일 때 이용하는 특성의 값. 이 특성의 유형은 해당 특성의 유형과 일치해야 하는데, 이를 테면, color에는 float3을 이용해야 하는 것과 같습니다.

death_value

파티클 나이가 point_age_limit 일 때 이용하는 특성의 값. 이 특성의 유형은 해당 특성의 유형과 일치해야 하는데, 이를 테면, color에는 float3을 이용해야 하는 것과 같습니다.

Output

out_particles

출력은 특성이 추가되었거나 수정된 파티클 객체입니다.

set_particles

simulate_particles 컴파운드 안에서 편집할 때, 이것을 이용해 파티클 솔버 객체의 파티클 특성을 설정할 수 있습니다.

Inputs

solver_object

이것은 simulate_particles 컴파운드 안에 있는 내부 solver_object에서 연결되어야 합니다.

particles

이것은 모든 파티클을 포함하고 있는 파티클 객체에 연결되어야 합니다.

Output

out_solver_object

이것은 simulate_particles 컴파운드 안에 있는 내부 solver_object에 연결되어야 합니다.

set_spin_value

이것을 이용해 입력 Euler 로테이션(도)을 기준으로 스핀 축의 각도 값을 정의할 수 있습니다. 이것은 vary_source_property의 스핀(spin)을 설정할 때 유용합니다.

Inputs

degrees

Euler 로테이션(도)

Order

이것은 각 축을 따라 로테이션이 진행되는 순서를 지정합니다.

- XYZ
- YZX
- ZXY
- XZY
- YXZ
- ZYX

Output

axis_angle

출력은 로테이션의 float4 축의 각도 값으로, 앞의 3 구성요소는 xyz 축 방향이고 4번째 구성요소는 라디안에서 축을 기준으로 한 로테이션 값입니다.

simulate_particles

이것은 메인 파티클 시스템 솔버 컴파운드입니다. 파티클을 방출한 후 나이가 들면 제거하고, 파티클을 객체에 충돌시키고, 필드를 적용하는 역할을 합니다.

대부분의 상황에서 이 노드를 블랙 박스로 삼을 수도 있는데, 물론, 고급 워크플로에서는 상황에 따라 이 노드를 편집 가능한 상태로 만들 수도 있습니다. 사용자 작업은 대부분 그래프의 서브스텝 루프 안에서 이루어져야 합니다. 이 컴파운드를 편집 가능하게 만든 후, 출발 지점을 다음과 같이 설정합니다. Simulate_particles: solve_particle: advance_one_frame: advance_on_timestep: custom_effects. 이 노드 안에서 원하는 파티클 특성에 손쉽게 액세스해 수정할 수 있습니다.

Inputs

sources

시뮬레이션으로 방출하려는 위치와 대상을 정의하는 source_particles 노드에서 출력되는 객체들의 어레이입니다.

collisions

파티클이 충돌하게 되는 collider 객체들의 어레이입니다.

influence_fields

힘과 미적 컨트롤을 정의하는 influence 필드들의 어레이입니다.

settings

particle_solver_settings 노드의 출력 객체에 연결되어야 하며, 이는 파티클 솔버의 전체 설정 항목들을 구성합니다.

Output

particles

시뮬레이션에서 산출된 파티클. 이것을 bifrostBoard 출력에 연결해 최종 파티클을 확인할 수 있습니다. 또는, 또 다른 파티클의 지오메트리나 공기 소스 노드에 연결할 수도 있습니다.

source_particles

이것을 사용하면 파티클을 simulate_particles 컴파운드로 방출할 수 있습니다.

이 노드를 이용하려면, 지오메트리 객체를 지오메트리 입력에 연결한 다음, source_particles.particle_source 출력을 simulate_particles 컴파운드의 sources 입력에 연결합니다.

Inputs

geometry

소스 지오메트리를 정의하는 입력 메쉬, 포인트, 볼륨들의 어레이

General

소스의 타임 바운드와 비활성화를 위한 전체 특성

start_frame

소스의 방출이 시작되는 프레임입니다.

use_end_frame

이것을 활성화하여 특정 프레임에서 방출이 멈추도록 설정할 수 있습니다. 활성화하지 않으면 시뮬레이션 내내 방출이 계속됩니다.

end_frame

소스의 방출이 끝나는 프레임입니다.

Spawning Properties

이런 컨트롤은 지오메트리 소스의 방출에 영향을 줍니다.

distribution

이것은 지오메트리에서 파티클들이 분포되는 방식을 제어합니다.

- Surface: 입력 지오메트리의 서피스에서 방출
- Volume: 입력 지오메트리의 내부에서 방출. 이때, 지오메트리를 복셀화해야 하며, volume_detail_size 특성을 이용해 바운딩 박스에 비례하여 얼마나 정밀하게 복셀화할 것인지를 결정합니다. 또, 이 컴파운드를 편집 가능한 상태로 만든 다음 안으로 들어가서 복셀화를 위해 geo_volume_mode와 geo_volume_offset을 설정할 수 있습니다.

volume_detail_size

이것은 Volume distribution 메소드를 위해 지오메트리를 복셀화할 때 이용하는 바운딩 박스 기준 복셀 사이즈입니다.

rate

이것은 rate_mode에 따라 각 스텝에서 방출되는 파티클의 수를 제어합니다.

rate_mode

이것은 rate가 정의되는 방식을 제어합니다.

• Density: 프레임당 파티클 개수는 월드스페이스 단위 면적(서피스 분포의 경우)과 단위 부피(볼륨 분포의 경우)를 기준으로 합니다. 일반적으로 rate 모드를 density로 바꿀 때는 rate를 먼저 낮추는 것이

BIFROST

좋은데, 매우 큰 객체는 공기의 양이나 부피가 상당히 커서 파티클 개수가 너무 많아질 수 있기 때문입니다.

- Count: rate는 각 프레임에서 방출되는 평균 파티클 개수를 나타냅니다.
- Count Per Point: 서피스나 볼륨이 아닌 정점에서 파티클이 방출됩니다. rate는 프레임마다 각 정점에서 방출되는 파티클의 개수입니다. 이 모드로 전환하는 경우, 메쉬에 정점이 너무 많은 경우에는 최종 파티클이 너무 많아질 수 있기 때문에 rate를 먼저 낮춰야 할 수 있습니다.

Particle Properties

이 컨트롤은 방출 시 각 파티클 특성의 초기 값에 영향을 줍니다.

speed

이것은 파티클의 초기 속도입니다. 이 값은 normal_speed와 direction 벡터 모두에 영향을 줍니다. 일반 속도의 단위는 유닛/초입니다.

direction

이것은 월드스페이스에서 고정 방향을 따르는 방출 속도의 구성요소입니다.

normal_speed

이것은 지오메트리 노멀을 따르는 방출 속도의 구성요소입니다. 지오메트리에 point_normal이 없다면, 이것은 모든 방향으로 방출됩니다.

spread

이것은 방출 방향에 영향을 줍니다. 값이 0이면 정해진 방향이나 노멀을 따릅니다. 값이 0.5이면 180도 범위에서 이 방향을 임의로 동요시킵니다. 값이 1.0이면 완전한 임의 방출 방향이 됩니다.

inherit_velocity

이것은 방출 속도에 소스 지오메트리의 속도가 더해진 값입니다. 값이 1.0이면, 이미터가 파티클에 최대 객체 속도를 추가합니다.

bounciness

이것은 파티클의 충돌 탄력성으로, 파티클이 얼마나 많이 튕길 것인지를 결정합니다. 콜라이더와 파티클의 탄력성이 모두 1.0인 경우, 완벽한 탄력 충돌을 일으켜 파티클들이 계속해서 같은 높이로 튕기고, 따라서 drag가 없는 상태가 됩니다. 값이 1.0보다 크면 물리적으로 불가능한 충돌을 일으킵니다.

live_forever

이것이 활성화되면, influence_field로 파티클을 제거하지 않는 한 age_limit가 무시되어 파티클이 사라지지 않습니다.

age_limit

이 시간(초)이 지나면 파티클이 사라지는 파티클의 나이

size

파티클 point_size에 설정하는 초기 값으로, 충돌(collision)과 당김(drawing)에 이용합니다.

orientation

파티클 로테이션을 위해 설정하는 초기 값

spin

파티클에 가해지는 스핀 속도

set_particle_properties

이 객체 포트를 이용해 각 파티클 특성을 개별적으로 추가할 수 있습니다. 전달되는 객체의 단순 특성은 모두 파티클 시스템의 어레이 지오메트리 특성으로 바뀝니다. 객체에 전달되는 값은 파티클이 방출 시 얻는 값이 됩니다. 예를 들어, set_property 노드를 만들고, 키(key)를 color 로 설정하고, 값을 float3으로 설정한 다음, 이를 set_particle_properties에 연결합니다. 이로써 파티클이 색상 특성을 갖게 됩니다.

Output

particle_source

출력은 파티클 솔버에 대한 소스 관련 입력의 총합을 포함하는 객체로, 이를 simulate_particles 노드의 sources 포트에 연결해야 합니다.

simulation

이 섹션의 페이지들

- collider
- compute_point_velocities
- generate_pulse
- should_simulate
- simulation_example
- vary_source_property

상위 페이지: 참조

collider

콜라이더(Collider)는 Bifrost 시뮬레이션 흐름의 장애물입니다. 따라서, 콜라이더를 이용해 연기가 흘러가야 하는 길에 장벽을 세우거나, 파티클이 중력에 의해 떨어지는 것을 막아주는 받침을 만들어놓거나, 애니메이트해서 각종 효과를 연출할 수 있습니다.

이 노드를 이용하려면, 이 노드의 출력을 <u>simulate_aero</u>, <u>simulate_particles</u>, <u>simulate_mpm</u> 등과 같은 <u>simulate</u> 노드의 <u>colliders</u> 입력에 연결합니다.

Inputs

geometry

장애물 역할을 할 기하학적 객체를 1개 이상 연결합니다. 포인트 객체, 메쉬, 볼륨 중 아무것이나 가능합니다.

General

이 설정은 모든 메소드에 적용됩니다.

enable

해제하면 충돌이 발생하지 않습니다.

method

충돌이 볼륨과 메쉬 중 무엇을 기준으로 할 것인지를 가리킵니다.

- Automatic: 공기와 연소 시뮬레이션일 경우 볼륨 기준 충돌을, 파티클과 MPM 시뮬레이션일 경우에는 메쉬 기준 충돌을 각각 이용합니다.
- Volume: 공기, 연소, 파티클 시뮬레이션에 이용할 수 있습니다. 입력 메쉬가 볼륨으로 자동 변환됩니다. 입력 볼륨에는 레벨 세트가 있어야 합니다(voxel_signed_distance).
- Mesh_For_Points: 파티클과 MPM 시뮬레이션에 이용할 수 있습니다. 이 메소드에서는 입력 볼륨을 지원하지 않습니다.

start_frame

충돌이 발생하는 첫 번째 프레임

use_end_frame

지정된 end_frame에서 충돌이 멈춥니다.

end_frame

시뮬레이션의 마지막 프레임

Common Properties

이 설정은 볼륨 기준 충돌과 메쉬 기준 충돌에 모두 적용됩니다.

detail_size

시뮬레이션에서 최소 분해능으로 나타낼 수 있는 디테일 사이즈

offset

지오메트리 서피스로부터의 충돌 거리(월드 스페이스 단위). 메쉬 기준 파티클 충돌의 경우에는 음의 값을 지원하지 않습니다.

inherit_velocity

충돌로 전달되는 콜라이더 지오메트리의 애니메이트된 움직임의 비. 값이 0.0이면 상속이 이뤄지지 않음을 의미합니다. 메쉬 기준 파티클 충돌의 경우에는 이 기능을 지원하지 않습니다.

Volume Collider Properties

이 설정은 볼륨 기준 충돌에 적용됩니다.

resolution_mode

볼륨 해상도를 좌우하는 메소드

- Absolute: detail_size는 월드 스페이스 단위입니다.
- Relative: detail_size가 충돌 지오메트리의 사이즈에 비례합니다.

geo_mode

입력 메쉬가 볼륨으로 변환되는 방식을 정의합니다.

- Solid: 최종 볼륨은 입력 메쉬의 솔리드 인테리어를 나타냅니다.
- Shell: 최종 볼륨은 메쉬의 두꺼워진 서피스만을 나타냅니다.

Mesh Collider Properties

이 설정은 메쉬 기준 충돌에 적용됩니다.

bounciness

충돌에 반영될 충돌 서피스에 대한 속도 노멀의 비. 이것은 예를 들어, 파티클이 땅에 부딪힌 다음 튀어 오르는 높이에 영향을 줍니다. 최종 값은 파티클과 콜라이더에 설정된 bounciness의 조합입니다. 두 값 모두 1.0인 경우에는 파티클이 완벽히 튀어 오르고 두 값 모두 0.0인 경우에는 전혀 튀어 오르지 않습니다.

friction

충돌 시 손실되는 충돌 서페이스의 속도 비. 예를 들어, 이 값이 1.0이면 파티클이 땅에 부딪힌 다음 굴러가지 않지만, 값이 0.0이면 파티클이 속도 저하 없이 굴러가게 됩니다.

roughness

반사각을 달리하면서 거친 표면에서의 충돌을 에뮬레이션 합니다. 값이 0.0이면 표면이 완전히 매끄러워집니다.

Output

collider 장애물 역할을 하는 콜라이더의 기하학적 객체들의 어레이

상위 페이지: 시뮬레이션

compute_point_velocities

특정 포인트 지오메트리에 대한 두 프레임 간의 속도를 산출해 그 결과를 지오메트리에 저장합니다.

Inputs

geometry

지오메트리(포인트 기반 객체) 어레이를 연결합니다. 포인트 객체나 메쉬 중에서 선택할 수 있습니다. 프레임 사이에서 포인트 개수는 변할 수 없습니다.

after_start

속도 계산에 피드백 포트를 사용할 것인지의 여부

frame_length 한 프레임의 시간(초)

Output

out_geometry

point_velocity라는 이름의 특성을 가진 입력 지오메트리 어레이

상위 페이지: 시뮬레이션

generate_pulse

이것은 소스에 펄스 속성이 포함되는 대신 파티클 특성에서 펄싱(pulsing) 애니메이션을 만드는 유틸리티입니다(그 밖의 용도에도 사용 가능).

Inputs

interval

두 펄스 간의 시간

on_value

펄스가 활성화될 때의 값

off_value

펄스가 해제될 때의 값

상위 페이지: 시뮬레이션
should_simulate

현재 시각이 유효한 시뮬레이션 스텝인지 여부에 따라 입력 object 또는 prev_object를 반환합니다. 이 노드의 기본 용도에 대해서는 simulation_example을 참조하십시오.

Inputs

object

시뮬레이션 컴퓨팅의 객체 또는 객체들의 어레이를 연결해 해당 시뮬레이션 스텝이 유효한지 여부를 반환합니다.

prev_object

객체 또는 객체들의 어레이를 연결해 시뮬레이션 스텝이 유효한지 여부를 반환합니다(예: 이전 시뮬레이션 컴퓨팅의 결과).

start_frame

시뮬레이션이 시작되는 프레임

backward 되돌아가 시뮬레이션 할 것인지의 여부

Output

out_object 입력 객체(단, 모든 조건을 충족하는 경우)

상위 페이지: 시뮬레이션

simulation_example

should_simulate의 기본 용도

Inputs

object

시뮬레이션 할 객체 또는 객체들의 어레이를 연결합니다.

prev_object

직전 프레임에서 시뮬레이션 된 객체. 이 포트에는 아무것도 연결할 필요가 없습니다. 시스템이 직전 프레임에서 해당 객체를 자동으로 복사합니다.

start_frame

시뮬레이션이 시작되는 프레임

Output

out_object

조건 충족 여부에 따라 시뮬레이션 된 객체 또는 이전 객체

상위 페이지: 시뮬레이션

vary_source_property

이 컴파운드는 시뮬레이션 소스에서 각 포인트의 값들을 설정하는 데 유용합니다. 예를 들어, 이 컴파운드를 이용해 파티클 시뮬레이션에서 생성되는 포인트의 사이즈를 다양하게 만들 수 있습니다.

이 컴파운드는 소스 특성을 임의로 바꿀 수 있는데, 그렇지 않으면, 장면 내 입력 지오메트리에 존재하는 컬러 세트를 이용할 수도 있습니다. randomize_value와 use_color_set 옵션을 이용해 이용할 메소드를 선택합니다. 두 옵션을 모두 활성화하면, 맵이 무작위화를 위한 곱수가 됩니다. 두 옵션 모두 활성화하지 않으면, 소스 노드의 값이 사용됩니다. 컬러 세트를 이용하려면, 호스트 장면에서 해당 이름을 지정합니다.

이 컴파운드는 입력 지오메트리와 소스 노드 사이에 연결할 수 있습니다. 소스에 구애 받지 않고 대부분의 float, float3, float4 특성에서 작동합니다. 주어진 소스 특성에 이용할 경우, 특성의 소스 노드 상의 값은 무시됩니다. vary_source_property 노드는 몇 개든 데이지 체인 방식으로 연결할 수 있습니다.

Inputs

geometry

소스 역할을 할 기하학적 객체를 1개 이상 연결합니다. 포인트 또는 메쉬 지오메트리 중에서 선택할 수 있습니다.

General

이 설정은 모든 메소드에 적용됩니다.

enable_variation

해제하면 변화(variation)가 생기지 않습니다.

property

변화할 수 있는 소스 특성의 이름으로, 소스 자체에 표시됩니다. 다음은 전체 목록입니다.

- <u>source_particles</u> : "rate", "speed", "direction", "normal_speed", "spread", "inherit_velocity", "bounciness", "age_limit", "size".
- source_air : "fog_density", "temperature", "inherit_velocity", "initial_speed", "initial_speed_direction".
- source_fuel : "ignition_temperature".
- source_mpm_sand : "volume_preservation", "viscosity", "vibration_speed", "friction", "cohesion",
 "initial_speed", "inherit_velocity".
- <u>source_mpm_snow</u> : "initial_firmness", "hardening", "volume_preservation", "viscosity", "vibration_speed", "friction", "cohesion", "initial_speed", "inherit_velocity".
- source_mpm_fluid : "surface_tension", "viscoelasticity", "relaxation_time", "viscosity", "vibration_speed",
 "initial_speed", "inherit_velocity".

소스 컴파운드로 들어가 소스 매개변수과 관련된 특성의 이름을 확인할 수 있습니다. 가령, 사이즈가 실제로는 point_size인 것을 확인할 수 있습니다. "size" 또는 "point_size"는 파티클의 사이즈를 변경합니다.

Random Variation

이 설정은 한 특성의 무작위화에 적용됩니다.

randomize_value

특성 값을 무작위화 할 것인지의 여부

min

특성의 최소 값. 유형은 소스 특성의 유형과 일치해야 합니다. 예를 들어, size는 float이고 initial_speed_direction은 float3입니다.

max

특성의 최대 값. 유형은 소스 특성의 유형과 일치해야 합니다.

bias

범위는 -1 ~ 1입니다. -1인 경우, min 값이 훨씬 더 큽니다. 0에서는 min과 max가 비슷한 수준입니다. 1에서는 max 값이 훨씬 더 큽니다.

animated

시간이 흐름에 따라 무작위화를 변화시킵니다.

Color Set Variation

이 설정은 특성에서 컬러 세트를 이용하는 경우에 적용됩니다. float 특성에서만 이용할 수 있습니다.

use_color_set

컬러 세트 특성 값을 이용할 것인지의 여부

color_set_name

입력 지오메트리에 설정된 색상의 이름. 와치포인트(watchpoint)를 추가하면 입력 지오메트리에 설정된 색상의 이름과 유형을 확인할 수 있습니다. 알파, RGB 또는 RGBA 채널을 색칠하여 컬러 세트를 float로 매핑할 수 있습니다.

multiplier

총 변화 효과의 최종 스케일링

Output

out_geometry 소스 역할을 하는 기하학적 객체들의 어레이

상위 페이지: 시뮬레이션

Bifrost Extension for Maya

Bifrost Extension은 Autodesk Maya용 플러그인입니다. 이 플러그인은 연기, 불, 폭발, 모래, 눈, 인스턴스 등 사용자 지정 효과를 연출할 수 있는 노드 기반 프레임워크를 제공합니다. 결과를 뷰포트에서 직접 미리 볼 수도 있고 Arnold for Maya로 렌더링할 수도 있습니다.



각종 샘플 그래프를 자기만의 효과를 만들기 위한 출발점으로 삼을 수도 있고, Bifrost에 대해 자세히 알아보기 위해 학습 대상으로 삼을 수도 있습니다.

Bifrost는 관련 업데이트 릴리스를 포함해 Maya 2018과 2019에서 모두 이용할 수 있습니다. 샘플 그래프는 모든 버전에서 실행됩니다.

🖋 참고:

만일, **Bifrost Graph**가 Maya의 **Create** 메뉴에 나타나지 않고 **Bifrost Browser**와 **Bifrost Graph Editor**가 Maya의 **Windows** 메뉴에 나타나지 않는다면, 이 플러그인이 로드되지 않은 것입니다. Bifrost Extension 플러그인 로드하기를 참조하십시오.

Bifrost FAQ

- Bifrost란?
- Bifrost는 누구를 위한 제품입니까?
- Bifrost는 어디서 구입할 수 있나요?
- Bifrost는 Bifrost Fluids와 같은 것입니까?
- 저는 Bifrost Fluids와 BOSS를 이미 사용하고 있습니다. Bifrost를 새로 설치하더라도, 이 기능들을 계속 사용할 수 있나요?
- Bifrost 그래프는 어떻게 렌더링하나요?
- Bifrost 라이센스는 어떻게 구할 수 있나요?
- Maya 멤버쉽에 가입하면 Bifrost를 무료로 이용할 수 있나요?
- Bifrost에 액세스할 수 있는 또 다른 방법이 있습니까?
- Bifrost에는 교육용 라이센스가 있나요?
- 영구 라이센스를 보유한 Maya LT 고객들도 Bifrost를 이용할 수 있나요?
- Bifrost에 필요한 최소 시스템 사양은 무엇인가요?
- Bifrost는 어떤 플랫폼을 지원하나요?
- Bifrost는 오토데스크가 아닌 타사의 3D 애니메이션 도구로 만든 3D 에셋을 지원하나요?
- Bifrost 커뮤니티 포럼은 어디서 찾을 수 있나요?
- Bifrost와 관련한 튜토리얼이나 기타 교육용 자료는 어디서 찾을 수 있나요?
- 도움이 될 만한 자료가 있나요?

Bifrost란?

Bifrost는 3D 아티스트와 TD를 대상으로 Maya에서 새로운 시각적 프로그래밍 환경을 이용해 복잡한 효과들을 쉽고 빠르게 연출할 수 있는 기능을 제공합니다. TD는 Bifrost Graph Editor에서 사용자 그래프를 만들어 아티스트들에게 패키지 형태로 제공하고, 아티스트들은 스튜디오에서 여러 가지 쇼, 장면, 샷에 그래프를 활용할 수 있습니다. 또, 아티스트들은 즉시 사용할 수 있는 다양한 그래프를 이용해 불, 연기, 폭발, 모래, 눈과 같은 효과들을 즉시 연출할 수 있습니다.

지금까지 Bifrost에는 다음과 같은 기능이 포함되어 있었습니다.

- FLIP (fluid implicit particle) 솔버를 이용해 시뮬레이션한 액체와 공기역학 효과 등을 연출할 수 있는 절차식 프레임워크 Bifrost Fluids.
- BOSS(Bifrost Ocean Simulation System).

Bifrost Extension for Maya로, 두 가지 기능이 추가되어 있습니다.

- 절차식 그래프를 만들어 모래, 불, 연기, 폭발 등의 효과를 연출할 수 있는 새로운 노드 기반 시각적 프로그래밍 환경 Bifrost Graph.
- 이런 절차식 그래프를 명령행에서 바로 실행하는 도구 bifcmd.

Bifrost는 누구를 위한 제품입니까?

Maya 아티스트와 TD.

Bifrost는 어디서 구입할 수 있나요?

Maya 2019.2 Update를 설치하면, Maya 설치관리자에 Bifrost Extension for Maya가 기본적으로 포함되어 있습니다. Maya의 모든 이전 버전에 대해서는, AREA에서 해당하는 설치관리자를 다운로드하십시오. Bifrost Extension for Maya는 Maya 2018과 Maya 2019 그리고 각각의 업데이트 릴리스에서 이용할 수 있습니다.

설치관리자를 선택할 때, Bifrost Extension for Maya 2019 설치관리자는 Maya 2019, 2019.1 업데이트, 2019.2 업데이트 버전에 모두 해당합니다. 마찬가지로, Bifrost Extension for Maya 2018 설치관리자는 Maya 2018과 이것의 업데이트 릴리스에 모두 해당합니다.

Bifrost는 Bifrost Fluids와 같은 것입니까?

Bifrost Extension for Maya는 기존의 Bifrost 기능인 Bifrost Fluids와 BOSS외에도, 절차식 그래프를 만들어 불, 연기, 폭발, 혹은 모래, 눈(snow) 등 다양한 효과를 연출할 수 있는 새로운 노드 기반 환경인 Bifrost Graph를 포함합니다. 그밖에, Maya와 관계없이 그래프를 실행할 수 있는 명령행 도구도 포함되어 있습니다.

현재, Bifrost Fluids와 Boss를 이용하고 있다면, 이들 기능을 Bifrost Extension for Maya와 함께 계속 사용할 수 있습니다.

저는 Bifrost Fluids와 BOSS를 이미 사용하고 있습니다. Bifrost를 새로 설치하더라도, 이 기능들을 계속 사용할 수 있나요?

물론입니다.

Bifrost의 구 버전 즉, Bifrost for Maya 2018과 Bifrost 1.5.x for Maya 2018/2019는 새 버전의 Bifrost Extension for Maya와 함께 설치할 수 있습니다. 이전 플러그인을 제거할 필요는 없지만 기본적으로 가장 최근에 설치한 버전만 로드됩니다. 만일, 다른 버전의 플러그인을 로드하고 싶다면, Maya에서 Plug-in Manager를 통해 해당 버전을 찾거나 bifrost.mod와 vnn.mod 모듈 파일을 편집해 원하는 버전을 가리키도록 하면 됩니다. .mod 파일은 아래 경로에서 찾을 수 있습니다.

- C:\Program Files\Common Files\Autodesk Shared\Modules\Maya (Windows)
- /usr/autodesk/modules (Linux)
- /Users/Shared/Autodesk/modules (macOS)

이전 플러그인 버전들을 제거할 경우, 새로운 Bifrost Extension for Maya 플러그인을 통해 Bifrost Fluids와 BOSS가 다시 설치됩니다.

좀 더 체계적으로 구성하기 위해 Maya 2018이나 이것의 업데이트 릴리스를 이용하고 있다면, Bifrost for Maya 2018 플러그인을 제거한 다음 Bifrost Extension for Maya 새 플러그인을 설치하고 싶을 수도 있는데, 이것은 이전의 Bifrost for Maya 2018 설치관리자가 .../Autodesk/Bifrost/Maya2018 디렉터리 안에 버전별 하위 폴더들을 생성하지 못했기 때문입니다.

Bifrost 그래프는 어떻게 렌더링하나요?

Maya 2019.2 Update는 Arnold로 Bifrost 콘텐츠를 렌더링할 수 있는 Arnold for Maya 버전을 포함하고 있습니다.

Maya 2019, Maya 2019.1 Update, Maya 2018, 혹은 Maya 2018 Update 릴리스 중 하나를 이용하고 있다면, Arnold for Maya Download Area에서 최신 Arnold for Maya 설치관리자를 다운로드할 수 있습니다.

Arnold for Maya는 최소 3.2.2 버전부터 절차식 그래프로 만든 Bifrost 콘텐츠를 렌더링할 수 있습니다.

Bifrost Fluids와 Boss를 렌더링할 때에는 이 최소 버전이 필요하지 않지만 그래도 최신 버전을 다운로드하는 것이 좋습니다.

Bifrost 콘텐츠를 직접 렌더링하는 것을 지원하지 않는 렌더러의 경우, 해당 콘텐츠를 Alembic이나 OpenVDB와 같은 표준 형식으로 내보내면 됩니다.

Bifrost 라이센스는 어떻게 구할 수 있나요?

Bifrost 라이센스는 플러그인 설치 시 동의해야 하는 EULA를 따릅니다. Maya 라이센스가 있다면 정식 Bifrost Extension for Maya 플러그인을 통해 Bifrost를 실행할 수 있고 그 밖에 세 가지 Bifrost Engine 라이센스 자격도 주어집니다.

bifcmd 도구를 이용해 Bifrost를 독립형 모드로 실행하면, Bifrost Engine 라이센스부터 적용됩니다.

Bifrost Engine 라이센스 1개는 한 시스템에서 bifcmd 작업을 무제한 진행할 수 있는 자격을 제공합니다. Bifcmd는 최대 3대의 컴퓨터에서 동시에 실행할 수 있는데, Maya 라이센스에 Bifrost Engine 라이센스 3개가 함께 제공되기 때문입니다.

다른 방법으로, ECP Collection 라이센스는 Bifrost Engine 라이센스를 15개 제공합니다.

회사에 Maya 라이센스가 10개 있다면, 최대 30대의 컴퓨터에서 bifcmd를 동시에 실행할 수 있습니다. ECP Collection 라이센스가 10개 있다면, 최대 150대의 컴퓨터에서 bifcmd를 동시에 실행할 수 있습니다. 이런 이용 권한들은 Bifrost Extension for Maya를 통해 Bifrost를 실행할 수 있는 권한에 추가로 포함됩니다.

예를 들어, 1개의 Maya 라이센스를 보유하고 있는 Maya 사용자는 다음과 같은 작업을 동시에 수행할 수 있습니다.

- Maya에서 Bifrost Extension for Maya 플러그인을 통해 Bifrost를 이용
- 3대의 다른 컴퓨터에서 bifcmd 작업 6가지를 계산

이때, 4대의 컴퓨터에서 Bifrost가 실행되는데, 1대는 Maya를 실행하고 나머지 3대는 bifcmd를 실행하지만 Maya 라이센스는 1개만 있으면 됩니다.

Maya 멤버쉽에 가입하면 Bifrost를 무료로 이용할 수 있나요?

위 'Bifrost 라이센스는 어떻게 구할 수 있나요?'를 참고하십시오.

Bifrost에 액세스할 수 있는 또 다른 방법이 있습니까?

물론입니다. AREA를 방문하면 현재 이용하고 있는 Maya 버전에 해당하는 Bifrost 최신 설치관리자를 다운로드할 수 있습니다.

Bifrost에는 교육용 라이센스가 있나요?

Bifrost 라이센스 중 교육용 라이선스는 별도로 없지만, 교육용 Maya 라이센스에 Bifrost가 포함되어 있습니다. 이 부분 역시 'Bifrost 라이센스는 어떻게 구할 수 있나요?'를 참고하십시오.

영구 라이센스를 보유한 Maya LT 고객들도 Bifrost를 이용할 수 있나요?

아닙니다.

Bifrost에 필요한 최소 시스템 사양은 무엇인가요?

Bifrost를 위한 최소 시스템 사양은 Maya의 경우와 같습니다.

Bifrost는 어떤 플랫폼을 지원하나요?

Bifrost Extension for Maya는 Windows, Linux, macOS 플랫폼에서 지원됩니다.

Bifrost는 오토데스크가 아닌 타사의 3D 애니메이션 도구로 만든 3D 에셋을

지원하나요?

메쉬는 Alembic 형식을 이용해, 볼륨과 포인트는 OpenVDB를 이용해, 볼륨은 Field3D를 이용해 각각 가져오거나 내보낼 수 있습니다.

Bifrost 커뮤니티 포럼은 어디서 찾을 수 있나요?

온라인 도움말 프런트 페이지의 Bifrost Graph Editor Help 메뉴에서 또는 AREA의 Community Hub에서 포럼 링크를 확인할 수 있습니다.

Bifrost와 관련한 튜토리얼이나 기타 교육용 자료는 어디서 찾을 수 있나요?

AREA와 YouTube를 방문하면 그래프를 구성하고 원하는 효과를 연출하는 방법을 학습할 수 있습니다. 또, 기타 학습용 콘텐츠에 대해서는 Bifrost Help를 참고할 수도 있습니다. Bifrost를 설치했다면, Windows > Bifrost Browser로 들어가 다양한 샘플 그래프를 찾아볼 수 있는데, 이런 그래프를 자신의 장면으로 가져오면 작업을 시작하기가 한결 더 수월해집니다.

도움이 될 만한 자료가 있나요?

현재, Bifrost에 대한 지원은 Maya 지원에 포함되어 있습니다.

Bifrost Extension 플러그인 로드하기

만일, **Bifrost Graph**가 Maya의 **Create** 메뉴에 나타나지 않고 **Bifrost Browser**와 **Bifrost Graph Editor**가 Maya의 **Windows** 메뉴에 나타나지 않는다면, Bifrost Extension 플러그인이 로드되지 않은 것입니다.

Maya에서 이 플러그인을 로드하는 방법:

- 1. Windows > Settings/Preferences > Plug-in Manager를 선택합니다.
- bifrostGraph.mll을 수동으로 로드하는데, 이것을 이후 세션에서는 자동으로 로드되도록 설정할 수 있습니다. Maya의 FX 메뉴 세트에 있는 Bifrost Fluids를 사용하기 위해서는 Bifrost 그룹 내 다른 옵션이 필요한데, 이 또한 선택적으로 로드할 수 있습니다.

만일, bifrostGraph.mll이 **Plug-in Manager**에 나타나지 않는다면, 이 플러그인이 설치되지 않은 것입니다. Bifrost Extension for Maya 설치하기를 참조하십시오.

bifrostGraph.mll이 로드되면 mayaVnnPlugin.mll이 자동으로 로드됩니다. 이것도 ell처럼 자동으로 로드되도록 설정할 수 있으나, 필수 사항은 아닙니다.

Get started with sample graphs샘플 그래프로 시작하기

각종 샘플 그래프를 자기만의 효과를 만들기 위한 출발점으로 삼을 수도 있고 Bifrost에 대해 자세히 알아보기 위해 연구 대상으로 삼을 수도 있습니다.

Maya에서 샘플 그래프를 로드하려면, Windows > Bifrost Browser를 선택합니다.



- 1. 왼쪽에서 카테고리를 선택합니다.
- 2. 그래프를 클릭해 선택하거나 더블클릭해 가져옵니다. 정보 아이콘 i를 클릭하면 포인터 아래의 샘플 설명을 볼 수 있습니다.
- 관련 장면 파일이 있다면, 여기서 그 파일을 클릭해 열 수 있습니다. 장면들은 그래프에 포함되지 않은 조명, 입력 지오메트리, 애니메이션 및 기타 요소를 포함하고 있을 수 있습니다.
- 4. 그래프를 클릭해 현재 장면으로 가져옵니다. Bifrost Graph Editor가 열립니다.

그래프의 최상위 레벨에서 선택된 노드가 하나도 없을 경우, **Info** 패널에서 그래프 정보를 확인할 수 있습니다.



자신만의 그래프를 게시해 다른 사용자들과 공유할 수 있습니다. 이렇게 게시해놓은 그래프는 **Bifrost Browser**에도 나타납니다. 효과를 그래프로 게시하기를 참조하십시오.

그래프 만들기

Maya 장면에서 Bifrost 그래프를 새로 만드는 방법:

- 1. 선택사항으로, Maya에서 메쉬나 nParticle 객체를 1개 이상 선택합니다.
 - 객체를 1개 선택한 경우, 그 객체는 그래프의 input 노드의 **Path** 섹션에 자동으로 추가됩니다. input 노드의 출력 port를 이용해 이 객체를 그래프에 있는 다른 노드에 연결할 수 있습니다.
 - 하지만, 객체를 다수 선택한 경우에는 input 노드 출력 port의 데이터 유형은 그와 다른 객체 어레이입니다.
- 2. Maya의 메인 메뉴에서 Create > Bifrost Graph를 차례로 선택합니다.

그러면, Graph Editor가 열리고 자체 input과 output 노드를 포함하는 새로운 그래프가 나타납니다. Graph Editor가 이미 열려 있는 경우에는 새로운 탭에 새로운 그래프가 열립니다.

Maya의 Outliner에 bifrostGraph 노드가 나타나는데, 여기서 이 노드를 선택할 수 있습니다.

노드를 추가하고 그래프를 구성하는 방법은 그래프 생성 및 수정을 참고하십시오.

🖋 참고:

nParticle 객체 1개를 선택한 경우, convertParticleToBifrost라는 이름의 두 번째 Bifrost 그래프가 만들어집니다. 이 탭은 Graph Editor에서 쉽게 닫을 수 있습니다.

그래프 열기

그래프를 열려면, Outliner에서 원하는 그래프를 선택한 다음 Window > Bifrost Graph Editor를 차례로 선택합니다.

Graph Editor가 이미 열려 있다면, 탭 표시줄에 있는 + 아이콘에서 기존의 그래프를 선택할 수 있습니다. 그러면, 그래프가 새 탭에 열립니다.

Create	Edit	Display	Options Help
bifrostGraph1 ×			bifrostGraph1
<	bifrostGrap	ostGraph	bifrostGraph2
			bifrostGraph3
			New Graph
	معيش		

장면에 있는 기하학적 객체 및 머티리얼 입력하기

그래프 입력값으로 폴리곤 메쉬나 nParticle 객체를 사용할 수 있습니다. 예를 들어, 장면의 객체들을 그래프로 변형하거나 시뮬레이션에서 파티클을 방출하는 것과 같은 용도로 사용할 수 있습니다.

그래프 상단의 각 입력 노드에는 **Path** 섹션이 있는데, 이 섹션을 이용해 장면에서 기하학적 객체를 가져올 수 있습니다.

path 매개변수는 장면을 조회하고 해당하는 객체 또는 객체 어레이를 반환하는 데 사용되는 문자열을 포함하고 있습니다. 장면에서 객체의 이름 또는 레벨을 변경하는 경우에는 이 문자열을 수동으로 업데이트해야 합니다.

장면에 있는 객체는 그래프의 최상위 레벨에만 입력할 수 있으며 컴파운드 안에 입력하는 것은 불가능합니다.

Maya의 Outliner에서 객체 끌어오기

마우스 가운데 버튼을 눌러 Maya의 Outliner에서 Bifrost Graph Editor의 그래프 최상위 레벨로 메쉬 객체나 메쉬의 형상을 드래그합니다. 또는, 마우스 왼쪽 버튼을 이용해 아이콘(이름이 아님)을 드래그할 수도 있습니다.

그러면, 객체와 동일한 이름의 새로운 input 타입 노드가 만들어집니다. 단, 계층구조는 지원되지 않으며 첫 번째 유효한 형상만 가져오게 됩니다.

여러 개의 객체를 선택해 그래프로 드래그한 경우, 노드의 이름은 기본적으로 Input_by_Path로 설정됩니다. 각 지오메트리 타입마다 별도의 출력이 있습니다. 동일한 유형의 객체가 여러 개 있을 경우, 그에 해당하는 출력은 객체 어레이입니다.

팁:

Node Editor 또는 Hypergraph에서 마우스 가운데 버튼으로 형상을 드래그할 수도 있습니다.

그래프로 머티리얼 끌어오기

기하학적 객체와 마찬가지로, 마우스 가운데 버튼을 눌러 머티리얼을 그래프로 끌어올 수 있습니다. 이것은 Node Editor와 Hypershade에서 볼 수 있는 머티리얼 노드와 Outliner(Display > Assigned Materials가 활성화된 경우)에서 볼 수 있는 머티리얼 노드에 해당됩니다.

객체를 수동으로 추가하기

객체를 그래프로 드래그하는 대신, input 노드를 수동으로 추가하여 객체를 지정할 수 있습니다.

- 1. 그래프 최상위 레벨에서, Tab을 누르고 input을 입력합니다. input을 클릭해 그래프에 노드를 추가합니다.
- 2. Parameter Editor에서, 형상의 이름이나 Path 박스에 있는 경로(예: "/pSphere1/pSphereShape1")를 입력합니다. 이름과 경로는 대소문자를 구분합니다.

다중 객체 및 와일드카드

다수의 객체를 어레이 하나로 지정하려면, 공백으로 구분된 목록(예: "/pCube1/pCubeShape1 /pSphere1/pSphereShape1")을 이용합니다.

와일드카드 "*"와 "?"도 지원됩니다. 임의의 심층 계층구조인 경우 "//"를 이용합니다. 예:

- "/*/*"는 그룹에 없거나 계층구조를 변형시키는 모든 지오메트리를 가리킵니다.
- "/group1/*/*"는 그룹1의 모든 지오메트리를 가리킵니다.
- "/p*/*"는 트랜스폼(transform) 노드의 바로 아래 하위 레벨에 속한 지오메트리 중 이름이 "p"로 시작하며 다른 그룹이나 계층구조에 속하지 않는 모든 지오메트리를 가리킵니다.
- "/*/*S*"는 이름에 "S"가 포함되며 다른 그룹이나 계층구조에 속하지 않는 모든 지오메트리를 가리킵니다.
- "//*"는 장면의 모든 지오메트리를 가리킵니다.
- "/group1//*"는 "group1" 내 계층구조에 있는 모든 지오메트리를 가리킵니다.

기하학적 객체를 장면으로 출력하기

객체를 장면으로 출력하려면, 그 객체를 최상위 레벨 그래프의 output 노드에 연결하면 됩니다.



그러면, 장면에서 Bifrost 지오메트리가 만들어집니다. 이름은 bif로 기본 설정됩니다.

참고:

- 객체 어레이도 출력할 수 있으나, 이 경우에는 장면에 지오메트리 1개가 만들어집니다.
- 그래프에서 재질 지정이 이루어지지 않는 한, 장면에서 입력된 메쉬는 자신의 원래 재질을 그대로 이용하며 그래프에 만들어진 다른 객체들은 기본 장면 재질을 이용합니다. 하지만, 출력 지오메트리에 다른 재질을 지정함으로써 이 설정을 변경할 수 있습니다.
- Bifrost 지오메트리를 직접 렌더링하려면, Arnold처럼 호환되는 렌더러를 사용해야 합니다. 또는, 그래프에 있는 지오메트리를 Alembic이나 VDB와 같은 표준 파일 형식으로 출력해서 렌더링하는 방법도 있습니다.

Bifrost 메쉬를 Maya 메쉬로 변환하기

Maya의 Node Editor에서는 Bifrost에서 출력한 메쉬를 Maya 메쉬로 변환할 수 있습니다.

- 1. Node Editor에 bifrostGeoToMaya 노드를 추가하고 Bifrost 그래프에서 출력한 메쉬를 연결합니다.
- 2. Maya Mesh 출력을 Maya 메쉬 형상 노드의 In Mesh 포트에 연결합니다.

🖊 참고:

컬러 세트를 옮기려면, Bifrost 메쉬를 확인하여 point_component 를 대상으로 하는 float4 (RGBA) 값들로 이루어진 point_color 라는 이름의 geo 특성이 있는지 확인합니다.



팁:

Node Editor에 mesh 노드를 추가하여 polySurfaceShape라는 이름의 새 Maya 메쉬를 만들 수 있습니다.

장면에 입력하거나 장면에서 출력하는 입력값 또는 출력값

장면에 값을 입력하거나 장면에서 값을 출력하려면, 그 값을 최상위 레벨 그래프의 input 또는 output 노드에 연결하면 됩니다.



Maya가 기본적으로 이해하는 유형이라면 어떤 유형이라도 그래프 입력 또는 출력이 될 수 있는데, 예를 들면, 부울, 정수, 부동 소수점 값, 벡터, 행렬 등이 있습니다. 이넘(Enum)이나 함수 곡선과 같은 특수화된 객체는 지원되지 않습니다.

입력값은 Maya의 Attribute Editor에서 그래프 노드의 Extra Attributes 섹션에 나타납니다.

출력값을 사용하려면, Node Editor에서 해당 값을 다른 Maya 노드에 연결합니다. 입력값도 이와 동일한 방식으로 연결할 수 있습니다.

렌더링 관련 도움말

Maya에서 Bifrost Extension의 객체를 렌더링할 때 몇 가지 염두에 두어야 할 사항이 있습니다.

배치-렌더링 시뮬레이션

Batch Render를 이용해 시퀀스를 렌더링하려면, 시뮬레이션의 첫 번째 프레임이 렌더링 범위의 첫 번째 프레임 앞에 올 수 없습니다.

해결책으로, 렌더링할 첫 번째 프레임까지만 시뮬레이션을 진행시킨 다음 Render Sequence를 이용해 렌더링하면 됩니다. 다른 방법으로, 시뮬레이션을 파일로 캐싱해 이를 렌더링할 수도 있습니다.

이 제한은 Bifrost Extension에만 해당합니다. 시뮬레이션이 먼저 시작될 경우 Bifrost Fluids가 렌더링할 첫 번째 프레임까지 자동으로 실행됩니다.

재질 속성을 뷰포트에서 직접 제어

Maya 2018은 geo 특성을 이용해 뷰포트에서 재질 속성을 제어할 때 color 채널만 지원합니다. 하지만, Maya 2019 및 이후 버전에서는 color 외에도 transparency, incandescence 등을 비롯해 float3 또는 float4 속성까지 지원합니다. geo 특성으로 재질 특성 제어를 참조하십시오.

Bifrost Extension for Maya 설치하기

Bifrost Extension for Maya 설치관리자에는 아래 항목들이 포함되어 있습니다.

- 절차식 그래프를 만들어 모래, 불, 연기, 폭발과 같은 효과를 연출할 수 있는 새로운 노드 기반 환경
- FLIP (fluid implicit particle) 솔버를 이용해 시뮬레이션한 액체와 공기역학 효과 등을 연출할 수 있는 절차식 프레임워크 Bifrost Fluids
- BOSS(Bifrost Ocean Simulation System)
- bifcmd(노드 기반 환경에서 내보낸 Bifrost 그래프를 실행하는 명령행 도구)

Bifrost Extension for Maya는 Maya 2018과 Maya 2019, 그리고 각각의 업데이트 릴리스와 호환됩니다.

Maya 2019.2 Update를 설치하면, Maya 설치관리자에 Bifrost Extension for Maya가 자동으로 포함됩니다. Maya의 모든 이전 버전에 대해서는, Bifrost Community Hub를 방문해 해당하는 설치관리자를 다운로드하십시오.

이전 버전 플러그인과의 호환성

Bifrost의 구 버전 즉, Bifrost for Maya 2018과 Bifrost 1.5.7.x for Maya 2019는 새 버전인 Bifrost Extension for Maya와 함께 설치할 수 있습니다. 이전 플러그인을 제거할 필요는 없지만 기본적으로 가장 최근에 설치한 버전만 로드됩니다. 만일, 다른 플러그인 버전을 로드하고 싶다면, Maya에서 Plug-in Manager를 통해 해당 버전을 찾거나 bifrost.mod와 vnn.mod 모듈 파일을 편집해 원하는 버전을 가리키도록 하면 됩니다. .mod 파일은 아래 경로에서 찾을 수 있습니다.

- C:\Program Files\Common Files\Autodesk Shared\Modules\Maya (Windows)
- /usr/autodesk/modules (Linux)
- /Users/Shared/Autodesk/modules (macOS)

이전 플러그인 버전들을 제거할 경우, Bifrost Extension for Maya 새 플러그인을 통해 Bifrost Fluids와 BOSS가 다시 설치됩니다.

사실, 좀 더 체계화하기 위해 Maya 2018 또는 이것의 업데이트 릴리스를 이용하고 있다면, Bifrost for Maya 2018 플러그인을 제거한 다음 Bifrost Extension for Maya 새 플러그인을 설치해야 할 수도 있는데, 이것은 이전의 Bifrost for Maya 2018 설치관리자가 ./Autodesk/Bifrost/Maya2018 디렉터리 내에 버전별 하위 폴더를 생성하지 않았기 때문입니다.

Bifrost Fluids를 포함하는 레거시 신

Bifrost Fluids를 포함하는 레거시 신을 로드할 때, 다음과 같은 오류가 발생할 수 있습니다.

// Warning: Bifrost: to use the updated Bifrost graphs in Maya, graph |bifrostLiquid1|bifrostLiquidContainer1
needs to be updated to the latest version, call mel script: replaceBifrostFXGraphsWithFactoryGraphs
|bifrostLiquid1|bifrostLiquidContainer1 ; //

// Error: bifrostLiquidContainer1: Emitters: Could set input port value. Failed to set input parameter
Emitters //

// Error: Bifrost evaluation error //

이 문제를 해결하려면, 메시지에 따라 MEL 명령을 실행합니다. 예를 들어, 액체 노드를 bifrostLiquid1이라고 할 때, 다음과 같은 명령을 실행합니다.

replaceBifrostFXGraphsWithFactoryGraphs |bifrostLiquid1|bifrostLiquidContainer1 ;

네트워크 상의 사용자 지정 위치에 Bifrost 설치하기

- 1. 설치 디렉터리 ../Bifrost/<MayaVersion>/<BifrostVersion>을 원하는 위치로 복사합니다.
- 2. MAYA_MODULE_PATH 환경 변수를 CustomPath/Bifrost/<MayaVersion>/<BifrostVersion>으로 설정합니다.

🖋 참고:

사용자 지정 위치에서는 설치 제거가 지원되지 않습니다. 설치 디렉터리에서 파일들을 수동으로 제거해야 할 수도 있습니다.

이 섹션 내 페이지 정보

- 플랫폼별 설치 도움말
- 렌더링을 위한 Arnold for Maya 플러그인 설치하기

플랫폼별 설치 도움말

Windows에 Bifrost Extension 설치하기

Bifrost Extension의 기본 설치 경로는 C:\Program Files\Autodesk\Bifrost\<MayaVersion>\<BifrostVersion>입니다.

Control Panel > Programs and Features에서 Bifrost를 제거합니다.

macOS에 Bifrost Extension 설치하기

Bifrost Extension의 기본 설치 경로는 /Applications/Autodesk/bifrost/<MayaVersion>/<BifrostVersion>입니다.

Linux에 Bifrost Extension 설치하기

Bifrost Extension의 기본 설치 경로는 /usr/autodesk/bifrost/<MayaVersion>/<BifrostVersion>입니다.

rpm을 통해 직접 설치하려면, Maya가 최신 버전을 알아서 인지하도록 rpm --force를 이용합니다.

Bifrost Community Hub를 통해 다운로드한 설치관리자는 확장자가 .run 파일로 되어 있고 다음과 같이 설치해야 합니다.

sudo sh ./Bifrost_2.0.1.0_Maya2018_Linux.run

Page Down을 눌러 라이선스 계약서를 읽은 후, 지시에 따라 Install 또는 Extract를 클릭합니다.

상위 페이지: Bifrost Extension for Maya 설치하기

렌더링을 위한 Arnold for Maya 플러그인 설치하기

Maya 2019.2 Update 설치관리자는 Arnold로 Bifrost 콘텐츠를 렌더링할 수 있는 Arnold for Maya 버전을 포함하고 있습니다.

Maya 2019, Maya 2019.1 Update, Maya 2018, Maya 2018 Update 릴리스 중 하나를 이용하고 있다면, Arnold for Maya Download Area를 방문해 Arnold for Maya 최신 설치관리자를 다운로드할 수 있습니다.

절차식 그래프로 만든 Bifrost 콘텐츠를 레더링할 수 있는 Arnold for Maya의 최소 버전은 3.2.2입니다. Bifrost 릴리스는 모두 이 버전 이상의 Arnold for Maya와 호환됩니다. 이 경우가 아니라면, 호환성 관련 정보는 릴리스 노트를 통해 제공할 것입니다.

상위 페이지: Bifrost Extension for Maya 설치하기