

Visual Effect Graph

작성자 : 김은규

목차

- 01. Changelog
- 02. Intro
- 03. Getting Start
- 04. Node

01. Changelog

- v0.1 : 초안 작성

02. Intro

시각 효과 그래프 사용

시각 효과 그래프를 사용하여:

- 하나 이상의 입자 시스템을 만듭니다.
- 정적 메쉬를 추가하고 셰이더 속성을 제어합니다.
- 장면에서 사용하는 인스턴스를 사용자 지정하는 속성을 만듭니다.
- 이벤트를 만들어 효과의 일부를 켜고 끕니다. 그런 다음 C# 또는 타임라인 을 통해 장면에서 이러한 이벤트를 보낼 수 있습니다.
- 일반적으로 사용하는 노드의 하위 그래프를 생성하여 기능 라이브러리를 확장합니다.
- 다른 시각 효과 그래프에서 시각 효과 그래프를 사용합니다. 예를 들어, 더 복잡한 그래프에서 간단하지만 구성 가능한 폭발을 재사용하고 사용자 정의할 수 있습니다.
- 변경 사항을 즉시 미리 볼 수 있으므로 다양한 속도로 효과를 시뮬레이션하고 단계별 시뮬레이션을 수행할 수 있습니다. Visual Effect Graph를 설치하는 방법에 대한 지침은 Visual Effect Graph 시작하기를 참조하십시오.

- 정적 메쉬 또는 셰이더 속성을 제어 가능
- C# 또는 타임라인을 통해 장면에서 이벤트를 보낼 수 있습니다.

02. Intro

- Visual Effect Graph는 2020.2 이상은 10.x, 2021.1 이상은 11.x를 사용할 수 있습니다.

Unity 에디터 호환성

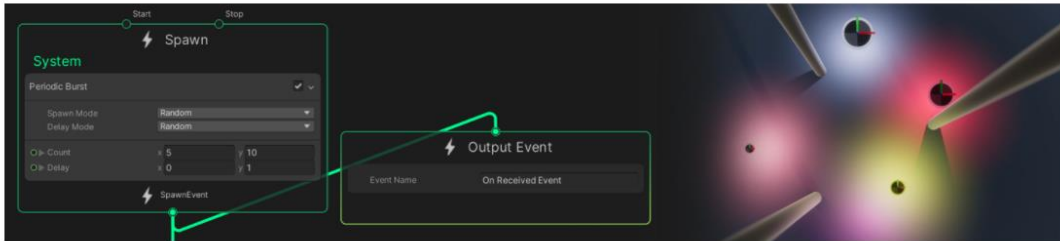
다음 표는 Visual Effect Graph 버전과 다양한 Unity 에디터 버전의 호환성을 보여줍니다.

패키지 버전	최소 Unity 버전	최대 Unity 버전
11.x	2021.1	2021.1
10.x	2020.2	2020.3
8.x / 9.x-미리보기	2020.1	2020.1
7.x	2019.3	2019.4
6.x	2019.2	2019.2

02. Intro

- Output Event로 C#으로 이벤트를 보낼 수 있습니다.

CPU 출력 이벤트



출력 이벤트는 트리거될 때 그래프에서 C#으로 이벤트를 보냅니다. 이를 사용하여 조명, 사운드, 물리적 반응 또는 게임 플레이를 시각 효과에 동기화할 수 있습니다.

이 기능에 대한 자세한 내용은 [출력 이벤트](#) 를 참조하십시오 .

이 기능을 사용하는 방법의 예는 [출력 이벤트 핸들러](#) 를 참조하십시오 .

02. Intro

특징

다음은 Unity 2021.1에 포함된 Visual Effect Graph 버전 11에 Unity가 추가한 기능 목록입니다. 각 항목에는 기능 요약과 관련 문서에 대한 링크가 포함되어 있습니다.

SRP 패키지는 핵심의 일부입니다.

Unity 2021.1 릴리스와 함께 그래픽 패키지가 Unity의 코어로 재배포됩니다. 이 이동은 새로운 Unity 그래픽 기능으로 작업하는 경험을 단순화할 뿐만 아니라 프로젝트가 항상 최신 검증된 그래픽 코드에서 실행되도록 합니다.

Unity의 각 릴리스(알파/베타/패치 릴리스)에 대해 그래픽 코드는 기본 Unity 설치 프로그램에 포함되어 있습니다. 최신 Unity 릴리스를 설치하면 최신 URP, HDRP, 셰이더 그래프, VFX 그래프 등도 함께 제공됩니다.

그래픽 패키지를 기본 Unity 릴리스에 더 가깝게 묶으면 더 나은 테스트를 통해 사용하는 그래픽 패키지가 다운로드한 Unity 버전에서 광범위하게 테스트되었는지 확인할 수 있습니다.

매니페스트 파일에서 재정의하여 그래픽 패키지의 로컬 복사본 또는 사용자 지정 버전을 사용할 수도 있습니다.

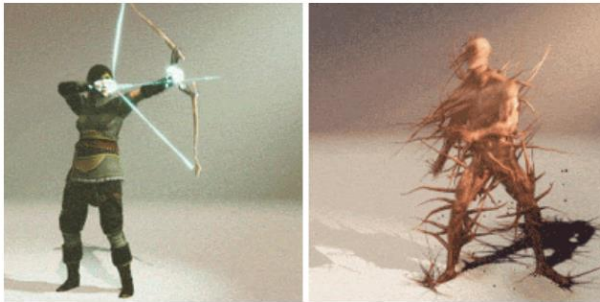
자세한 내용은 포럼의 다음 게시물을 참조하십시오. [SRP v11 beta is available now](#).

- Unity 2021.1 릴리스와 함께 그래픽 패키지가 Unity의 코어로 재배포됩니다.(11.x)

02. Intro

- SkinnedMeshRenderer의 표면에 생성되는 파티클의 예입니다.(11.x)

SkinnedMeshRenderer 샘플링



SkinnedMeshRenderer의 표면에 생성되는 파티클의 예입니다. [Mixamo.com](https://mixamo.com)의 모델 및 애니메이션.

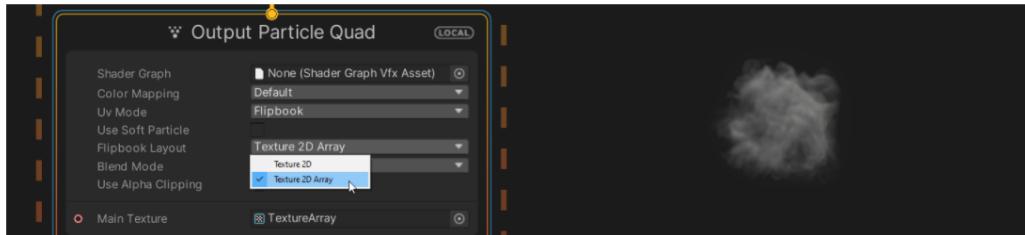
이 버전의 Visual Effect Graph에는 [SkinnedMeshRenderers](#)를 샘플링하는 기능이 추가되었습니다. 이를 통해 스킨된 지오메트리에서 정점 데이터를 검색하고 애니메이션 캐릭터에 입자를 생성하는 것과 같은 다양한 목적으로 사용할 수 있습니다.

이 기능에 대한 자세한 내용은 [샘플 메시](#)를 참조하십시오.

02. Intro

- Texture2DArray를 사용할 수 있습니다.(11.x)

Texture2DArray 플립북

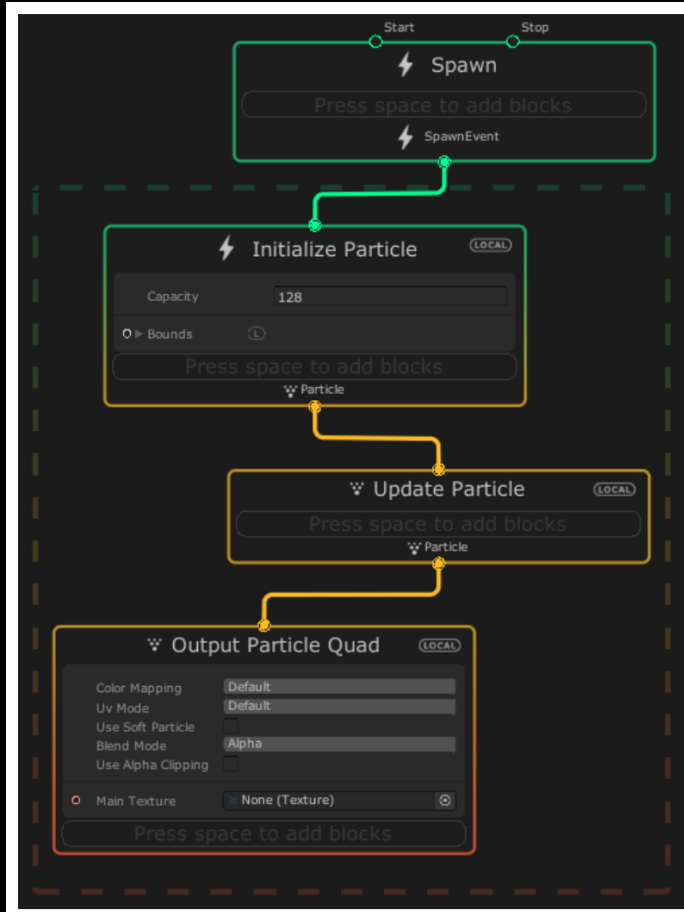


Texture2DArray 옵션이 있는 플립북 레이아웃 속성.

이 버전의 Visual Effect Graph에서 Texture2DArray 자산을 플립북으로 사용할 수 있습니다. Texture2DArrays를 사용하면 플립북 프레임 사이의 텍스처 번짐을 방지할 수 있습니다. Texture2DArrays를 사용하려면 출력 컨텍스트의 **Flipbook 레이아웃**을 **Texture 2D Array**로 설정하십시오. 이를 통해 Texture2DArray 자산을 컨텍스트의 텍스처 포트에 할당할 수 있습니다. 텍스처의 각 슬라이스는 플립북 내의 프레임에 해당합니다. 플립북을 재생하려면 플립북 **플레이어** 블록을 사용하세요. 일반적으로 Texture2DArray 플립북과 상호 작용하려면 **Tex Index** 속성을 사용하십시오. 이것은 입자당 무작위 플립북 프레임을 사용하는 것과 같이 애니메이션 이외의 목적으로 플립북을 사용하는 경우에 유용합니다.

많은 출력 컨텍스트가 이 설정을 공유하므로 자세한 내용은 [공유 출력 설정 및 속성](#)을 참조하세요.

02. Intro



- 기본 구조는 사진과 같습니다.

02. Intro

- Visual Effect Graph System

Visual Effect Graph 내에서 여러 시스템이 서로 상호 작용할 수 있습니다.

- 스폰 시스템은 하나 이상의 파티클 시스템에서 파티클을 스폰 할 수 있습니다. 이것은 파티클을 생성하는 주요 방법입니다.
- 파티클 시스템은 GPU 이벤트를 사용하여 다른 파티클 시스템에서 파티클을 생성할 수 있습니다. 이 대체 방법은 파티클 사망과 같은 시뮬레이션 이벤트를 기반으로 다른 파티클에서 파티클을 생성할 수 있습니다.
- Spawn 시스템은 다른 Spawn 시스템을 활성화 및 비활성화 할 수 있습니다. 이를 통해 다른 생성 시스템을 관리하는 마스터 생성 시스템을 사용하여 파티클 방출을 동기화할 수 있습니다.

02. Intro

- Visual Effect Graph System

Visual Effect Graph 내에서 여러 시스템이 서로 상호 작용할 수 있습니다.

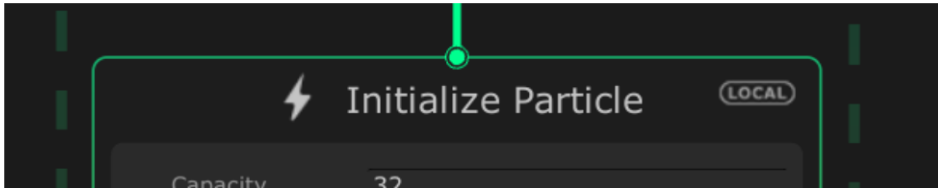
- 스폰 시스템은 하나 이상의 파티클 시스템에서 파티클을 스폰 할 수 있습니다. 이것은 파티클을 생성하는 주요 방법입니다.
- 파티클 시스템은 GPU 이벤트를 사용하여 다른 파티클 시스템에서 파티클을 생성할 수 있습니다. 이 대체 방법은 파티클 사망과 같은 시뮬레이션 이벤트를 기반으로 다른 파티클에서 파티클을 생성할 수 있습니다.
- Spawn 시스템은 다른 Spawn 시스템을 활성화 및 비활성화 할 수 있습니다. 이를 통해 다른 생성 시스템을 관리하는 마스터 생성 시스템을 사용하여 파티클 방출을 동기화할 수 있습니다.

02. Intro

• 시스템 시뮬레이션 공간 설정

시스템 시뮬레이션 공간 설정

시스템은 구성되는 각 컨텍스트의 오른쪽 상단 모서리에 시뮬레이션 공간을 표시합니다. 이것은 시스템의 시뮬레이션 공간 식별자입니다. 컨텍스트가 시뮬레이션 공간에 의존하는 프로세스를 사용하지 않으면 시뮬레이션 공간 식별자를 표시하지 않습니다.



시스템의 시뮬레이션 공간을 변경하려면 시스템의 시뮬레이션 공간 식별자를 클릭하여 호환 가능한 공간을 순환합니다.

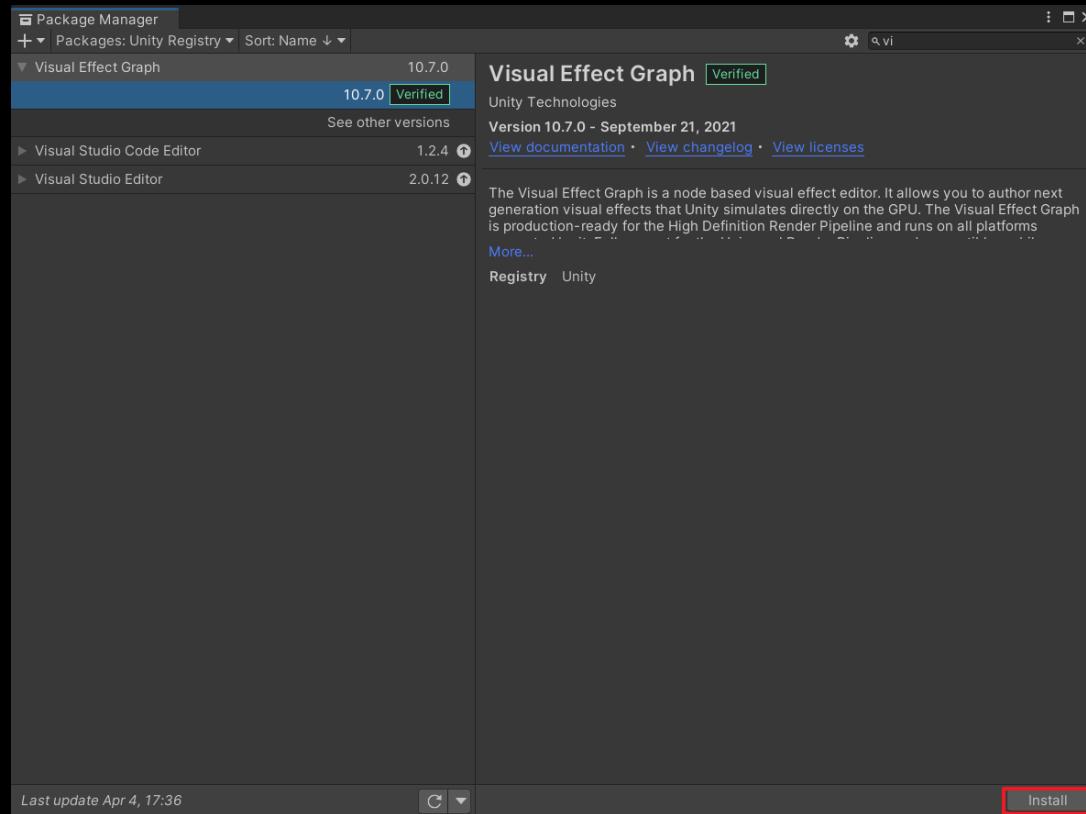


속성의 시뮬레이션 공간 식별자

일부 공간 가능 특성은 시뮬레이션 공간 식별자의 더 작은 버전을 표시합니다. 이것은 시스템의 시뮬레이션 공간을 변경하지 않고 대신 시스템의 시뮬레이션 공간과 다른 공간에서 값을 표현할 수 있습니다. 예를 들어 시스템은 세계 공간에서 시뮬레이션할 수 있지만 속성은 로컬 위치일 수 있습니다.

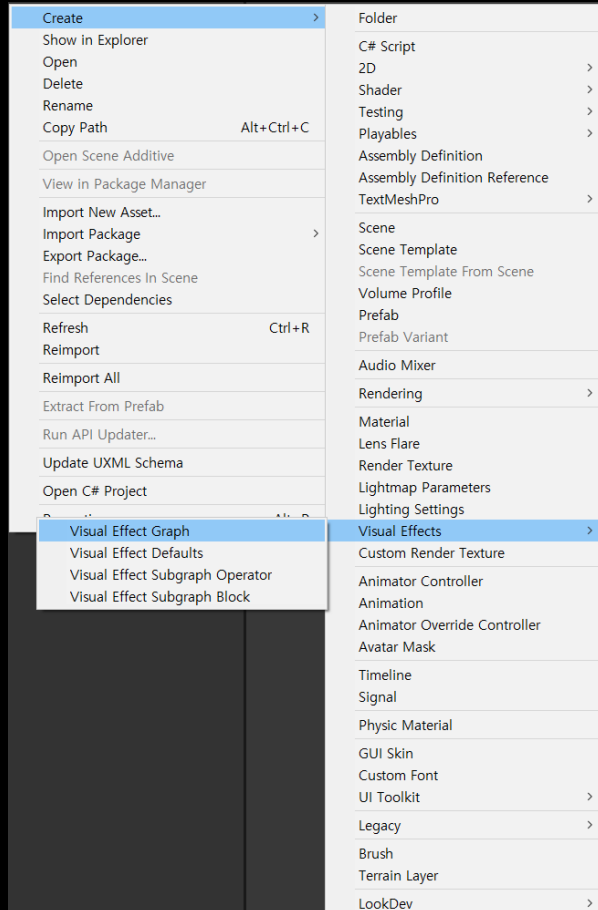


03. Getting Start



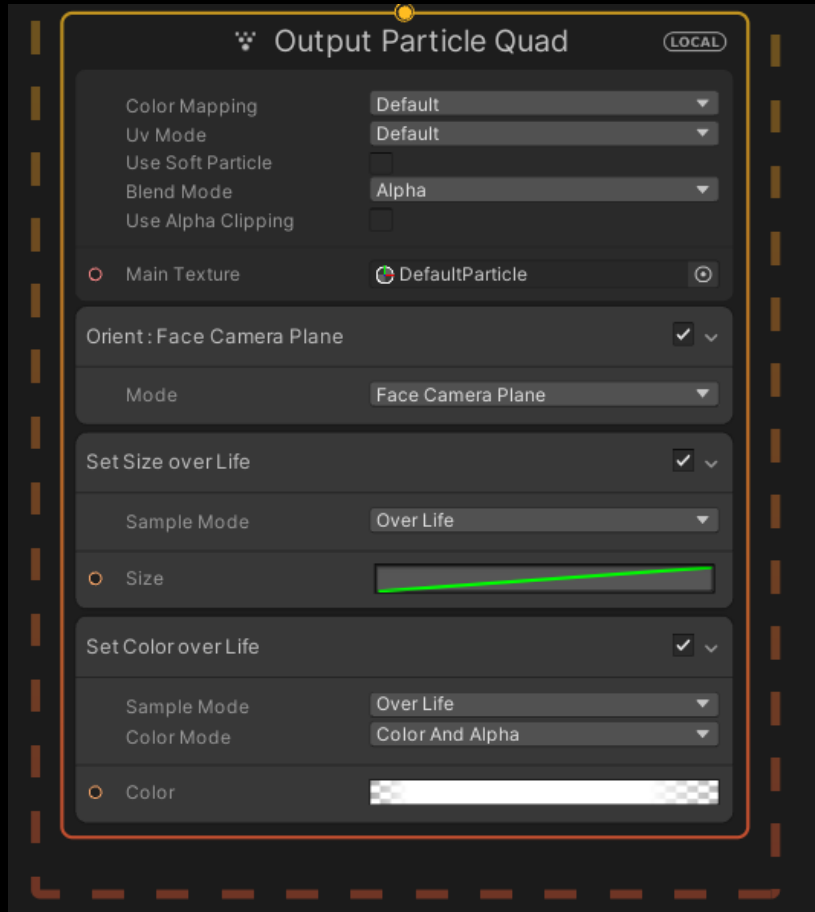
- Package Manger에서 설치 합니다.

03. Getting Start



- Visual Effect Graph를 생성합니다.

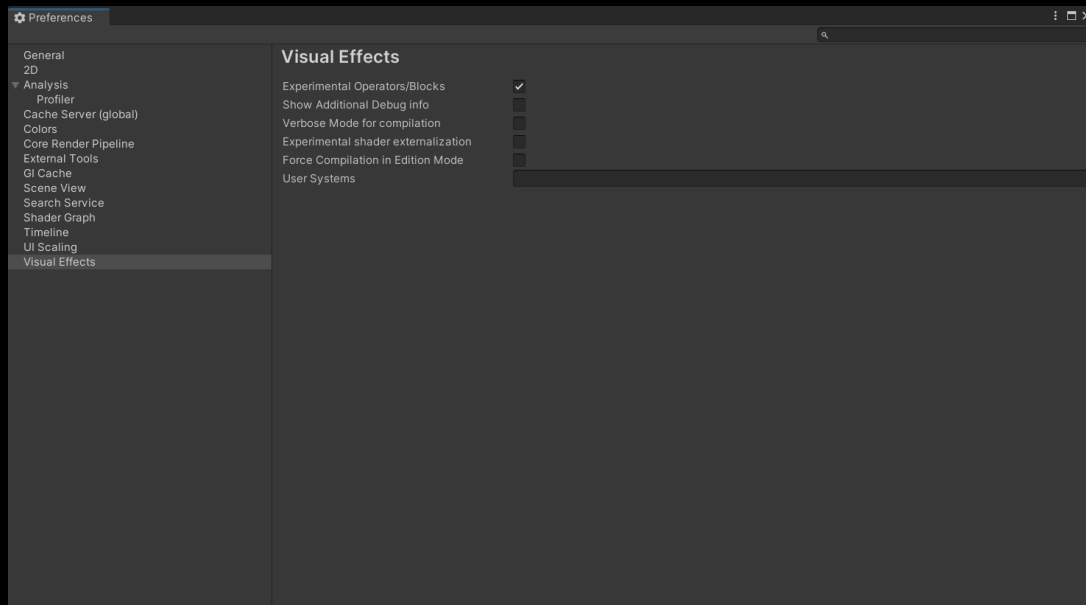
03. Getting Start



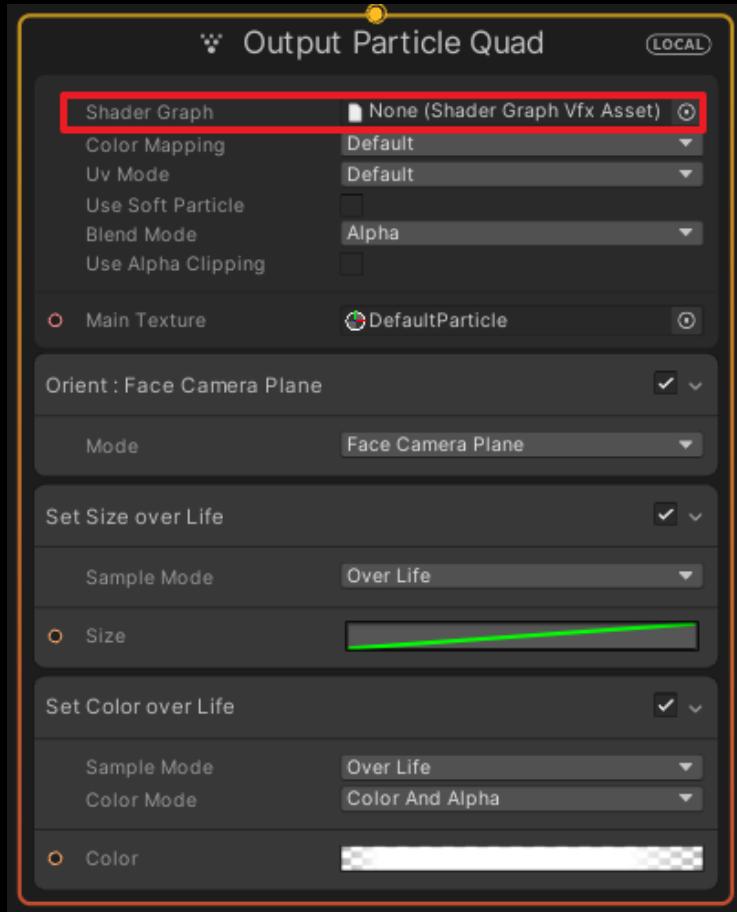
- Shader Graph가 없는 경우, Experimental Operator/Blocs를 체크합니다.

03. Getting Start

- Shader Graph가 없는 경우, Experimental Operator/Blocs를 체크합니다.

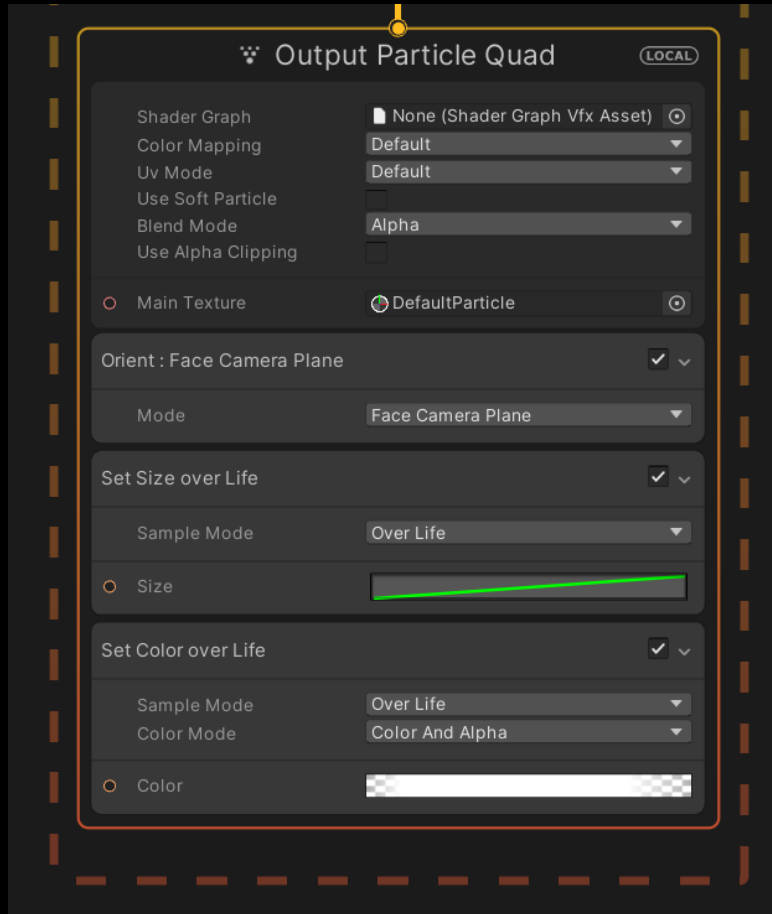


03. Getting Start



- 상단에 Shader Graph가 나타난 것을 알 수 있습니다.

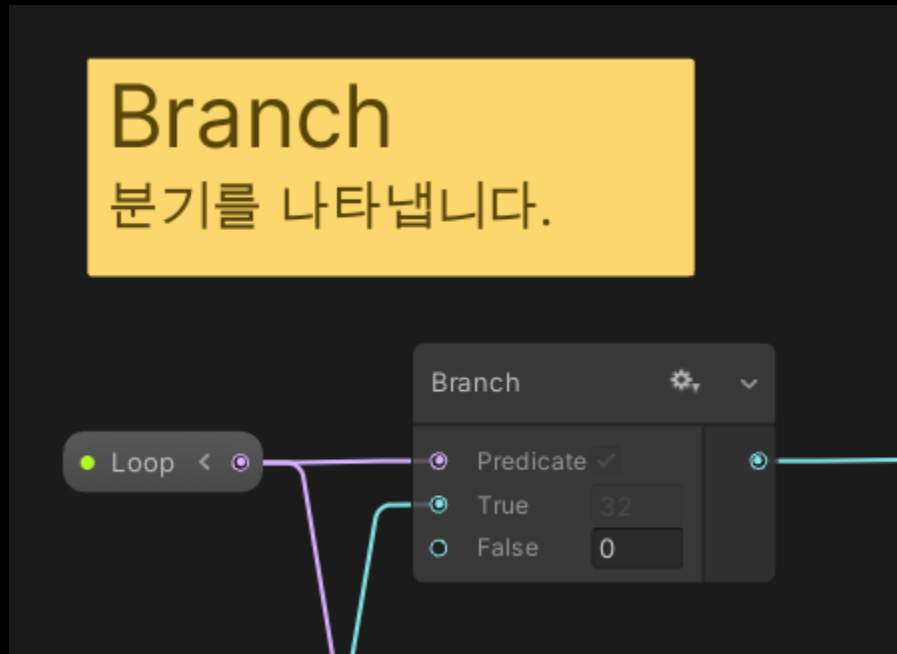
03. Getting Start



- 정적 메쉬 또는 셰이더 속성을 제어 가능하고, C# 또는 타임라인을 통해 장면에서 이벤트를 보낼 수 있습니다.

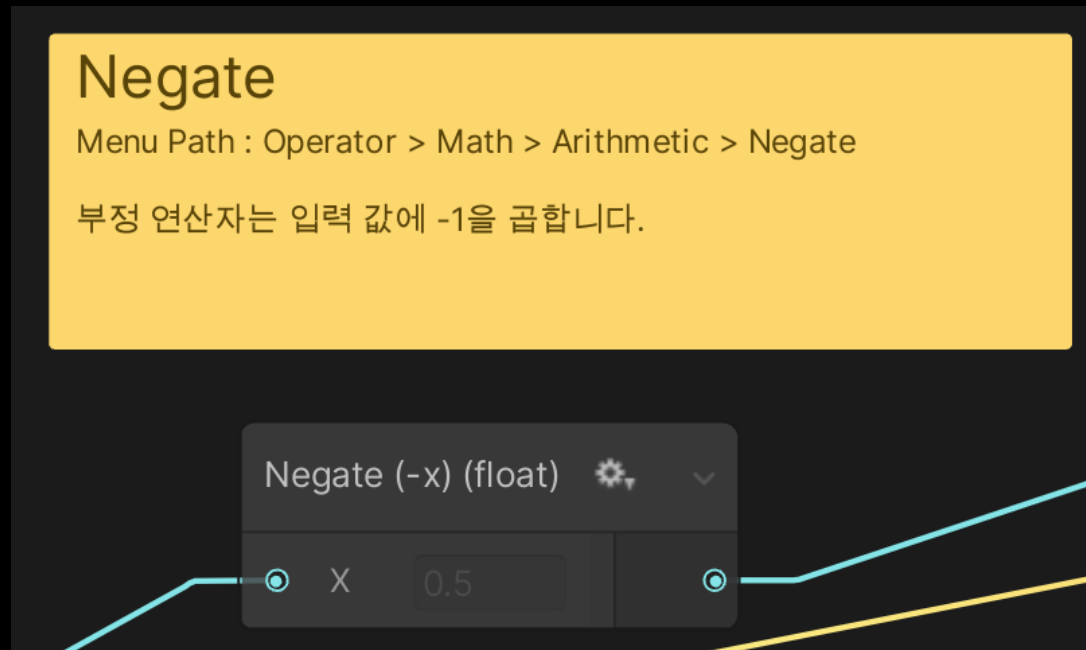
04. Node

- Branch



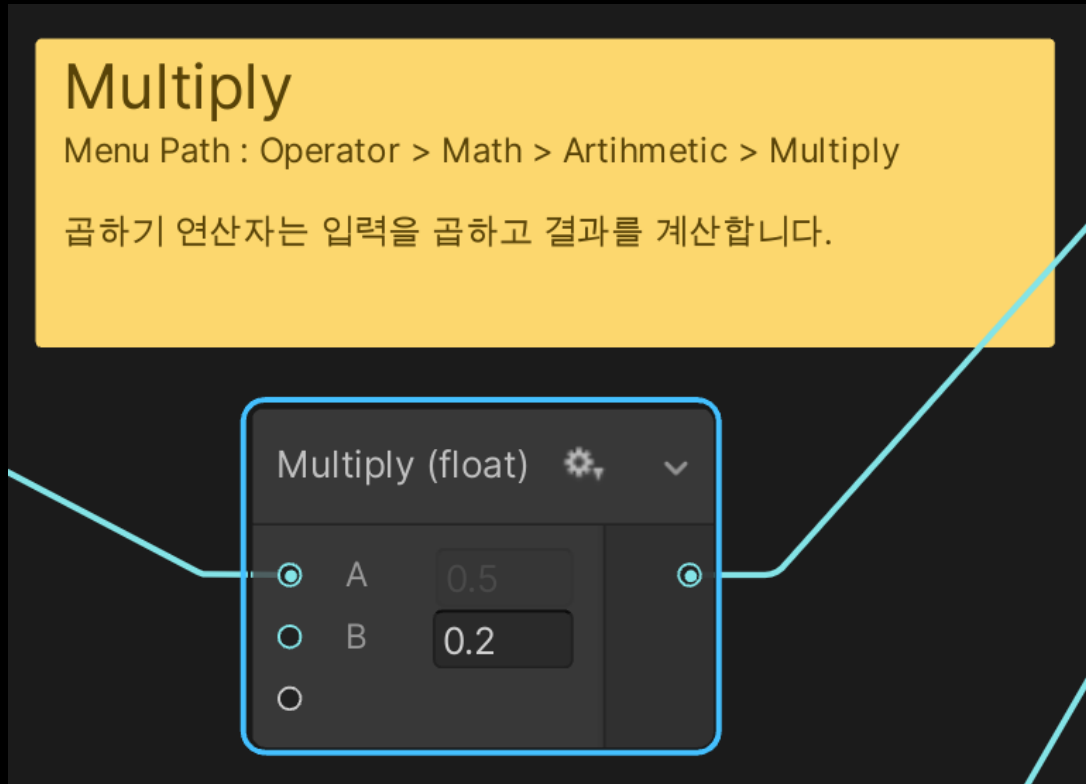
04. Node

- Negate



04. Node

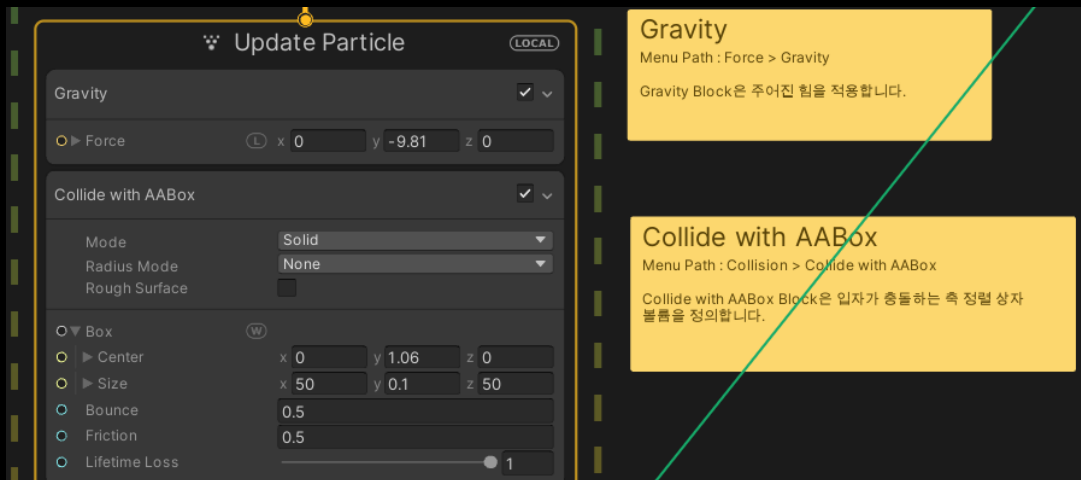
- Multiply



04. Node

- Gravity

- Collide with AABox



04. Node

- Gravity

- Collide with AABox

