A character with long brown hair, wearing a brown sleeveless top and a blue gauntlet, stands on a rocky outcrop. They are looking out over a vast, dark, and jagged landscape under a dramatic sunset sky with orange and purple hues. In the distance, there are tall, dark rock formations and some industrial structures.

게임 엔진

LEC 12 인터랙션 (1)



LEC 11 복습

Animation Montage

■ 다양한 애니메이션 편집 구현

- 여러 개의 애니메이션 시퀀스들을 하나의 애셋으로 만든 후, 일부분들을 다양한 방식으로 조합하여 재생할 수 있음.

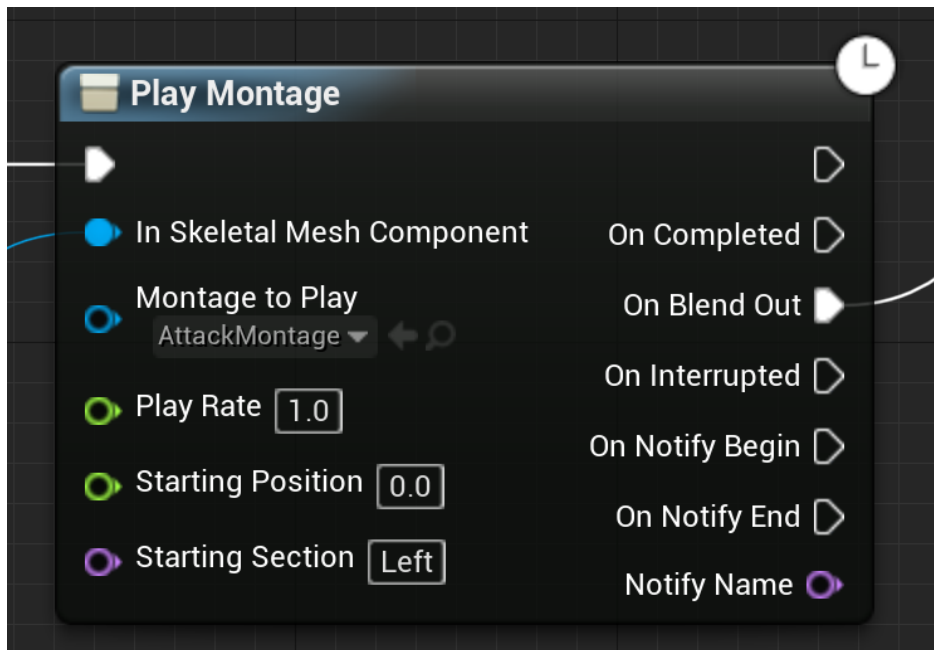
■ 코드 및 블루프린트와 상호 연결하여 애니메이션을 컨트롤

■ 애니메이션의 지능형 루프, 로직기반 애니메이션 전환, 루트 모션 핸들링 등

- 애님 블루프린트의 이벤트 그래프 안에서 애니메이션 재생
- 복잡한 애니메이션 시퀀스를 엮어 하나의 애니메이션으로 간주
- 코드나 블루프린트 스크립트를 통해 애니메이션의 특정 부분이나 다수의 애니메이션 루프
- 코드나 블루프린트 스크립트를 통해 다수의 애니메이션에 대한 이벤트 기반 전환 처리
- 복잡한 애니메이션 시퀀스를 네임드 슬롯에 할당하여 코드나 블루프린트에서 전환
- 코드에 따라 또는 블루프린트 스크립트로 다양한 애님 시퀀스 사이의 정교한 전환

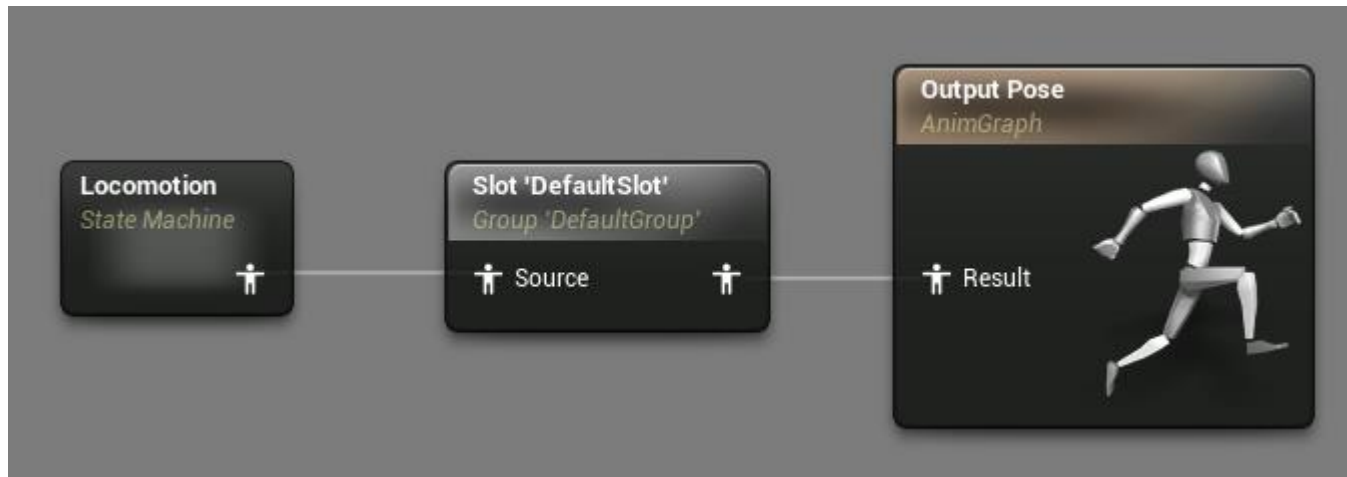
Play Montage 노드

- 몽타주 재생의 모든 기능을 이용할 수 있음.
- 타겟은 스켈레탈 메시



Slot 노드

- 재생 중인 몽타주 슬롯의 애니메이션 데이터를 출력



ComboAttack 몽타주 구성

The screenshot displays the ComboAttack montage editor interface. The main window shows a timeline with a grid from 0 to 110. A red vertical line marks the current position at 20. A blue vertical line marks the end of the current section at 65. The timeline is divided into three sections: RightPunch (0 to 20), LeftPunch (20 to 65), and RightKick (65 to 110). The actions are represented by green bars on the timeline. The left sidebar contains a search filter (20+), a dropdown menu for Montage (DefaultG), a dropdown menu for DefaultGroup.Def, a dropdown menu for Timing, a dropdown menu for Notifies, a dropdown menu for 1, and a dropdown menu for Curves (0). The bottom of the interface shows playback controls and a progress bar.

ComboAttack

Filter 20+

Montage (DefaultG)

DefaultGroup.Def

Timing

Notifies

1

Curves (0)

20* (0.68) (18.22%)

RightPunch

LeftPunch

RightKick

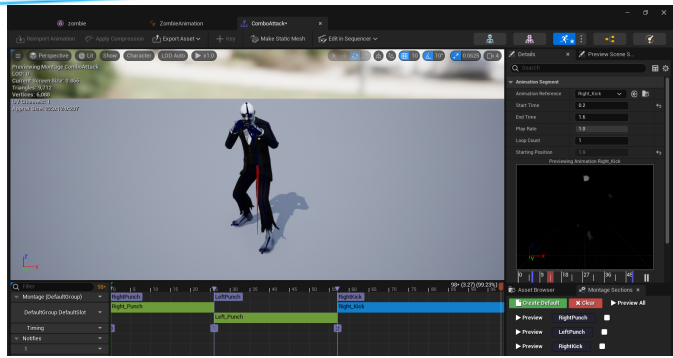
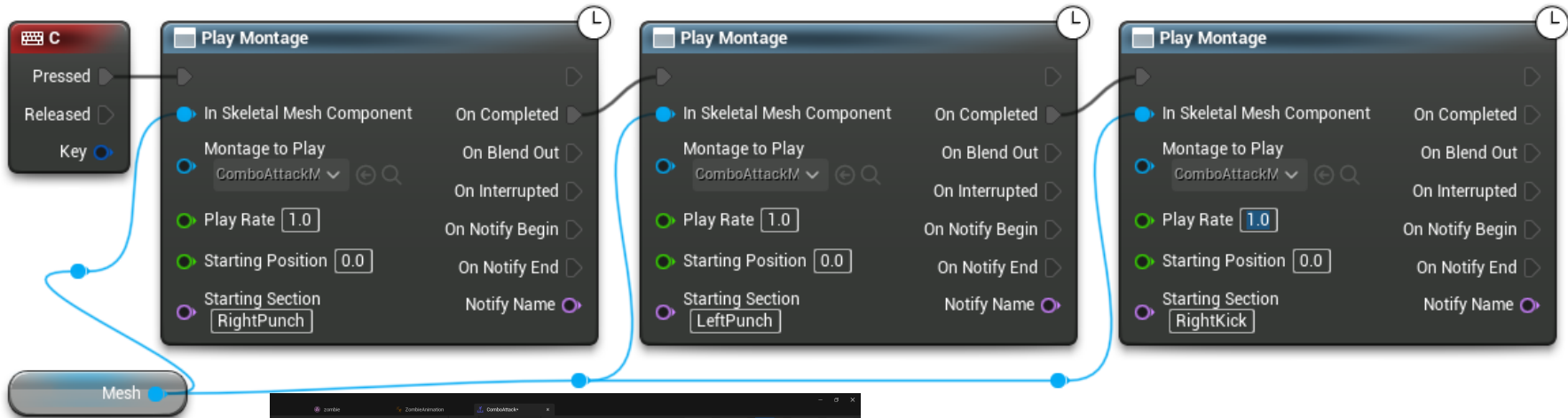
Montage Sections

Create Default Clear Preview All

Preview RightPunch

Preview LeftPunch

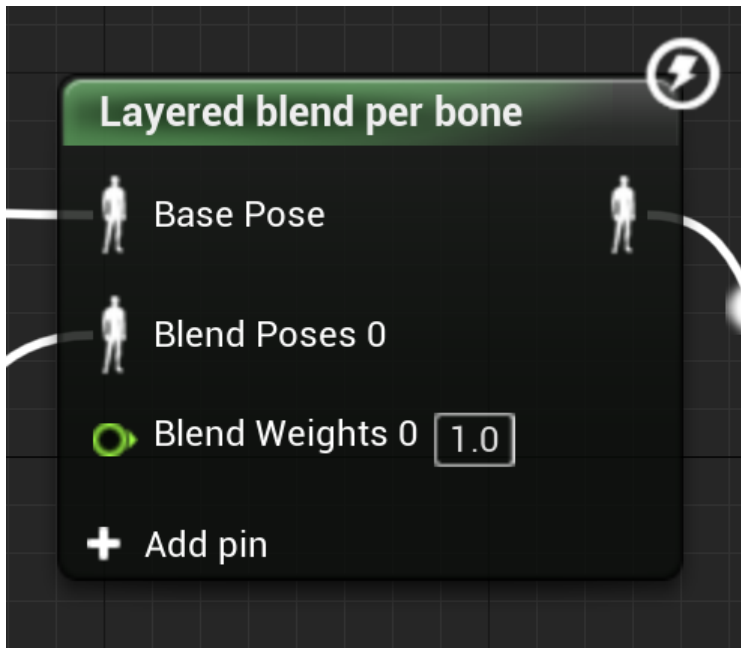
Preview RightKick



Start Time

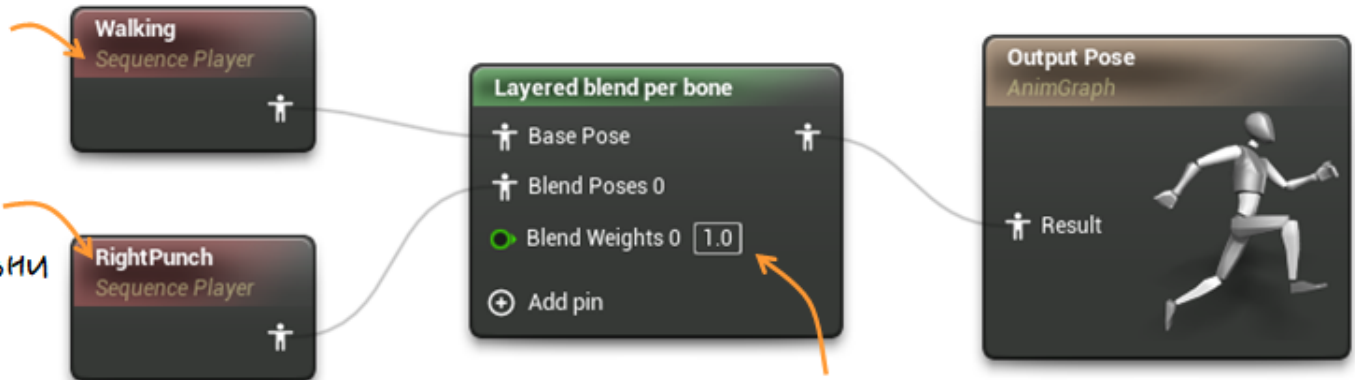
Layerd Blend per Bone 노드

- 스켈레탈 메시의 특정 bones 기준으로 애니메이션을 분리해서 블렌딩



기본 동작은 walking 입니다.

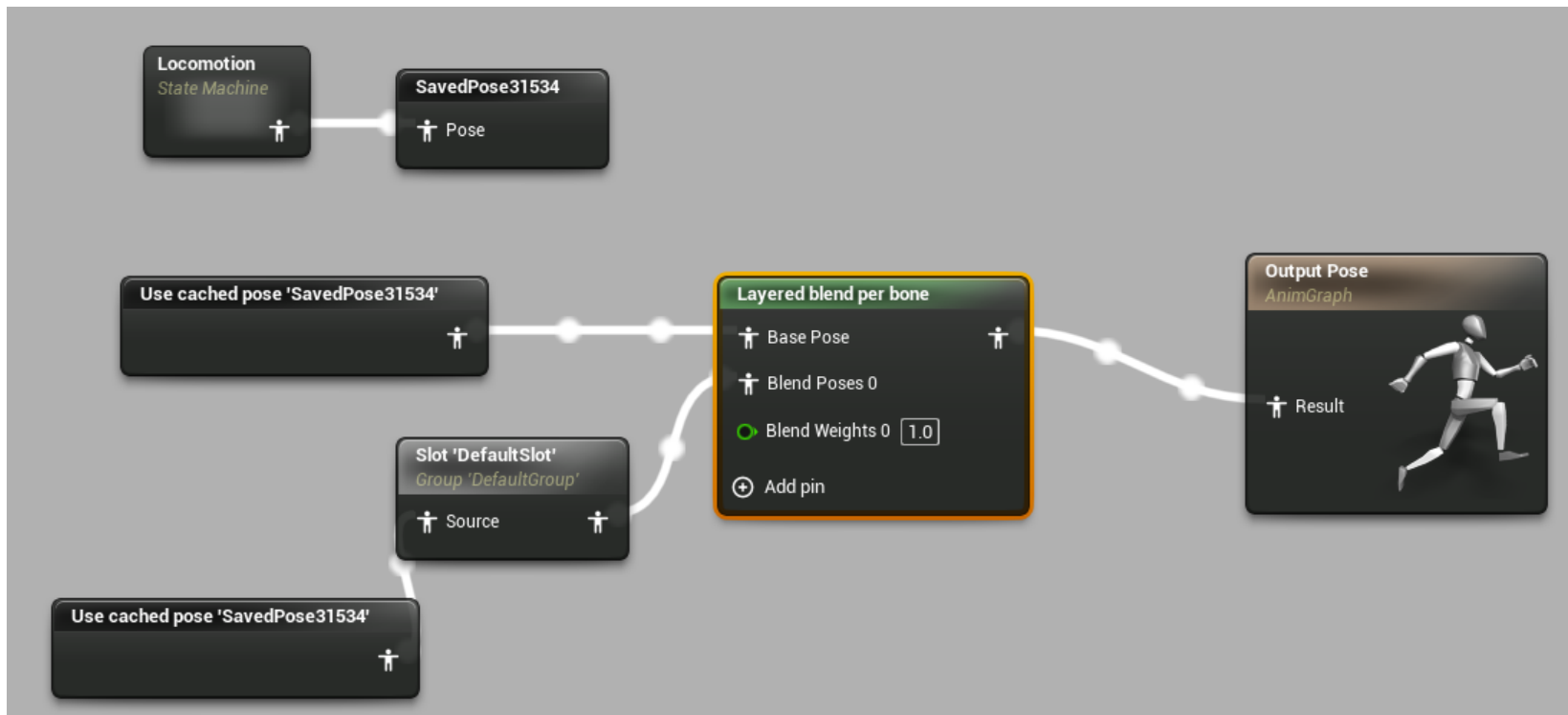
상체에 대해서, 즉 상체를 구성하는 bone에 대해서는 Right Punch 애니메이션이 블렌딩됩니다.



이 bone 레이어 비율이 1.0, 즉 100% 이므로, 상체는 Right Punch 애니메이션만 적용됩니다.

캐시 포즈

- 애니메이션 결과 출력을 임시로 저장하고, 추후에 다른 노드의 입력 데이터로 활용



학습 내용

- 인터랙션의 종류와 구현 방식
- 충돌 검사 및 충돌 처리 절차
- 칼질 구현
- 공질 구현

인터랙션(Interaction)

- 게임 안에서 발생하는 액터 간의 상호 작용

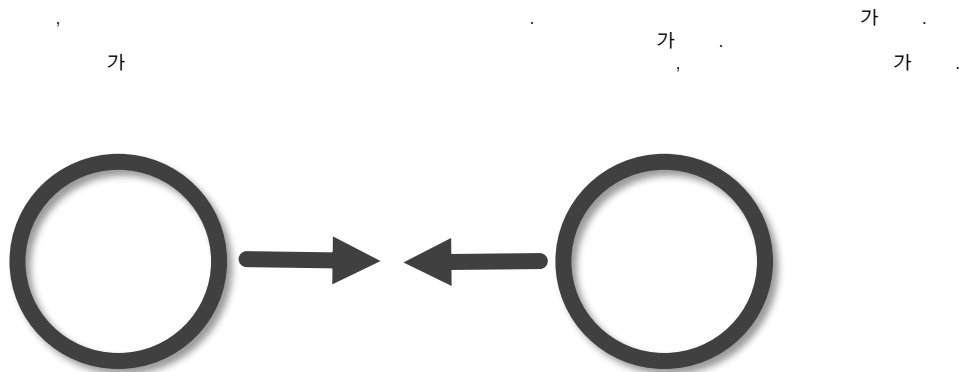
- 대표적인 인터랙션

- 공질 - 야구 피칭, 수류탄 투척 A가 가 B . 가
- 칼질 - 칼공격, 펀칭 A B . 가 .
- 총질 가 . 가 , .

인터랙션 구현

■ 물리(Physics) 엔진에 위임

- 전적으로 현실 세계의 물리 법칙을 따르도록 하는 방법
- 충돌 검사 및 충돌 처리 모두 물리 엔진에 의존
- 정교한 만큼, 계산 및 실행 부하가 매우 큼

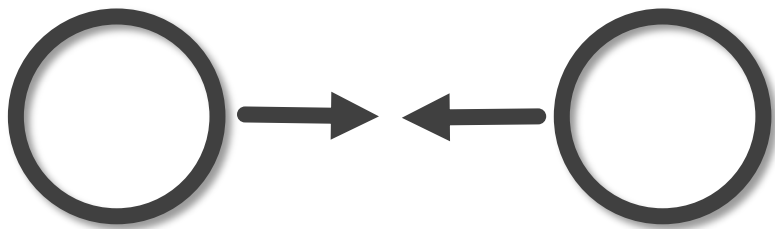


공과 공이 충돌하면, 서로 튕겨나감.

인터랙션 구현

■ 직접 개입해서 일일이 처리

- 액터의 움직임 - 애니메이션 또는 Transform을 통해 직접 액터 제어
- 충돌 검사와 충돌 처리를 필요한 만큼의 정확도만 갖도록 구현 - 계산 부하가 작음.
- 충돌 검사 - 물리적 접촉 확인(Volume Check, Ray Check) <<
- 충돌 처리 - 접촉에 따른 처리(Collision Response)
 - 객체 간 communication
 - A와 B가 충돌했을 때, 처리의 주체는? A도 될 수 있고, B도 될수 있고, 둘다도..



공과 공이 충돌하면, 한쪽이 사라짐.

가 ' 가 Rotate 가

스태틱 메시의 충돌 설정

- Simple Shape – primitive 를 이용한 근사적 설정
- Complex Shape – 3D 모델 mesh polygon
- 스태틱메시액터 자신이 물리적으로 운동을 하려면, 반드시 Simple Shape 이 설정되어 있어야 함.

물리 시뮬레이션 충돌 처리

- **대원칙**
 - 자신이 움직이는 상황에서 자신에 대한 물리 시뮬레이션을 할 때는 simple collision shape을 사용함.
 - 자신이 가속되는 상황에서는 simple collision 이 사용됨. 따라서 simple collision 사용이 활성화되어야 함.
 - 자신이 고정되어 있는 즉, 시뮬레이션 되어 있지 않은 상황에서는 simple, complex 모두 사용 가능
 - 결국 complex vs complex 에서 둘 다 운동이 되는 상황은 처리되지 않음.
 - Simple collision 이 셋업되어 있지 않으면 자신에 대한 피직스 시뮬레이션 불가.
 - Primitive 영역은 simple collision 을 설정하는 영역임.
- **Use simple as complex : simple로 다 처리**
- **Use complex as simple: complex 로 다 처리. 따라서 phy simulation 불가.**

충돌 검사 및 처리 설정 기본 절차

■ 1. 객체마다 오브젝트 유형을 설정

오브젝트 유형	설 명
WorldStatic	벽돌집의 벽과 같이 어딘가에 고정되어 이동 불가능한 물체입니다.
WorldDynamic	이동가능한 또는 애니메이션될 수 있는 물체를 나타냅니다.
Pawn	말 그대로 폰 역할을 하는 물체입니다.
PhysicsBody	물리 시뮬레이션을 통해서 움직이게 되는 물체입니다
Vehicle	차량 물리 시뮬레이션을 통해서 움직이는 물체입니다.
Destructible	파괴 가능한 물체입니다.

충돌 검사 및 처리 설정 기본 절차

2. 객체 별로 충돌 반응과 트레이스 반응을 설정

- 반응의 종류는 Blocking, Overlap, Ignore 세가지
- Collision Response(충돌 반응)
 - 객체의 Collision Volume과 다른 객체의 Collision Volume 과 만났을 때의 반응
- Trace Response(트레이스 반응)
 - 객체의 Collision Volume이 사전 정의된 어떤 궤적과 만났을 때의 반응.

	Ignore	Overlap	Block
Collision Responses ?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Trace Responses			
Visibility	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Camera	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Object Responses			
WorldStatic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
WorldDynamic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pawn	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PhysicsBody	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vehicle	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Destructible	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

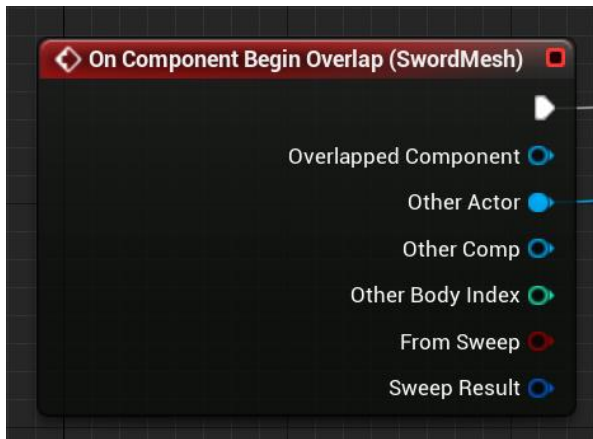
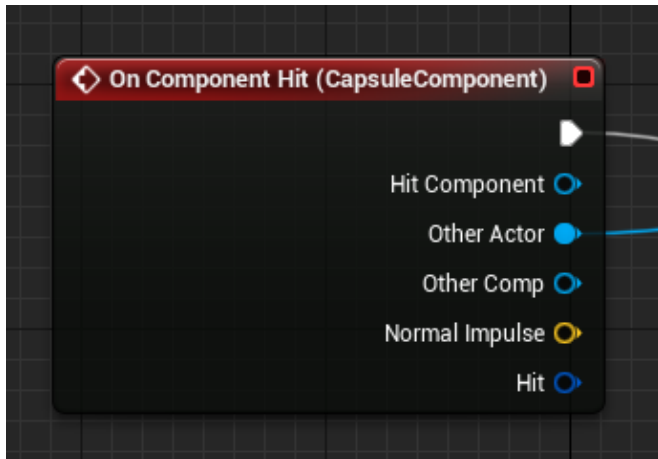
충돌 검사 및 처리 설정 기본 절차

3. 충돌 검사 여부와 수행 방식 설정

프로퍼티	설명
No Collision	콜리전 없음 - 이 바디는 물리 엔진 내 어떠한 표현이 없습니다. 공간 쿼리(레이캐스트, 스윕, 오버랩) 또는 시뮬레이션(리깃 바디, 컨스트레인트)에 사용할 수 없습니다. 이 세팅은 특히나 움직이는 오브젝트에 최적의 퍼포먼스를 냅니다.
Query Only	쿼리 전용 - 이 바디는 공간 쿼리(레이캐스트, 스윕, 오버랩)에만 사용됩니다. 시뮬레이션(리깃 바디, 컨스트레인트)에는 사용할 수 없습니다. 이 세팅은 물리 시뮬레이션이 필요치 않은 오브젝트와 캐릭터 동작에 좋습니다. 물리 시뮬레이션 트리 내 데이터를 감소시키는 것으로 퍼포먼스를 약간 개선시킬 수 있습니다.
Physics Only	피직스 전용 - 이 바디는 물리 시뮬레이션(리깃 바디, 컨스트레인트)에만 사용됩니다. 공간 쿼리(레이캐스트, 스윕, 오버랩)에 사용할 수 없습니다. 이 세팅은 본 단위의 감지가 필요치는 않은 캐릭터의 이차 시뮬레이션 동작에 좋습니다. 쿼리 트리의 데이터를 감소시키는 것으로 퍼포먼스를 약간 개선시킬 수 있습니다.
Collision Enabled	콜리전 켜짐 - 이 바디는 공간 쿼리(레이캐스트, 스윕, 오버랩)과 시뮬레이션(리깃 바디, 컨스트레인트)에도 사용할 수 있습니다.

충돌 검사 및 처리 설정 기본 절차

■ 4. Hit 이벤트 및 Overlap 이벤트에 대한 처리



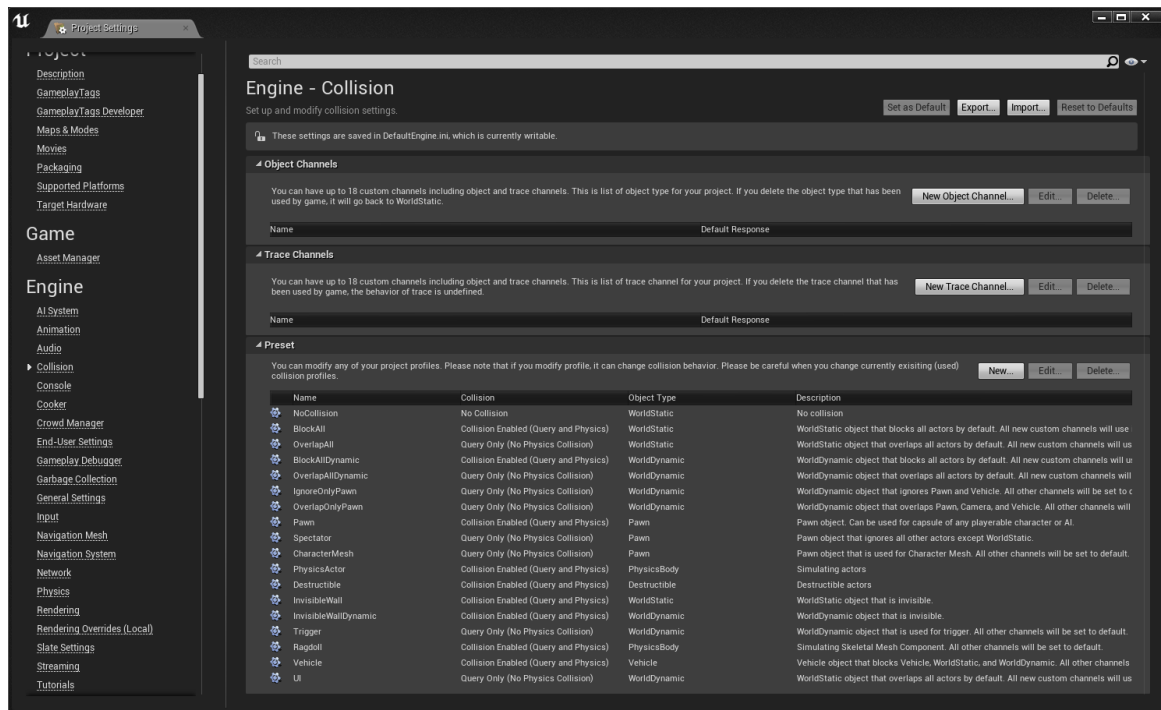
충돌 프리셋

- 다른 컴포넌트와의 충돌 반응을 사전에 정의한 것. 18개 기본 정의.



프리셋 설정 내용

- Project Settings → Engine → Collision 에서 정의됨.
- 필요에 따라 사용자가 커스텀 프리셋을 만들 수 있음.



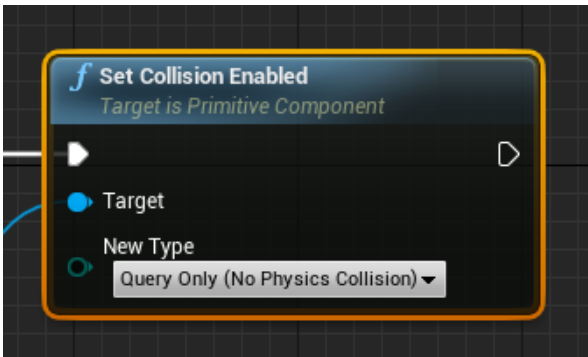
충돌 설정 변경

- 실행 중, 액터 전체의 충돌 설정을 변경할 수 있음.



충돌 설정 변경

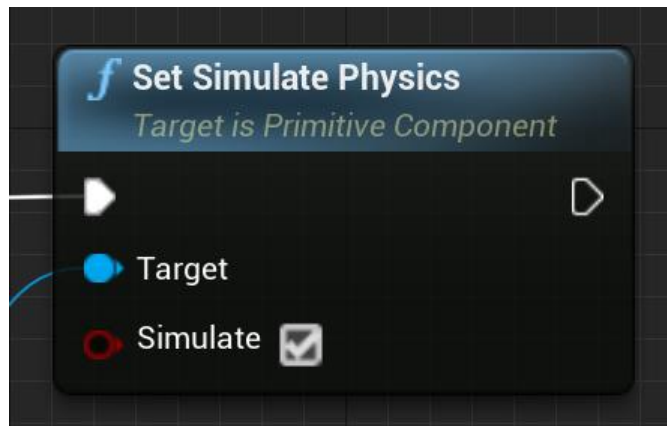
- 실행 중, 컴포넌트의 충돌 여부를 변경할 수 있음.



Overlap 이벤트 발생 여부 변경



물리 시뮬레이션 여부 설정 변경



충돌 영역 표시 : Alt+C





실습 칼질

실습 목표

- 애니메이션 특정 구간을 알리는 애니메이션 노티파이 활용
- Overlap 이벤트에 의한 충돌 처리

Animation Montage

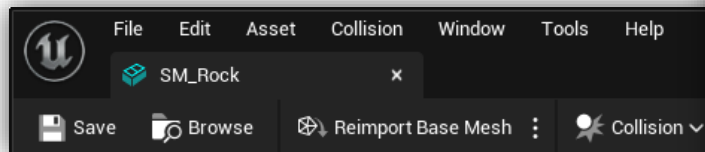
- Mixamo 애니메이션 준비
- 몽타주 생성
- 사운드 추가
- 액션 매핑 - Left Mouse 버튼
- Slash 애니메이션 재성



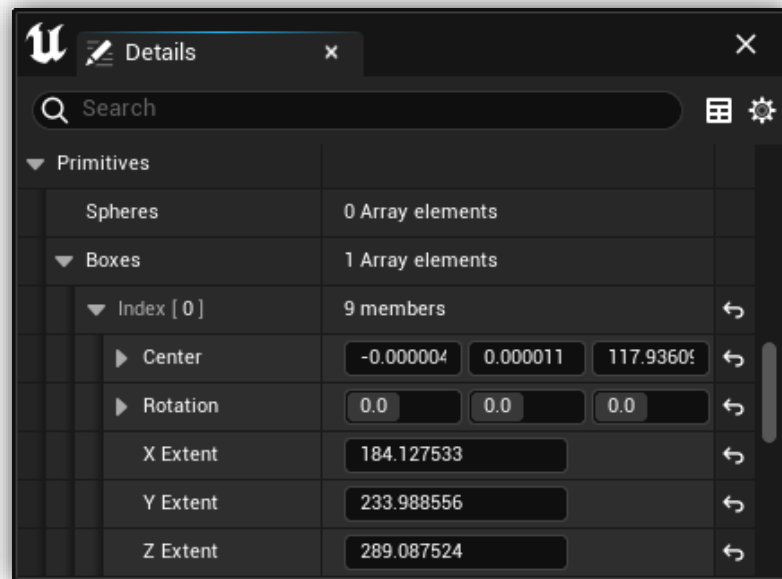
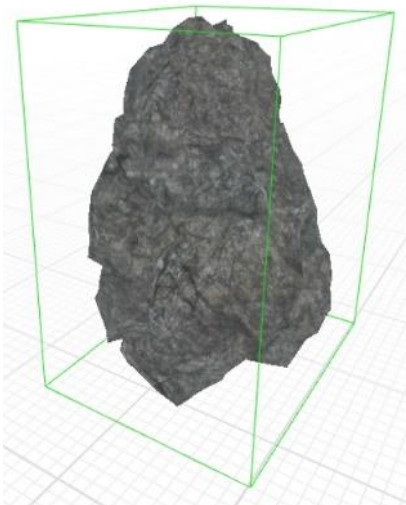
Stable Sword Outward Slash



바위 충돌 영역 설정



- Add Sphere Simplified Collision
- Add Capsule Simplified Collision
- Add Box Simplified Collision
- Add 10DOP-X Simplified Collision
- Add 10DOP-Y Simplified Collision



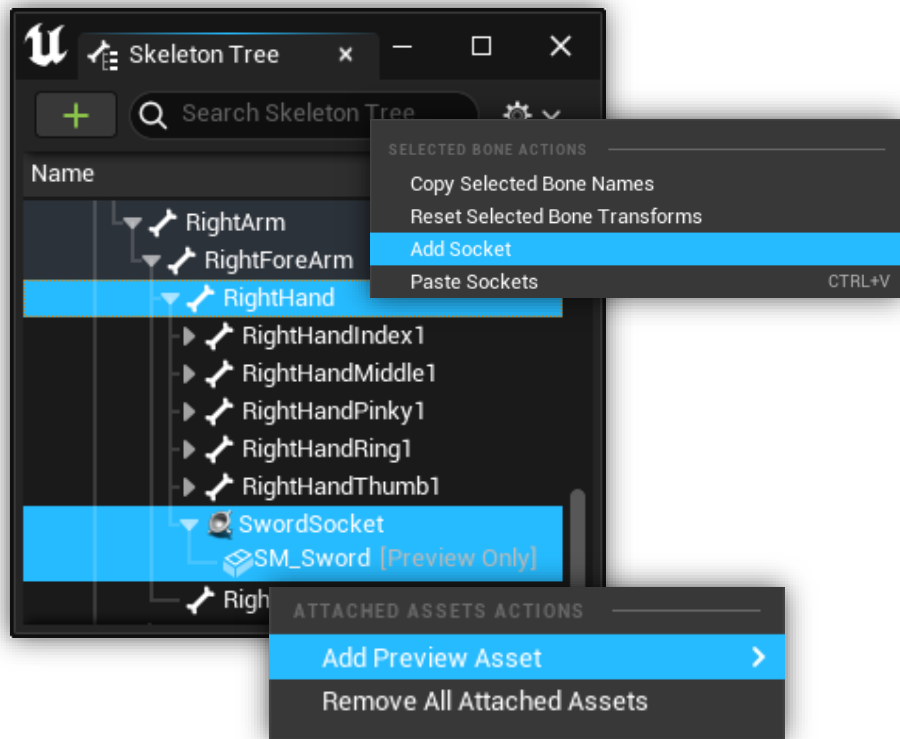
대검 총돌 영역 설정



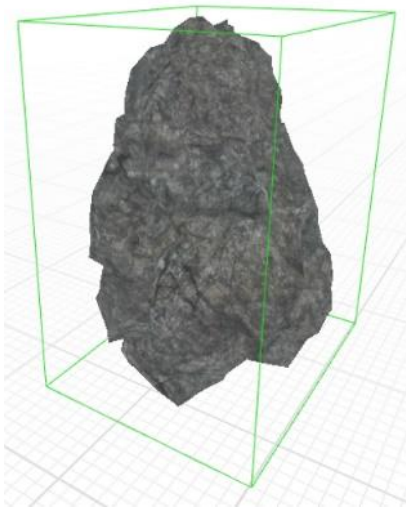
▼ Primitives			
Spheres	0 Array elements		
▼ Boxes	1 Array elements		
▼ Index [0]	9 members		↶
▶ Center	0.000008	-5.0	75.0 ↶
▶ Rotation	0.0	0.0	0.0 ↶
X Extent	5.446594		↶
Y Extent	20.0		↶
Z Extent	70.0		↶

Infinity Blade:Weapons 에서 다운로드

소켓 추가



오브젝트 유형 설정



WorldStatic



WorldDynamic

충돌 반응 설정

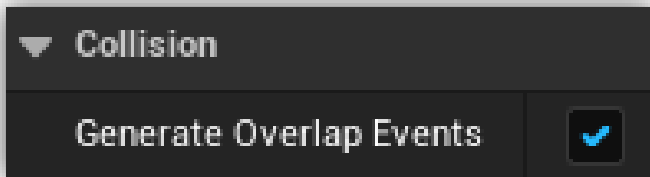
Object Responses			
WorldStatic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
WorldDynamic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pawn	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PhysicsBody	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vehicle	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Destructible	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

바위

	Ignore	Overlap	Block
Collision Responses ?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Trace Responses			
Visibility	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Camera	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Object Responses			
WorldStatic	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
WorldDynamic	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pawn	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PhysicsBody	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Vehicle	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Destructible	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

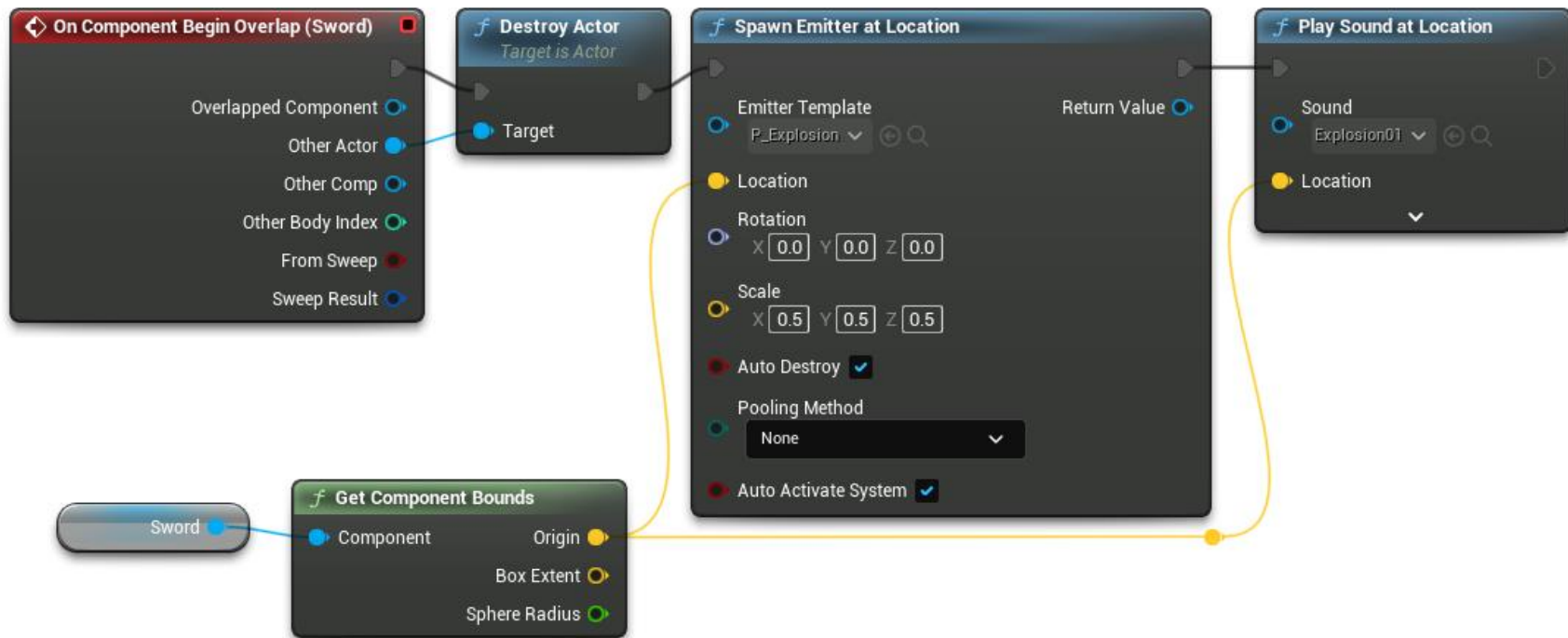
대검

오버랩 이벤트 발동 여부 설정



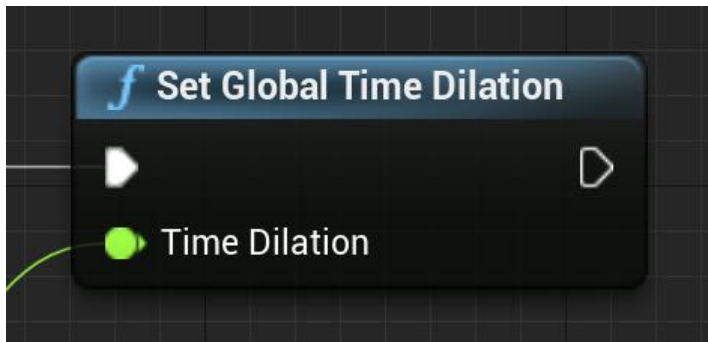
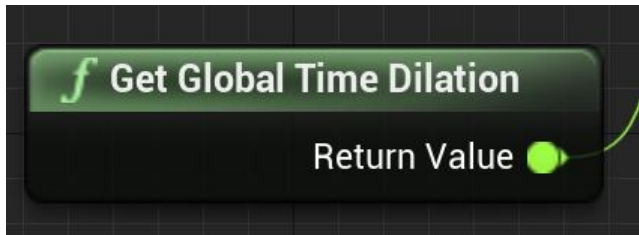
바위, 대검 양쪽 모두 발동 설정이 되어
있어야 함.

대검에 맞는 물체의 파괴

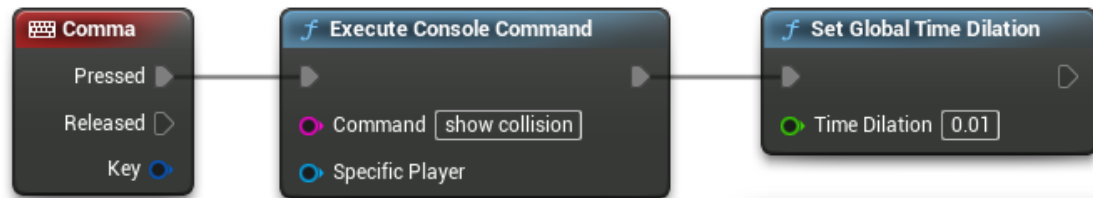


Global Time Dilation 설정

- 시간 단위의 축소 확대를 통해 게임 실행 속도를 가속 또는 감속시킬 수 있음.
- 값이 1보다 크면, 원래보다 빠르게 진행됨.
- 값이 1보다 작으면, 원래보다 천천히 진행됨.

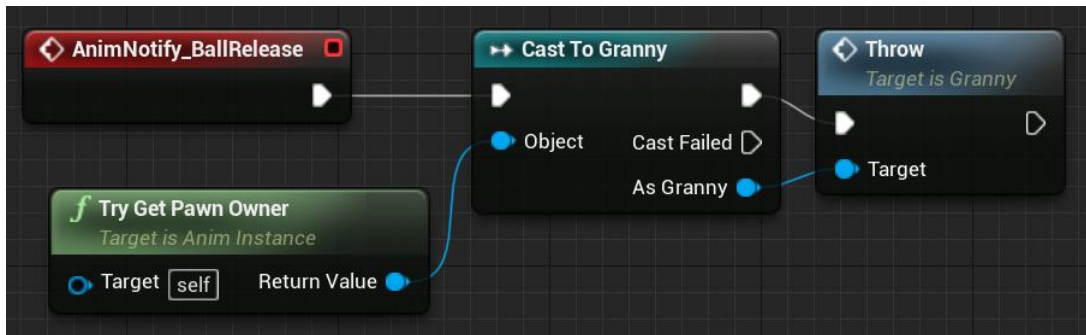
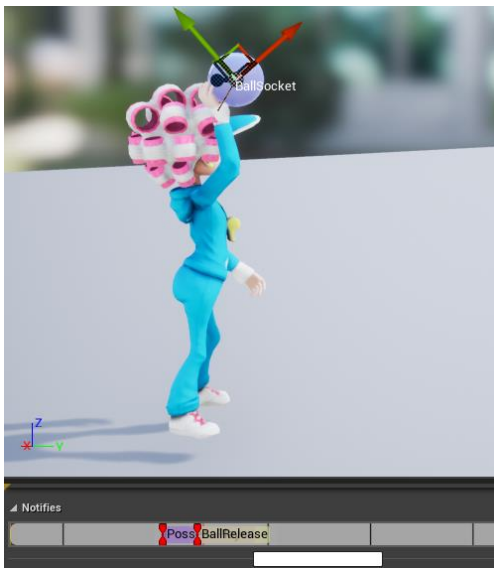


디버그 표시 설정



애니메이션 노티파이(Animation Notify)

- 애니메이션 도중의 특정 지점에서 이벤트를 발생시킴.
 - 사용사례) 걷기, 달리기 도중 발소리 사운드 재생
- 애니메이션 블루프린트 안에서 이벤트 처리가 이루어짐.

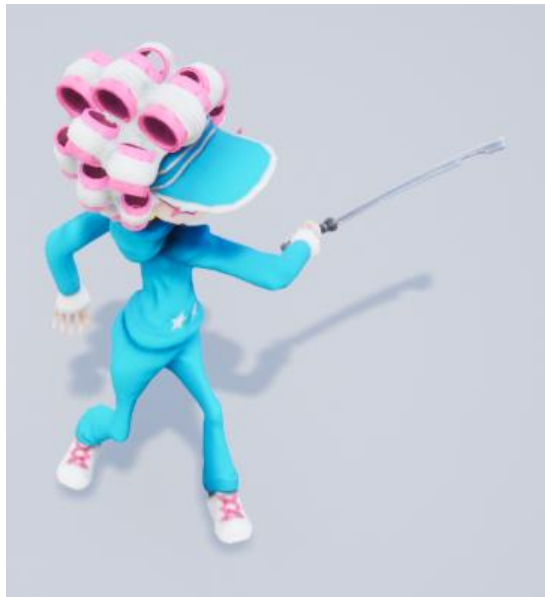


타격 시작점과 끝지점에 Animation Notifys 설정 – 애니메이션 몽타주에서 설정

- 이 구간에서만 Overlapping 확인

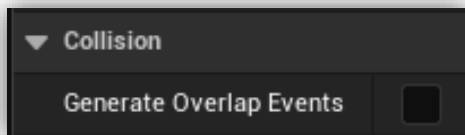


Slash Start

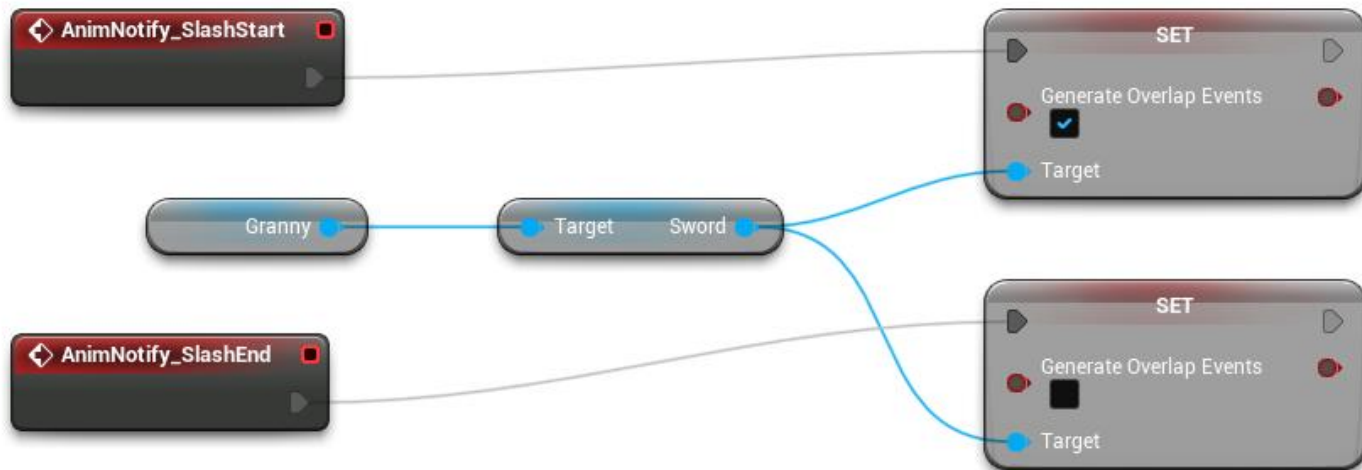


Slash End

시작과 끝 지점에서 오버랩 발생 제어



기본 설정은 이벤트 무발생



DAY 4-2: Projectile Movement



실습 공 던지기

실습 목표

- ProjectileMovement 컴포넌트를 이용한 공 던지기 구현
- 애니메이션 노티파이를 이용한 인터랙션 동기화



Ball 블루프린트의 공 궤적 구현

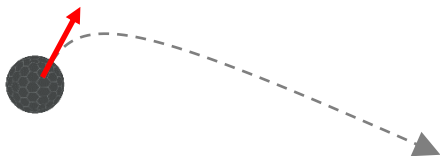
- 방법 #1: 직접적인 물리 엔진 활용

- 중력, 질량을 설정하고, 충격량을 가함.

- 방법 #2: ProjectileMovement 컴포넌트 활용.

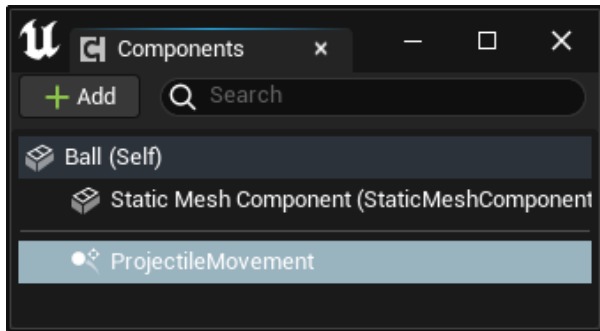
- 날라가는 총알과 같은 발사체(Projectile)의 물리적인 움직임을 시뮬레이션

발사체 운동 컴포넌트를 통해 적색 화살표 방향으로 초기 속도를 부여하면, 공은 그 방향으로 이동을 시작하다가 중력에 의해서 점차 아래로 떨어지게 됩니다.



공 블루프린트

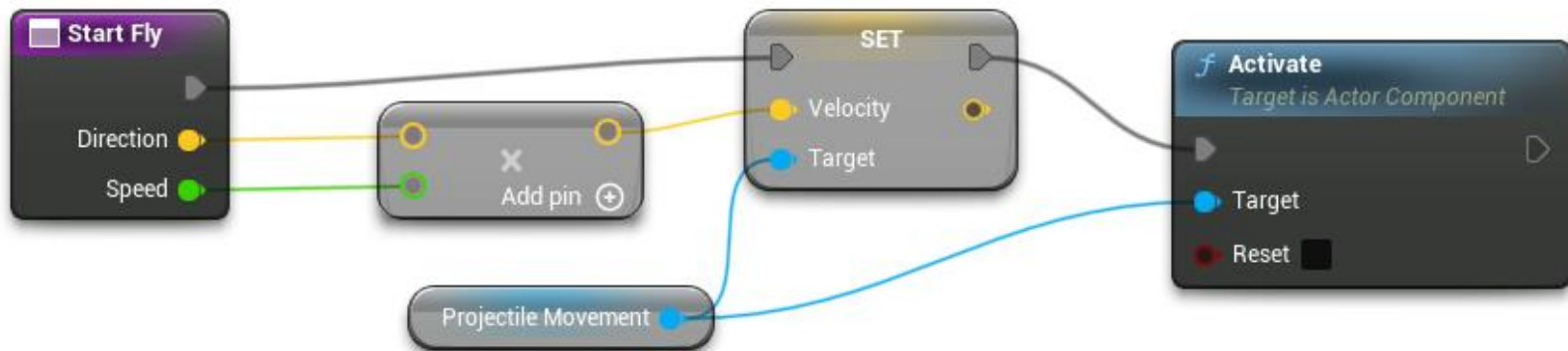
속도값은 기본적으로 공의 로컬 좌표계가 기준입니다. 따라서 할머니가 바라보는 목적지 방향으로 공을 던지려면 이 값을 False로 변경해야 합니다.



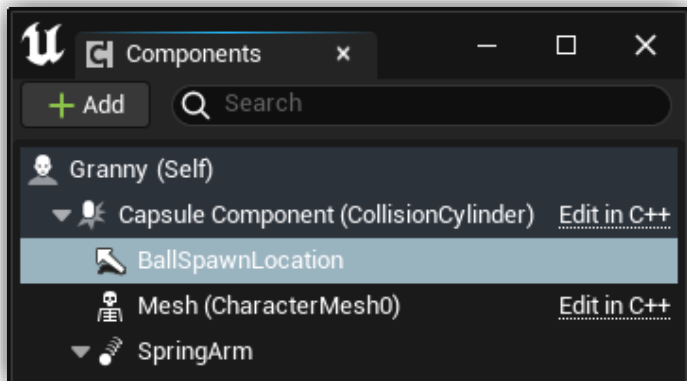
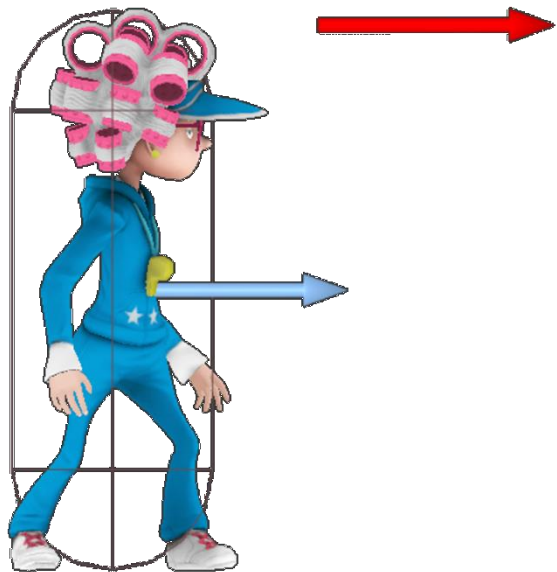
속성	값
Initial Velocity in Local Space	<input checked="" type="checkbox"/>
Auto Activate	<input checked="" type="checkbox"/>

이 값을 False로 변경해서 자동으로 공의 운동이 시작되는 것을 막도록 합니다.

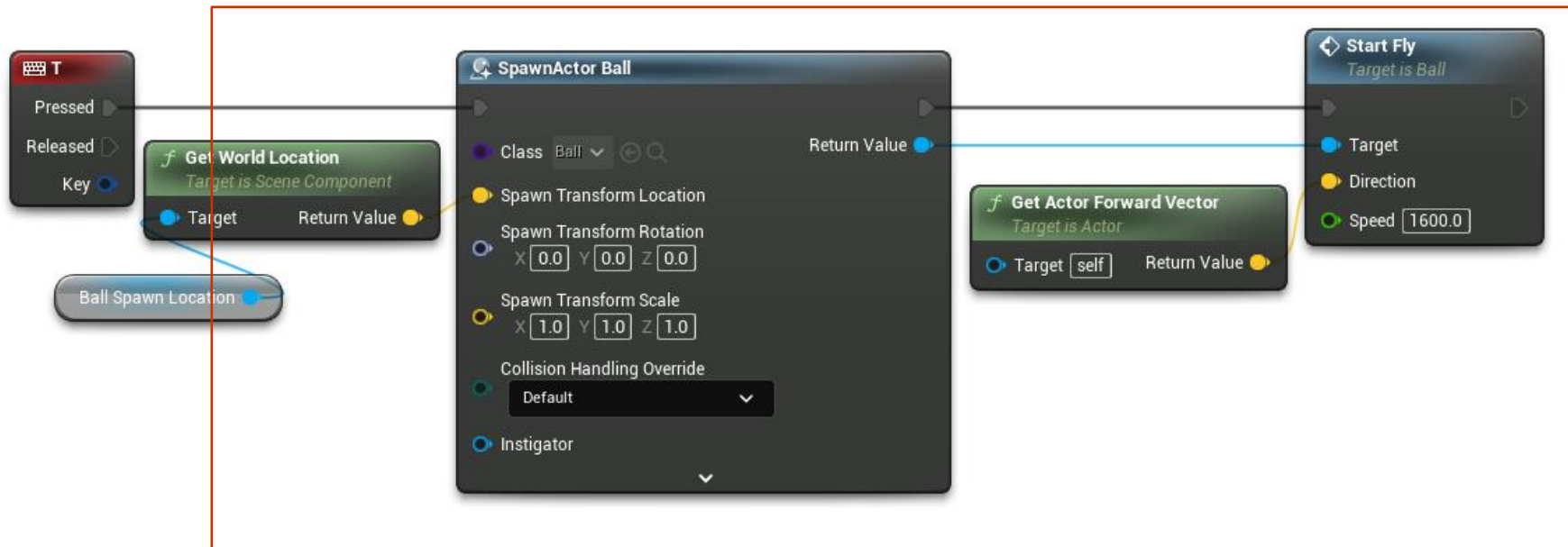
Projectile Movement 컴포넌트의 활성화



공 생성 위치 지정



공 발사와 비행



Throw 함수로 변환

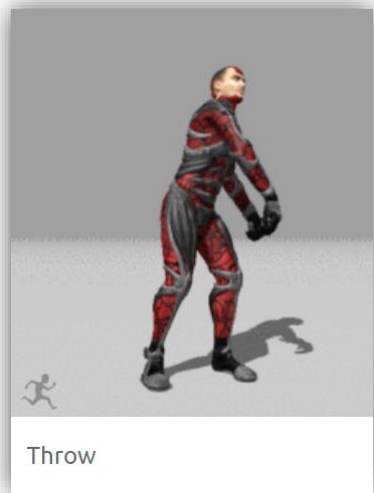
공의 오브젝트 유형 및 충돌 반응 설정

▼ Collision Presets		BlockAllDynamic ▼	↩
Collision Enabled	Collision Enabled (Que ▼)		
Object Type	WorldDynamic ▼		
	Ignore	Overlap	Block
Collision Responses ?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Trace Responses			
Visibility	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Camera	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Object Responses			
WorldStatic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
WorldDynamic	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pawn	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PhysicsBody	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vehicle	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Destructible	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Simulation Generates Hit Events	<input type="checkbox"/>		

공과 충돌되는 액터들의 제거

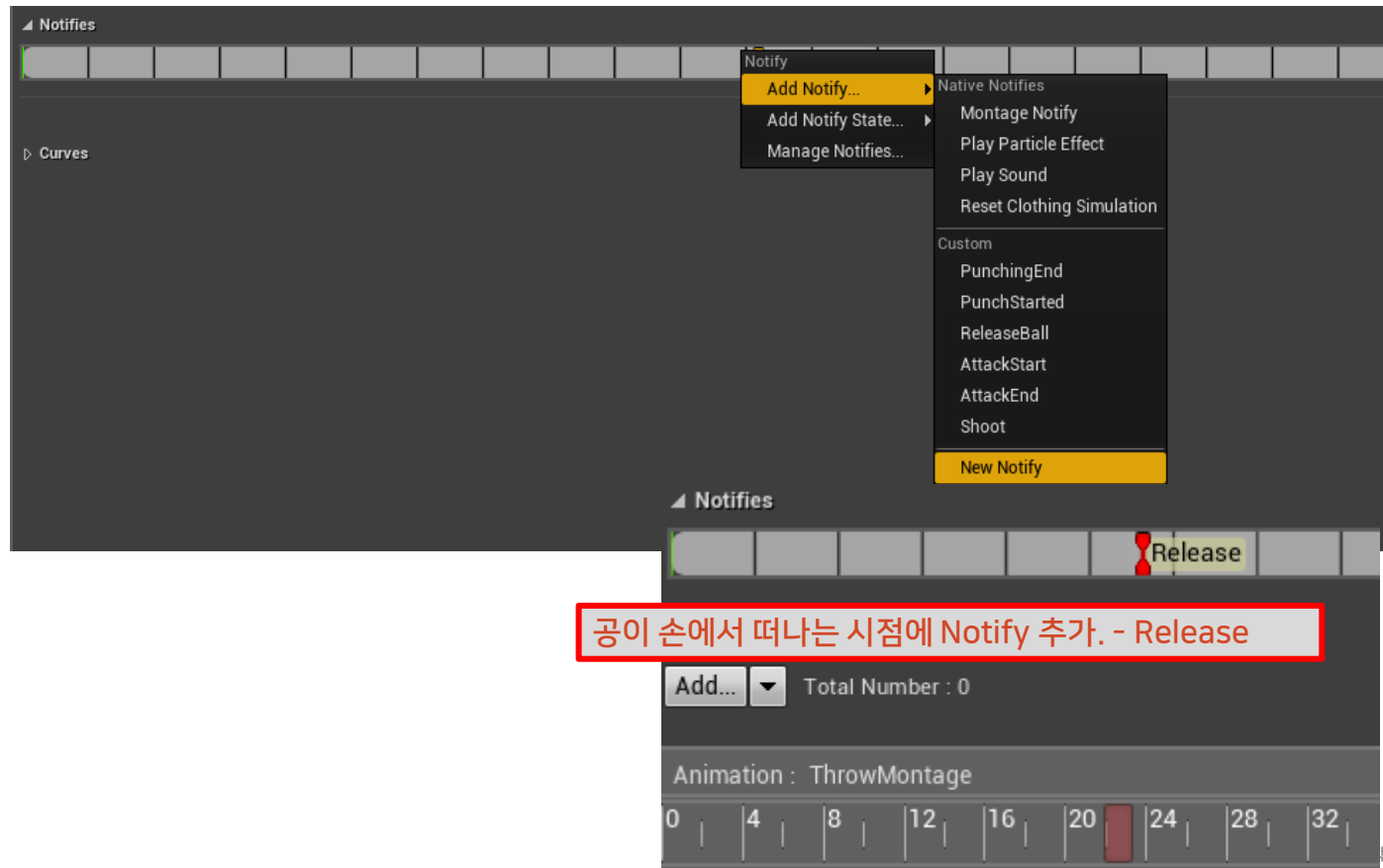


던지기 애니메이션

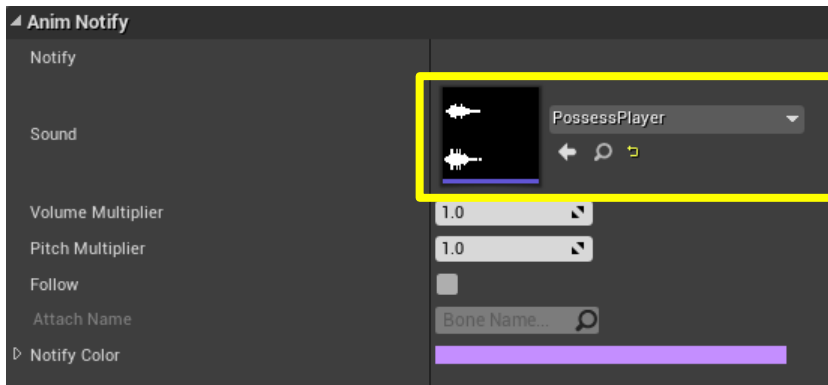


Notifies									
2				▶ ThrowBall					
2				▶ PossessPlayer					

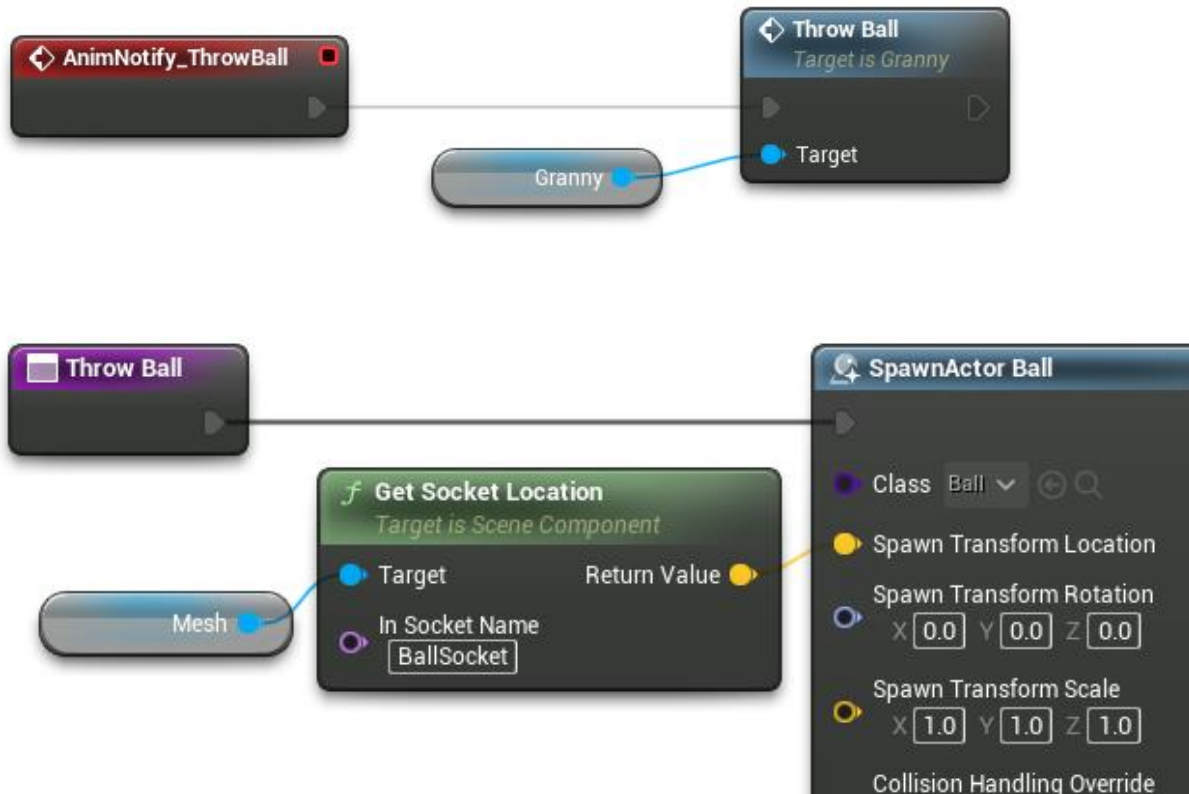
공을 놓는 시점 통보



Sound Notify - 공 던지는 사운드 플레이



애니메이션 노티파이에 따른 Throw Ball 호출



충돌 처리 관련 유의 사항

- 두개의 오브젝트가 서로를 막으려면, 둘다 모두 Block 설정이 되어 있어야 함.
 - 하나는 Block 이고, 다른 하나는 Overlap이면, 겹치는 것은 가능하지만, 차단은 일어나지 않음.
- Block으로 설정된 두 액터가 있을 때, Simulation Generate Hit Events 옵션이 켜져 있는 쪽에 Event Hit 가 발생함. (액터가 물리 시뮬레이션에 의해 움직이는 경우)

충돌 처리 관련 유의 사항

- 액터 설정이 **Overlap**으로 되어 있으면, 액터들은 서로의 충돌을 무시(**Ignore**)하게 됨. 따라서, **Generate Overlap Events** 옵션 설정이 없으면, **Overlap** 과 **Ignore**는 서로 동일한 효과임.
- 한쪽이 **Ignore**, 한쪽이 **Overlap** 이면, **Overlap Event**는 발생하지 않음.

충돌 처리 관련 유의 사항

- 속도가 빠른 경우라면, 물체가 다른 물체를 막는 경우에도 Overlap event는 발생 가능함.
 - 물체가 Collision event와 Overlap event를 둘 다 처리하는 것은 권장되지 않음.
- 100% Block을 보장하려면, CCD(연속 충돌 검출:Continuous Collision Detection) 옵션을 활성화 하 함.



- Overlap 이 100% 보장되지는 않음. 객체의 속도가 매우 빠른 경우 Overlap 여부가 검출되지 않을 수 있음.
- 트레이스 반응은 100% 보장됨.

실습 시험 #2

- 주제: 공중 침투 좀비 소탕 게임

- 제한 시간 : 70분

- 게임 개요

- 플레이어는 비행접시를 타고 적진에 침투하여 좀비들을 사살하면 게임 완료

- 최소 게임 기능

- 비행 접시 위에 서있을 때는 비행접시를 컨트롤 해서 조종.
- 비행 접시 운행 중, 스페이스를 누르면 주인공은 낙하함.
- 낙하 중에, 그냥 바닥에 떨어지면 즉사. 바닥에 근접했을 때, 일정 거리 내에서 스페이스키이를 누르면, 착지를 함.
- 땅에 착지 후에는, 캐릭터 이동 모드(WASD 와 마우스로 이동. SPACE 는 점프)
- 캐릭터 이동시 Dash 할 수 있음 - Shift 키이를 누른 상태로 이동하면 DASH 모드 - 기존 속도보다 2배로 빠른 주행.
- 땅에 떨어져 있는 총을 획득해서 장착.
- 적진에 있는 좀비들을 총으로 모두 사살하면, 게임 종료.

시작 프로젝트

- e-class에 제공되는 TestStarter 프로젝트에서 시작.
 - 플레이어 캐릭터와 좀비 캐릭터의 3D 모델과 애니메이션은 기본 제공됨.
- 에디터 실행 후, StarterContent 팩 추가

채점표

학번		이름		
NO	채점 기준	배점	자기평가	평가
1	비행 접시 이동(WASD + 마우스)	1		
2	낙하 및 착지	1		
3	플레이어 이동과 점프(WASD+마우스,스페이스)	2		
4	DASH	1		
5	착지 사망 시 게임 재시작, 좀비 전부 사살시 게임 종료	1		
6	총 획득 및 장착	1		
7	총질	1		
8	좀비 사망	1		
9	총 장착시 캐릭터 이동 방식의 변경	2		
10	JUICY FACTOR	3	X	
총점		14		