A cinematic screenshot from the video game The Last of Us. A female character, Ellie, is seen from behind, standing on a rocky, desolate landscape. She is looking towards a distant, ruined city with tall, jagged spires under a dramatic, sunset-colored sky. The scene is atmospheric and serves as a background for the title text.

게임 엔진

LEC 14 인터랙션 (3)



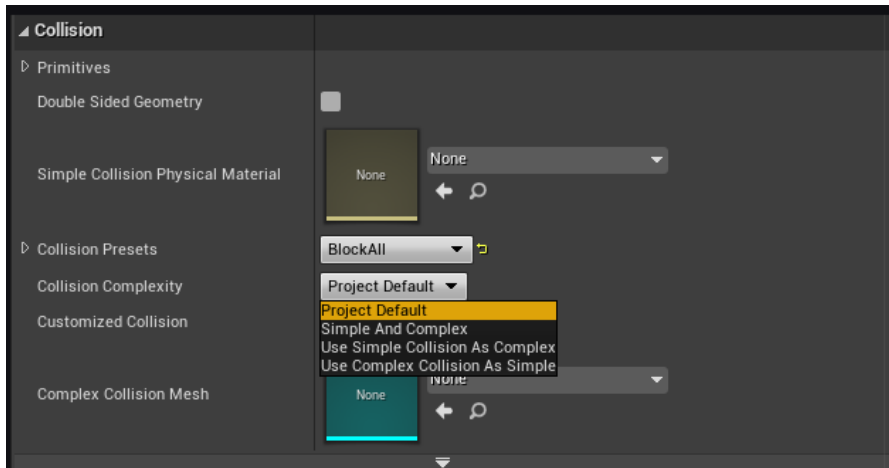
한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

스태틱 메시의 충돌 설정

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

Collision complexity 설정



세팅

설명

Default

기본 - 단순 콜리전 요청이 단순 콜리전을, 복합 요청은 복합 콜리전을 사용하도록 하는 "기본" 동작입니다.

UseSimpleAsComplex

단순을 복합으로 사용 - 복합 쿼리 요청시에도 엔진은 단순 모양에 대한 쿼리를 계속, 기본적으로 트라이메시를 무시합니다. 트라이메시를 구을 필요가 없기에 메모리가 절약되고, 콜리전 지오메트리만 단순해 지면 퍼포먼스가 향상될 수도 있습니다.

UseComplexAsSimple

복합을 단순으로 사용 - 단순 쿼리 요청시에도 엔진은 복합 모양에 대한 쿼리를 계속, 기본적으로 단순 콜리전을 무시합니다. 이를 통해 물리 시뮬레이션 콜리전에 트라이메시를 사용할 수 있습니다. 참고로 **UseComplexAsSimple** 사용중인 경우 오브젝트 시뮬레이션은 불가능하나, 다른 시뮬레이션 (단순) 오브젝트와의 충돌은 가능합니다.

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 shape, complex shape 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 수행 방식

프로퍼티	설명
No Collision	콜리전 없음 - 이 바디는 물리 엔진 내 어떠한 표현이 없습니다. 공간 쿼리(레이캐스트, 스윕, 오버랩) 또는 시뮬레이션(리짓 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(리짓 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(리짓 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(리짓 바디, 컨스트레인트)에도 사용할 수 있습니다.

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, 연속적인 히트에 대해서도)**

충돌 처리 관련 유의 사항

- 속도가 빠른 경우라면, 물체가 다른 물체를 막는 경우에도 **Overlap event**는 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.

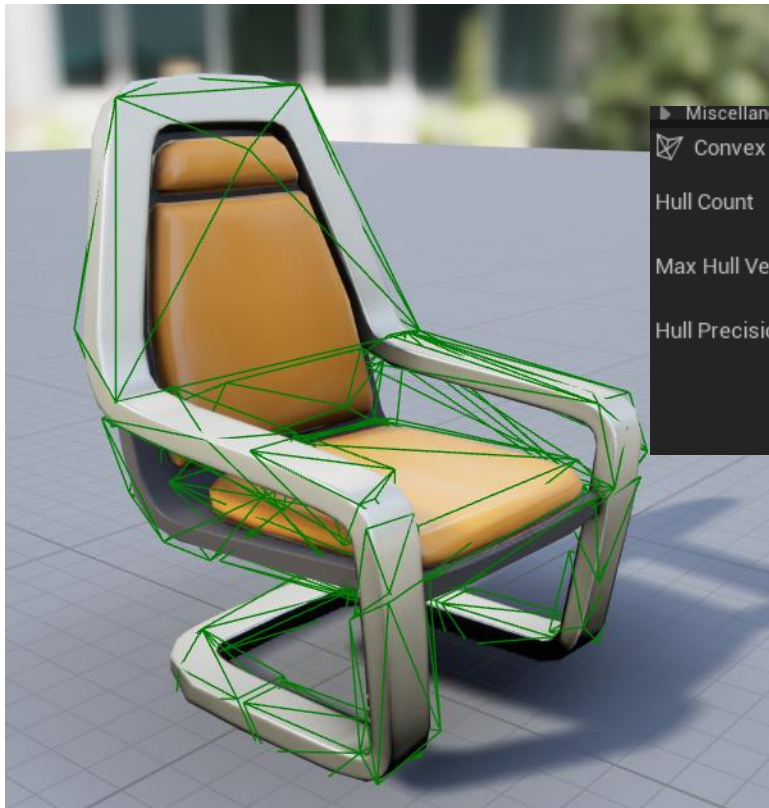
오버랩 관련 유의 사항

- 어느 한쪽이라도 Overlap으로 되어 있으면, 충돌(Hit)은 발생하지 않음.
- 한쪽이 Ignore, 한쪽이 Overlap 이면, Overlap 은 발생하지 않음.
- Generate Overlap Events 옵션이 양쪽에 다 활성화되어 있어야 이벤트가 발생함.

언리얼 엔진의 물리 시뮬레이션 한계 실험



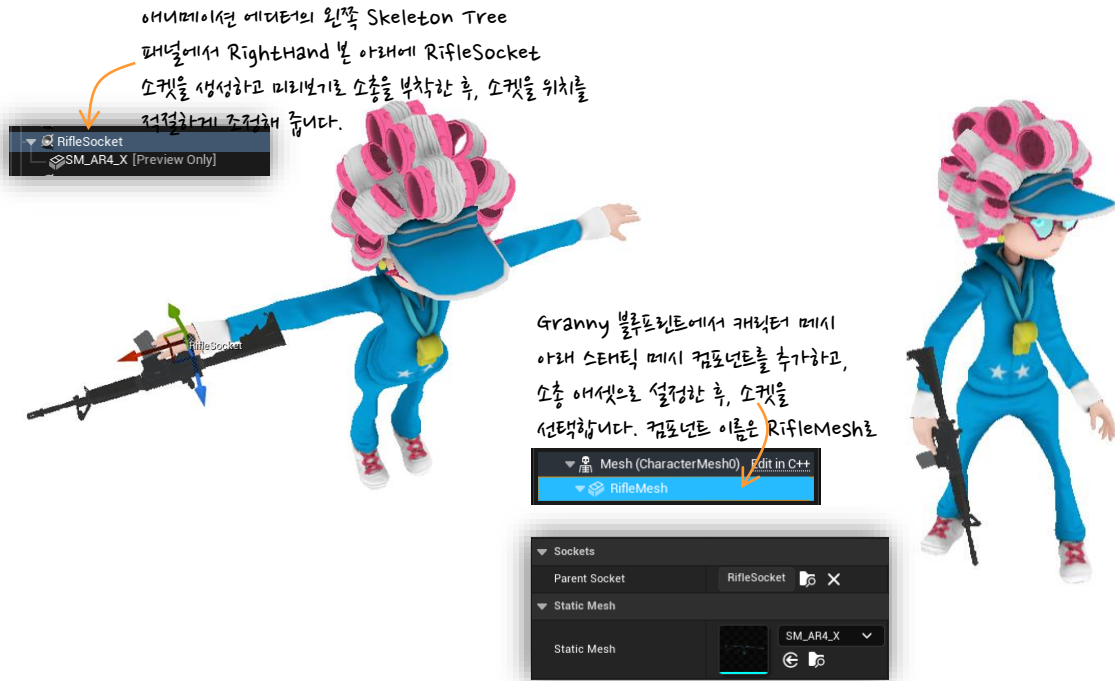
Simple collision 자동 생성



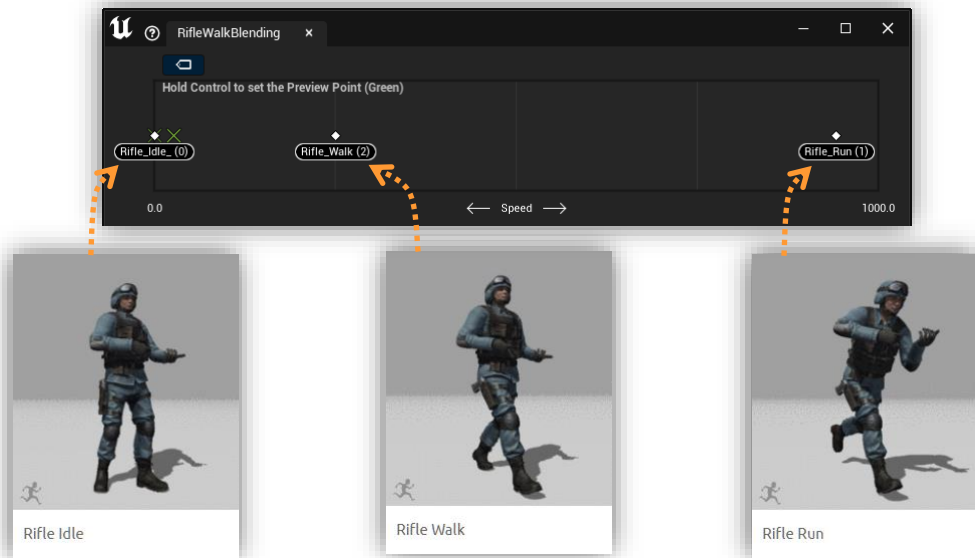


실습 총질 구현

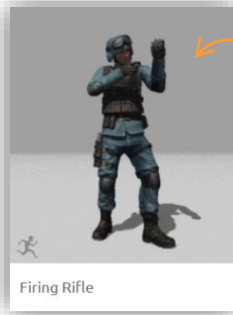
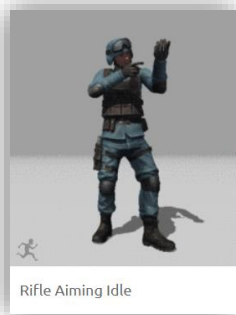
소총 부착



소총 장착 애니메이션 블렌드 스페이스



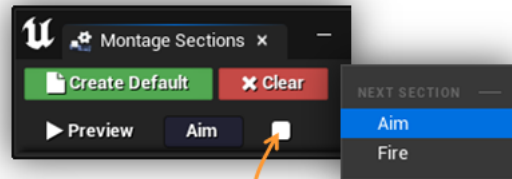
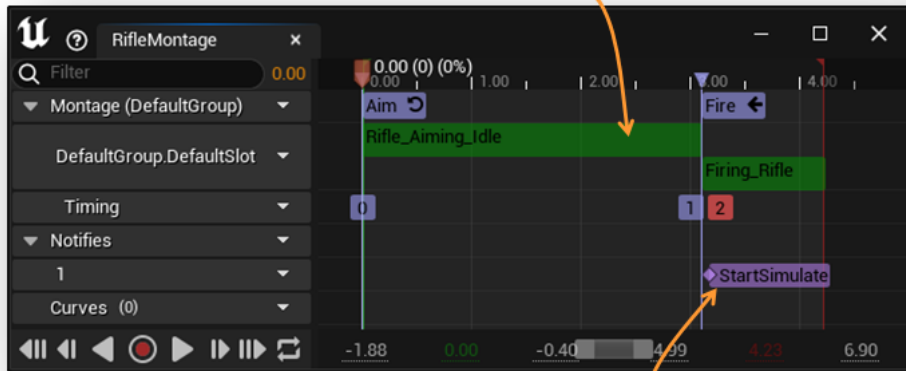
조준과 사격 애니메이션



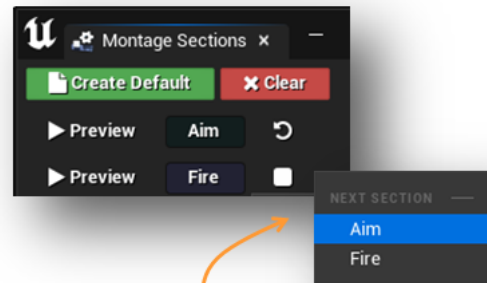
사격 애니메이션이 여러 개 있습니다. 이
중에서 한발만 쏘는 애니메이션을 찾으도록
하세요.

애니메이션 몽타지

- 1 Rifle Aiming_Idle과 Firing_Rifle 애니메이션을 트랙에 차례로 배치합니다. 그리고 Aim 섹션과 Fire 섹션을 생성하고 위치를 알맞게 조절해 줍니다.



- 2 이곳을 눌러 Aim의 다음 섹션을 지정할 수 있습니다. 다음 섹션으로 Aim, 즉 자신을 선택하면 Aim 섹션을 반복 재생하게 됩니다.

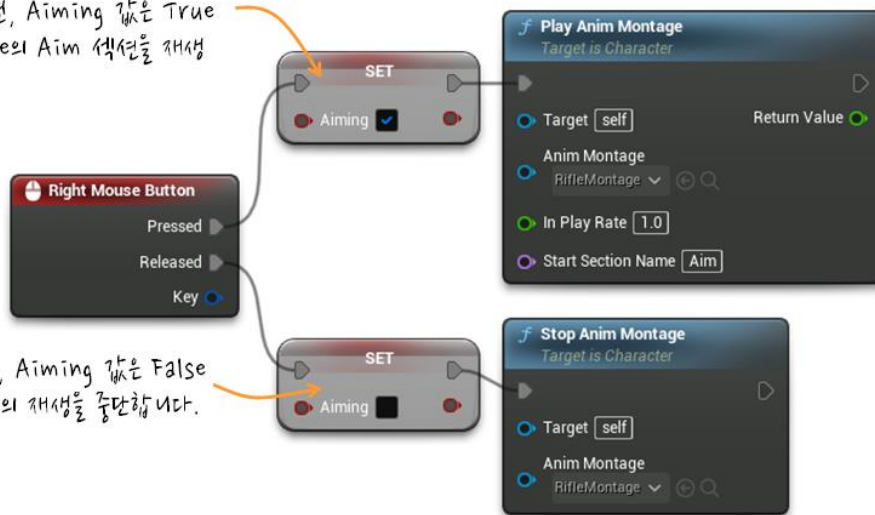


- 4 발사시 총성을 내기 위해 사운드 노티파이를 Fire 섹션의 전반부에 추가합니다. 엔진 내부 사운드 애셋인 StartSimulate를 선택하면 그걸 듣게 됩니다.

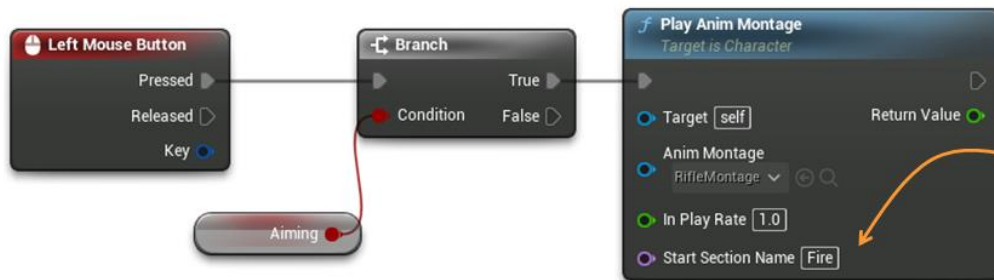
- 3 마찬가지로 보령으로 이번에는 Fire 섹션의 다음 섹션을 지정합니다. Aim 섹션을 다음 섹션으로 지정하면, Fire 섹션 다음에 Aim 섹션을 재생하게 됩니다.

조건 및 발사 입력 연결

오른쪽 마우스 버튼을 누르면, Aiming 값은 True가 되고, RifleMontage의 Aim 섹션을 재생합니다.

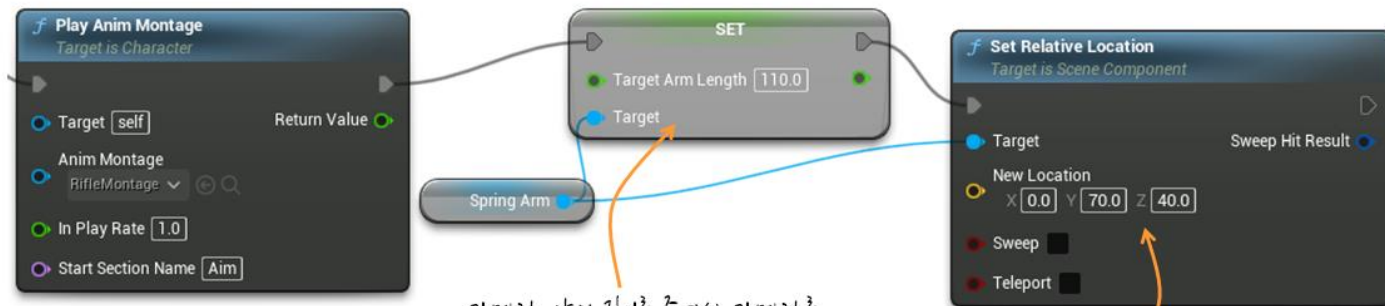


오른쪽 마우스 버튼을 떼면, Aiming 값은 False가 되고, RifleMontage의 재생을 중단합니다.

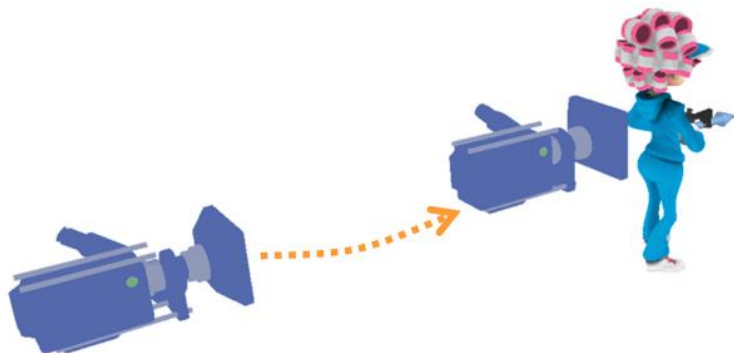


조건 상태에서 왼쪽 마우스 버튼을 누르면 Fire 섹션을 재생합니다.

조준 모드



카메라 암의 길이를 줄여서 카메라를
캐릭터에 가까이 이동합니다.

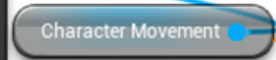


캐릭터 뒤에서 봤을 때, 카메라의 위
치를 우측 상단으로 이동합니다.



이제 캐릭터는 화면 기준으로 전진 방향을 항상 보고 있어야 하므로, 진행 방향에 따라 회전하는 기능을 비활성화합니다.

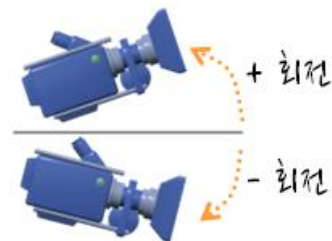
캐릭터가 플레이어 컨트롤러에 따라서 회전하도록 설정합니다. 현재 카메라 암 역시 플레이어 컨트롤러를 따르도록 되어 있기 때문에, 캐릭터와 함께 카메라도 같이 회전합니다.



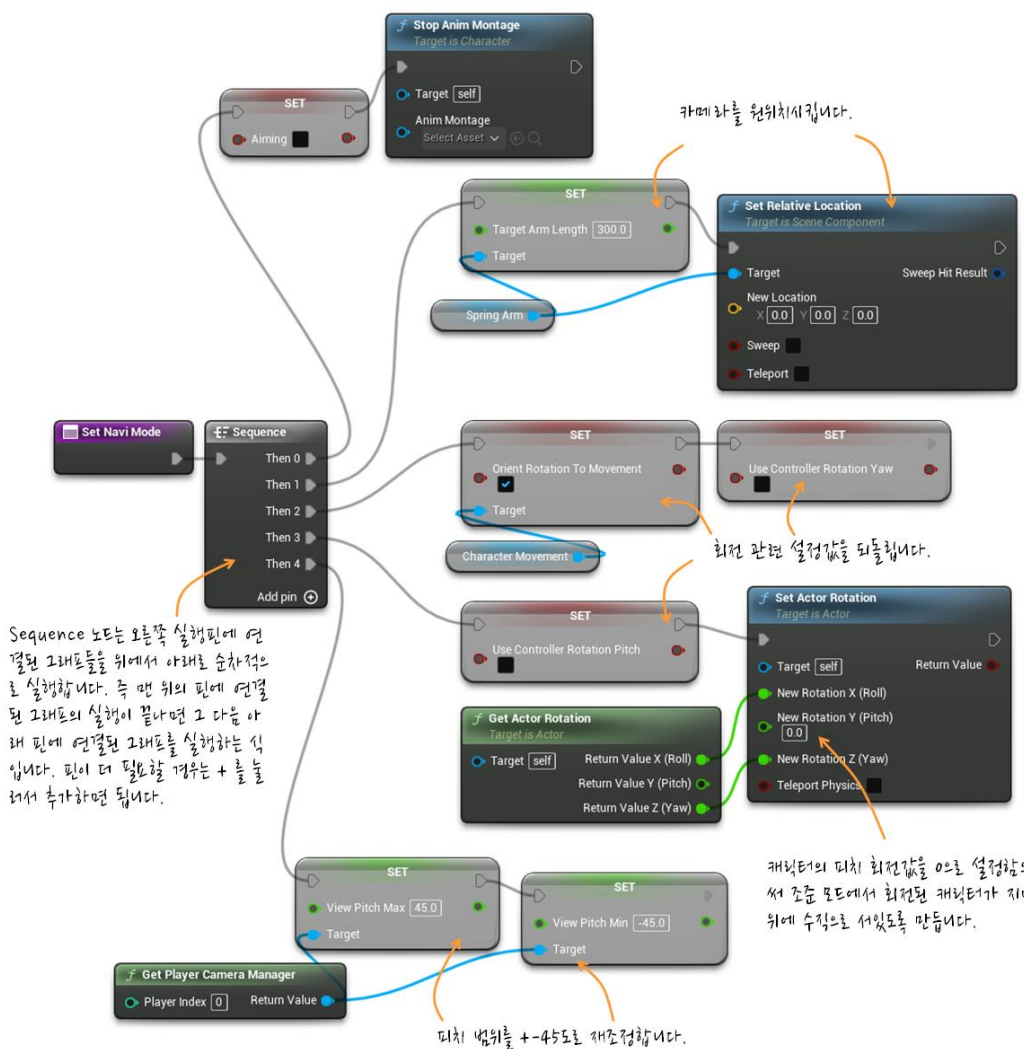
왼쪽의 컴포넌트 패널에서 character Movement 컴포넌트를 드래그해서 배치합니다.

카메라 피치 제한

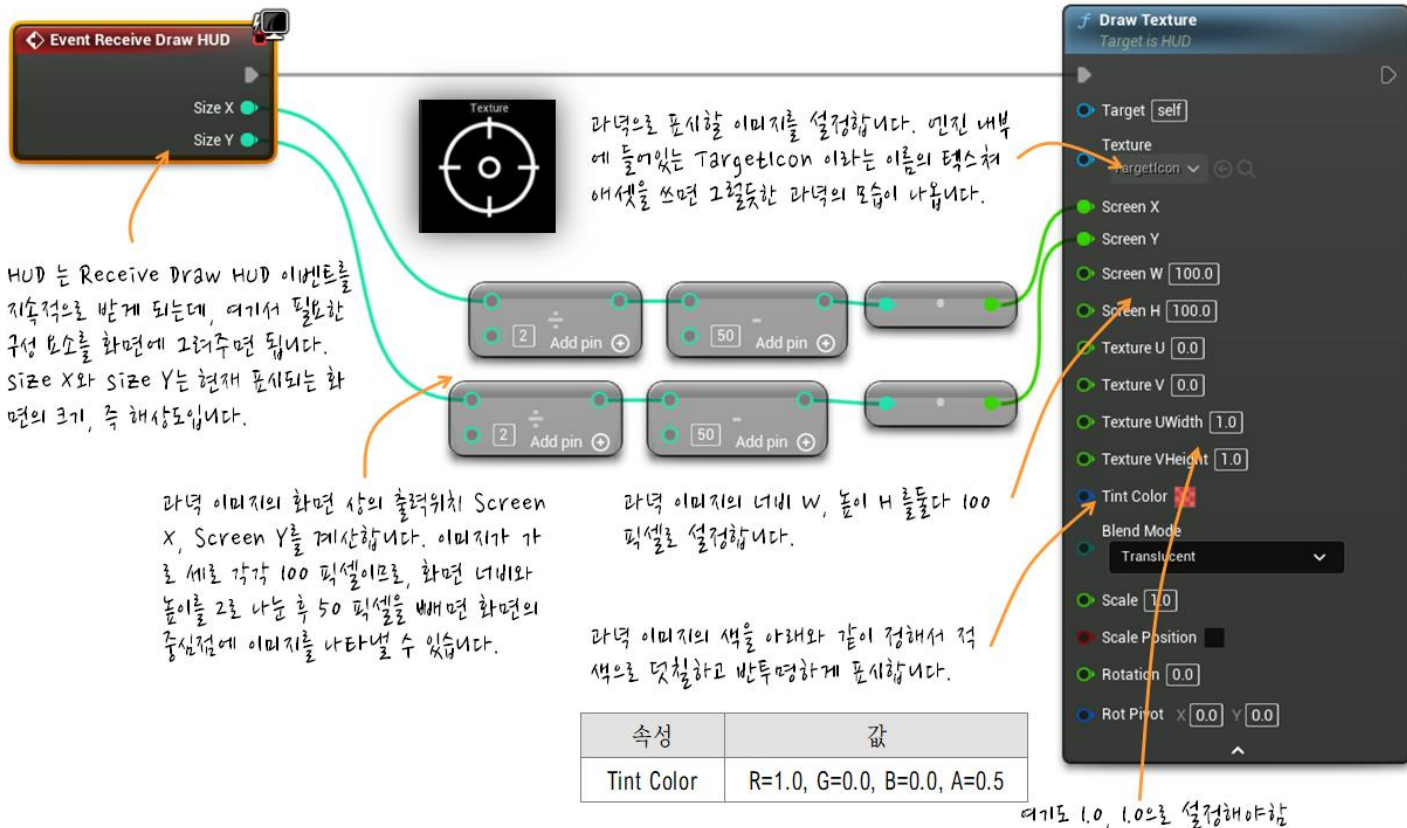
피치각도의 최대 최소 범위를 설정합니다.
여기서는 ± 10 도 정도면 적당합니다.



네비 모드



과녁 HUD



HUD 조절

플레이어가 시작되면 처음에는 HUD가 표시되지 않아야 하므로, BeginPlay 이벤트에서 처리합니다.



TargetHUD 변수를 하나 만들고, 여기에 현재 화면에 나오고 있는 HUD 액터를 구해서 저장합니다.

'Show HUD'를 False로 설정해서 HUD가 표시되지 않도록 만듭니다.



조준 모드에서는 HUD를 표시합니다.

주행 모드에서는 HUD를 숨깁니다.

