A character with dark hair in a bun, 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.

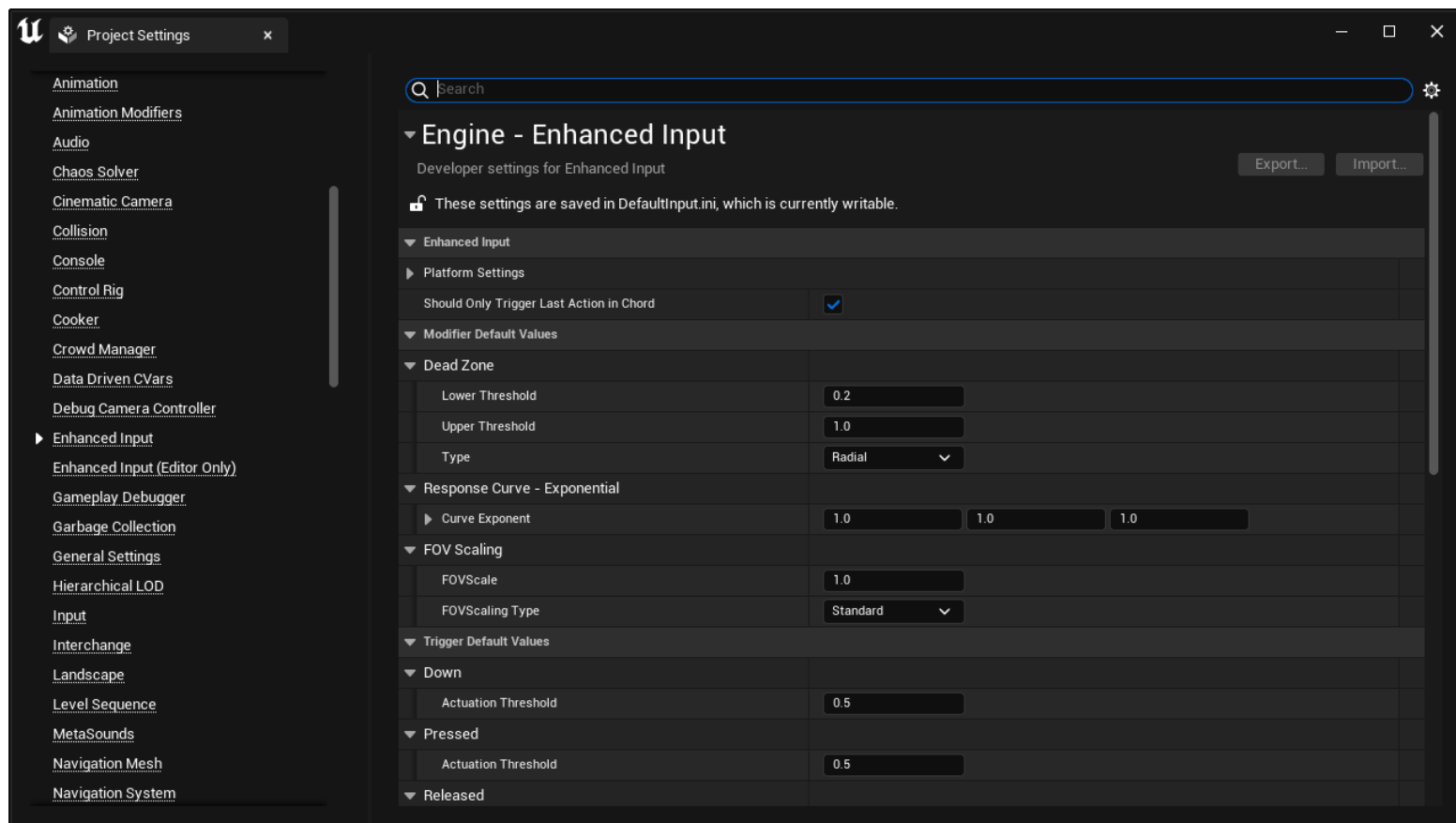
게임 엔진

# LEC 09 캐릭터



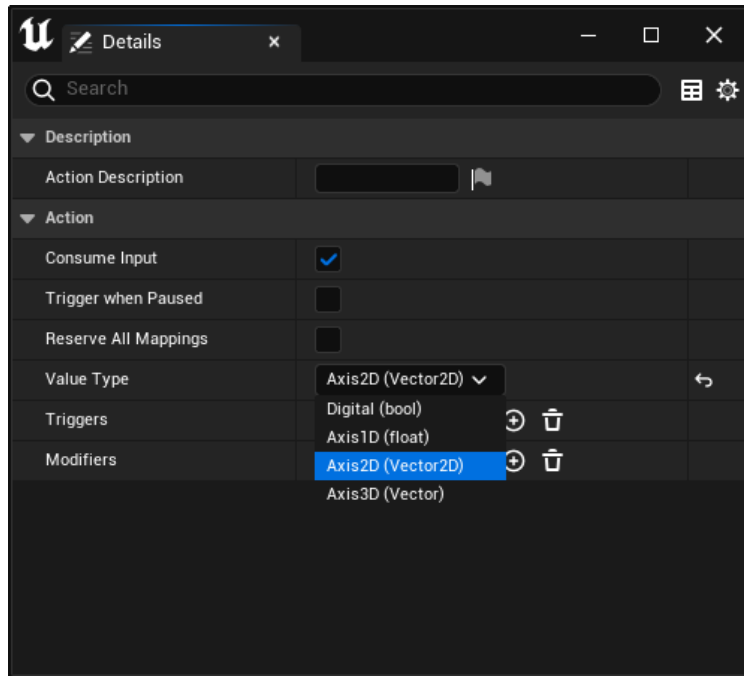
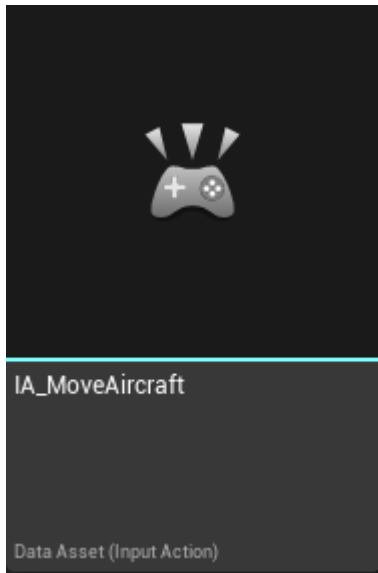
# LEC 08 복습

# 언리얼 엔진 입력 시스템 - UE5 - Enhanced Input



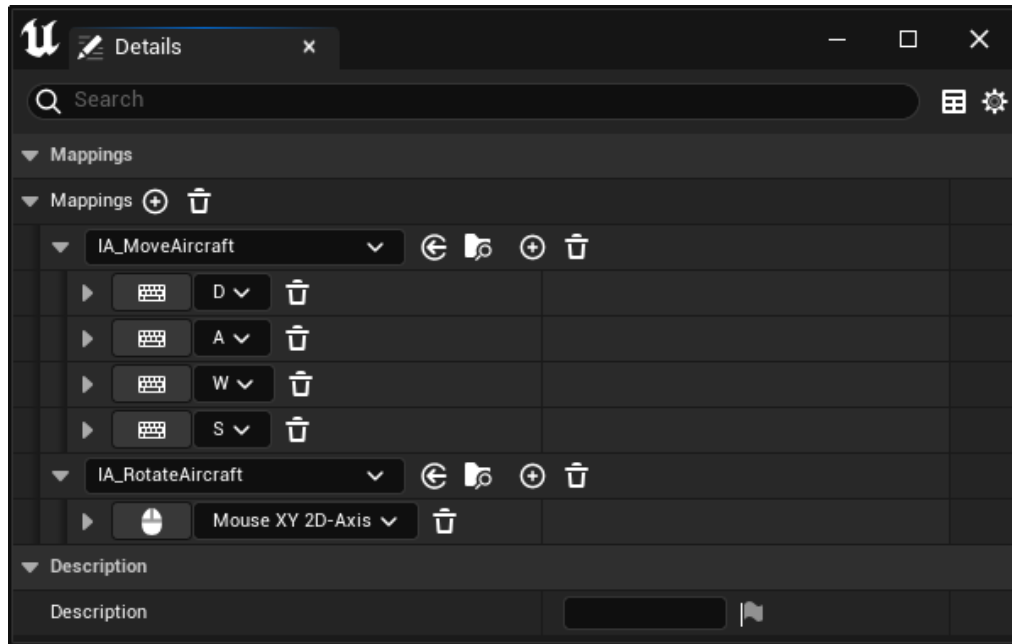
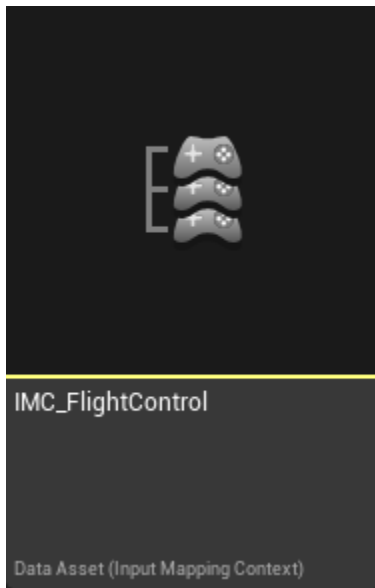
# Input Action

- 입력의 이름과 타입을 정의

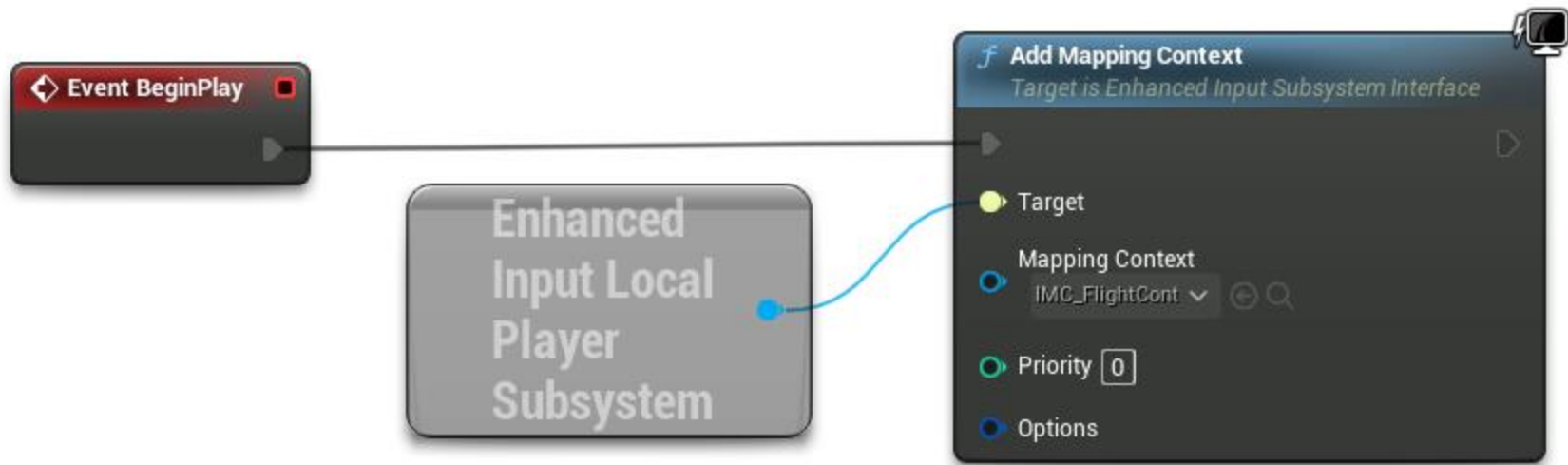


# Input Mapping Context

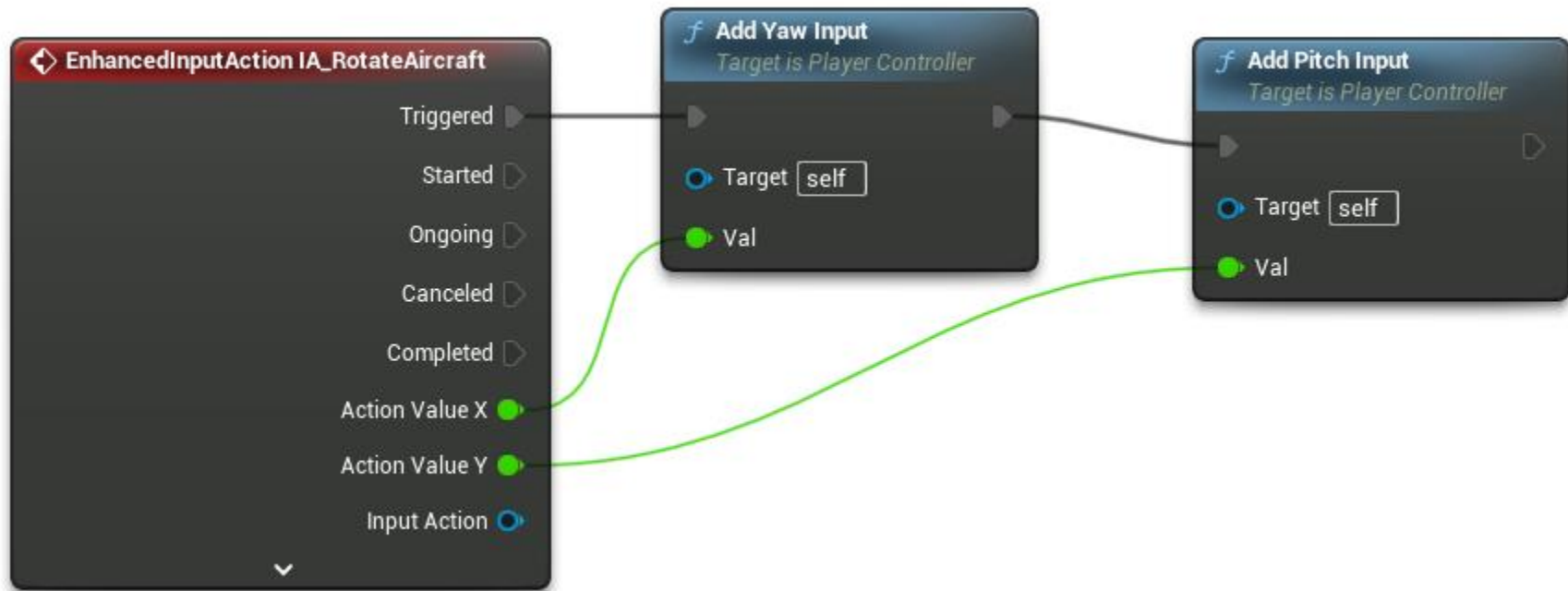
- Input Action과 실제 입력(키보드, 마우스, 조이스틱)을 연결
- 입력 인가(Trigger) 조건을 설정
- 입력값을 추가 조정(Modifier)



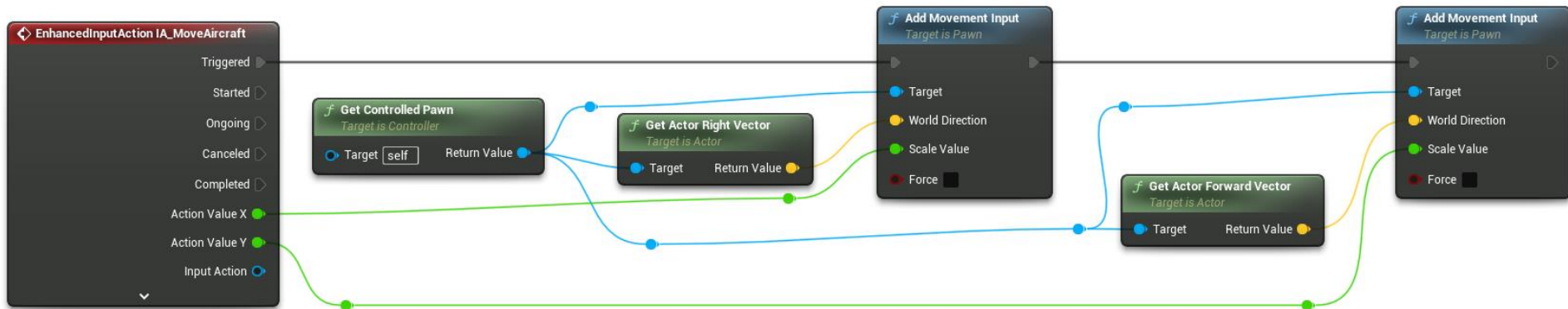
# AircraftController – 입력 시스템 연결



# AircraftController – 회전

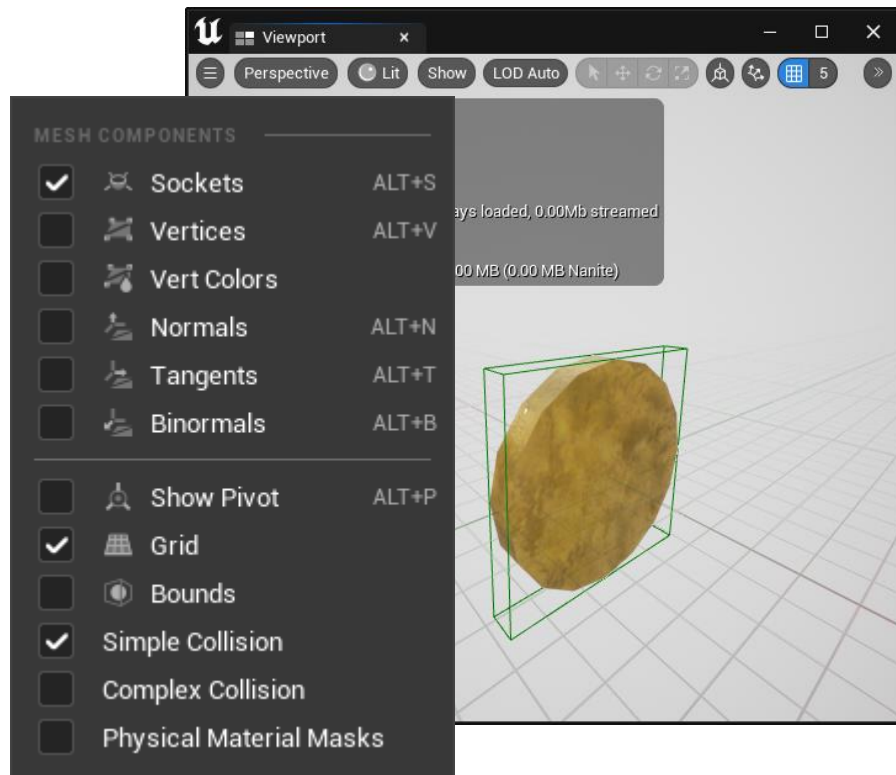


# AircraftController – 이동

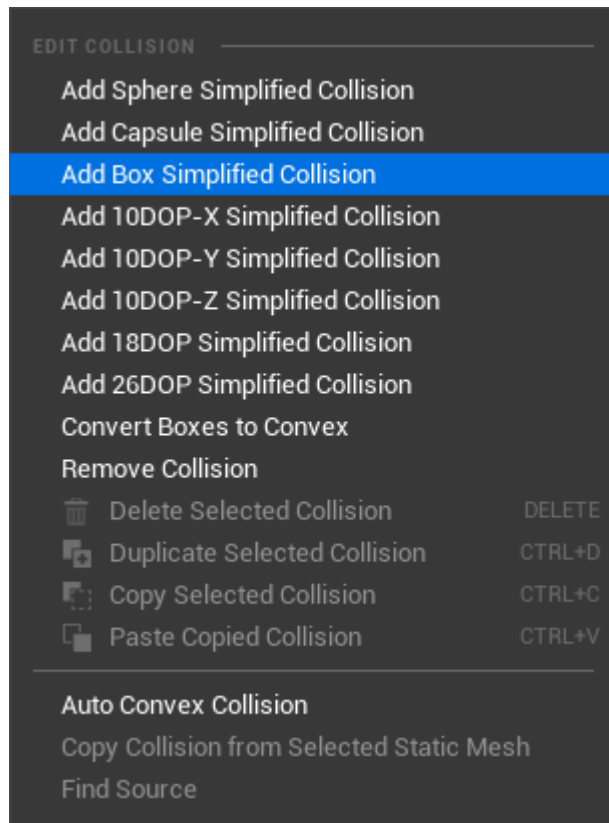




# Coin 메쉬의 collision box 확인



Collision 이 설정되어 있지 않으면, 물리  
시뮬레이션이 되지 않음.

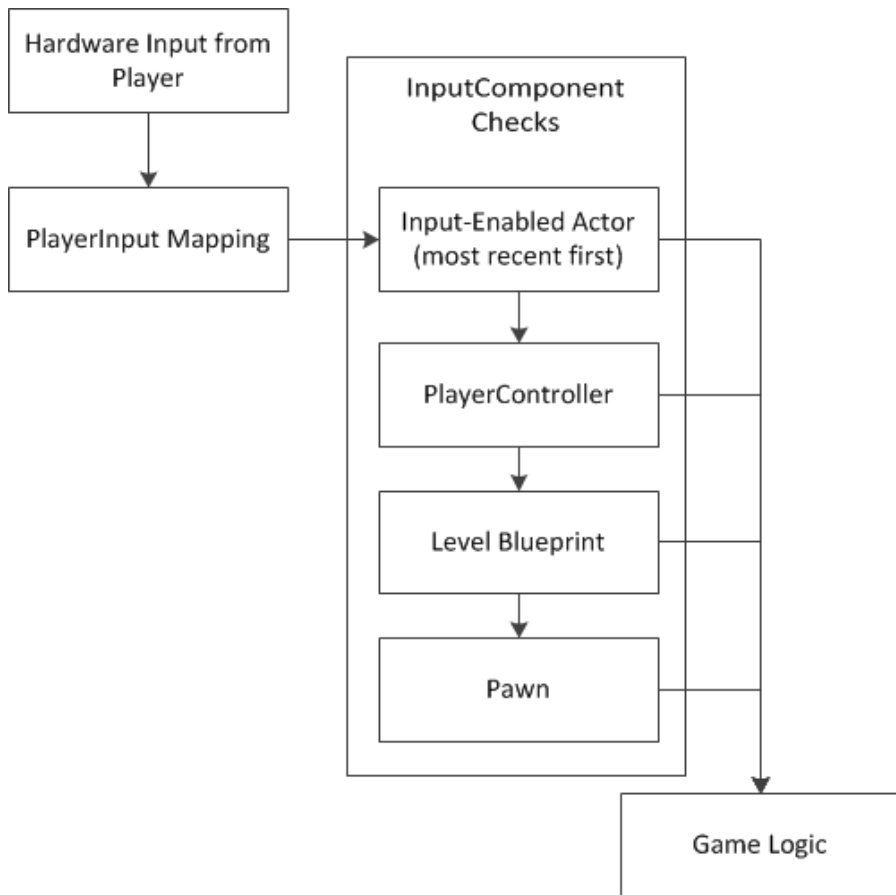


# IA\_FireCoin 입력에 따른 코인 폭탄 생성

SpawnActor From Class 를 활용하여, 원하는 블루프린트 클래스의 액터를 생성할 수 있음.



# Input Processing Procedure



Pawn은 "Possess"되지 않으면,  
입력을 아예 받지 못함.

# 목차

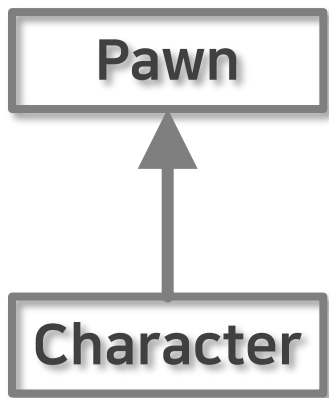
---

- 캐릭터 클래스
- Default Pawn 클래스
- Aircraft 조종 실습

# “Character” 클래스

---

- 걷기, 달리기, 점프, 비행, 수영 등이 가능한 직립 객체를 나타내는 클래스
- 기본 네트워킹 및 입력 처리 기능도 내장
- Pawn 의 자식 클래스

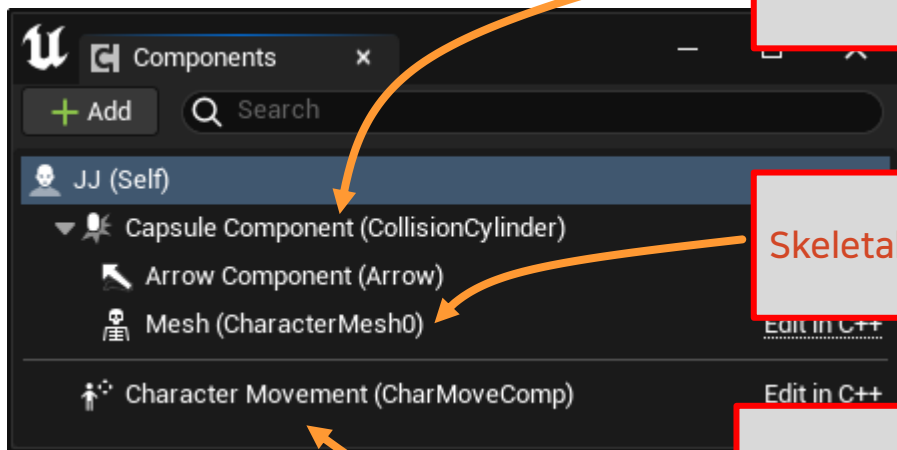


<http://api.unrealengine.com/KOR/Gameplay/Framework/Pawn/Character/index.html>

# “Character” 클래스

- Pawn 에 클래스에 세개의 Component가 추가됨.

- Capsule Component
- Skeletal Mesh Component
- Character Movement Component



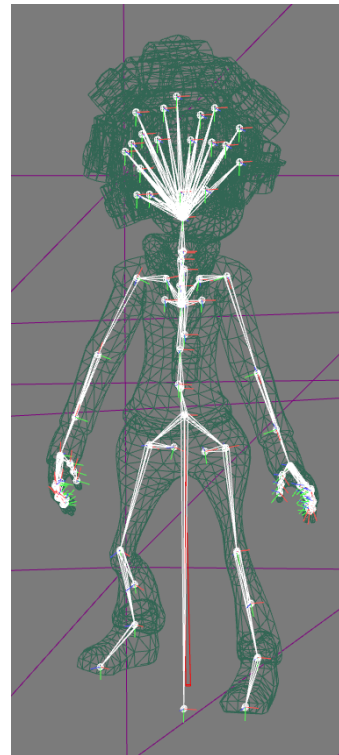
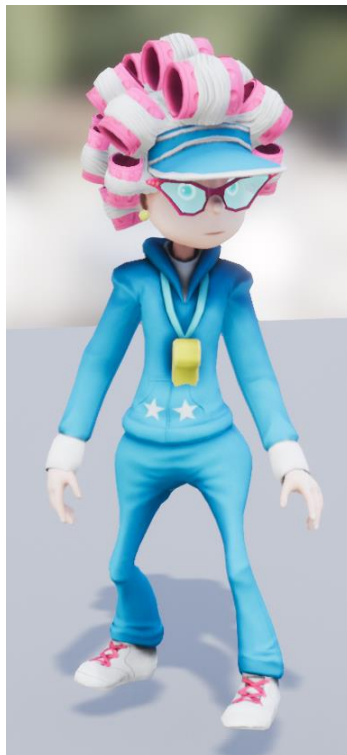
물리 담당:  
캐릭터의 물리적 존재(공간에서 차지하는  
영역)를 표현

Visual 담당:  
Skeletal Mesh와 Animation을 이용하여,  
시각적 표현을 담당

운동 담당:  
운동과 관련된 여러가지 속성을 정의하여,  
캐릭터의 움직임을 구현함.

# Skeletal Mesh

- 캐릭터 또는 애니메이션 객체를 표현.
- 3D 모델의 Mesh 를 구성하는 점들을 뼈대(Skeleton) 과 연결시킴.
- 뼈대가 움직이면, 여기에 붙어있는 점들도 함께 움직임.
- 자료 구조
  - Skeletal mesh
  - Skeleton



# 캐릭터 셋업 절차

---

- 캐릭터 3D 모델링 & 애니메이션 - FBX로 저장
- импорт(Import)
- 플레이어 컨트롤러 만들기
- Pawn 또는 캐릭터 블루프린트 만들기
- 애니메이션 블루프린트 만들기
- 게임 모드 셋업
- 월드 세팅

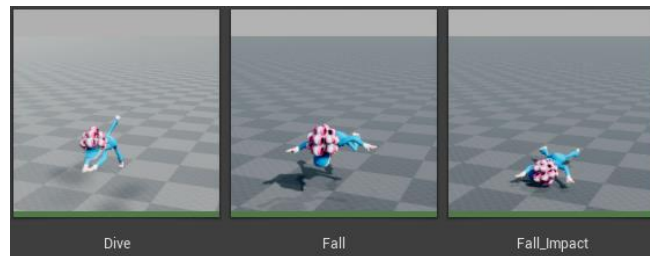
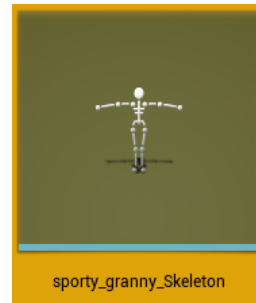
<http://api.unrealengine.com/KOR/Engine/Animation/CharacterSetupOverview/index.html>



# 3D 모델의 제작, Export 및 Import



Mesh 제작  
텍스처링 & UV 매핑  
리깅 & 스키닝 작업  
애니메이션 제작

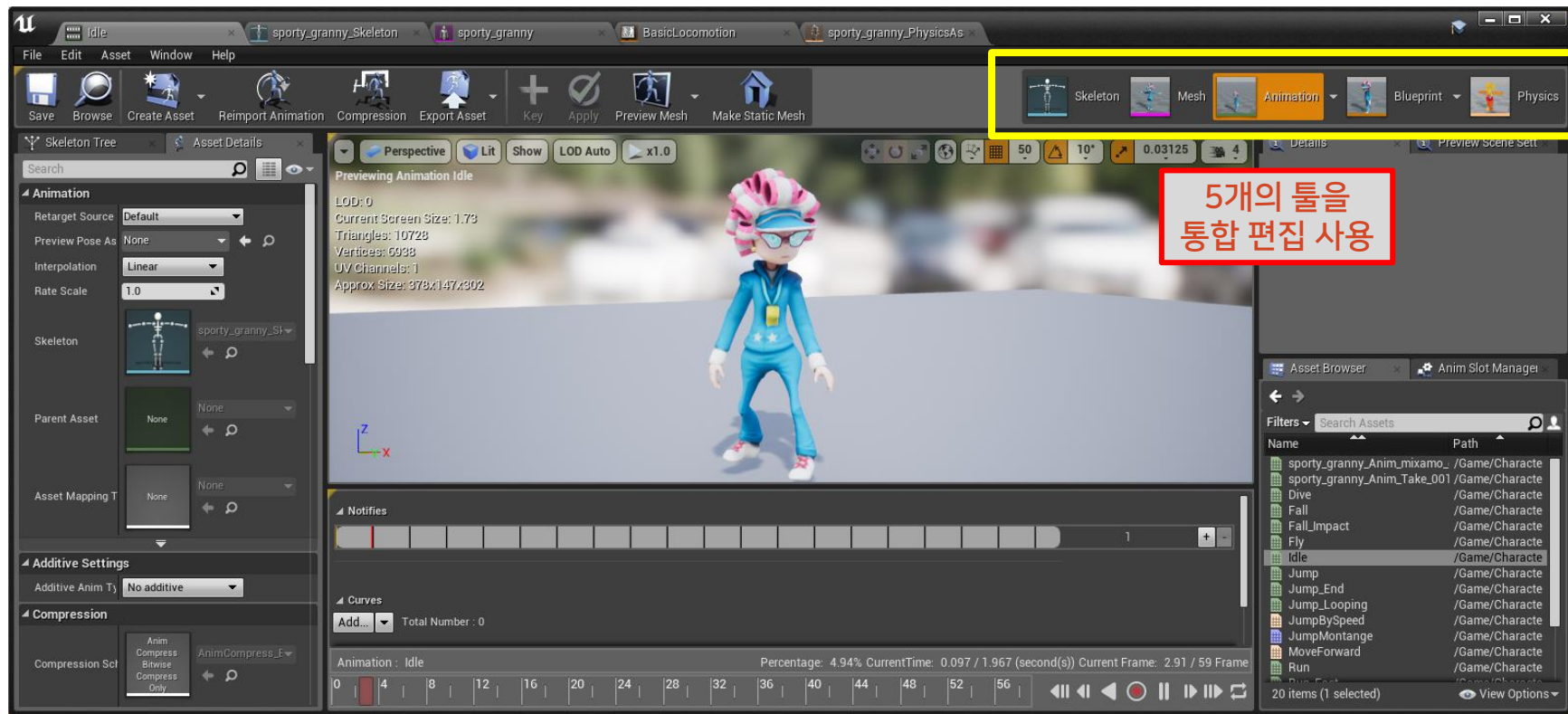


# 애니메이션 애셋

---

- **애니메이션 시퀀스(Animation Sequence)**
  - 특정 동작에 대한 키프레임 애니메이션 데이터
- **애니메이션 컴포짓(Animation Composite)**
  - 여러 애니메이션 애셋을 합쳐 하나의 단위로 취급
- **블렌드 스페이스(Blend Space)**
  - 특정 속성 또는 조건에 따른 애니메이션 블렌딩 처리
- **애니메이션 몽타주(Animation Montage)**
  - 애니메이션 에셋들을 다양한 방식으로 편집 및 재생

# 애니메이션 에디터 - 페르소나



<http://api.unrealengine.com/KOR/Engine/Animation/Persona/index.html>



실습

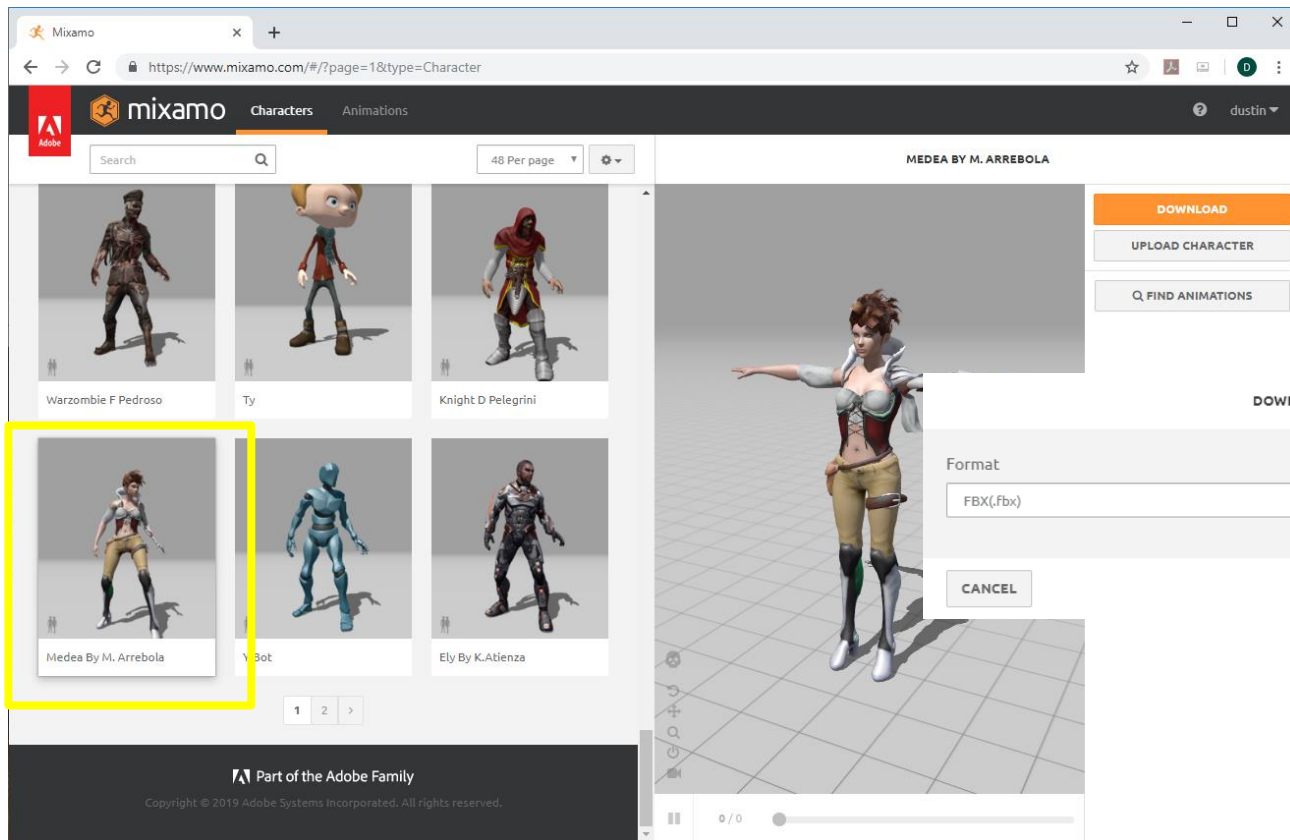
## 3D 모델 & 애니메이션 импорт

# 실습 목표

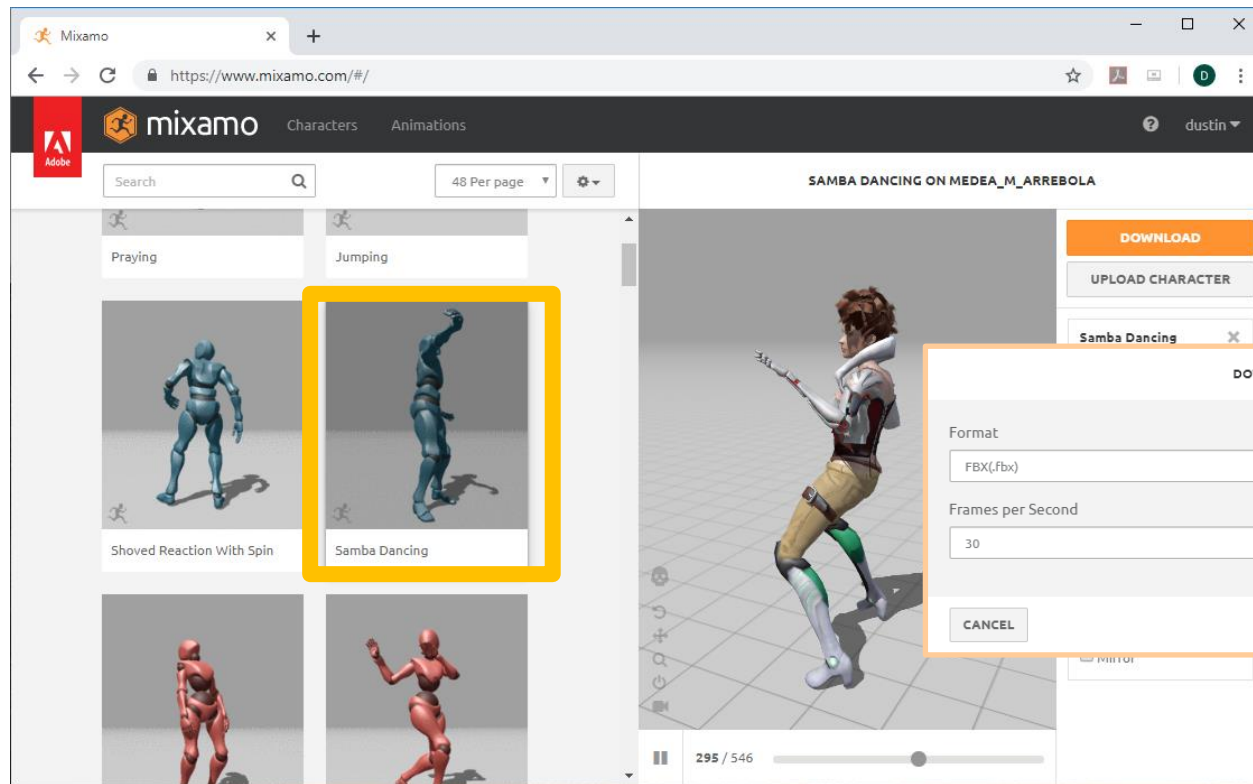
## ■ 믹사모를 활용한 3D 모델의 생성과 импорт



# Mixamo - 캐릭터 다운로드



# 애니메이션 다운로드 - idle, walk, slow run, fast run



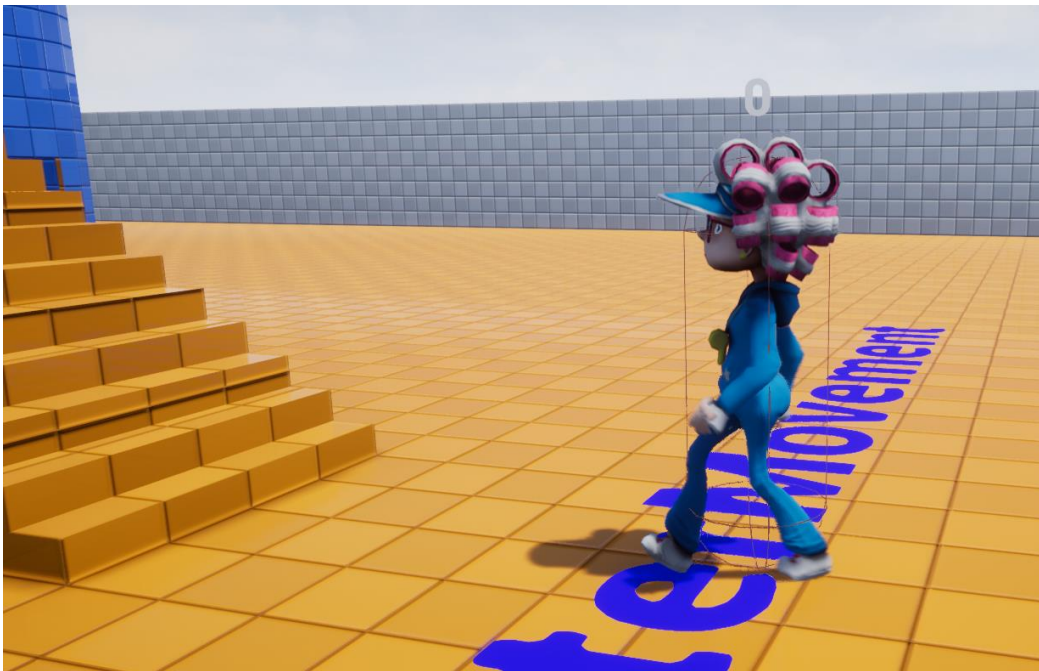


## 실습 캐릭터 이동



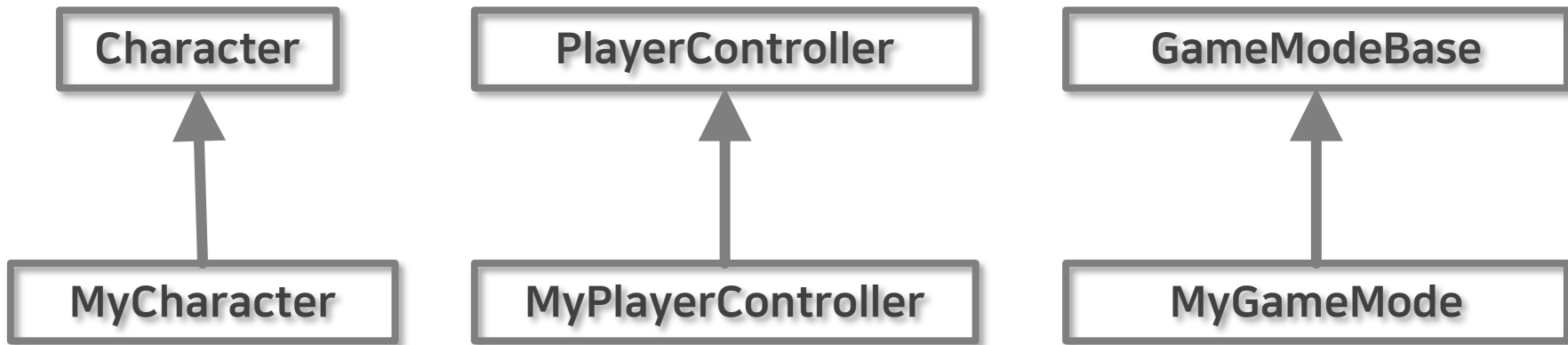
# 실습 목표

- 게임 모드 구성
- 캐릭터의 전후좌우 이동 구현



# 게임 모드 구성

---



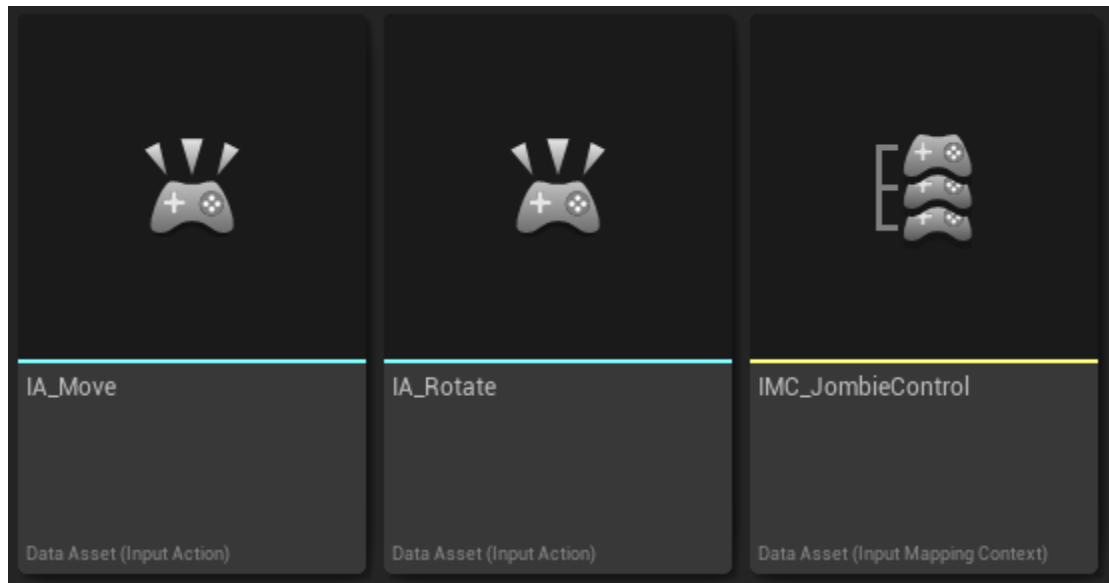
# PlayerInput 매핑(Mapping)

---

- 키보드, 마우스, 게임컨트롤러의 입력을 "읽기 쉬운 " 이름으로 표시
- Action Mapping
- Axis Mapping

<https://docs.unrealengine.com/en-US/Programming/Tutorials/PlayerInput/2>

# Project Input 설정



# 캐릭터 이동 및 회전 제어 방식

---

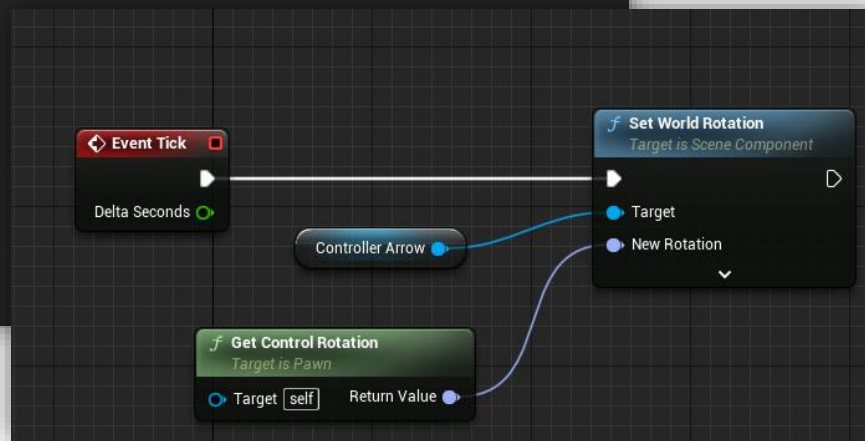
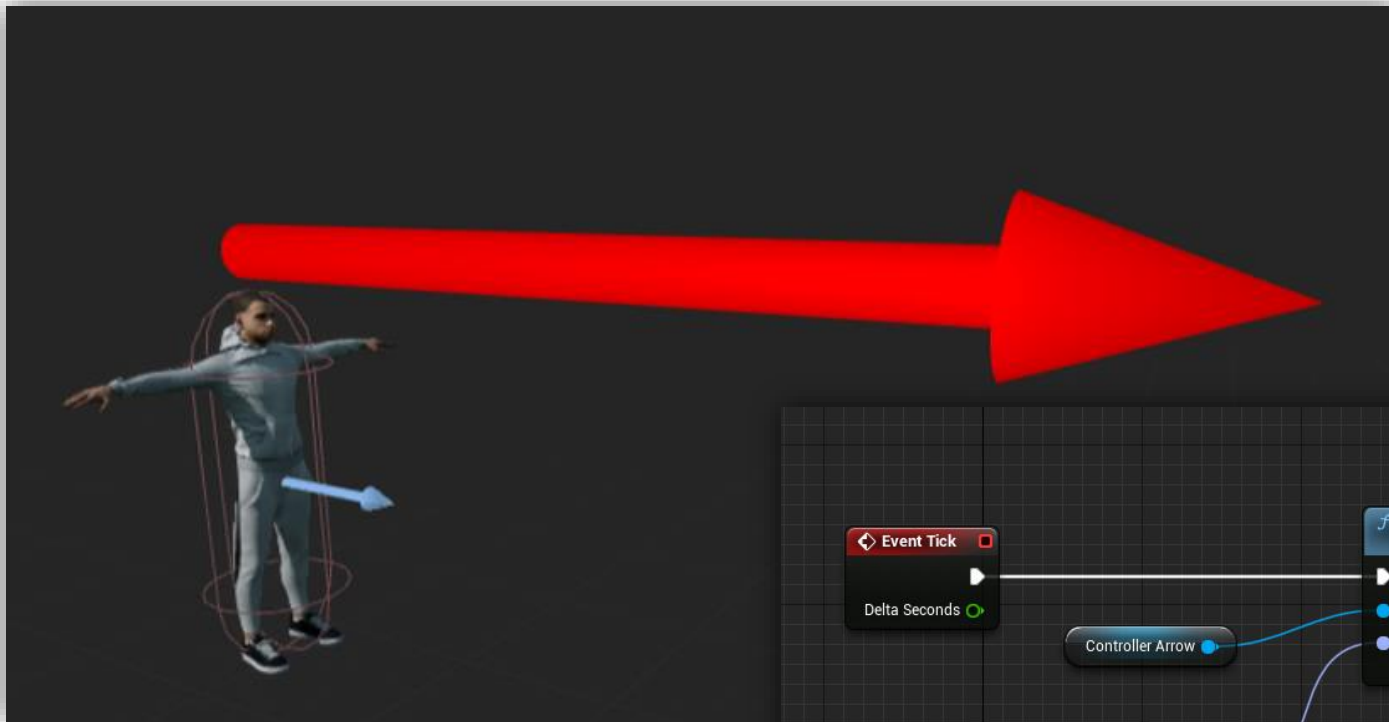
- **PlayerController를 사용한다.**
- **회전 – Rotation**
  - PlayerController 자체를 회전시킨다. - Add Yaw Input, Add Pitch Input, Add Roll Input 활용
  - Pawn의 회전을 Controller와 일치시킬지 여부를 결정한다.
- **위치 이동 – Translation**
  - PlayerController에서 Pawn을 움직인다. - Add Movement Input 활용

# 캐릭터와 카메라 조작 방식

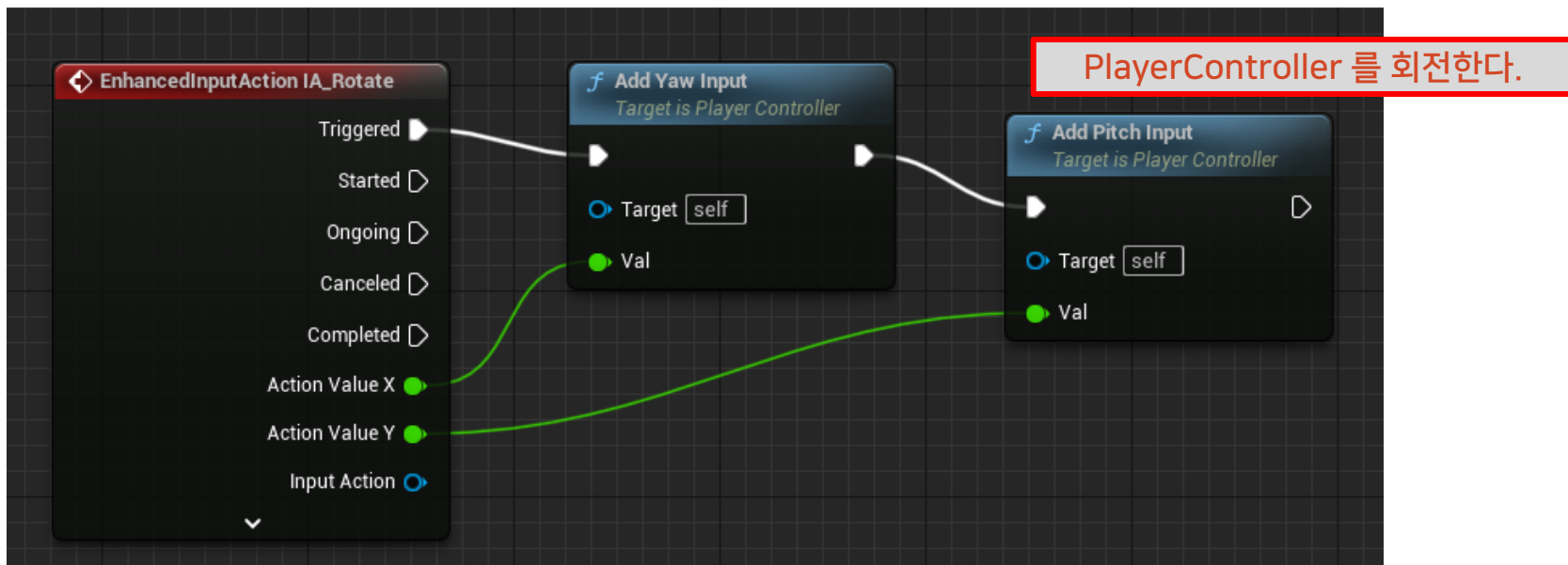
---

- WASD 와 마우스 이용
- 회전할 때 캐릭터 회전 ? 카메라 회전 ?
- 전후좌우 이동 시 캐릭터의 방향?

# PlayerController



# 캐릭터 회전(조작 스타일 #1)

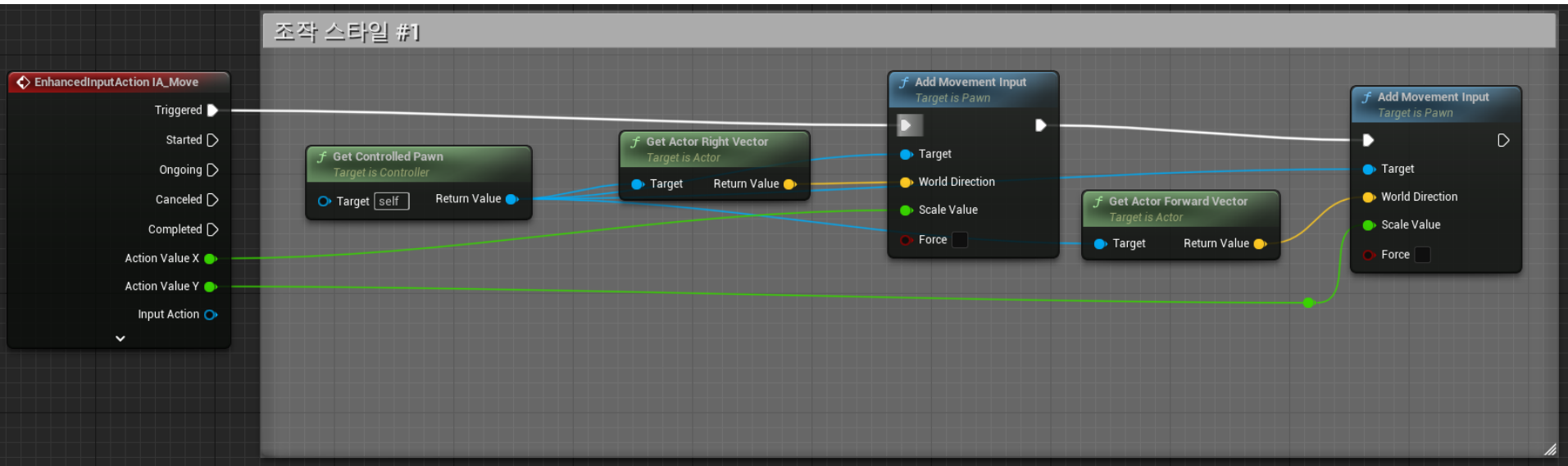


Character 의 Class Defaults 속성

▼ Pawn	
Use Controller Rotation Pitch	<input type="checkbox"/>
Use Controller Rotation Yaw	<input checked="" type="checkbox"/>
Use Controller Rotation Roll	<input type="checkbox"/>



# 캐릭터 이동(조작 스타일 #1)



# 캐릭터 회전 (조작 스타일 #2)

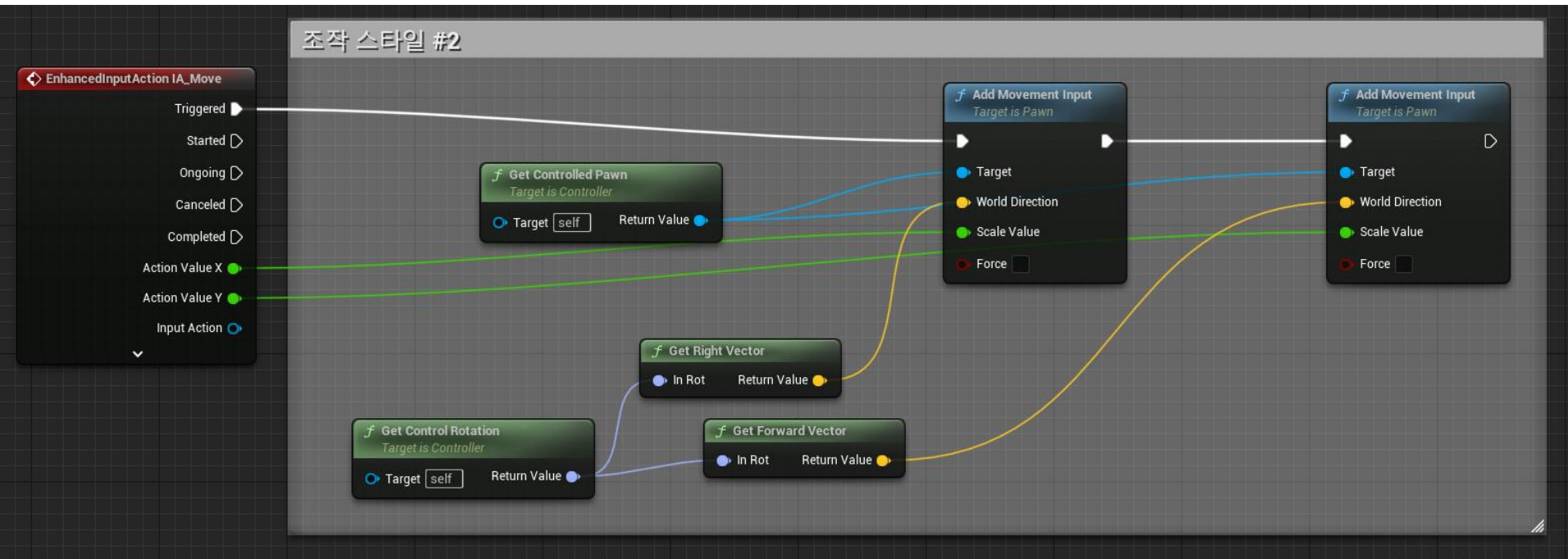
▼ Pawn

Use Controller Rotation Pitch	<input type="checkbox"/>
Use Controller Rotation Yaw	<input type="checkbox"/>
Use Controller Rotation Roll	<input type="checkbox"/>

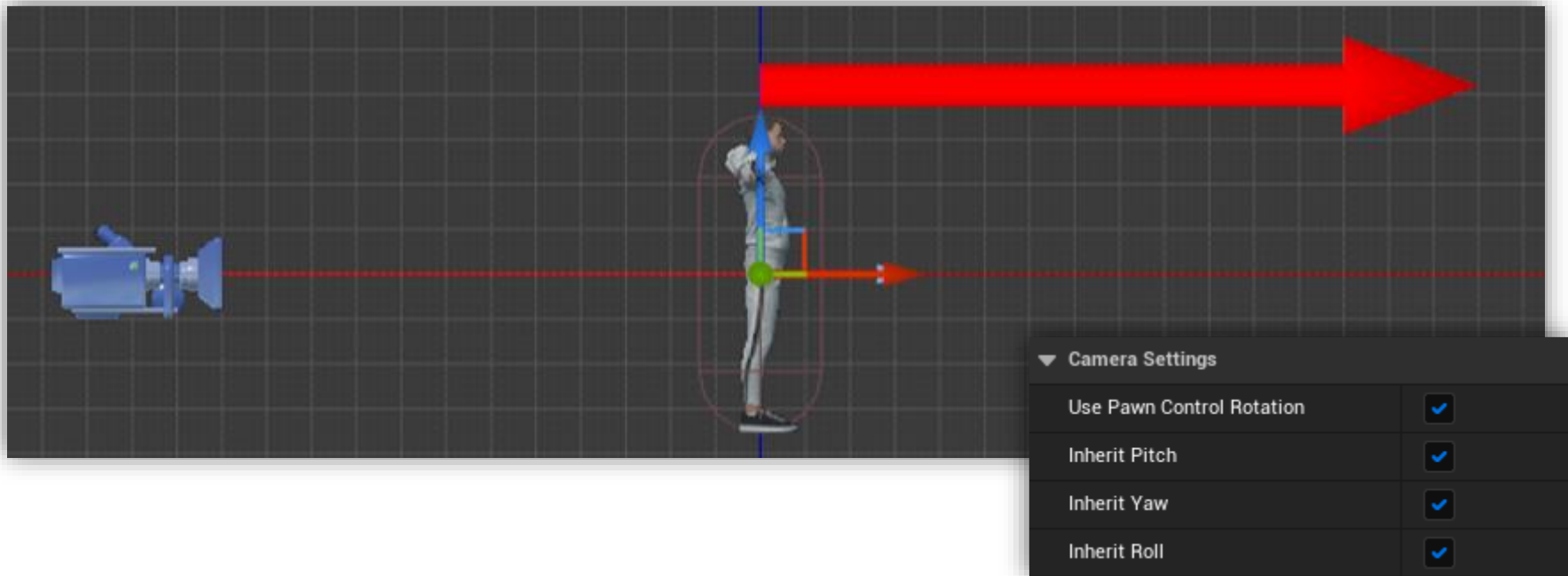
▼ Character Movement (Rotation Settings)

▶ Rotation Rate	0.0	0.0	360.0	
Use Controller Desired Rotation	<input type="checkbox"/>			
Orient Rotation to Movement	<input checked="" type="checkbox"/>			↶

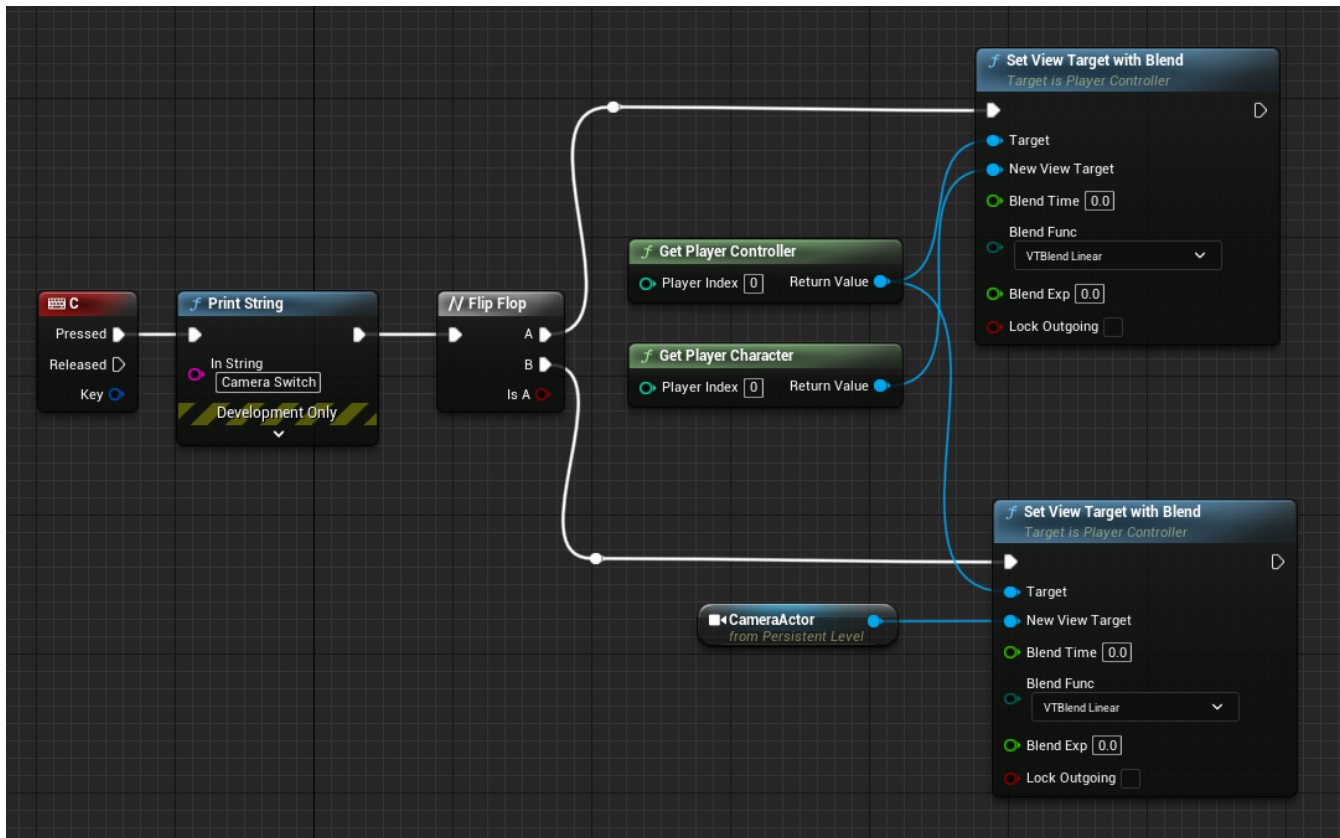
# 캐릭터 이동 ( 조작 스타일 #2 )



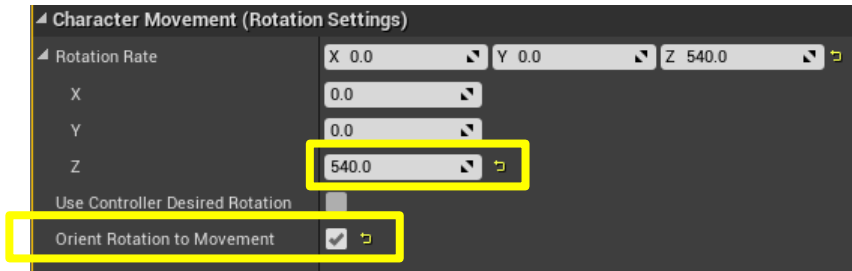
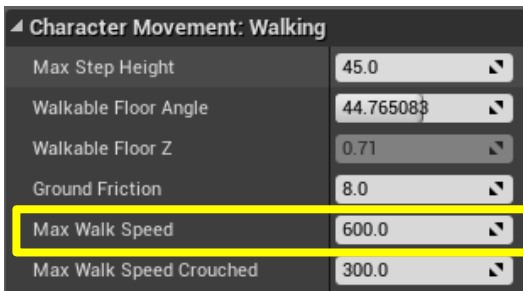
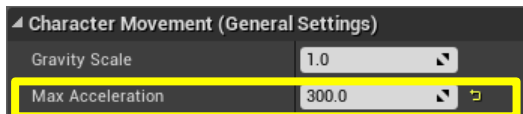
# 카메라 설치



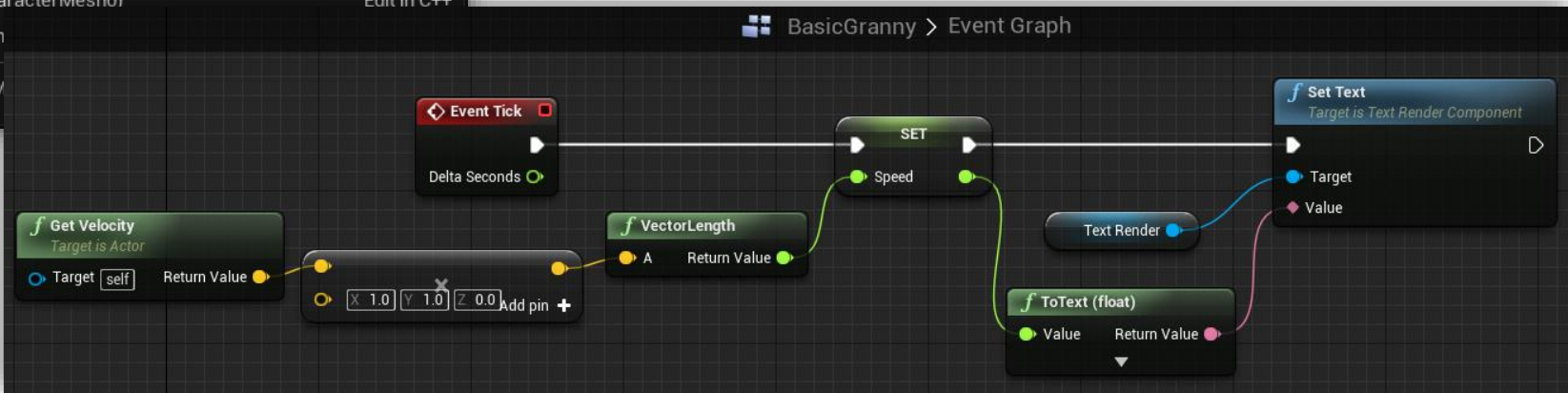
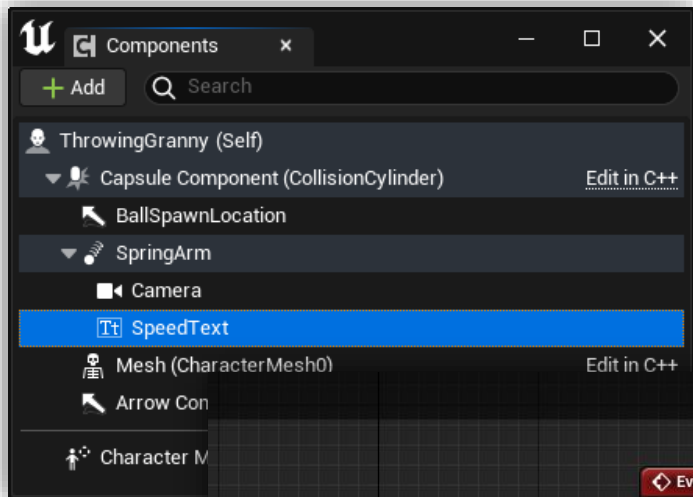
# 카메라 스위치 (디버깅) - Level Blueprint



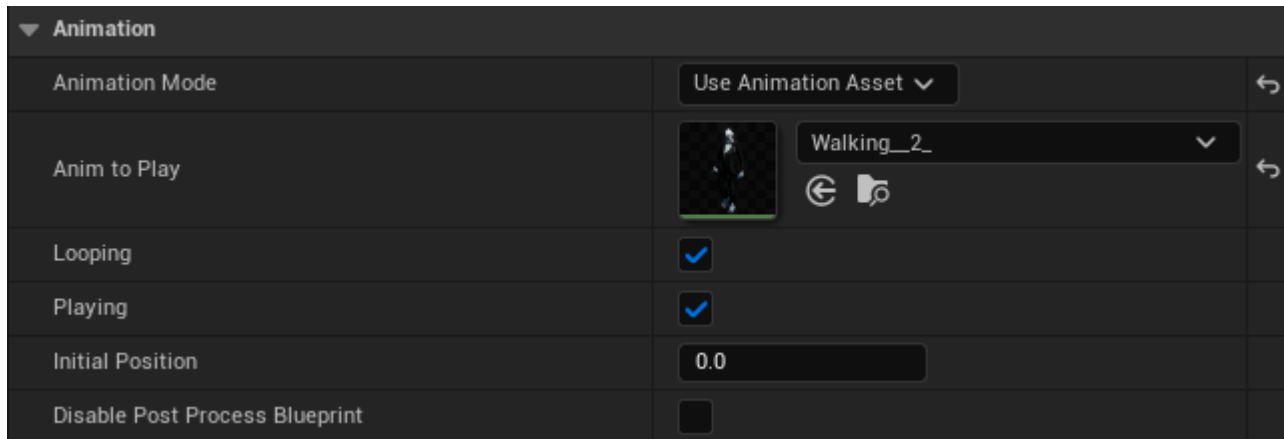
# CharacterMovement 컴포넌트 설정



# 그라운드 스피드 측정 및 실시간 표시



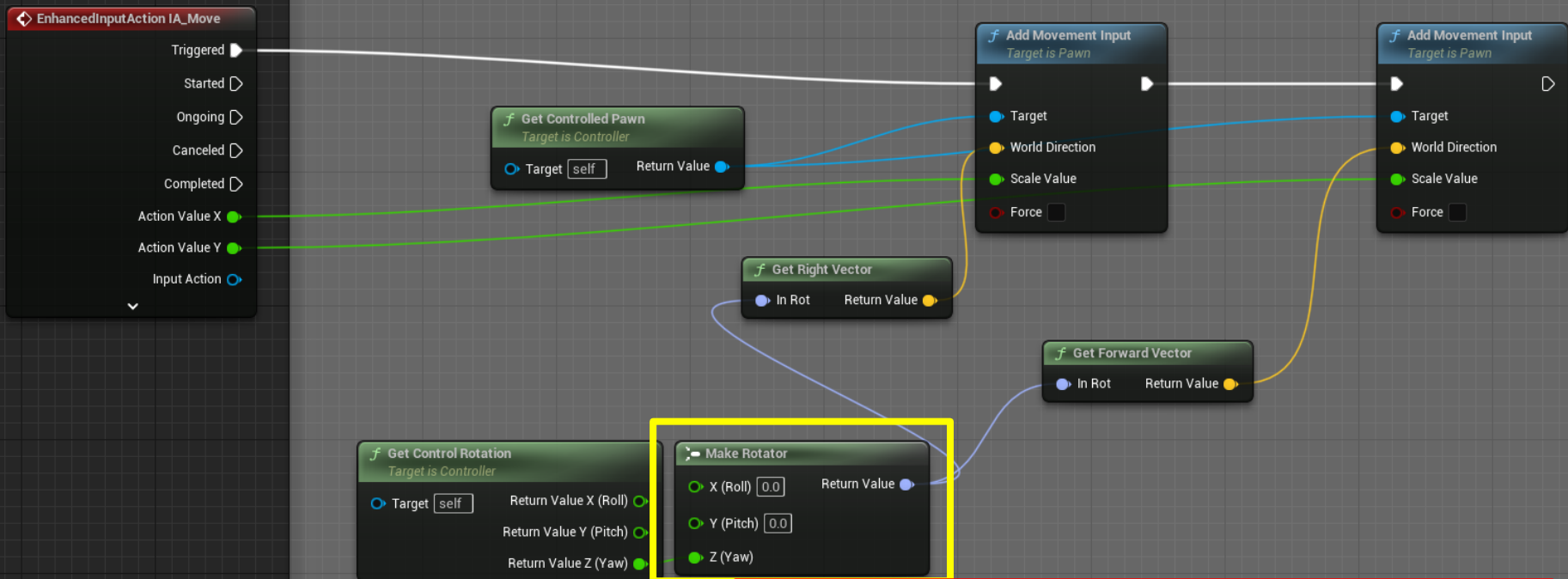
# 액터 초기 애니메이션 설정





# 플레이어 컨트롤러

## 조작 스타일 #2-FINAL



XY 평면 방향으로만 전진, 후진하기 위해서, Z 축 성분만 추출.

## 문제점? 물리와 Visual의 불일치 → 불편하고 어색함.



항상 걷고있다 ππππ

속도가 빨라지면 발이 미끄러진다 ππππ