A cinematic scene 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 out over a vast, ruined city with tall, jagged spires and crumbling structures. The sky is a mix of dark, heavy clouds and a bright, orange-hued sunset or sunrise, creating a dramatic and atmospheric scene. The overall tone is somber and contemplative.

게임 엔진

LEC 10 애니메이션



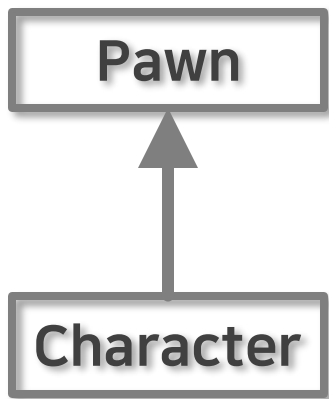
한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

LEC 09 복습

“Character” 클래스

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

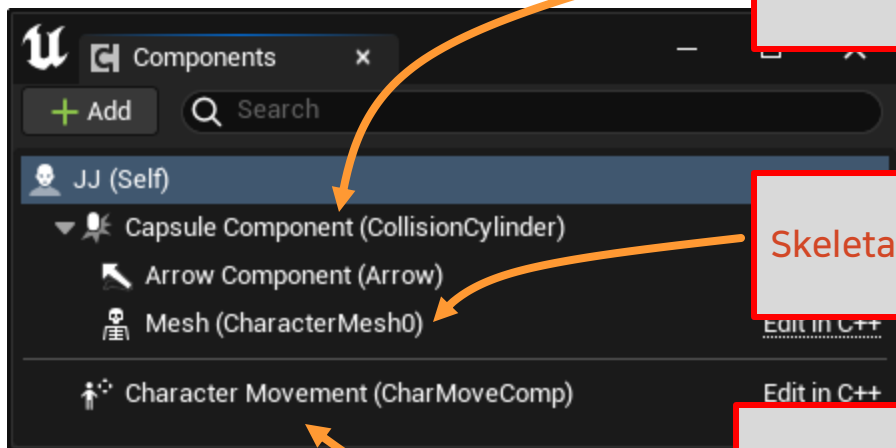


<http://api.unrealengine.com/KOR/Gameplay/Framework/Pawn/Character/index.html>

“Character” 클래스

- Pawn 에 클래스에 세개의 Component가 추가됨.

- Capsule Component
- Skeletal Mesh Component
- Character Movement Component



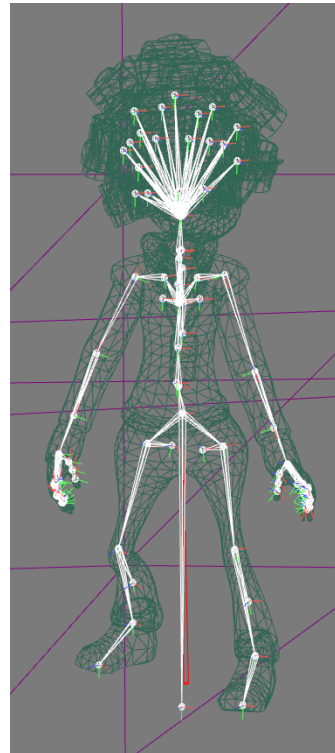
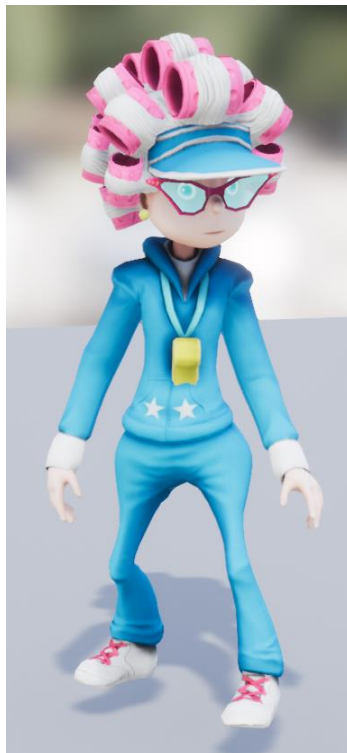
물리 담당:
캐릭터의 물리적 존재(공간에서 차지하는
영역)를 표현

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

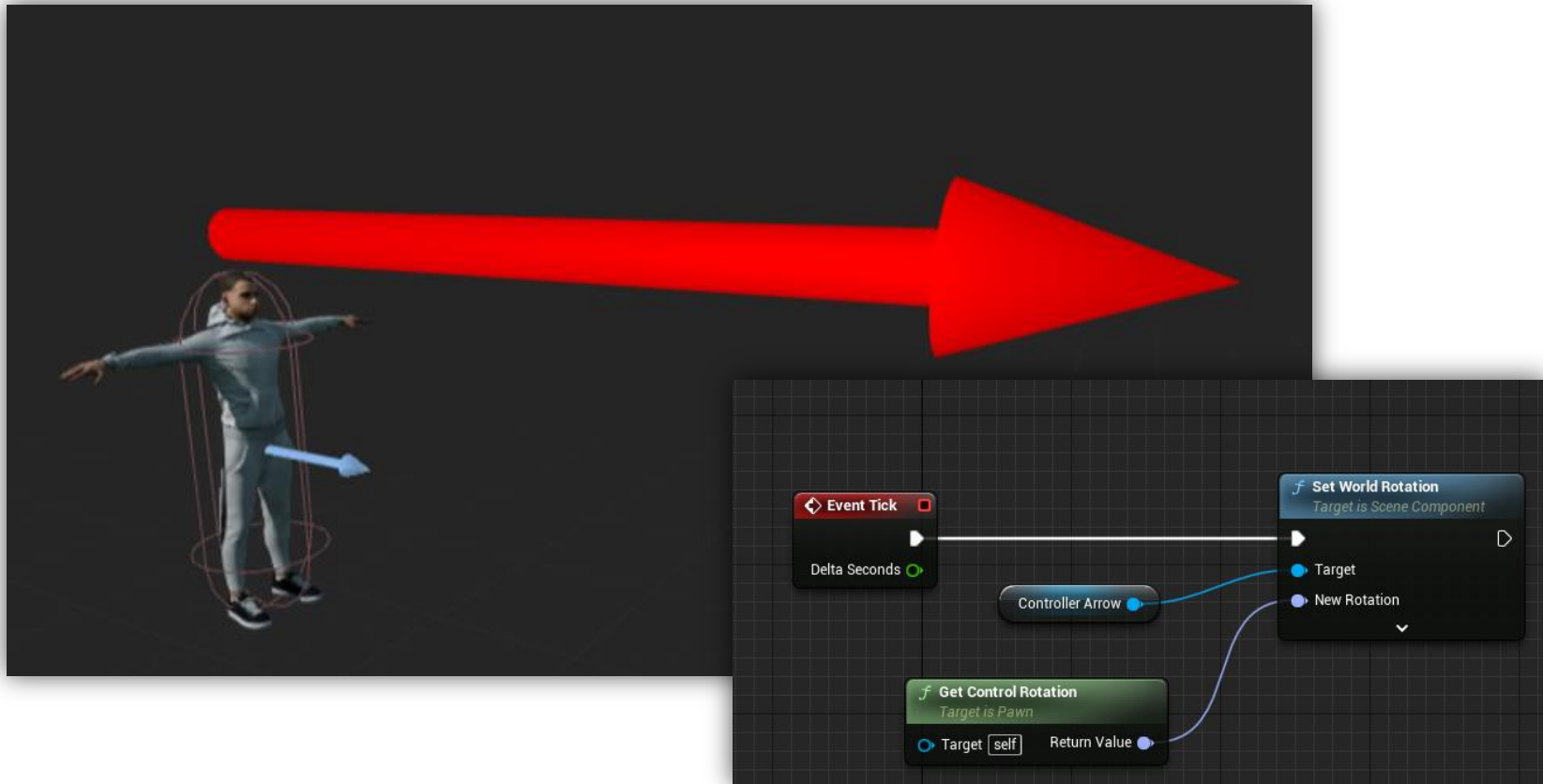


캐릭터 셋업 절차

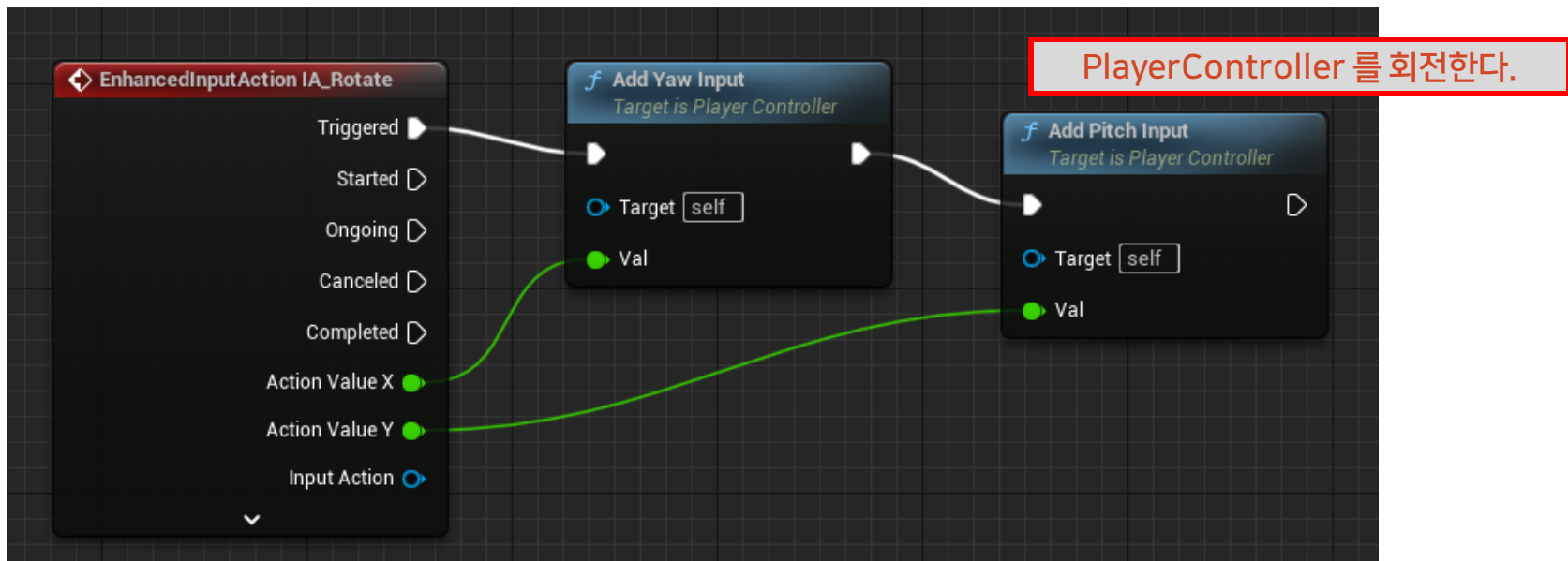
- 캐릭터 3D 모델링 & 애니메이션 - FBX로 저장
- импорт(Import)
- 플레이어 컨트롤러 만들기
- Pawn 또는 캐릭터 블루프린트 만들기
- 애니메이션 블루프린트 만들기
- 게임 모드 셋업
- 월드 세팅

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

PlayerController



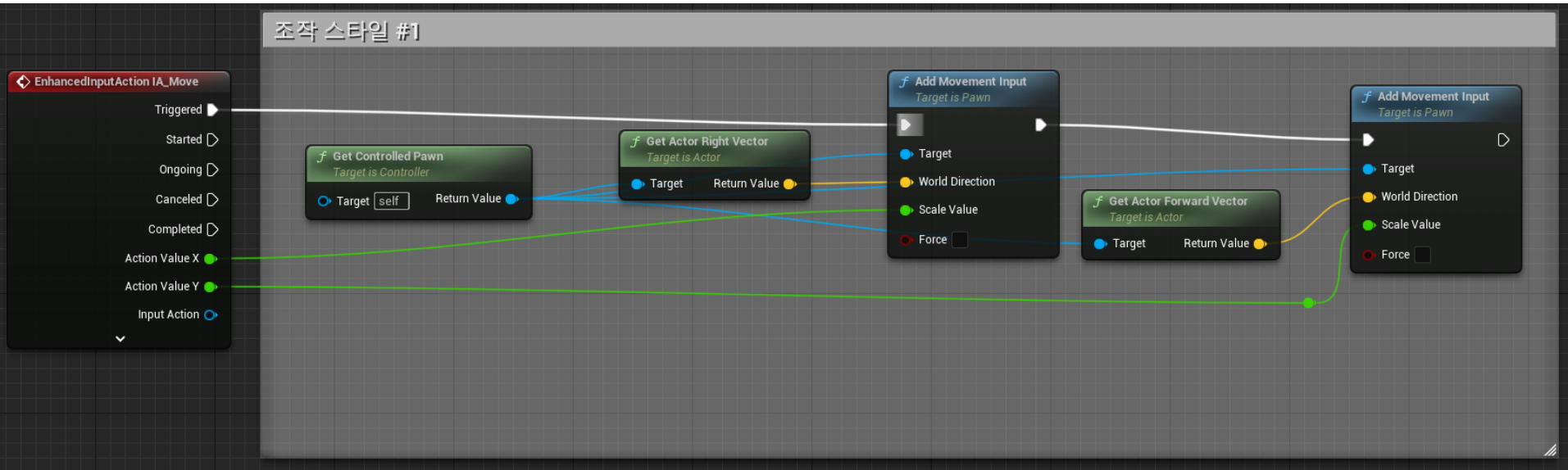
캐릭터 회전(조작 스타일 #1)



Character의 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)

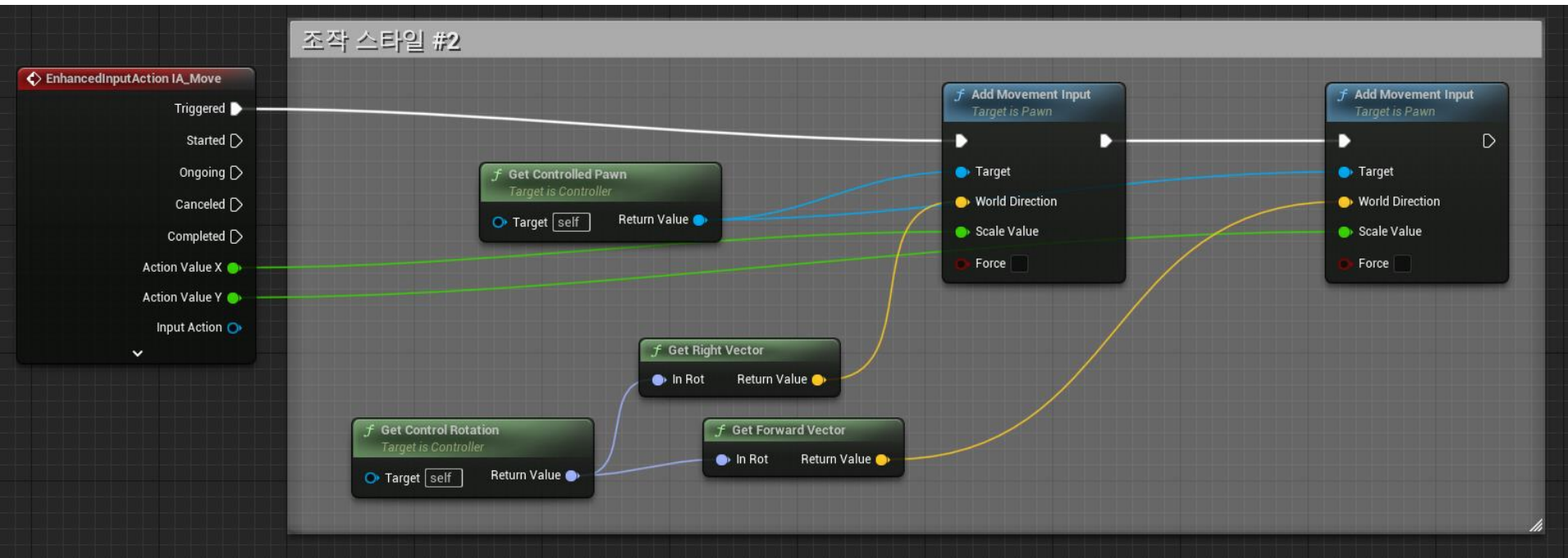
▼ Pawn

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)

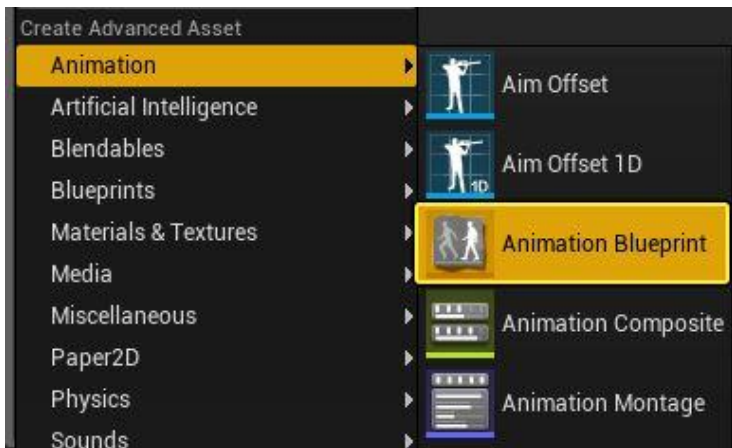


목차

- 애니메이션 블루프린트
- 애니메이션 블렌딩
- 블렌드 스페이스
- 애니메이션 스테이트 머신

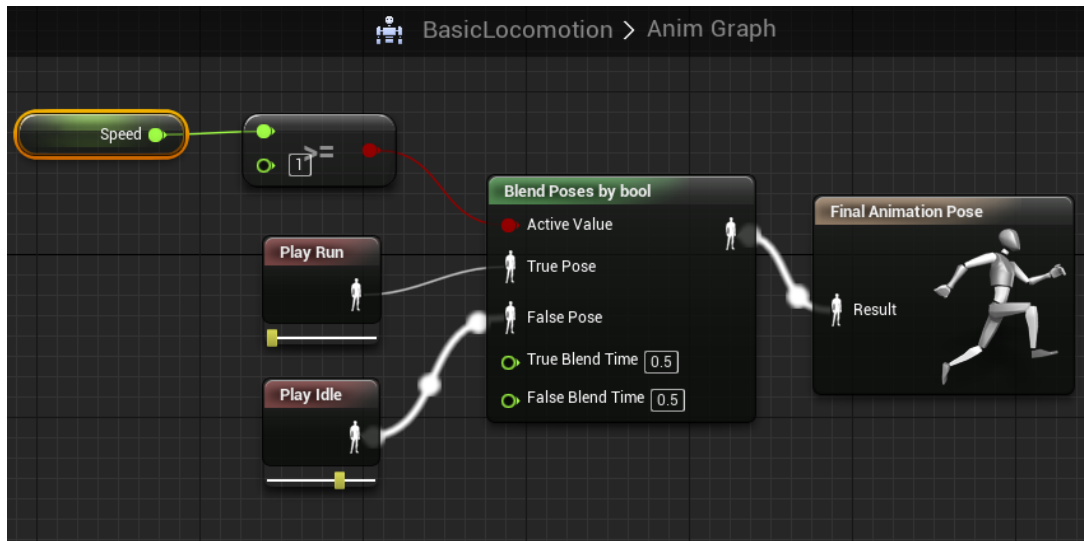
애니메이션 블루프린트

- 스켈레탈 메시의 애니메이션을 제어하는 블루프린트
- 애니메이션 블루프린트 에디터에서 편집
- 애니메이션 블렌딩, 스켈레톤의 본 직접 제어, 로직을 구성하여 각 프레임에 사용할 스켈레탈 메시의 최종 애니메이션을 최종 결정



애님 그래프

- 여러 애니메이션들을 섞을 수 있는, (블렌딩할 수 있는) 그래프

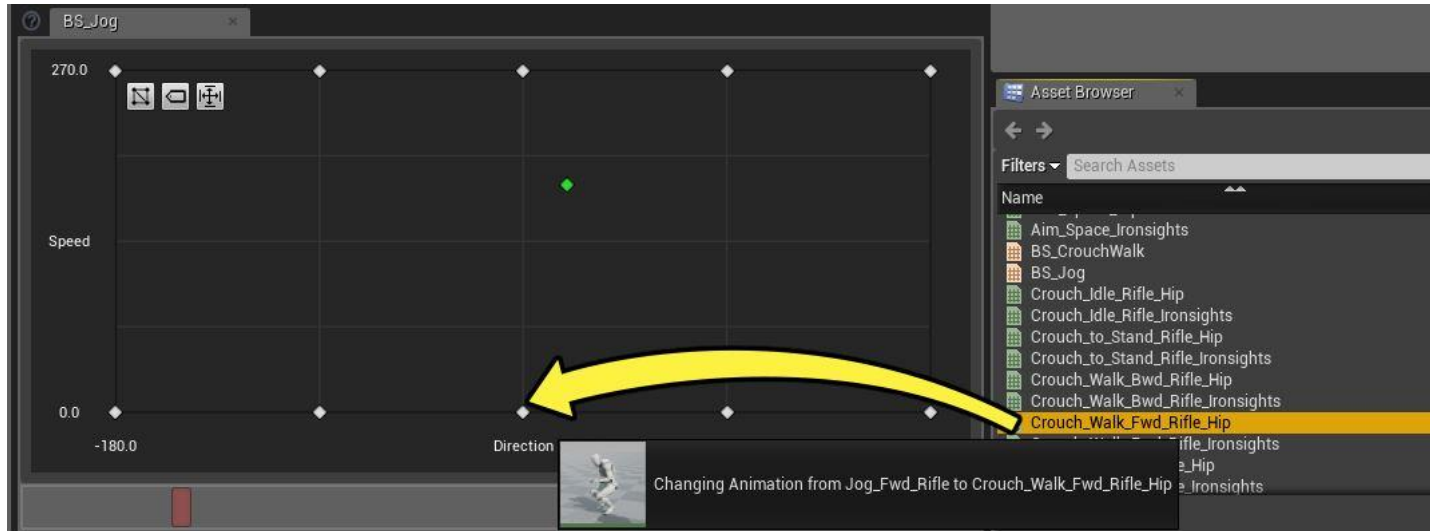


애니메이션 블렌딩(Animation Blending)

- 애니메이션들간의 부드러운 전환을 하는 것.
- 두가지 방법
 - 블렌드 스페이스
 - 애니메이션 블루프린트

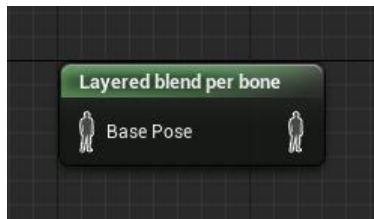
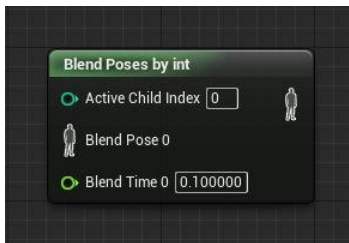
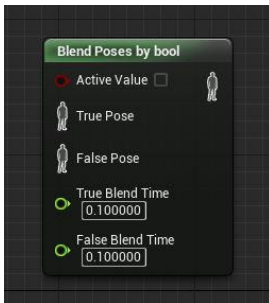
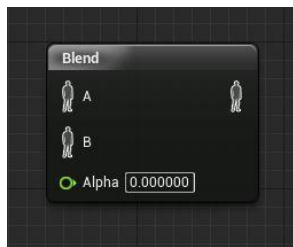
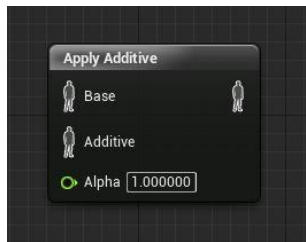
블렌드 스페이스

- 다수의 입력값에 따라 애니메이션을 블렌딩시켜주는 기능.
- 특정 속성이나 조건에 따라 블렌딩할 때 마다 복잡한 하드 코드하는 수고를 덜기 위함.
- 현재 지원되는 스페이스 : Blend Space (2축), Blend Space 1D (1축)



Blend 노드

- Blend 노드는 여러 애니메이션을 블렌딩하는 데 쓰임.
- Animation Blueprint (애니메이션 블루프린트)의 AnimGraph (애님 그래프)에서만 놓아 사용.
- 블렌드 노드는 여러 포즈와 알파 또는 웨이트(weight) 값을 받아 최종 결과물에서 각 포즈에 대한 비중을 계산.





실습

정지 및 보행 애니메이션 (애니메이션 블루프린트)

실습 목표

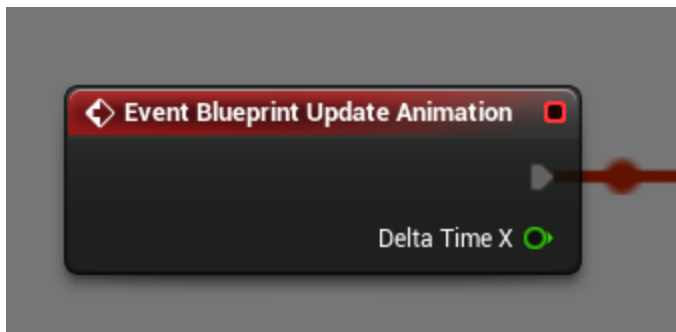
- 정지 및 보행 간의 부드러운 애니메이션 구현
- 애니메이션 블루프린트와 블렌딩 노드 사용



스피드에 따라서 IDLE \leftrightarrow WALK 의 전환

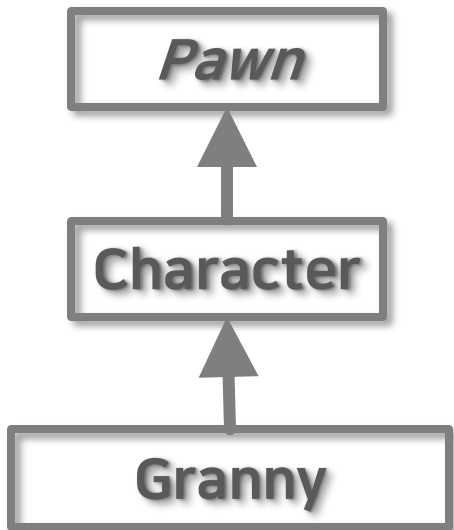
Blueprint Update Animation 이벤트 노드

- 애니메이션 블루프린트 안에 존재하는 이벤트
- 액터의 Event Tick과 마찬가지로, 반복적으로 호출됨.
- 애니메이션 재생에 요구되는 다양한 작업을 처리.



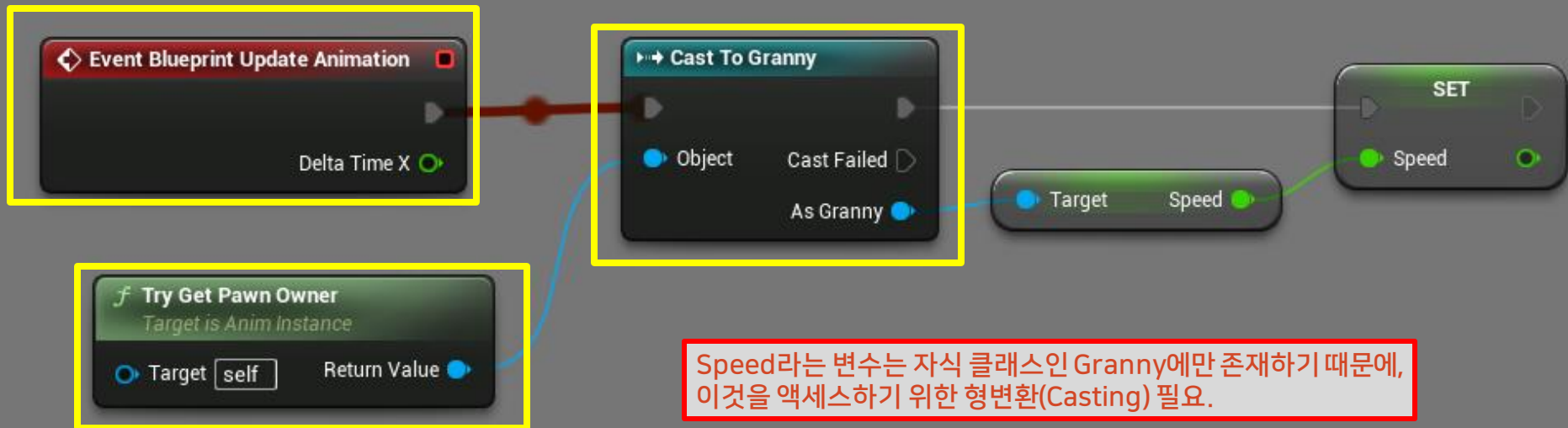
형변환 캐스팅 역할

- 자식 클래스 내의 변수 또는 함수에 액세스할 수 있게끔 함.



애니메이션 블루프린트의 이벤트 그래프

액터의 Event Tick과 같이, Event Blueprint Update Animation 이벤트는 지속적으로 호출됨.

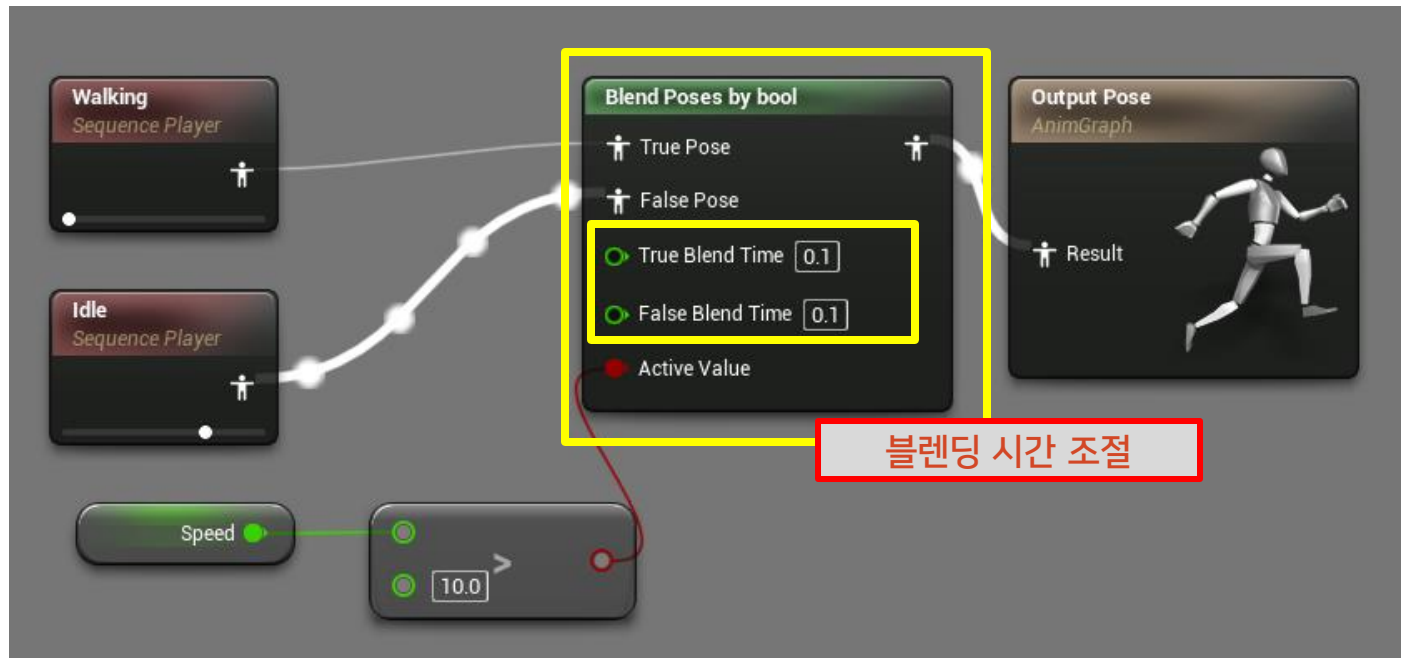


Speed라는 변수는 자식 클래스인 Granny에만 존재하기 때문에, 이것을 액세스하기 위한 형변환(Casting) 필요.

애니메이션 블루프린트를 사용하고 있는 Pawn의 획득.

Blend Poses by bool 노드

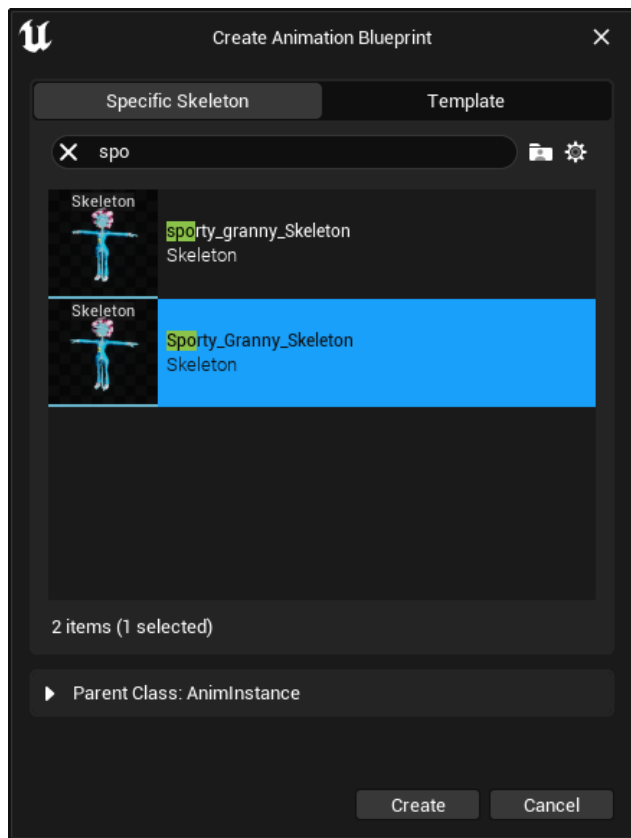
- 두개의 입력 포즈 중, 하나를 선택해서 출력.
- 포즈 전환 시, 블렌딩이 이루어짐.



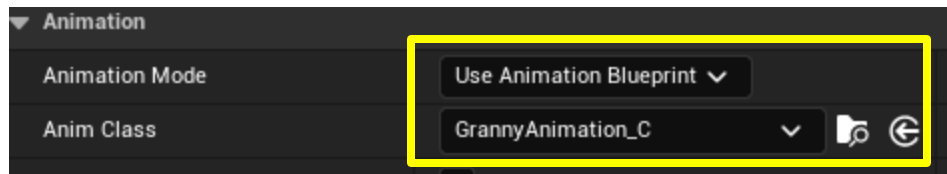


스피드에 따라서 IDLE \leftrightarrow Walk 의 전환

애니메이션 블루프린트 스켈레톤 설정



애니메이션 블루프린트 모드 설정





실습

부드러운 이동 애니메이션 (블렌드 스페이스)

앞선 구현의 문제점



빠르게 달릴 경우, 달리는 속도와 진행거리의 차이가 눈에 띈다.
천천히 뛰고 있는데 상당히 많은 거리를 이동함.

실습 목표

■ 블렌드 스페이스를 활용한 블렌딩 구현

IDLE



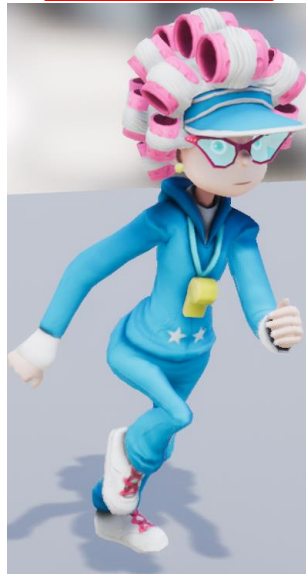
WALK



RUN

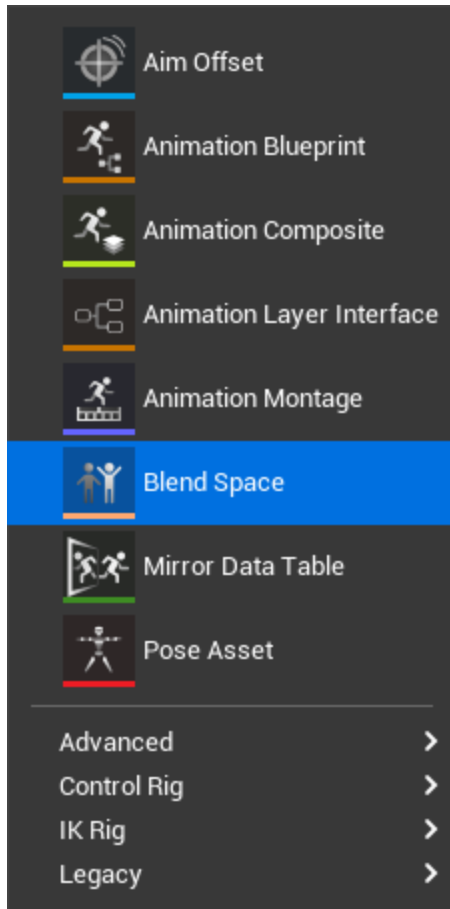


RUN_FAST



스피드에 따라서 애니메이션 포즈간 지속적인 블렌딩

Speed 를 축으로 하는 Blend Space 생성



축 지정

▼ Horizontal Axis	
Name	Speed
Minimum Axis Value	0.0
Maximum Axis Value	428.0

축 상에 애니메이션 포즈 지정

- 애니메이션 브라우저에 끌어다 놓으면 됨.

The image shows a workflow in Unreal Engine for assigning animation poses to movement states. On the left, a 'GrannyGroundMoveBySp' component is shown with a timeline. The timeline has four states: 'Idle (0)', 'Walking (1)', 'Running (2)', and 'Fast_Run (3)'. Below the timeline, the states are labeled 'WALKING', 'RUNNING', and 'FAST_RUN' (the last one is highlighted with a red box). An orange arrow points from the 'FAST_RUN' label to the 'Fast_Run (3)' state in the timeline. On the right, the 'Asset Browser' window is open, displaying a list of assets. The 'Fast_Run (0)' asset is selected, and its properties are shown in the right panel. The 'Speed' property is set to 426.84848. Below this, the 'Idle (1)' and 'Running (2)' assets are also shown, with their 'Speed' properties set to 357.0 and 172.054932 respectively. The 'Walking (3)' asset is also visible.

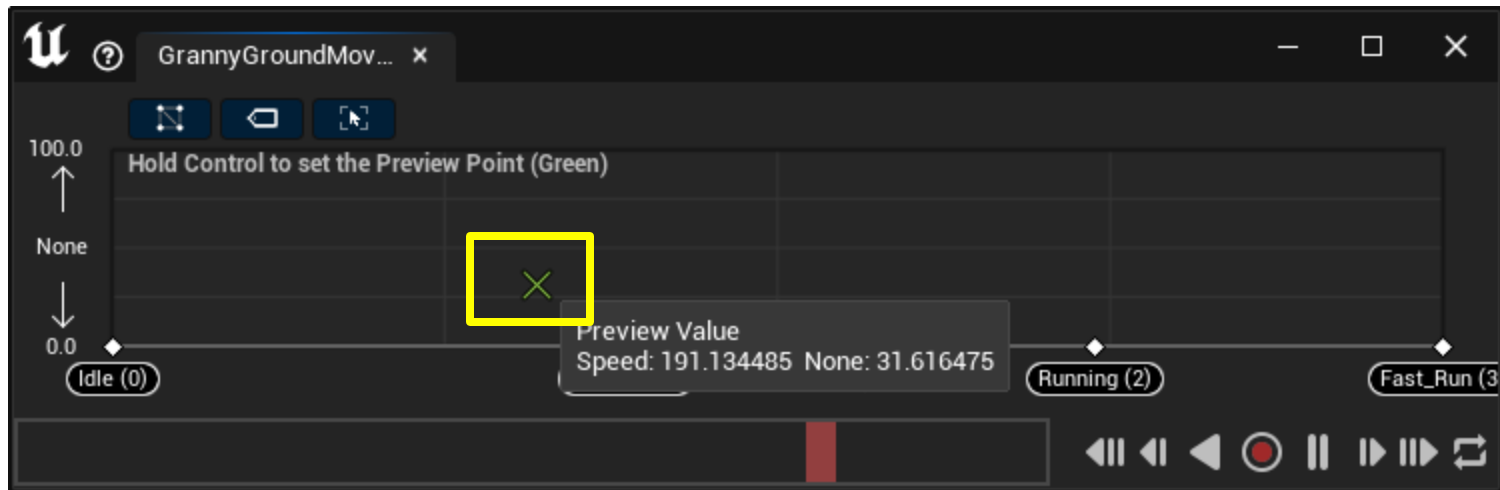
Name	Path	Disk Size	Has Virtualized D
Falling_Idle	/All/Game/AA	111.46 KiB	False
Fast_Run	/All/Game/AA	91.821 KiB	False
Idle	/All/Game/AA	258.7 KiB	False
Jump	/All/Game/AA	145.701 KiB	False
Jumping_Down	/All/Ga		
Running	/All/Ga		
Walking	/All/Ga		
GrannyGroundMoveBySp	/All/Ga		

Property	Value
Speed	426.84848
None	0.0
Animation	Fast_Run
Rate Scale	1.0
Include in Analyse All	<input checked="" type="checkbox"/>

Property	Value
Speed	357.0
None	0.0
Animation	Running
Rate Scale	1.0
Include in Analyse All	<input checked="" type="checkbox"/>

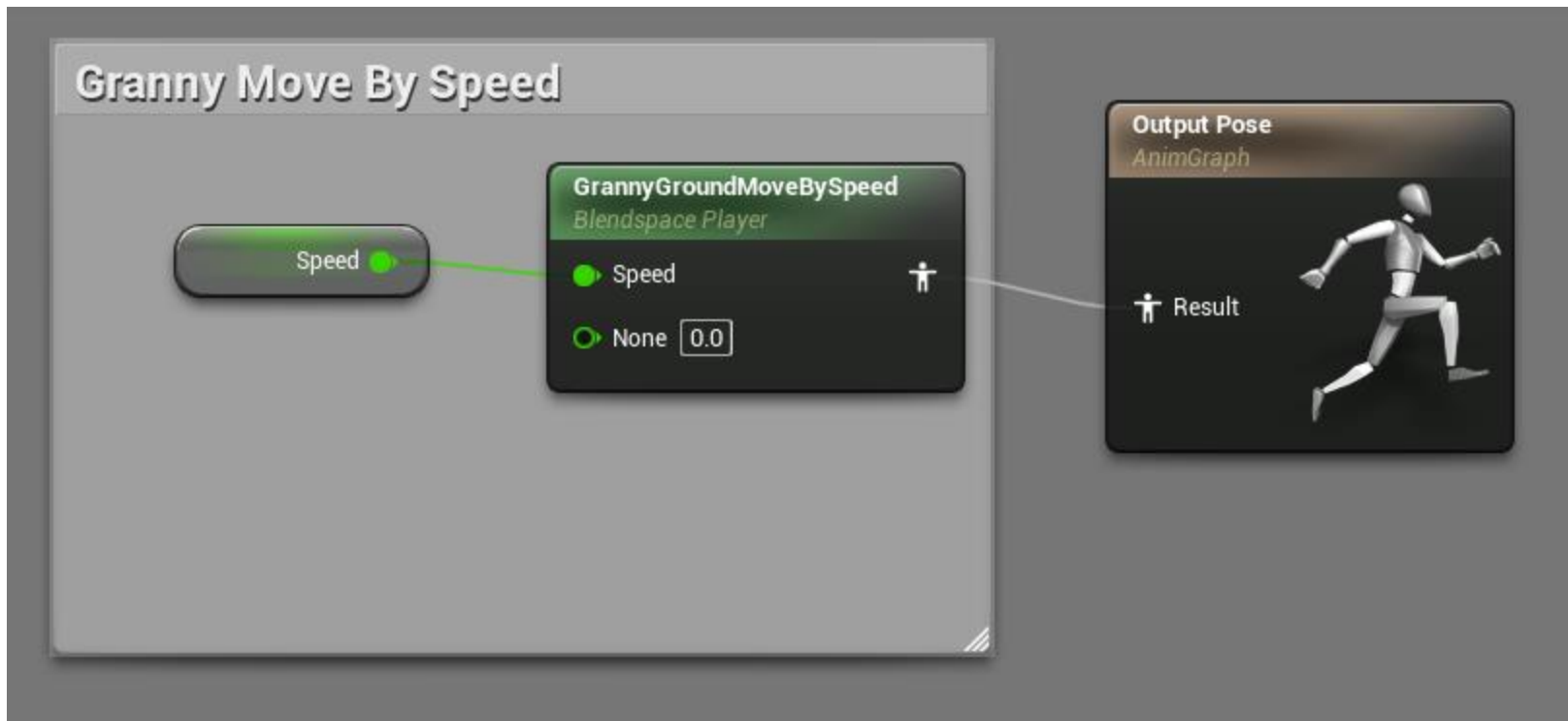
Property	Value
Speed	172.054932

블렌드 스페이스 미리 보기




Ctrl 키를 누르고 초록 다이아몬드를 끌어 여러 포즈 사이 블렌딩을 확인 가능.

블렌드 스페이스 사용

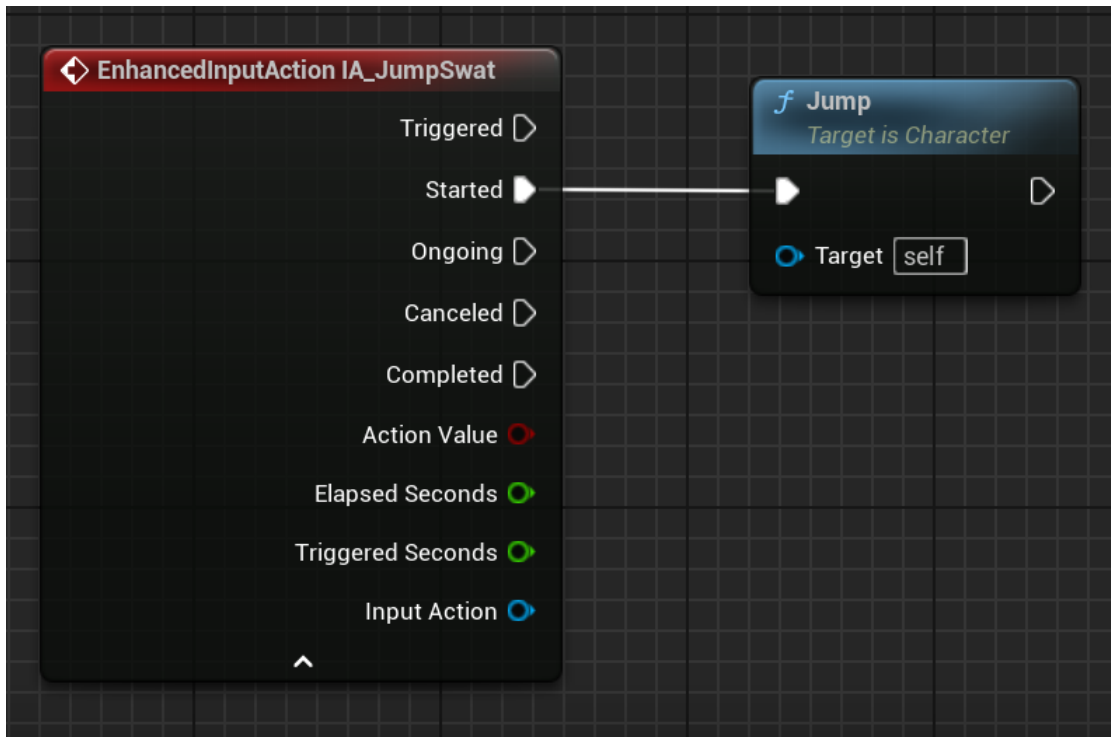


Input Action , IMC

▼ Action	
Consume Input	<input checked="" type="checkbox"/>
Trigger when Paused	<input type="checkbox"/>
Reserve All Mappings	<input type="checkbox"/>
Value Type	Digital (bool) ▼
Triggers	0 Array elements + -
Modifiers	0 Array elements + -

▼	IA_JumpSwat ▼	↶	🔍	+	-
▼	 Space Bar ▼	-			

점프 구현



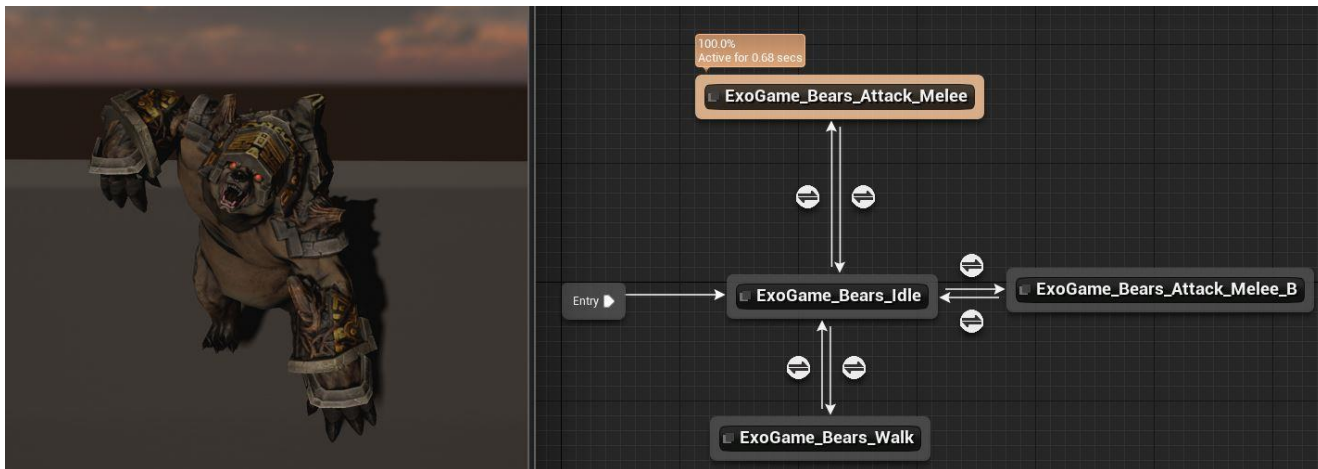
CharacterMovement Component의 Jump 기능 활용.



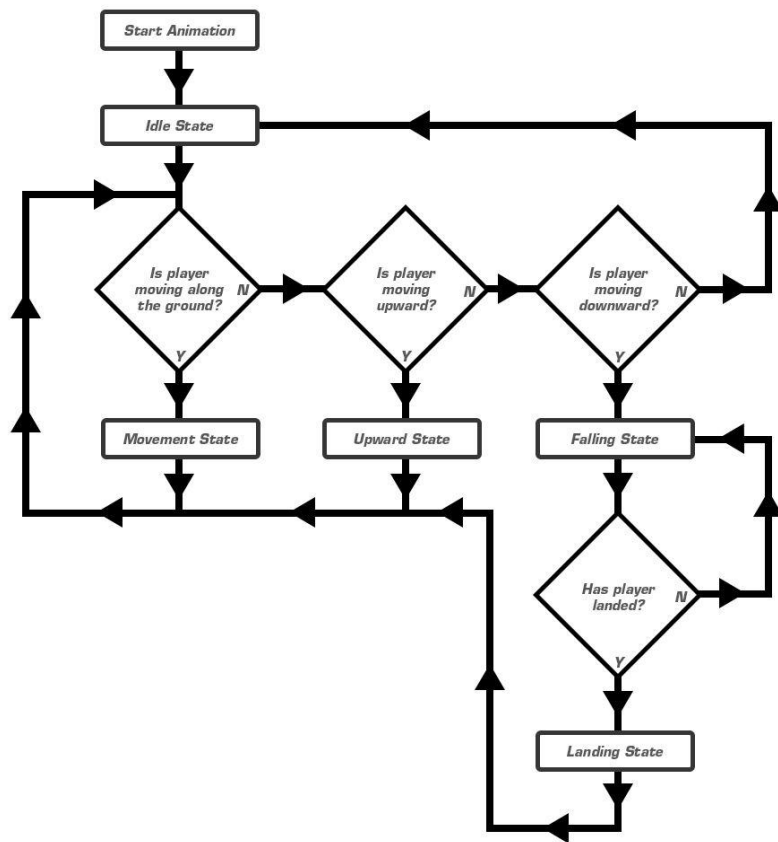
캐릭터가 점프하지만, 점프 애니메이션이 없어 부자연스러움

스테이트 머신(State Machine)

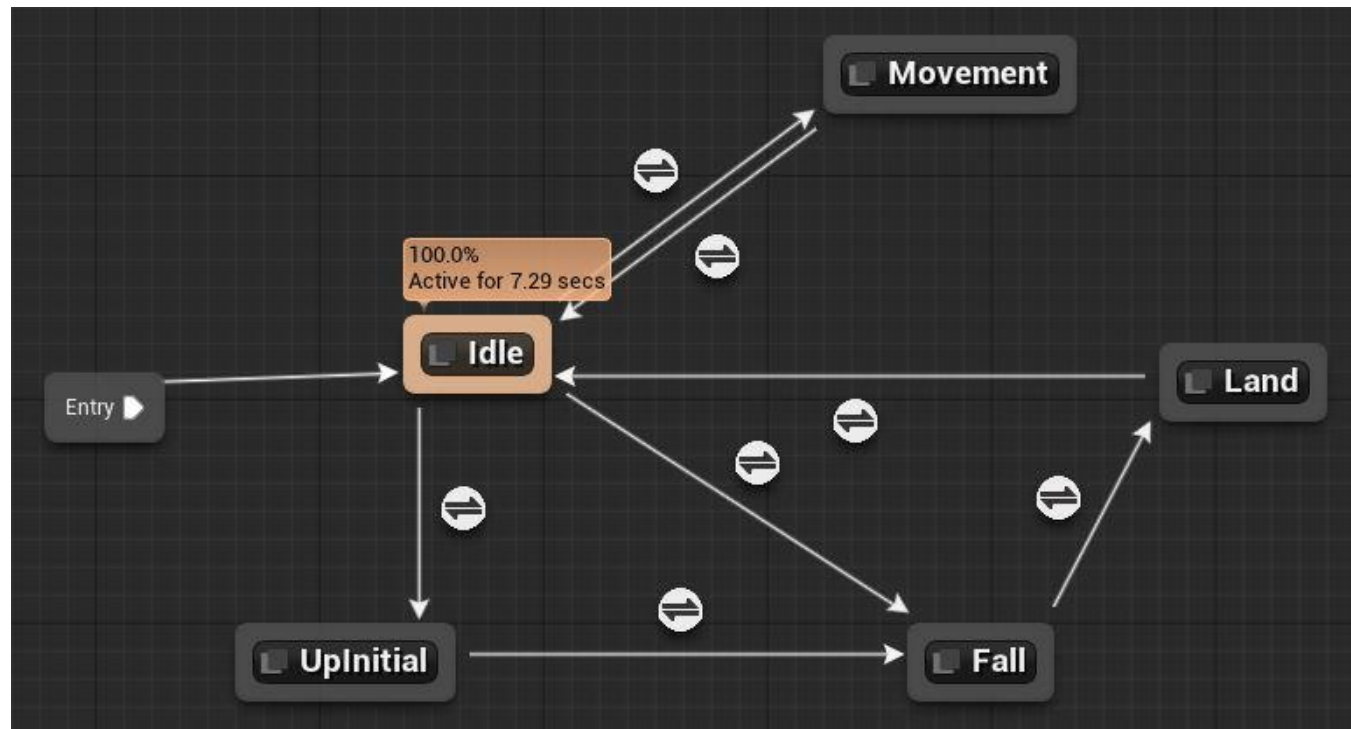
- 애니메이션을 여러 개의 스테이트로 분해하여 그래프로 표시
- Transition Rules (트랜지션 룰)의 관리 하에 한 스테이트에서 다른 스테이트로의 블렌딩을 처리
- 복잡한 블루프린트 망을 만들지 않고도 캐릭터의 다양한 애니메이션 흐름을 그래프로 표현하여 쉽게 제어
→ 프로세스를 크게 단순화시켜주는 도구



전형적인 캐릭터 애니메이션 스테이트 다이어그램



스테이트 머신





실습

점프 애니메이션

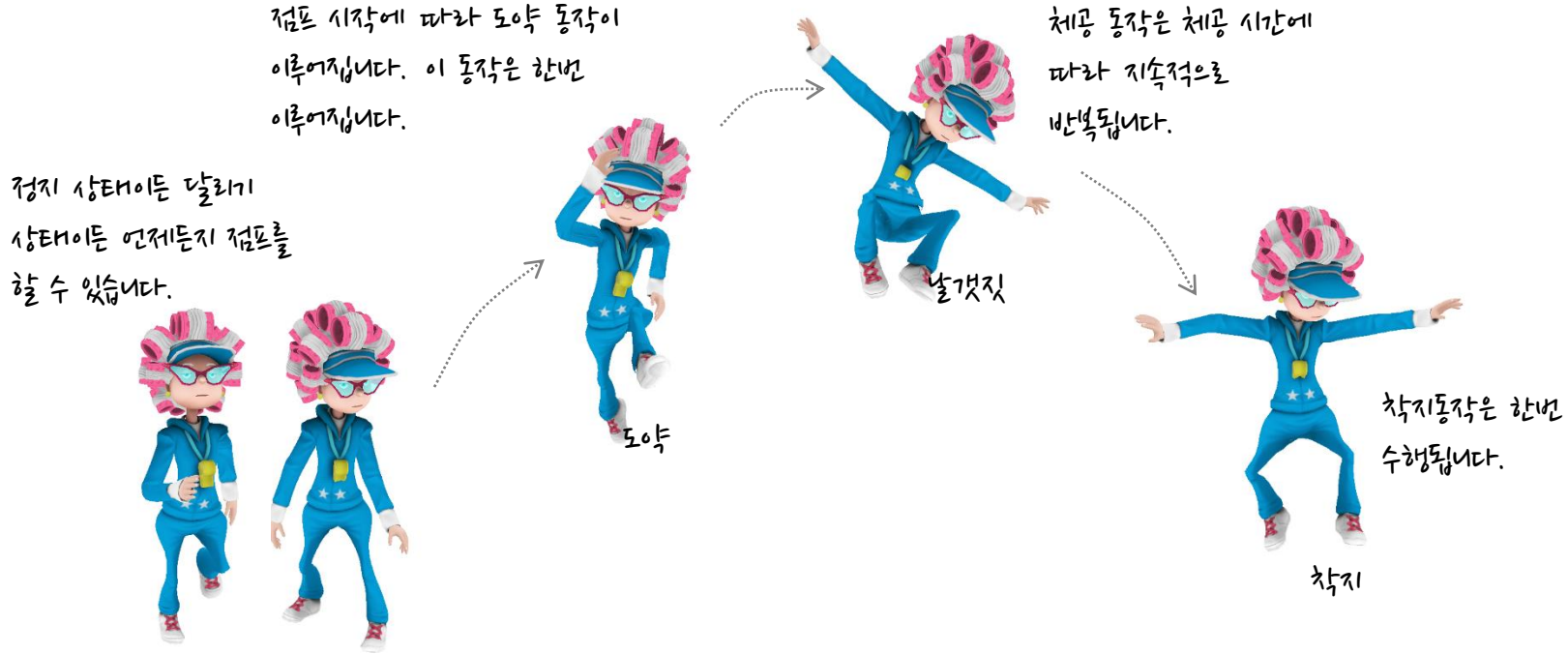
(애니메이션 스테이트 머신)

실습 목표

- 애니메이션 스테이트 머신을 활용한 점프 애니메이션 구현



점프 애니메이션 상태 분석



지민

애니메이션 애셋 준비



Jump
Description: Forward Running
Jump

도약 동작. "Forward
Running Jump"로 검색



Falling Idle
Description: Mid-Air Falling
Idle

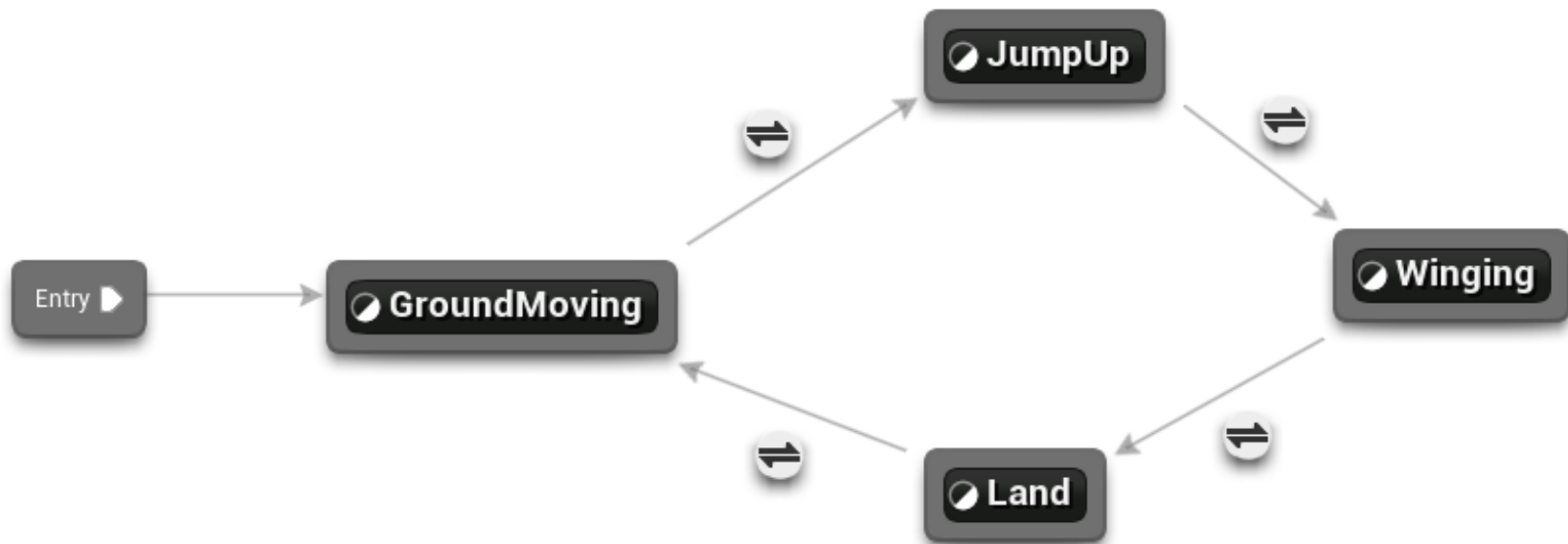
날갯짓 동작. "Falling
Idle"로 검색.



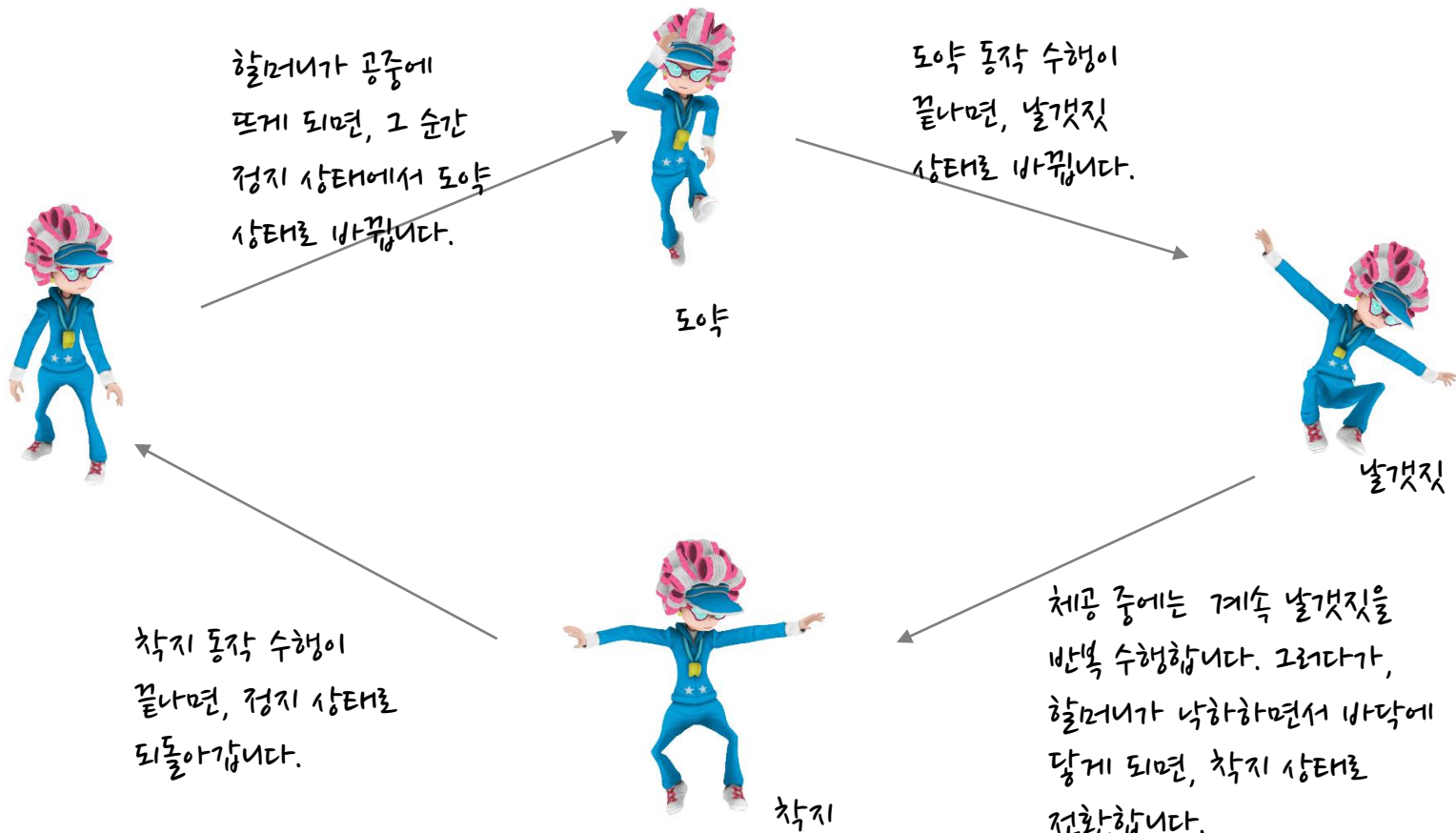
Jumping Down
Description: Jumping Down
From Falling Idle Game Blend

착지 동작. "Jumping
Down"으로 검색.

애니메이션 스테이트 머신

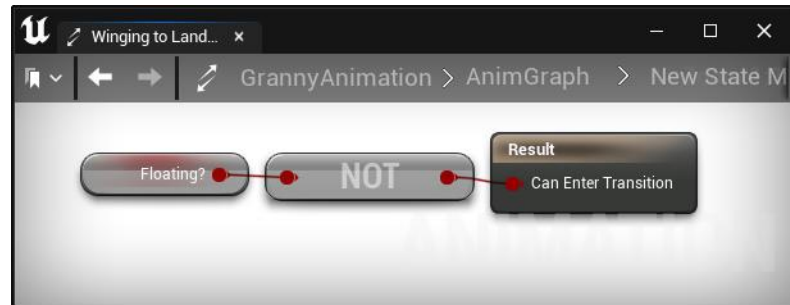
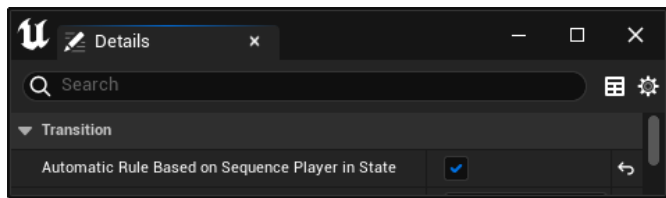
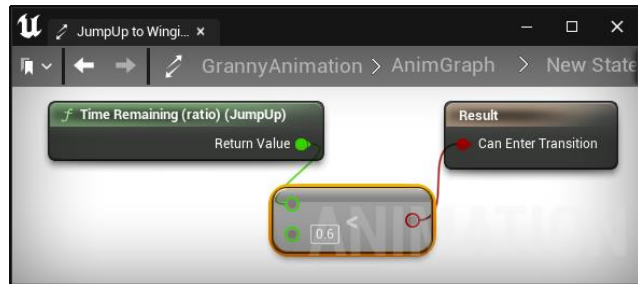
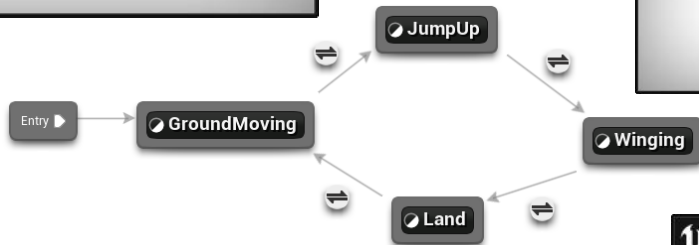
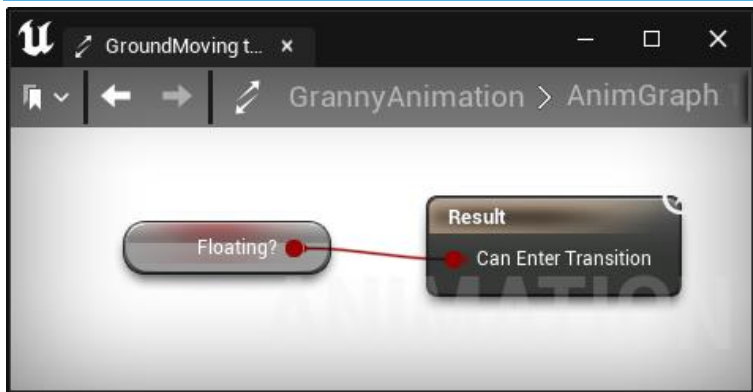


Transition Condition

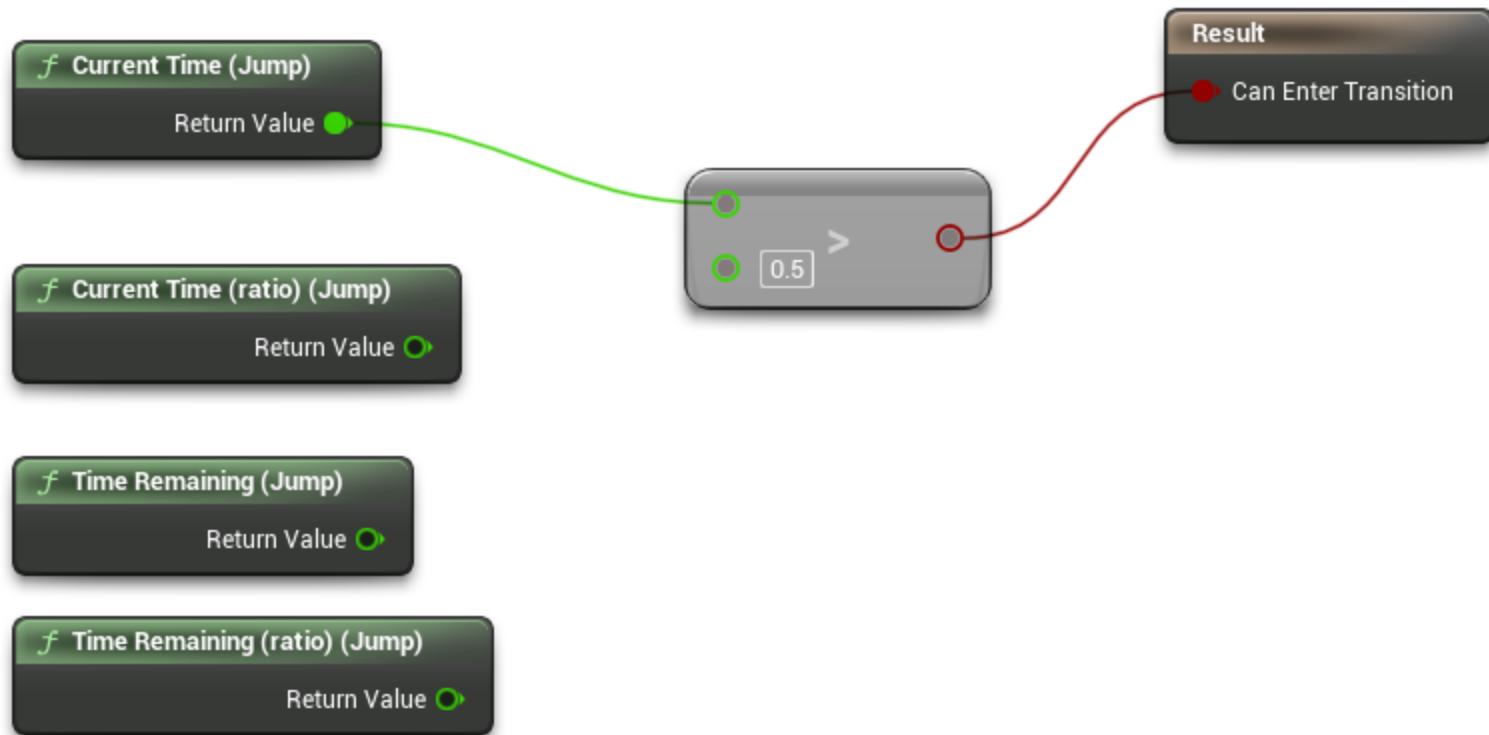


Transition Condition

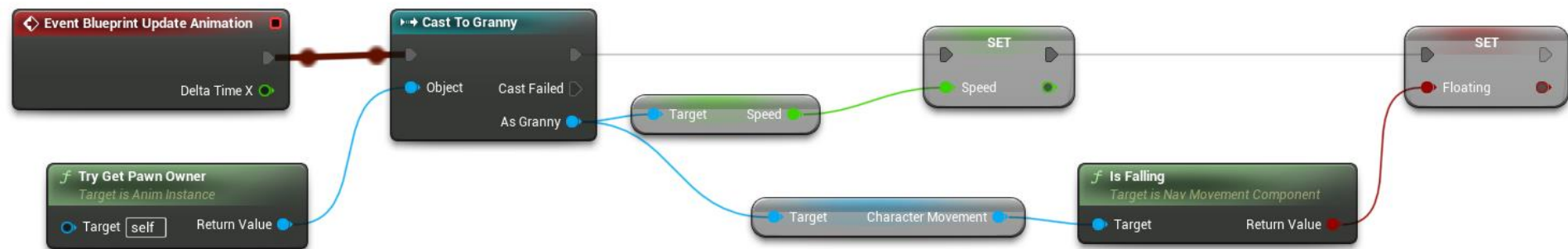
애니메이션 재생 잔여 시간 비율이 60% 보다 낮으면, 다시 말하면 재생 진행 비율이 40%를 넘으면 Winging 상태로 전환이 이루어지도록 했습니다. 왜냐하면 도약 애니메이션을 보면, 재생 시간 비율이 40%인 시점에서 캐릭터가 최고점에 이르고, 그 다음에는 낙하하기 때문입니다. 최고점에 이르는 동작까지만 애니메이션 재생을 하기 위한 조치입니다.



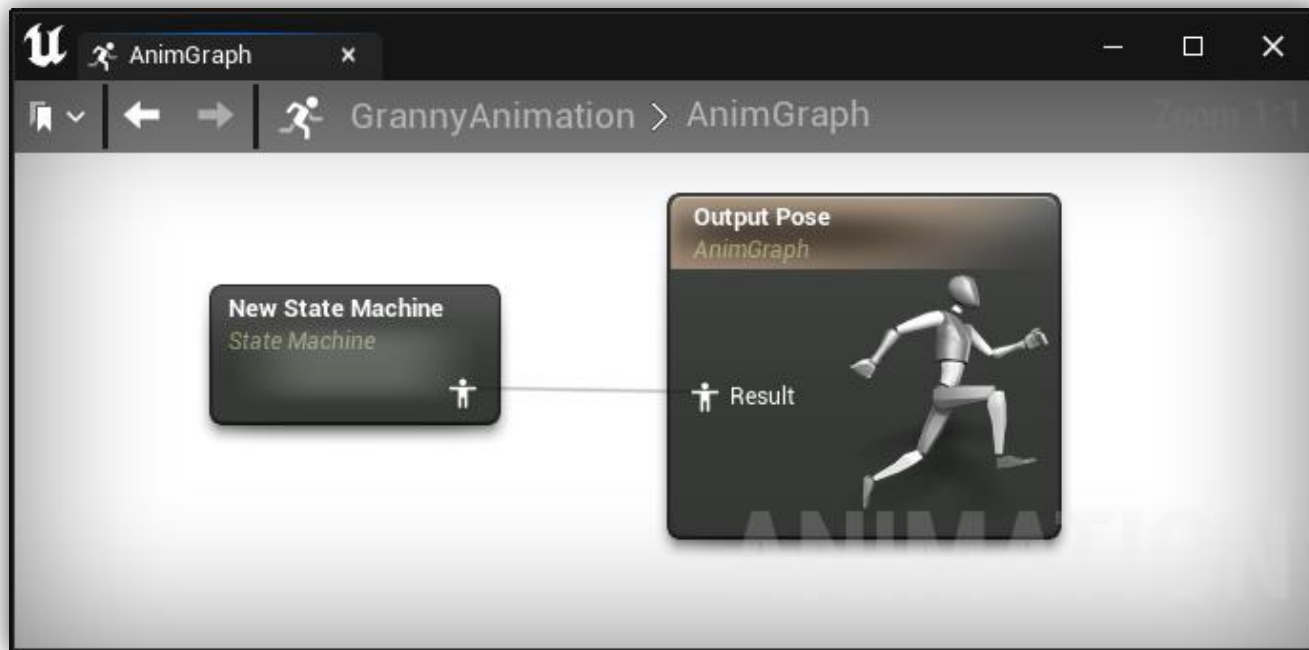
애니메이션 시간 판단



Floating 설정



State Machine 추가와 연결



언리얼 엔진 애니메이션 구현 프로세스

