

# 소스 코드 개요

Unreal4 C++ Portfolio

박서원

네이버 블로그 : <https://blog.naver.com/routine430/222537991966>

유튜브 : <https://www.youtube.com/watch?v=82BDIBmFEu8>

깃허브 : [https://github.com/SeoWon430/Unreal\\_Cpp.git](https://github.com/SeoWon430/Unreal_Cpp.git)

(작성한 cpp 파일명에는 모두 접두사 C가 붙어 있습니다)

# 순차

1. Components
2. Character
3. Actions(Common, Attack)
4. Objects
5. Weapon
6. BehaviorTree
7. Notifies
8. Widget
9. Utilities

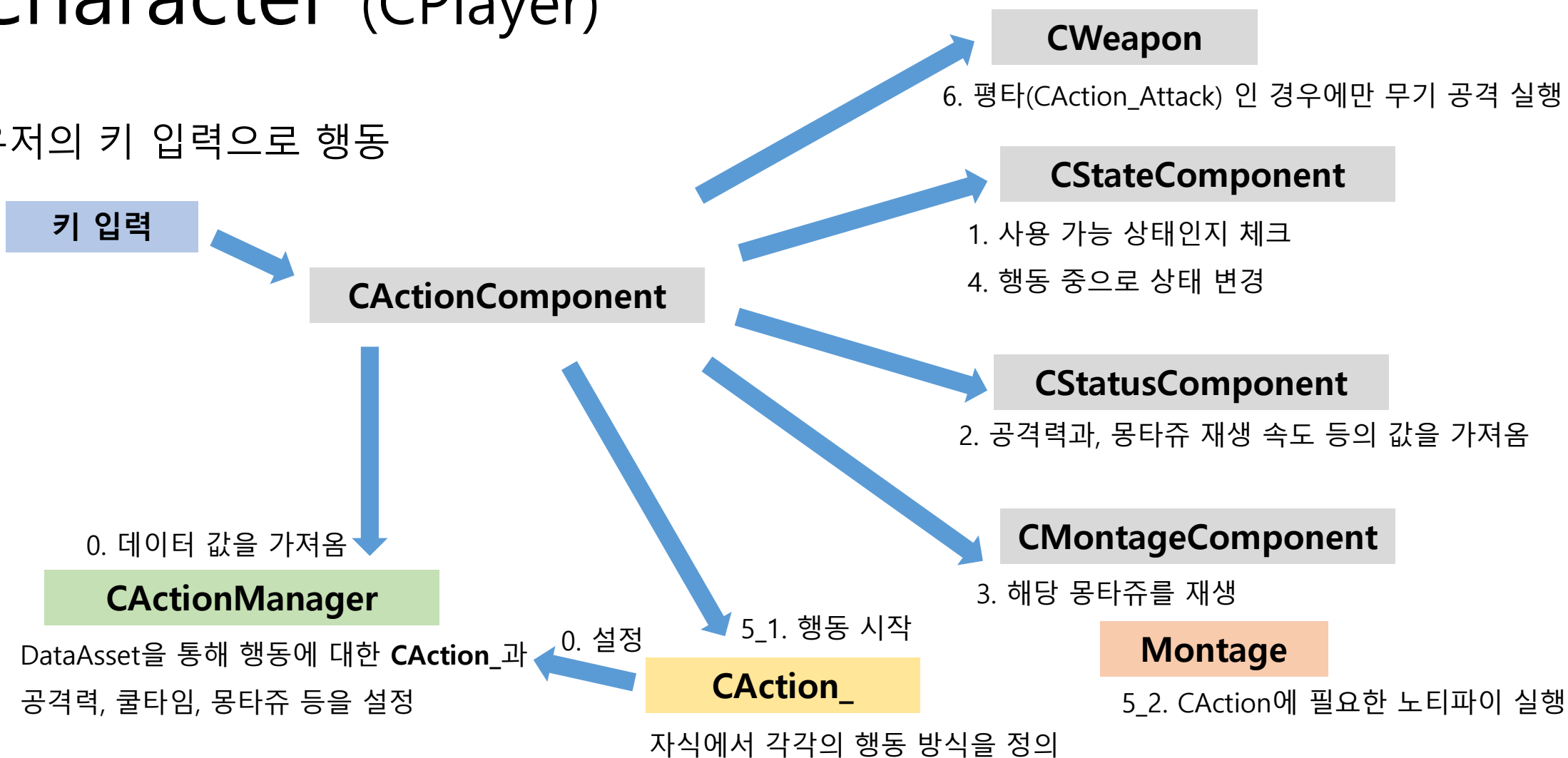
# Components

캐릭터들간 사용 할 공통적 처리를 컴포넌트로 관리

- CActionComponent : 키 입력에 따른 행동
- CBehaviorComponent : AI 행동 경우의 수 판단
- CMontageComponent : 애님몽타쥬 재생
- CPatrolComponent : AI의 Behaviour에서 순찰 시 처리 할 경로 관리
- CStateComponent : 캐릭터의 상태 관리
- CStatusComponent : 캐릭터의 스탯(체력, 공격력 등) 관리
- CWidgetDamageTextComponent : 피격시 데미지 텍스트 표시
- CWidgetPlayerComponent : 플레이어의 인터페이스 표시

# Character (CPlayer)

유저의 키 입력으로 행동



# Character (CPlayer)

컴포넌트 외 플레이어 캐릭터가 가지는 추가적 기능

- 벽타기(Wall) : [LineTrace](#)를 통해 벽을 판단하고, 캐릭터의 위치와 회전 값을 바꿈
- 스캔(Scan) : [Postprocess](#)와 [MaterialParameterCollection](#), [Custom Depth](#)를 사용하여 특정 대상 탐색
- 죽음 효과(End\_Dead) : [DestructibleMesh](#)와 [SceneCaptureComponent2D](#)를 이용하여 죽었을 때 순간적으로 화면이 깨지는 효과를 표현



벽타기



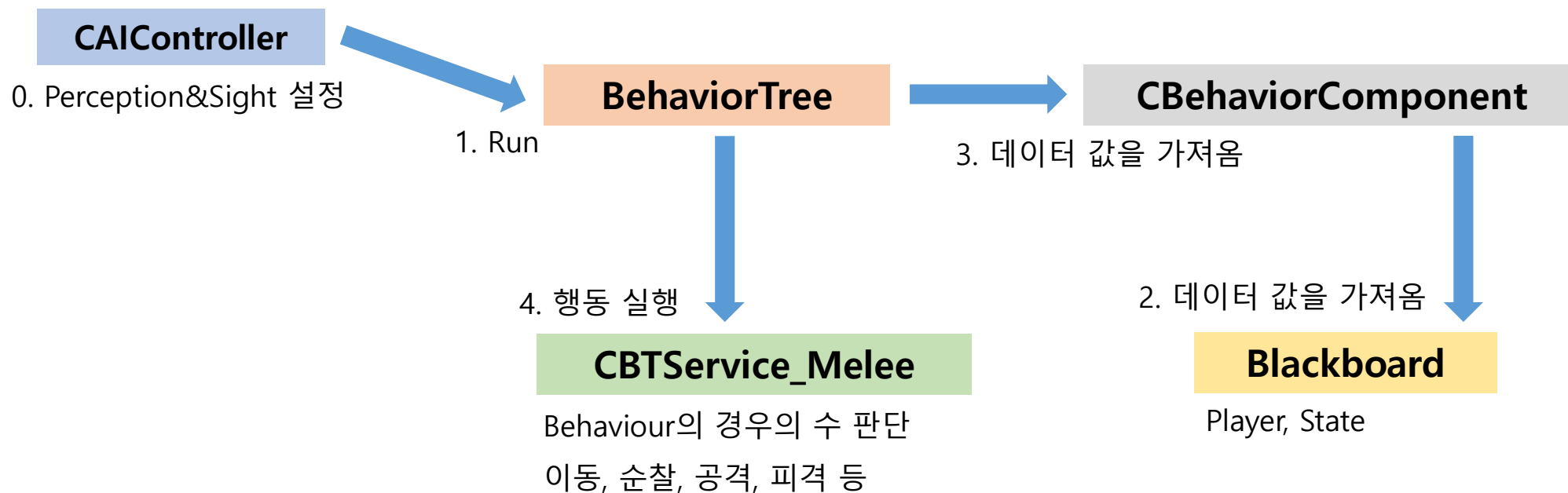
스캔



죽음 효과

# Character (Cenemy\_AI)

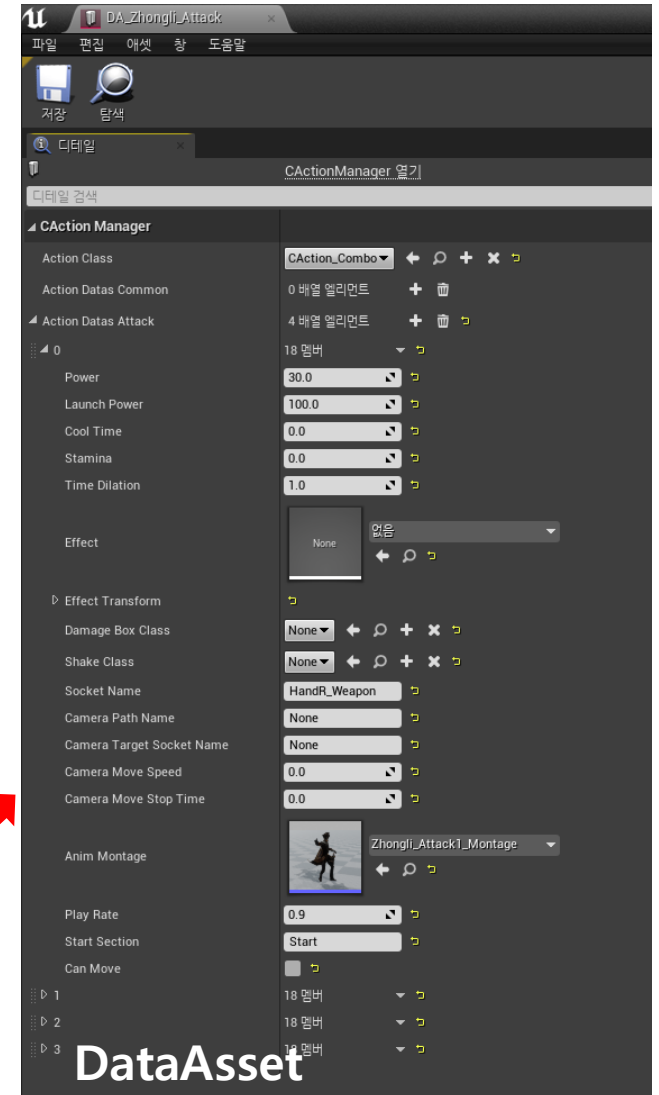
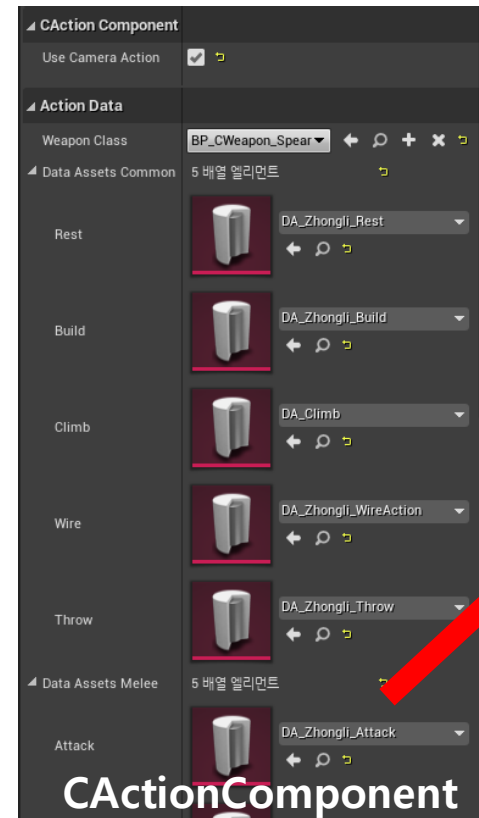
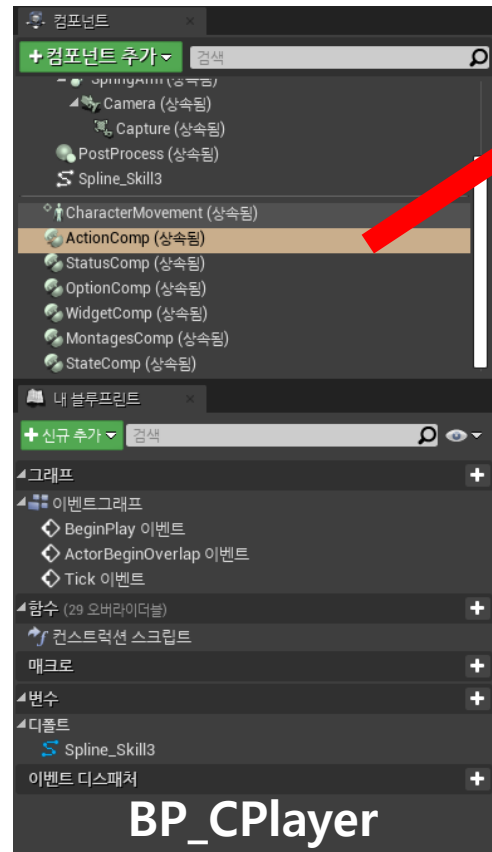
AI Controller에 의해 행동



# Actions

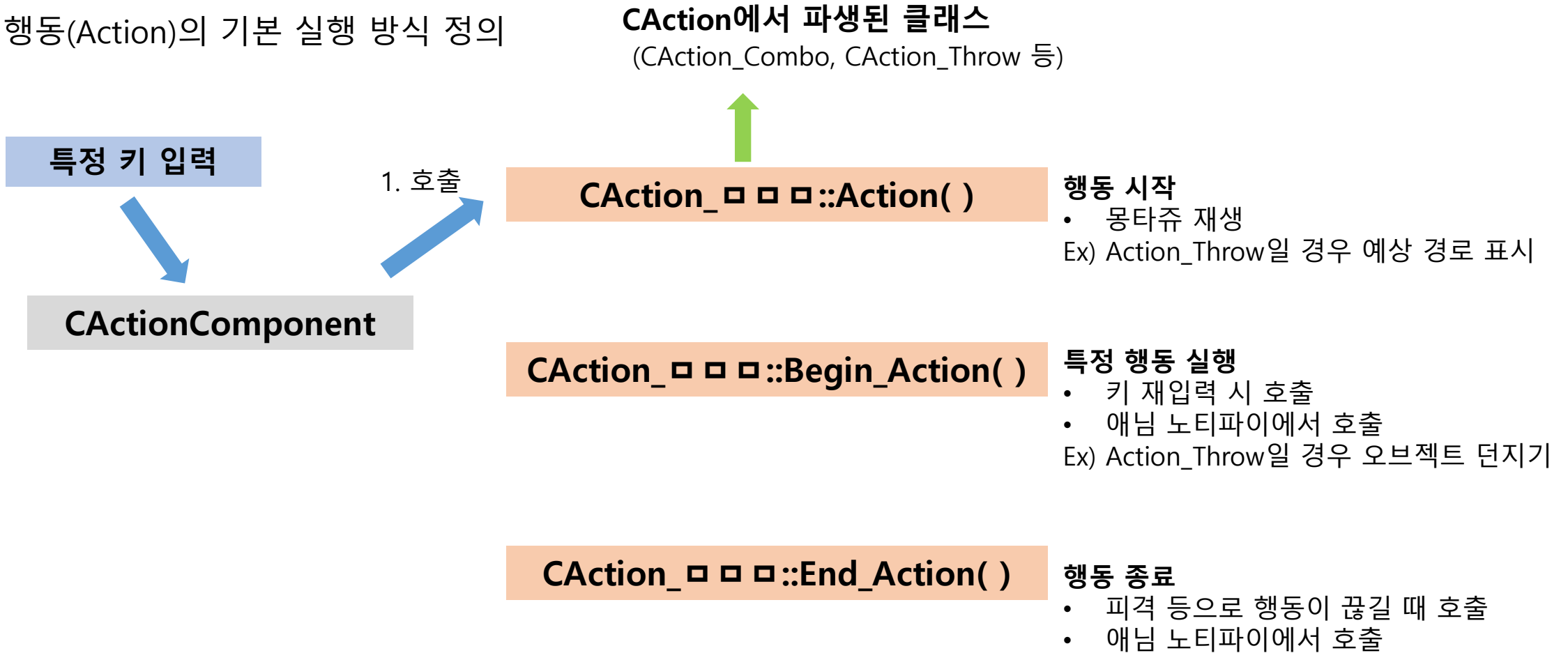
**CActionData** : Action에 필요한 데이터 정의 (공격력, 몽타주 등)

**CActionManager** : [DataAsset](#)으로 CActionData에서 정의한 데이터를 설정



# Actions

행동(Action)의 기본 실행 방식 정의





# Actions

CAction을 상속하여 만들어진 클래스

- CAction\_Build, CAction\_Throw, CAction\_Combo, CAction\_Cutter등

행동 방식에 따라 2가지로 분류

## 1) Common : 일반 행동

- DataAsset에서 **FActionData\_Common**구조체를 사용 (hp회복량, Build정보 세팅)
- Build, Throw등이 해당

## 2) Attack : 공격 행동

- DataAsset에서 **FActionData\_Attack**구조체를 사용 (공격력, 쿨타임 등 세팅)
- Combo, Cutter 등이 해당

# Actions (Common)



## **CAction\_Build** : 오브젝트를 설치

- **LineTrace**로 설치 위치를 정하고, 설치 가능여부를 판단
- 위에서 구한 **Normal**에 따라 벽, 바닥을 판단하여 설치 가능/불가능을 판단
- 설치 가능 여부에 따라 위 오브젝트의 색상을 녹색/적색으로 변경
- 키 재입력 시 취소, 마우스 좌클릭 시 설치



## **CAction\_Climb** : 정면의 벽이나 언덕을 뛰어 넘음

- **LineTrace**로 정면 벽의 두께와 높이를 파악
- 두께가 얇으면 뛰어 넘고, 두꺼우면 넘어섬

# Actions (Common)



**CAction\_Throw** : 오브젝트를 던짐

- [PredictProjectilePath](#)를 이용하여 던지기 예상 경로를 구함
- 위에서 구한 경로를 CLaser로 표시
- 최종 지점의 [Normal](#)을 구하여 표시 할 Decal의 방향 수정



**CAction\_WireAction** : 오브젝트를 던지고 해당 위치로 날아감

- **CAction\_Build**처럼 오브젝트를 설치, 설치 할 때 해당 방향으로 오브젝트를 던짐
- 던지는 오브젝트는 [ProjectileMovement](#)에 의해 날아감
- [SetActorLocation](#)과 [Sin](#)함수를 이용하여 곡선을 그리며 날아가도록 함

# Actions (Attack)



## **CAction\_Combo** : 연속 공격

- 플레이어의 기본 공격 (총 4타로 구성)
- CWeapon\_Spear를 통해 각 타수의 기능을 실행
- Cweapon에 **바인딩** 되어 무기가 적에 충돌 시데미지를 줌



## **CAction\_Buff** : 스테이터스 보너스

- 플레이어의 스킬2
- CStatusComponent의 Status구조체의 추가(Add) 값 설정
- 원래 스탯(Origin)과 추가 스탯(Add) 구조체로 관리
- Timmer에 의해 일정 시간이 지나면 다시 돌아옴

# Actions (Attack)



**CAction\_SpawnObject** : 오브젝트 생성

- 플레이어의 스킬1, 3
- CSpawnObject를 스폰하여 특정 위치로 오브젝트를 이동 시킴



**CAction\_Continuous** : 키 지속 동안 이펙트 생성

- 플레이어의 스킬4
- CSpawnEffect를 스폰하여 특정 위치에 Niagara를 스폰
- 키 입력이 지속 된 상태에서만 이펙트가 유지 됨

# Actions (Attack)



**CAction\_Cutter** : 적을 행동 불가 상태로 만들고 레이저로 자름

- 플레이어의 스킬5
- 적중 시 적의 [CustomTimeDilation](#)을 0으로 하여 기능을 정지
- CProceduralSkeletalMesh를 스폰하여 적의 SkeletalMesh에서 [ProceduralMesh](#)생성
- 6개의 기둥을 설치하고 6개의 CLaserCutter를 생성
- CLaserCutter는 날아가며 위에서 생성 한 [ProceduralMesh](#)를 [Slice](#)하고 기둥에 반사 되어 다시 [ProceduralMesh](#)에 돌아와 [Slice](#)함

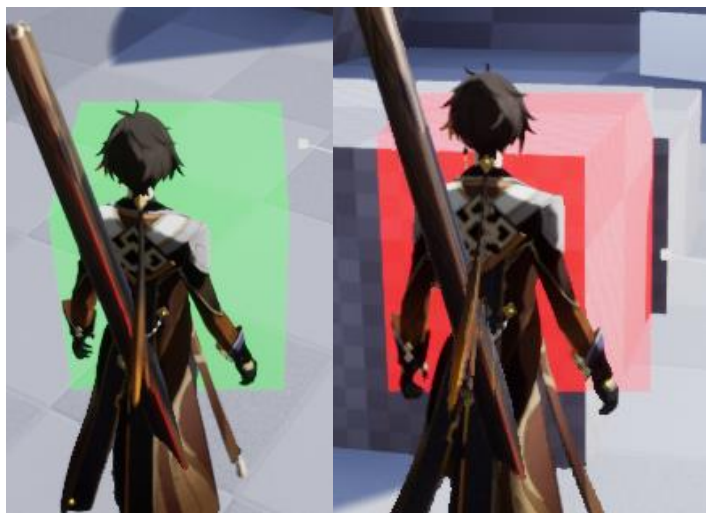


# Objects



## CLaserCutter

- CAction\_Cutter에서 사용
- Niagara에서 Ribbon Render를 이용하여 렌더
- 각 레이저는 직선만 표현하며, 반사를 위해 다수의 NiagaraComponent를 가짐
- Niagara Parameter 값을 조정하여 레이저 거리와 방향을 설정
- LineTrace를 통해 부딪힌 지점의 위치와 방향을 파악
- 위에서 구한 방향을 통해 반사각에 맞게 다음 레이저를 렌더



## CBuildObject

- CAction\_Build에서 사용
- 설치 할 오브젝트를 일단 스폰
- 설치 가능 여부에 따라 녹색/적색 반투명으로 오브젝트를 렌더
- DynamicMaterial을 통해 런타임 중 메터리얼의 색상을 변경
- 최종 설치가 되면 오브젝트의 메터리얼을 원본으로 바꿈

# Objects



## CThrowDestination

- CAction\_Throw에서 사용
- [PredictProjectilePath](#)를 이용하여 던지기 예상 경로를 파악
- 예상 경로는 Vector3 Array로 이 위치 값들을 CLaser에 적용하여 궤도의 선을 표현
- 도착 지점에는 [Decal](#)을 표시하며, 바닥이나 벽의 방향에 맞게 회전하여 [Decal](#)을 그림

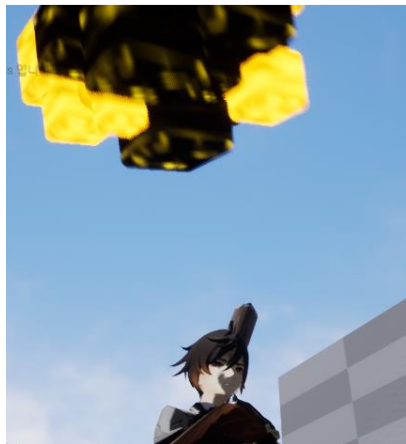


## CThrowObject

- CAction\_Throw에서 사용
- [ProjectileMovement](#)를 통해 날아감
- 위의 예상경로에서 사용 했던 방향과 힘의 값이 같아야 예상 경로대로 날아감
- 예상 경로의 곡선이 그려지게 할려면 힘의 값이 낮아져야 하고,  
이 오브젝트의 날아가는 속도도 느려짐
- 위의 문제를 해결하기 위해 날아 갈 때, [CustomTimeDilation](#)을 3배로 하여 빠르게 날아가도록 함

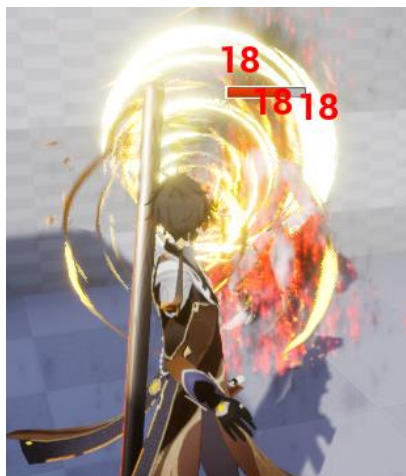
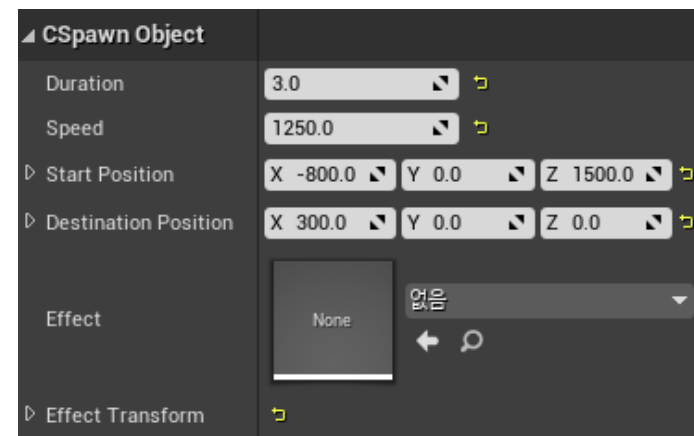


# Objects



## CSpawnObject

- CAction\_SpawnObject에서 사용
- 특정 오브젝트(StaticMesh)를 생성
- 스폰 후 설정한 위치를 향해 이동 (플레이어 기준 상대 좌표로 이동)



## CSpawnEffect

- CAction\_Continuous에서 사용
- 특정 이펙트(Niagara)를 생성
- Overlap된 대상 모두 일정 간격 데미지를 줌

# Objects



## CProceduralSkeletalMesh

- CAction\_Cutter에서 사용

- ProceduralMesh는 기본적으로 StaticMesh에서만 생성 가능

1. SkeletalMesh는 애니메이션이 있지만 한 순간만은 StaticMesh와 같음
2. SkeletalMesh에서 **Weight**를 고려하여 특정 순간, 애니메이션이 적용된 모습을 만들 수 있음
3. 이 Weight는 Mesh Model을 만들 때 적용된 Bone Skinning정보로, 이 값을 통해 현재 실행중인 애니메이션에 대한 Bone정보를 가져 옴
4. 위의 과정으로 한 순간 애니메이션이 적용된 SkeletalMesh를 StaticMesh처럼 만들어 **ProceduralMesh**로 만듦

우측 사진은 각기 다른 2개의 애니메이션으로 만들 **ProceduralMesh**이며, CLaserCutter로 **Slice**하고 있음

# Weapon

캐릭터가 사용 할 무기 (CWeapon에서 상속 받으며, 현재는 Spear만 구현 됨)

- **CWeapon\_Spear** : 기본 공격(1~4타)의 공격 방식을 정의

- CAction\_Combo 에서 현재 실행중인 타수에 따라 아래 무기 행동을 실행
- CAction\_Combo 에서 재생한 몽타주에서 CAnimNotifyState\_WeaponAction 에 의해 CWeapon::Action(UINT)가 호출 됨

- CWeapon\_Spear::Action(UINT)

- 원신 '종려'의 공격 방식 참고
- 1타 : 창을 앞으로 밀며 타격
- 2타 : 창을 들고 횡베기
- 3타 : 창을 위에서 내려꽂기
- 4타 : 창을 회전시키며 앞으로 날리기

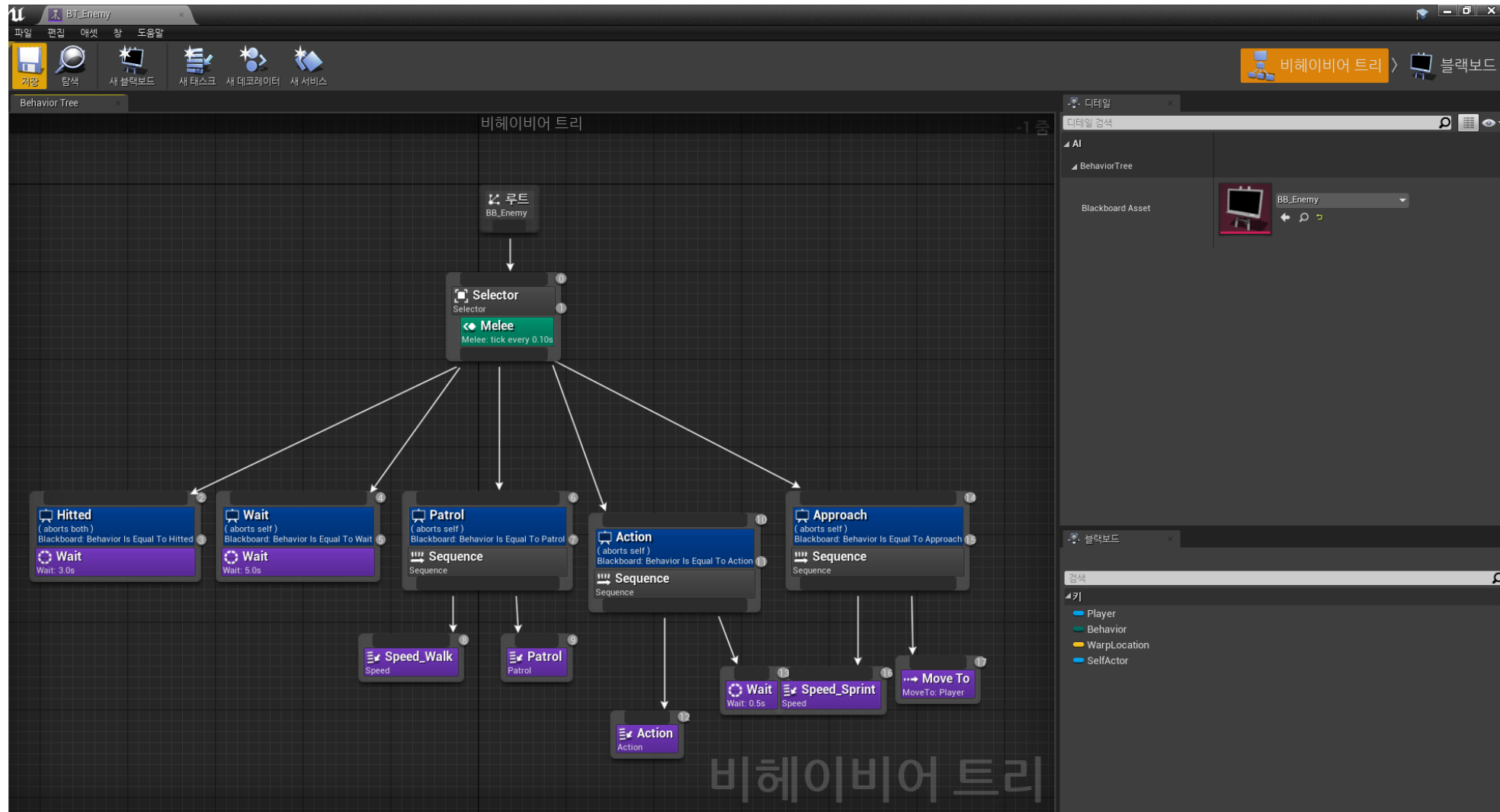


# BehaviorTree

CEnemy의 AI Controller에서 사용할 행동 방식의 경우를 정의

- CBTSERVICE\_Melee : 현재 실행 할 행동의 경우를 판단  
피격, 순찰, 공격, 추적 등
- CBTTASKNODE\_Action : CActionComponent를 통해 Action 실행
- CBTTASKNODE\_Patrol : CPatrolComponent를 통해 경로를 따라 이동
- CBTTASKNODE\_Speed : 캐릭터의 현재 움직이는 속도 설정

# BehaviorTree

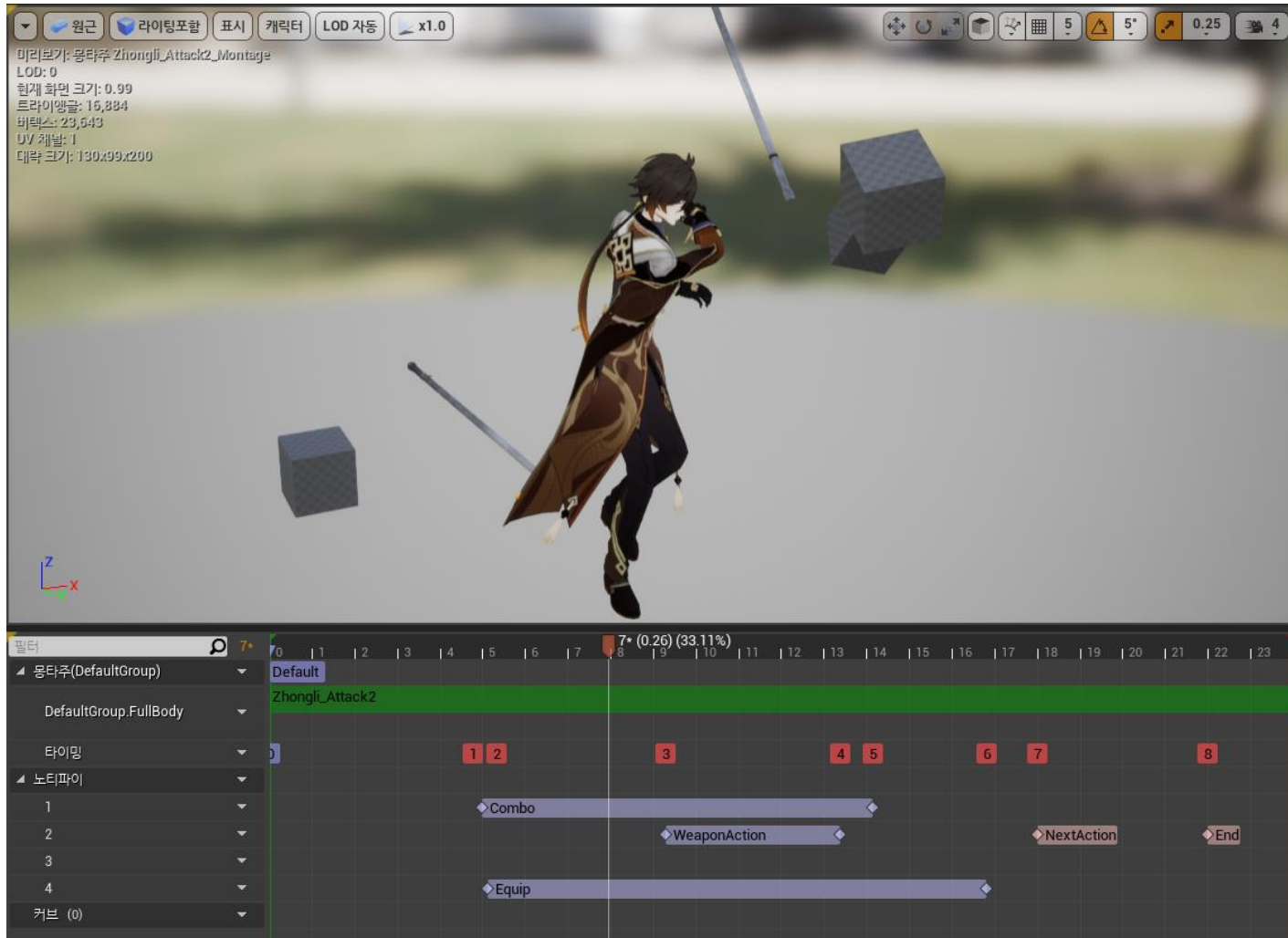


# Notifies

애니메이션 몽타쥬 재생 동안 호출될 노티파이

- CAnimNotify\_BeginAction : 현재 실행 중인 CAction에 대해 Begin\_Action() 호출
- CAnimNotify\_End : 현재 실행 중인 CAction에 대해 End\_Action() 호출
- CAnimNotify\_NextAction : CAction\_Combo에서 다음 동작( Begin\_Action() ) 실행
- CAnimNotifyState\_Combo : 유지시간 내 키 추가 입력 시 CAction\_Combo에서 다음 동작 실행 가능
- CAnimNotifyState\_Collision : 유지시간 내 CAction\_Combo에서 무기의 충돌 가능여부 설정
- CAnimNotifyState\_Equip : 유지시간 동안 무기를 특정 소켓에 Attach
- CAnimNotifyState\_WeaponAction : 유지시간 동안 CAction\_Combo에서 각 동작의 무기 공격 실행

# Notifies



End : CAnimNotify\_End

NextAction : CAnimNotify\_NextAction

WeaponAction : CAnimNotifyState\_WeaponAction

Combo : CAnimNotifyState\_Combo

Equip : CAnimNotifyState\_Equip

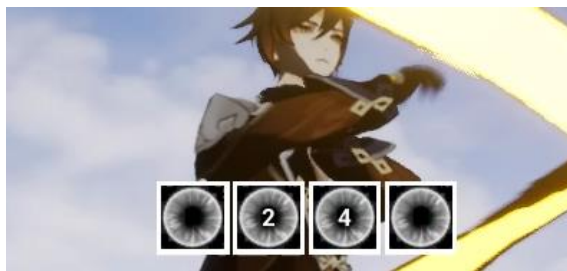
# Widgets

위젯 관련 클래스 (필요 한 함수를 BlueprintImplementableEvent로 선언만 한 후 BP에서 정의)

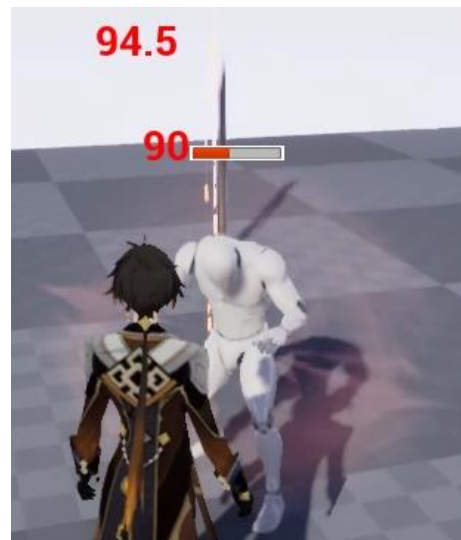
- CUserWidget\_DamageText : 캐릭터 피격시 데미지 텍스트를 띄움
- CUserWidget\_EnemyHealth : 적 캐릭터의 Hp를 머리 위에 표시
- CUserWidget\_PlayerStatus : 플레이어의 Hp, Stamina를 표시하는 UI
- CUserWidget\_SkillSet : 플레이어의 스킬 쿨타임을 표시하는 UI



PlayerStatus



Skillset



DamageText  
EnemyHealth



# Utilities

미리컴파일, 매크로, 로그 관련 정의

- CHelpers : Component 생성, Asset과 Class가져오기 등을 정의
- CLog : Log관련 정의
- Defines : 매크로 정의 (객체 Null, 변수 True/False 체크)