08_Character 클래스

1. Character

1) Character

- Character (캐릭터)는 걸어다닐 수 있는 능력을 지닌 특수 유형 Pawn을 말한다.

- 이족보행을 하는 인간형 오브젝트에 매우 적합하다.
- CharacterMovementComponent, CapsuleComponent,SkeletalMeshComponent 의 추가를 통해 Pawn 클래스는 고도로 특화된 Character (캐릭터) 클래스로 확장된다.
- 캐릭터는 월드에서 걷기, 달리기, 점프, 비행, 수영 등이 가능한 플레이어를 표현한다.
- 이 클래스에는 기본적인 네트워킹이나 인풋 모델 구현도 추가되어 있다.

2) CapsuleComponent

- CapsuleComponent 는 운동 콜리전에 사용된다.
- CharacterMovementComponent 에 대한 복잡한 지오메트리 계산을 위해, Character 클래스의 콜리전 컴포넌트는 직립된 캡슐 모양이라는 가정을 한다.

3) SkeletalMeshComponent

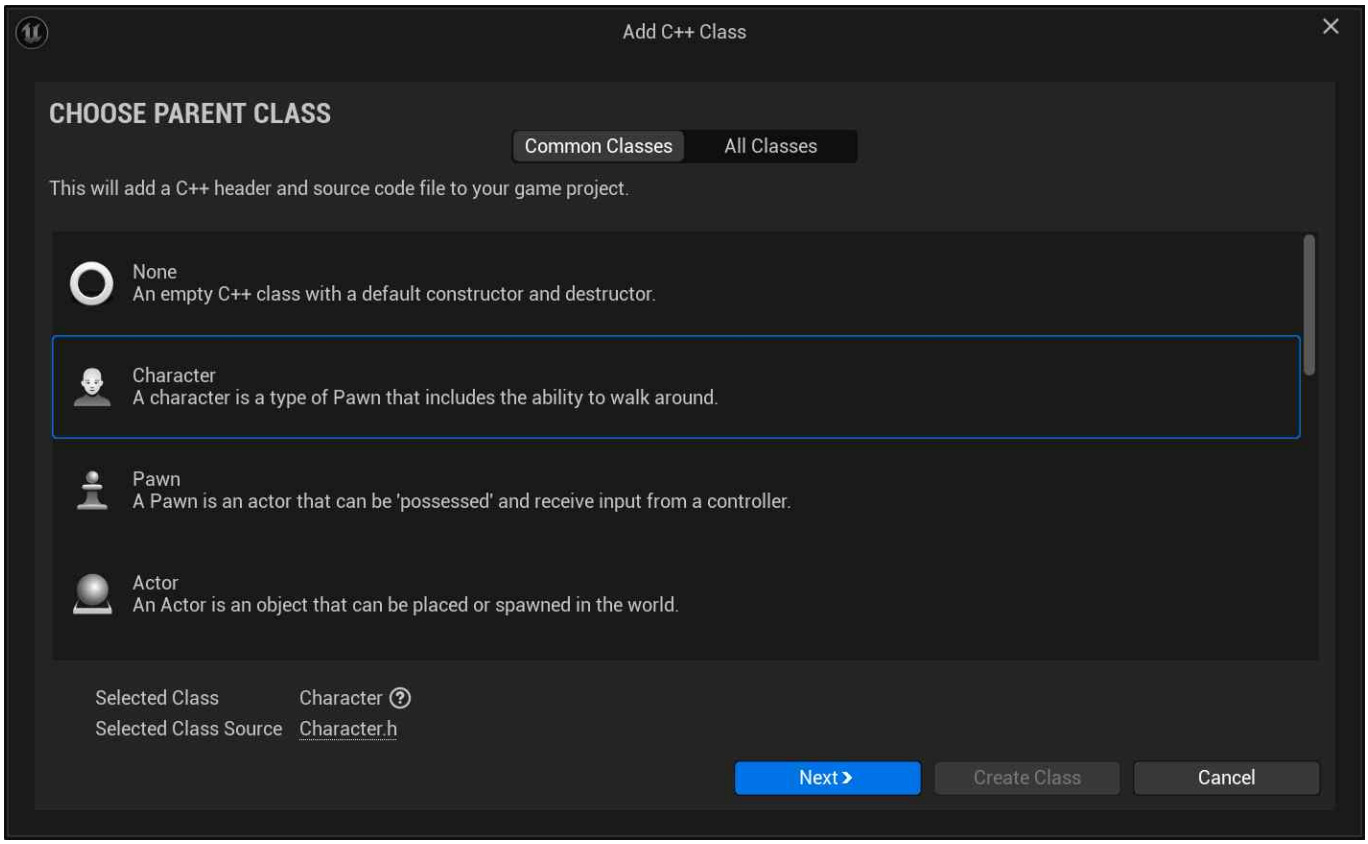
- 폰과 달리 캐릭터에는 스켈레톤을 사용하는 고급 애니메이션을 위한 SkeletalMeshComponent 가 딸려온다.
- 다른 스켈레탈 메시를 Character 파생 클래스에 추가시키는 것도 가능하지만, 캐릭터에 관련된 주 스켈레탈 메시는 SkeletalMeshComponent 이것이다.

4) CharacterMovementComponent

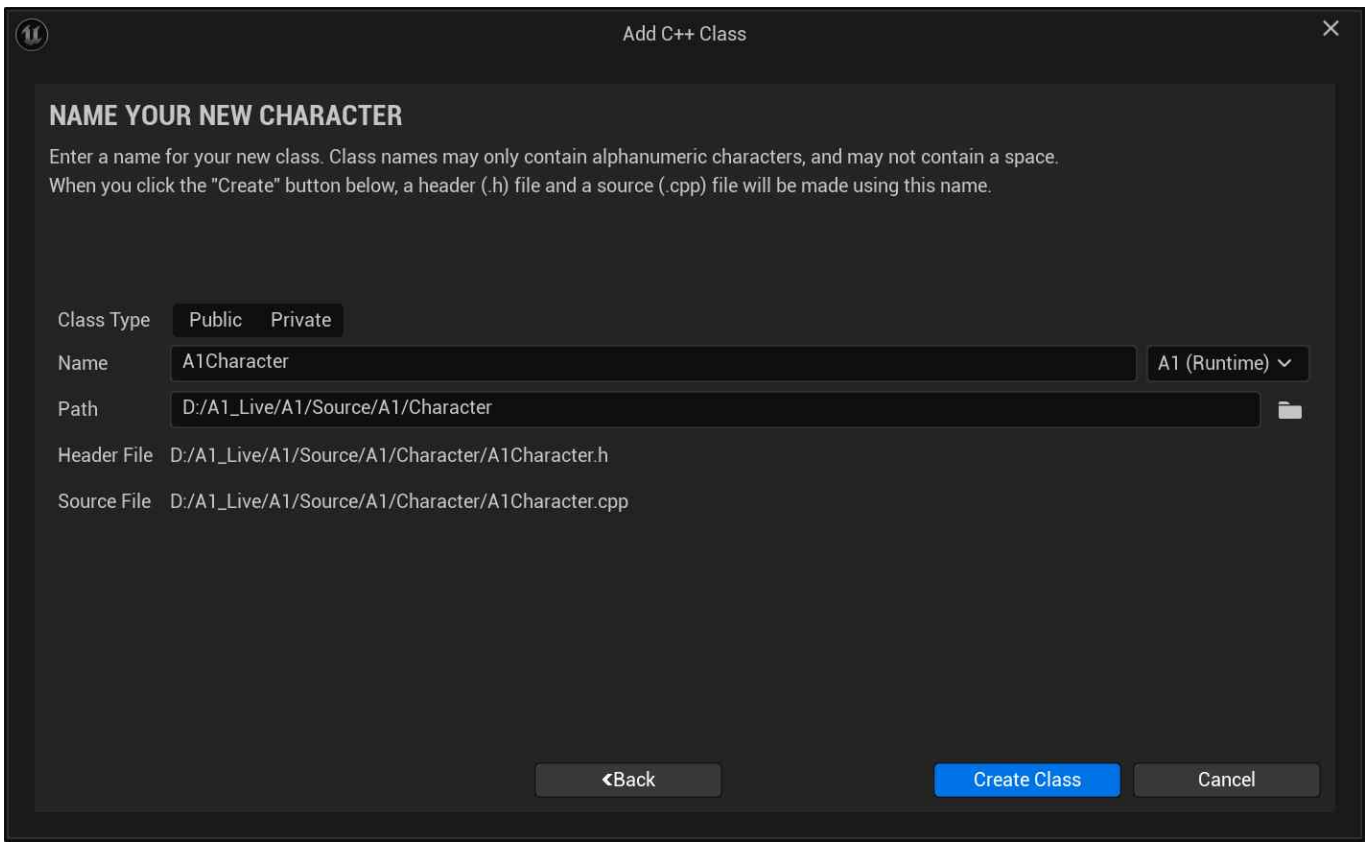
- CharacterMovementComponent 는 아바타가 걷기, 달리기, 점프, 낙하, 수영 등으로 이동할 때 리지드 바디 피직스를 사용하지 않도록 할 수 있다.
- 캐릭터에 국한된 것으로, 다른 클래스에서 구현할 수 없다.

5) A1Character 클래스 추가하기

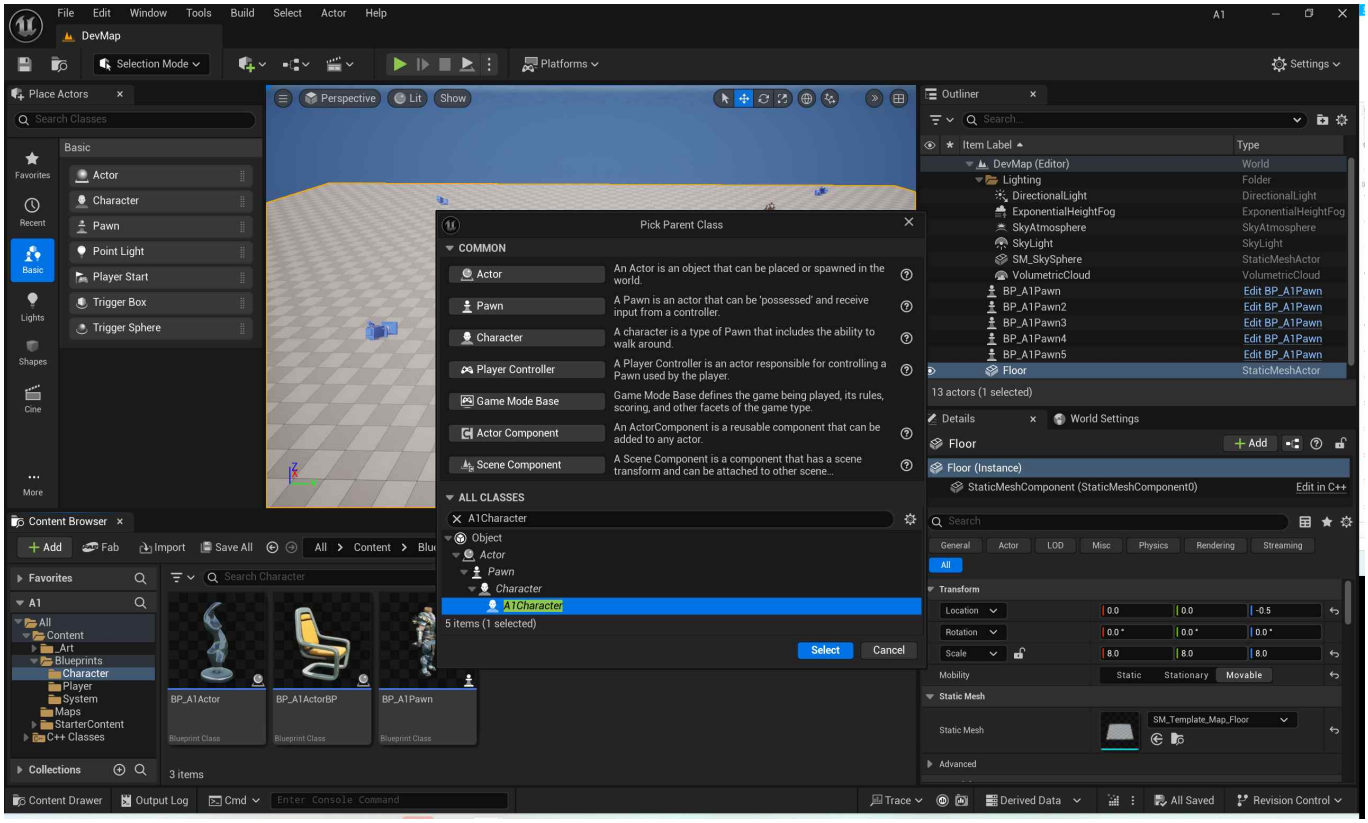
- Character 클래스를 상속받은 A1Character 클래스를 추가해보자.
- **[Tools] - [New C++ Class...]** 통해 Character를 상속받은 C++ 클래스를 추가하자.



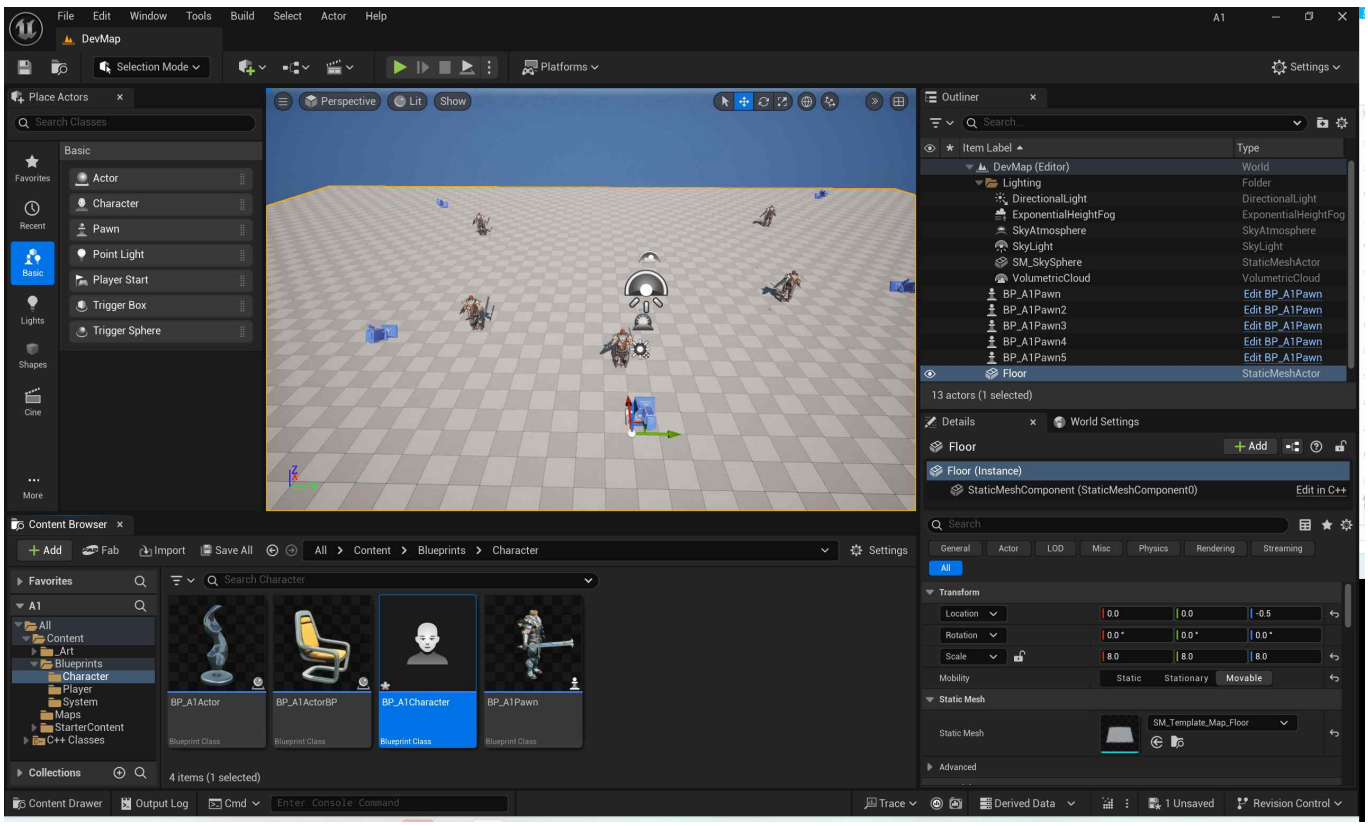
- 추가 할 **A1Character**의 위치는 **A1폴더** 하단의 **Character 폴더**에 위치시키자.



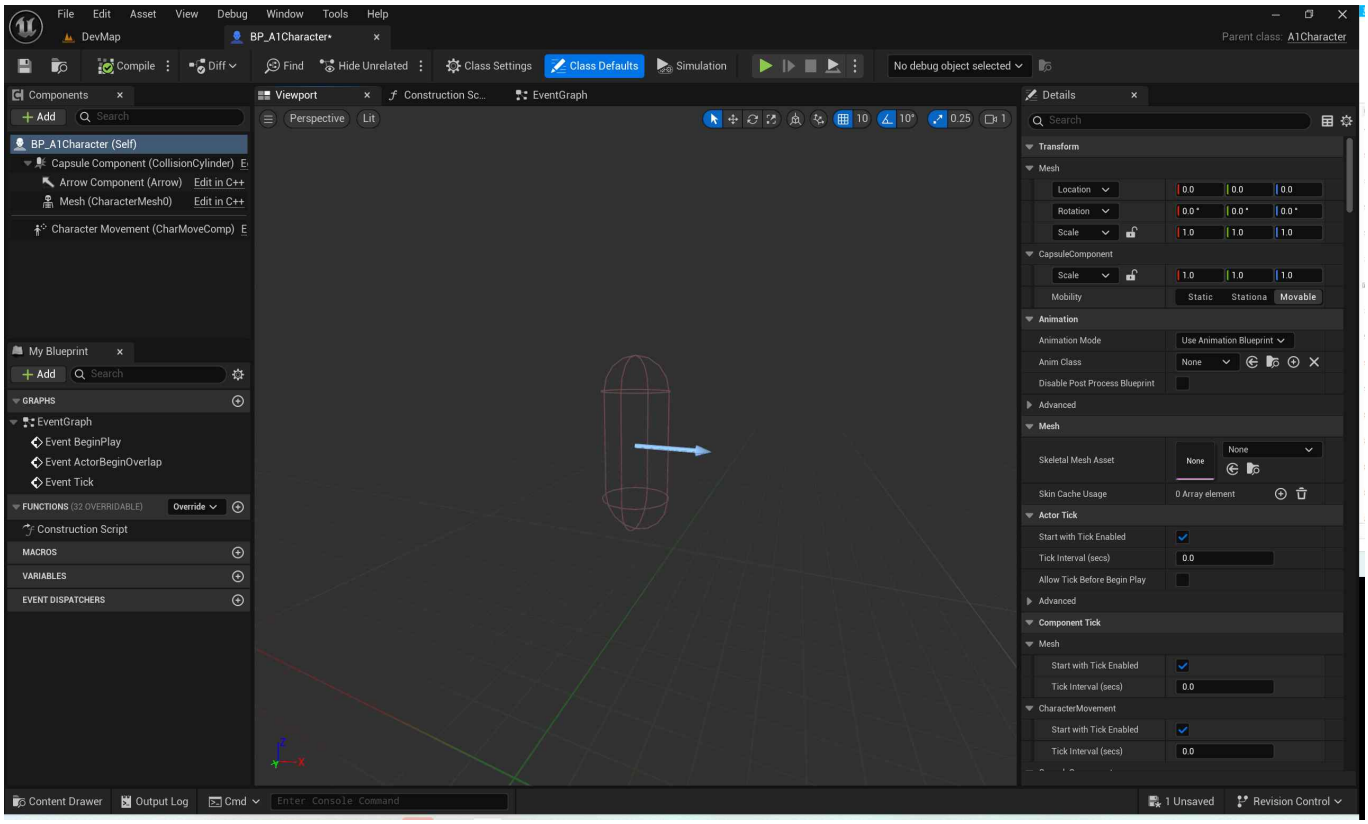
- 생성된 A1Character 클래스를 살펴보도록 하자.
- **A1Character**를 상속받은 **BP_A1Character** 블루프린트 클래스를 **Character폴더**에 생성하도록 하자.



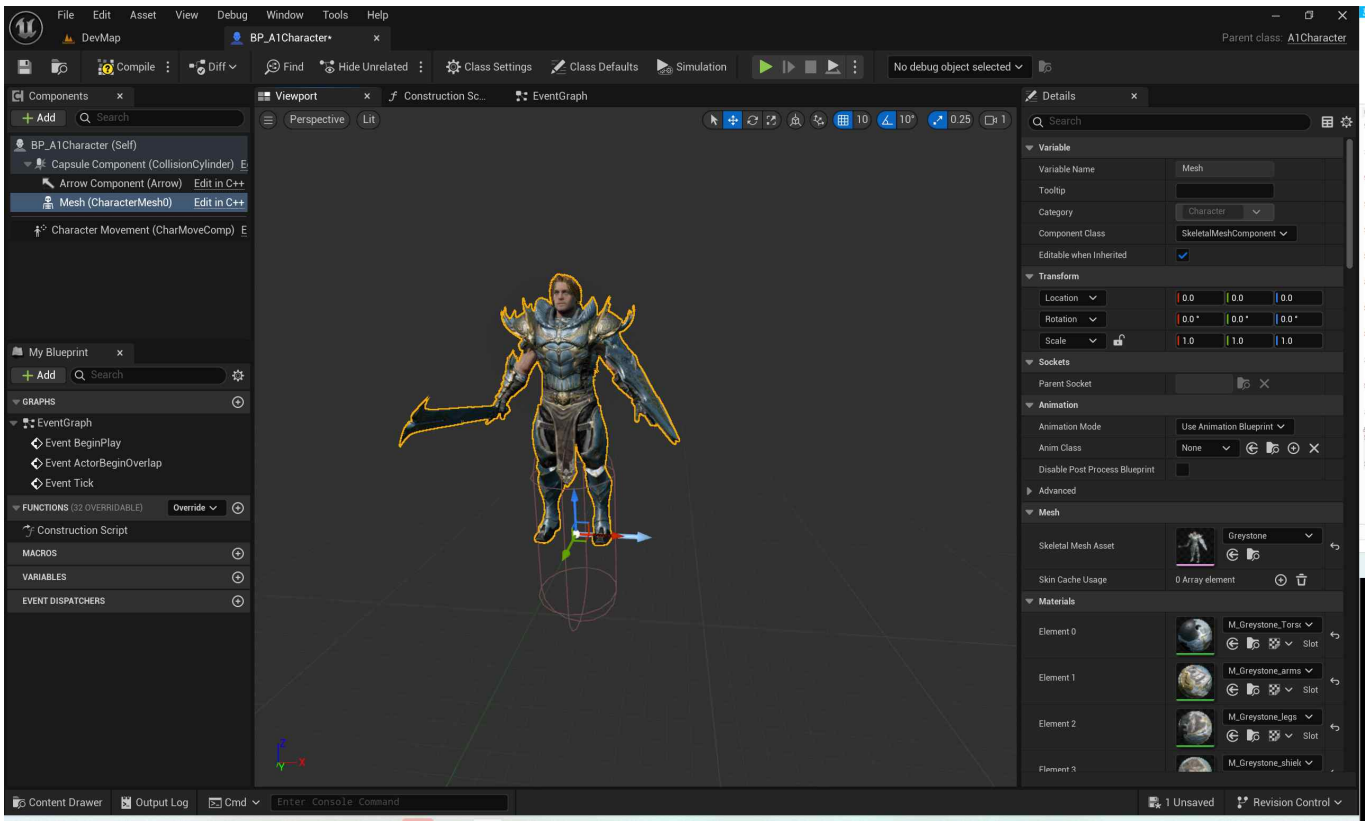
- 생성한 블루프린트의 이름은 **BP_A1Character**로 설정하자.



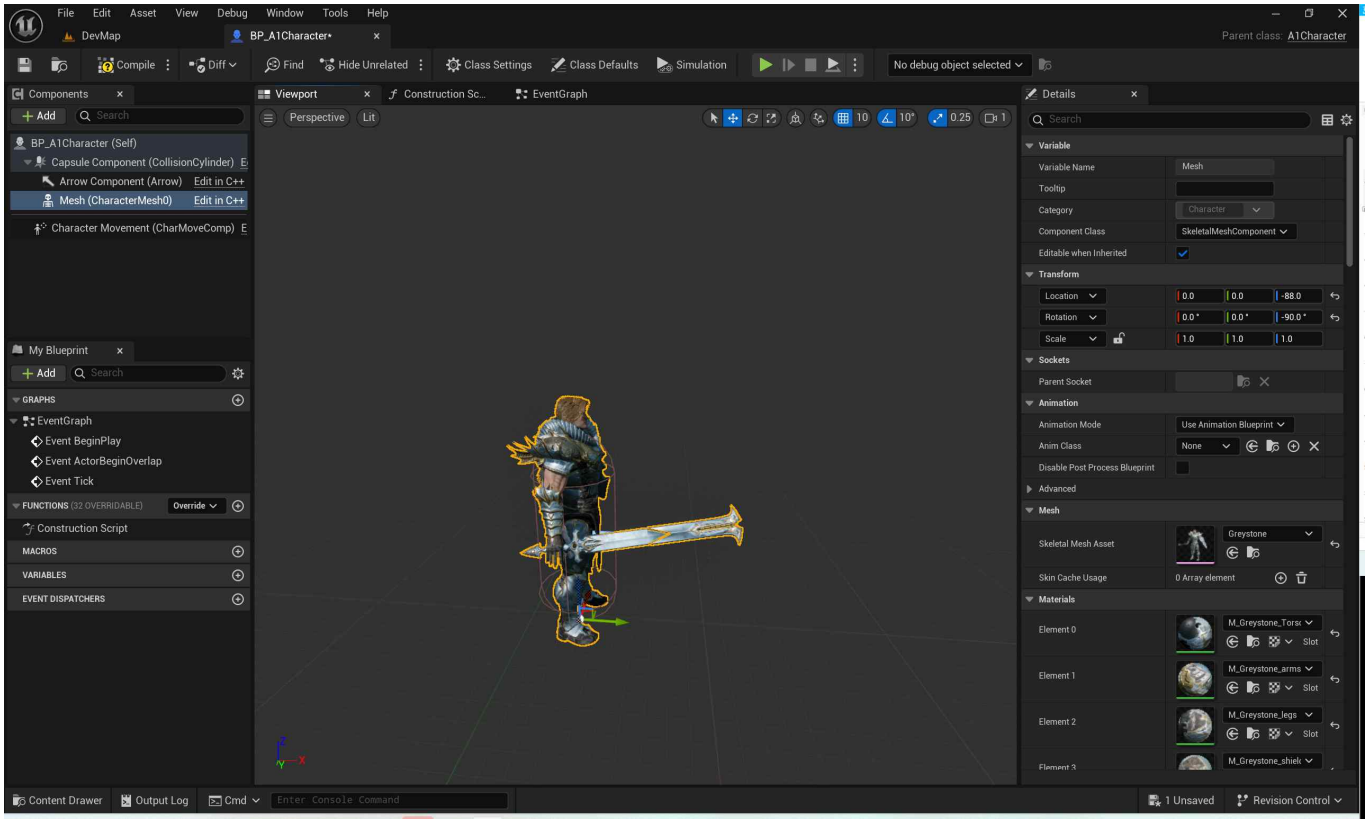
- 생성한 **BP_A1Character** 블루프린트를 열어서 내부 변수를 확인해 보자.
- Capsule Component, Arrow Component, Mesh(Character Mesh), Character Movement 컴포넌트가 기본적으로 추가되어 있다.



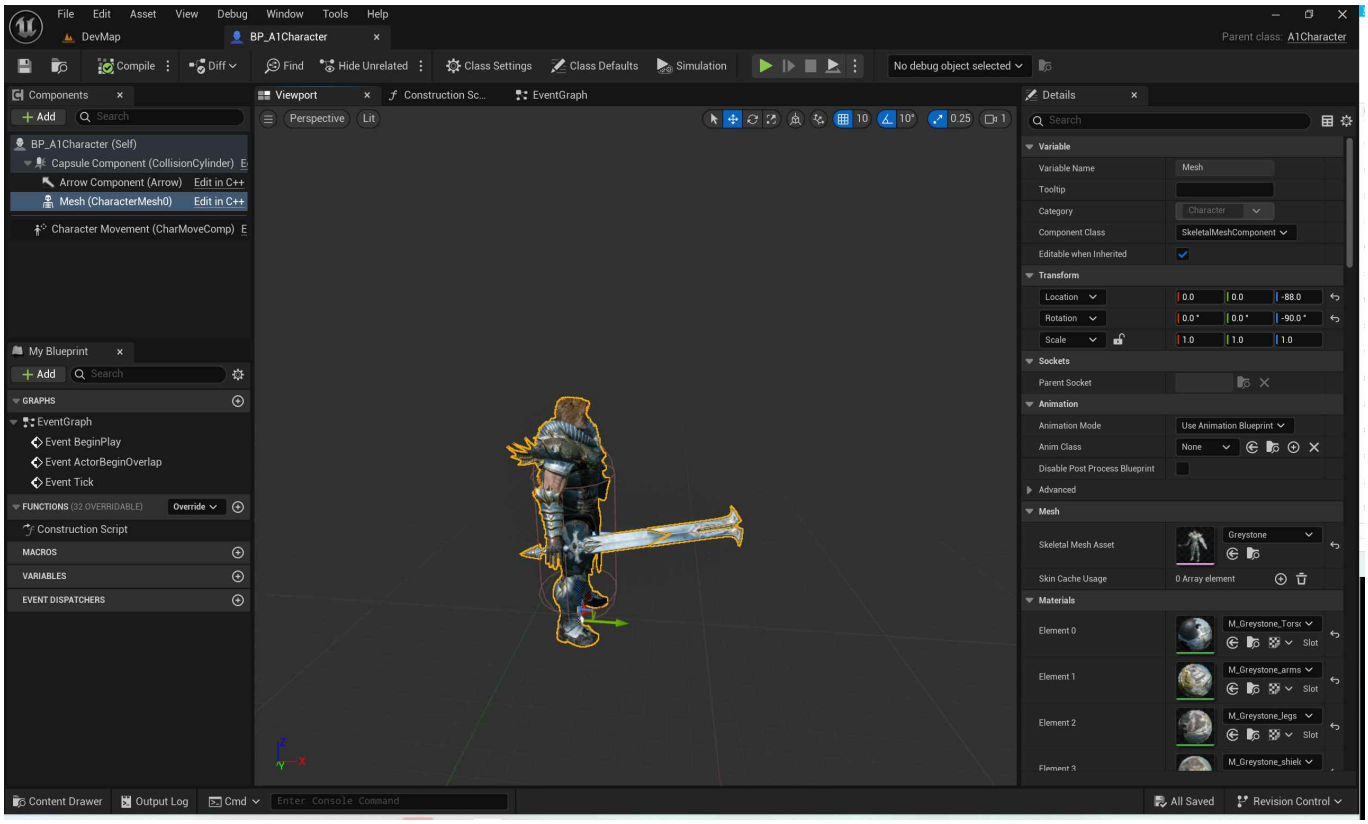
- Mesh부터 설정을 해보자.
- Mesh는 Character가 기본 Skeletal Mesh를 의미한다. **Greystone**로 설정하자.



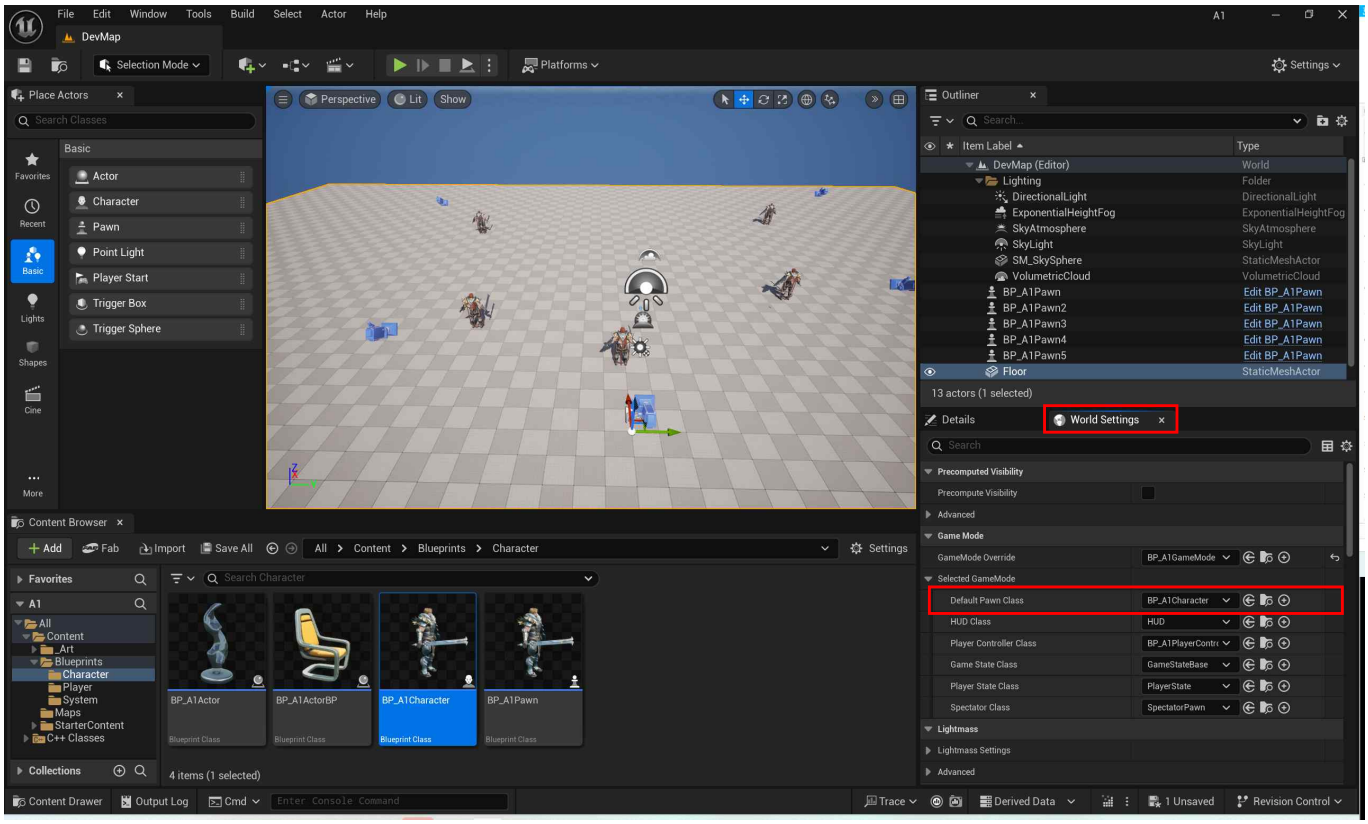
- 추가한 Greystone의 위치인 Location은 0.0, 0.0, -88.0으로 설정하자.
- 추가한 Greystone의 회전인 Rotation은 0.0, 0.0, -90.0으로 설정하자.



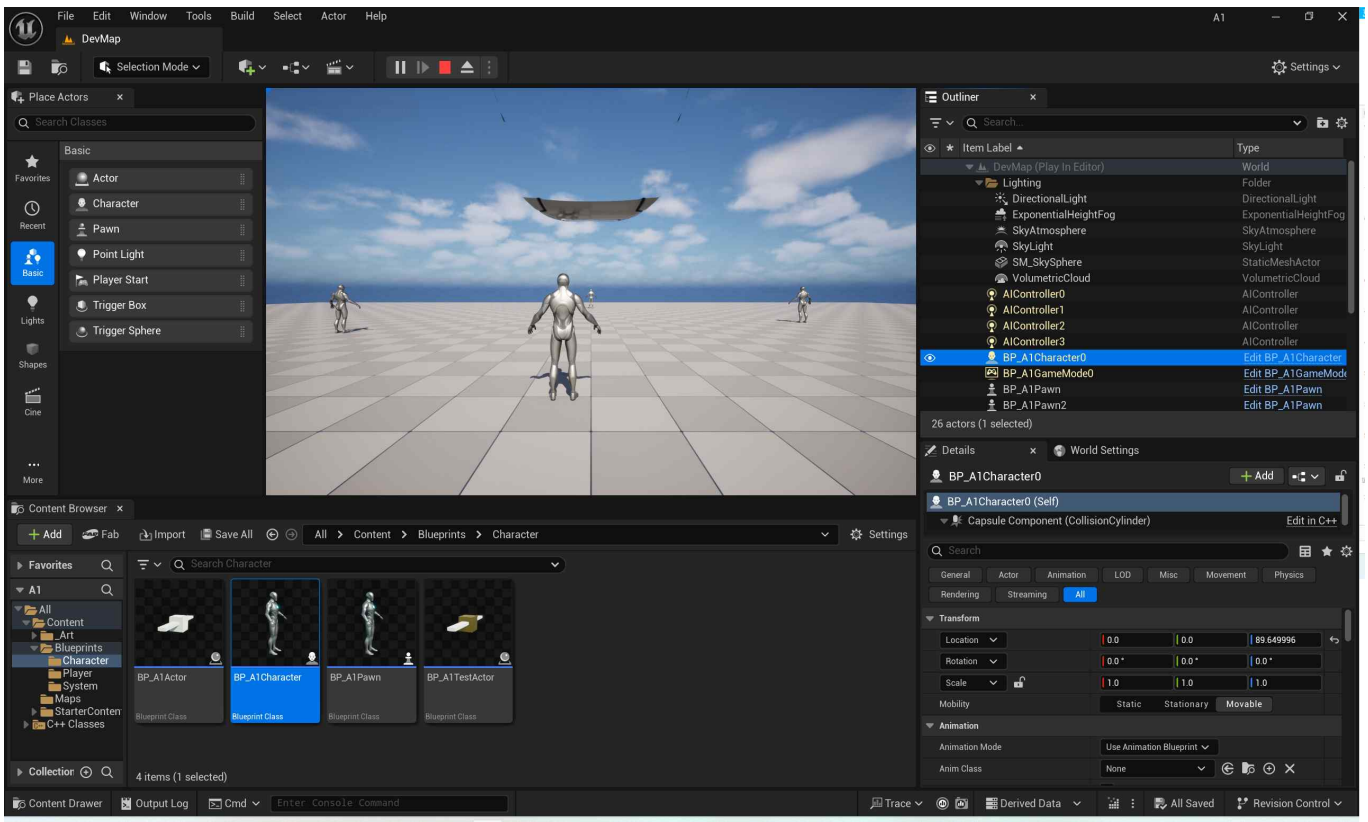
- 컴파일을 하자.



- **World Settings**에서 **GameMode**의 **Default Pawn Class**를 **BP_A1Character**로 설정하자.



- 게임을 실행해 보자.
- BP_A1Character0이 소환된 것을 확인할 수 있다.
- 하지만, 카메라가 없어서 캐릭터의 몸통에서 카메라가 렌더링 되고 있다.



- A1Character 클래스에도 A1Pawn 클래스와 마찬가지로 SpringArm과 Camera가 없다.
- A1Pawn에 작성했던 **SpringArm Component**와 **Camera Component**를 복사해서 붙여넣자.

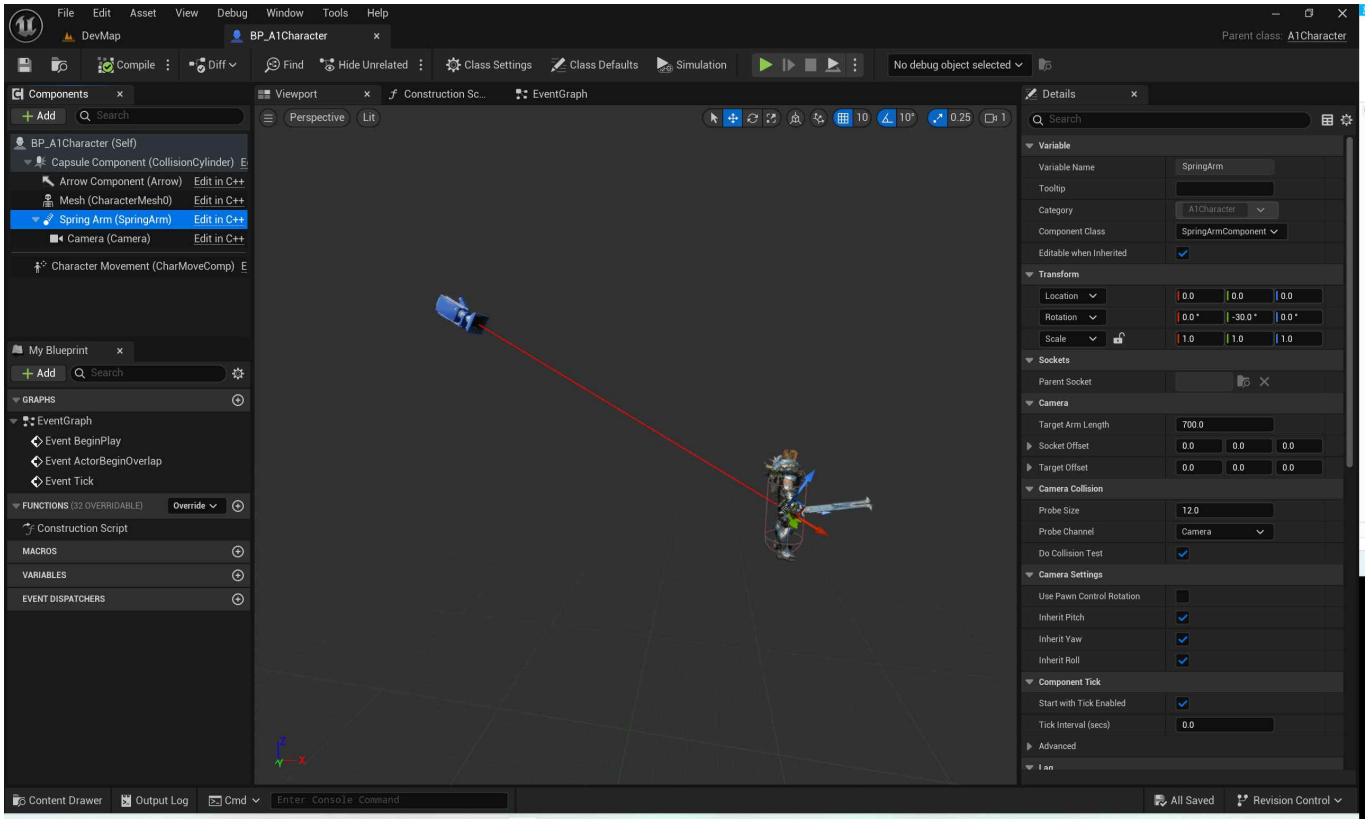
A1Character.h

```
// Fill out your copyright notice in the Description page of Project Settings.

#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Character.h"
#include "A1Character.generated.h"

UCLASS()
class A1_API AA1Character : public ACharacter
{
	GENERATED_BODY()

public:
	// Sets default values for this character's properties
	AA1Character();

protected:
	// Called when the game starts or when spawned
	virtual void BeginPlay() override;

public:
	// Called every frame
	virtual void Tick(float DeltaTime) override;

	// Called to bind functionality to input
	virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;

protected:
	UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
	TObjectPtr<class USpringArmComponent> SpringArm;

	UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
	TObjectPtr<class UCameraComponent> Camera;
};
```

A1Character.cpp

```
// Fill out your copyright notice in the Description page of Project Settings.


#include "Character/A1Character.h"
#include "GameFramework/SpringArmComponent.h"
#include "Camera/CameraComponent.h"

// Sets default values
AA1Character::AA1Character()
{
	// Set this character to call Tick() every frame.  You can turn this off to improve performance if you don't need it.
	PrimaryActorTick.bCanEverTick = true;

	SpringArm = CreateDefaultSubobject<USpringArmComponent>(TEXT("SpringArm"));
	SpringArm->SetupAttachment(GetRootComponent());
	SpringArm->TargetArmLength = 700.0f;
	SpringArm->SetRelativeRotation(FRotator(-30.0f, 0.0f, 0.0f));

	Camera = CreateDefaultSubobject<UCameraComponent>(TEXT("Camera"));
	Camera->SetupAttachment(SpringArm);
}

// Called when the game starts or when spawned
void AA1Character::BeginPlay()
{
	Super::BeginPlay();

}

// Called every frame
void AA1Character::Tick(float DeltaTime)
{
	Super::Tick(DeltaTime);

}

// Called to bind functionality to input
void AA1Character::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
{
	Super::SetupPlayerInputComponent(PlayerInputComponent);

}
```

- 빌드 후, 언리얼 에디터를 실행해서 살펴보자.
- BP_A1Character 블루프린트를 열어서 확인해보자.



- 게임을 실행해 보자.