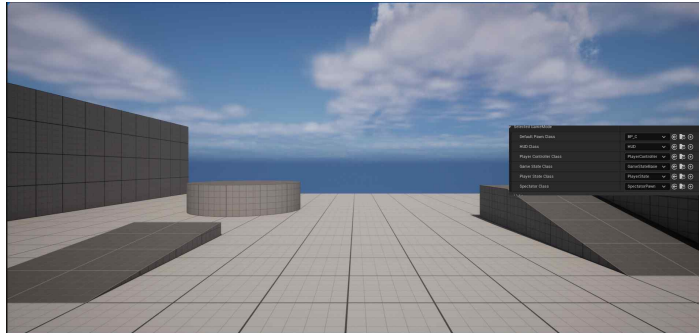


Day 3. 오류 해결

3-1. 라이브 코딩 문제



전날 애니메이션에 관련된 사소한 문제 몇 가지를 고치고 BP로 만들 캐릭터에 대한 C++ class를 본격적으로 설계하였습니다. 상속받을 캐릭터 클래스들을 위해 basecharacter에 기본적인 mesh나 springarm, camera등을 설정하려고 했는데 이상하게 런타임하면 오류가 발생했습니다. C++ 솔루션 빌드도 문제 없고 라이브코딩도 문제 없다고 디버깅 메시지는 나오는데 언리얼에서 시뮬레이션을 돌리면 적용이 되지 않거나 게임모드에서 설정한 defaultpawn으로 설정한 캐릭터가 생성되지 않거나 했습니다. 솔루션 정리와 다시 빌드를 해도 고쳐지지 않았고 계속 문제를 찾아보다가 헤더 파일에서 새로운 class를 선언하면 문제가 발생함을 알게 되었고 혹시나 하여 다른 프로젝트에서 완전히 똑같은 코드로 작성해보니 이번에는 잘되었습니다. 문제를 해결할 방법이 없어 새롭게 프로젝트를 만들어야 하나 고민하고 있었는데 계속해서 문제를 찾아본 결과 언리얼 내 라이브코딩 연동이 생각보다 좋지 않다는 글이 있었습니다. 이 글을 참조하여 라이브코딩은 사용하지 않고 언리얼을 종료한 상태에서 VS에서 솔루션 빌드 이후 언리얼을 실행하니 해당 문제가 고쳐졌습니다. 앞으로는 라이브코딩을 사용하지 않는 방향으로 프로젝트를 진행해야 할 것 같습니다.

3-2. 기본 세팅

```
GetMesh()->SetRelativeLocationAndRotation(FVector(0.f, 0.f, -88.f), FRotator(0.f, -90.f, -0.f));

SpringArm = CreateDefaultSubobject<USpringArmComponent>(TEXT("SpringArm"));
Camera = CreateDefaultSubobject<UCameraComponent>(TEXT("Camera"));

SpringArm->SetupAttachment(GetCapsuleComponent());
Camera->SetupAttachment(SpringArm);

SpringArm->TargetArmLength = 500.f;
SpringArm->SetRelativeRotation(FRotator(-35.f, 0.f, 0.f));
SpringArm->SocketOffset = (FVector(0.f, 120.f, 75.f));
SpringArm->bUsePawnControlRotation = true;
```

문제를 해결하고 캐릭터의 컴포넌트들에 대한 기본적인 세팅을 끝마쳤습니다. 다양한 값을 넣어보면서 캐릭터를 조작하기에 제일 괜찮았던 springarm transform 값들을 넣었고 추후 무기를 사용할 예정이기 때문에 socket을 설정했습니다. 이외에도 캐릭터의 시점을 바꾸고 상하좌우로 이동하는 기본 조작과 입력을 Bind로 넣었는데 현재는 기본적인 틀을 만들고 확장해가는 방향으로 진행하고 있어서 8방향에 관한 움직임이나 시점이 바뀔 때 캐릭터가 돌아가는 정밀한 값들은 추후에 추가할 예정입니다.