

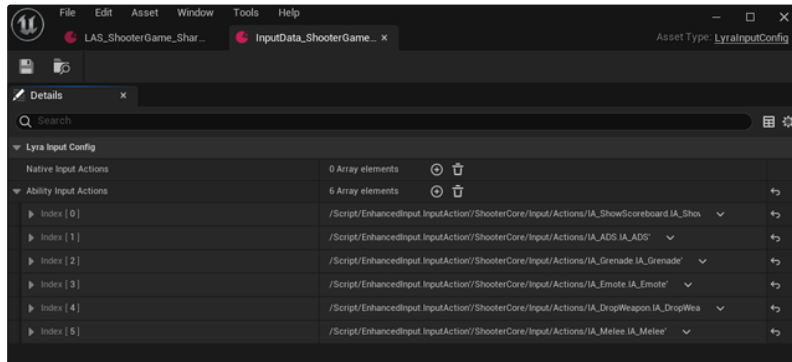
이만택 - 1

헤더에 선언할 때 private로 선언하지 않고 protected로 선언해서 교수님의 습관인줄 알았더니...

```
1 error : BlueprintReadOnly should not be used on private members
```

안되는 것이었음

input을 c++로 연결하지 않고 좀 스마트(?)하게 하는 방법이 없을까...



라이라에서는 이렇게 등록해서..

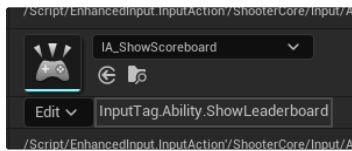
```
1 ULyraHeroComponent* HeroComponent = Pawn->FindComponentByClass<ULyraHeroComponent>();
2 if (HeroComponent && HeroComponent->IsReadyToBindInputs())
3 {
4     for (const TSoftObjectPtr<const ULyraInputConfig>& Entry : InputConfigs)
5     {
6         if (const ULyraInputConfig* BindSet = Entry.Get())
7         {
8             HeroComponent->AddAdditionalInputConfig(BindSet);
9         }
10    }
11 }
```

AddAdditionalInputConfig 안에서 연결..하는데 결국 c++이네?

```
1 UEnhancedInputLocalPlayerSubsystem* Subsystem = LP->GetSubsystem<UEnhancedInputLocalPlayerSubsystem>();
2 check(Subsystem);
3
4 if (const ULyraPawnExtensionComponent* PawnExtComp = ULyraPawnExtensionComponent::FindPawnExtensionComponent(Pawn))
5 {
6     LyraIC->BindAbilityActions(InputConfig, this, &ThisClass::Input_AbilityInputTagPressed, &ThisClass::Input_AbilityInputTagReleased, /*out*/ BindHandles);
7 }
```

```
1 template<class UserClass, typename PressedFuncType, typename ReleasedFuncType>
2 void ULyraInputComponent::BindAbilityActions(const ULyraInputConfig* InputConfig, UserClass* Object, PressedFuncType PressedFunc, ReleasedFuncType ReleasedFunc)
3 {
4     check(InputConfig);
5
6     for (const FLyraInputAction& Action : InputConfig->AbilityInputActions)
7     {
8         if (Action.InputAction && Action.InputTag.IsValid())
9         {
10             if (PressedFunc)
11             {
12                 BindHandles.Add(BindAction(Action.InputAction, ETriggerEvent::Triggered, Object, PressedFunc, Action.InputTag).GetHandle());
13             }
14
15             if (ReleasedFunc)
16             {
17                 BindHandles.Add(BindAction(Action.InputAction, ETriggerEvent::Completed, Object, ReleasedFunc, Action.InputTag).GetHandle());
18             }
19         }
20     }
21 }
```

물론 안에서 인풋을 누르면 스펙을 찾아서 태그에 지정된 함수를 쓰겠지...



캐릭터 만들다가 뭔가 더 만들어보려했으나 빠르게 교수님강의를 뒀음...

애니메이션 많길래 하려했으나 시간이 부족하였....

만들다가 의문인점

```
1 class UNREAL5STUDY_API UUSRenderTarget : public UActorComponent
2
3 UUSMiniViewWidget::UUSMiniViewWidget(const FObjectInitializer& ObjectInitializer) : Super(ObjectInitializer)
4 {
5     //RenderTarget = CreateDefaultSubobject<UUSRenderTarget>(TEXT("PlayerRenderTarget"));
6 }
```

이렇게 생성하였더니...생성하다 죽음 왜일까..

일단 NewObject로 생성하니 넘어가긴했는데..gpt는 이렇다는데 크래시까지..? 아무것도 없는 컴데기인데.. ui라서 outer actor가 없어서? owner였나..

```
1 CreateDefaultSubobject와 NewObject는 Unreal Engine에서 객체를 생성하는 두 가지 주요 방법입니다. 이 두 함수는 유사해 보일 수 있지만, 사용 목적과 적용 시나리오에 있어
2
3 CreateDefaultSubobject
4 사용 시점: CreateDefaultSubobject는 주로 클래스의 생성자에서 사용됩니다. 이는 컴포넌트나 다른 서브 오브젝트들이 해당 객체의 일부로서 초기화되어야 할 때 사용됩니다.
5 목적: 이 함수는 주로 액터의 고정된 컴포넌트를 생성하는 데 사용됩니다. 예를 들어, AActor의 서브클래스에서 메시 컴포넌트나 콜리전 컴포넌트 등을 초기화할 때 사용됩니다.
6 특징: CreateDefaultSubobject로 생성된 객체는 해당 부모 객체의 생명주기에 종속됩니다. 즉, 부모 객체가 파괴될 때 자동으로 정리됩니다. 또한, 이 방법으로 생성된 컴포넌트는 Un
7 제한사항: 이 함수는 오직 생성자에서만 호출될 수 있으며, 동적인 컴포넌트 생성에는 적합하지 않습니다.
8 NewObject
9 사용 시점: NewObject는 게임의 실행 도중 언제든지 호출될 수 있습니다. 이는 런타임에 동적으로 객체를 생성할 필요가 있을 때 사용됩니다.
10 목적: NewObject는 더 일반적인 객체 생성 함수로, 다양한 유형의 객체를 동적으로 생성하는 데 사용됩니다. 이는 게임 플레이 중에 필요에 따라 추가되는 컴포넌트나 다른 유형의 객체
11 특징: NewObject로 생성된 객체는 생성 시 지정된 부모 객체에 종속될 수 있습니다. 부모 객체가 지정되지 않은 경우, 객체의 수명 관리는 개발자에게 달려 있습니다. NewObject는 또
12 제한사항: NewObject로 생성된 객체는 에디터에서 시각적으로 편집할 수 없으며, 블루프린트에서 직접 노출되지 않습니다(단, 블루프린트에서 생성 및 반환하여 사용하는 것은 가능).
13 결론
14 고정 컴포넌트 생성: 액터의 고정 컴포넌트나 초기화 시 필요한 서브 오브젝트는 CreateDefaultSubobject를 사용하여 생성자에서 초기화합니다.
15 동적 객체 생성: 게임 플레이 중에 필요에 따라 객체를 생성하거나, 더 복잡한 초기화 로직이 필요한 경우 NewObject를 사용합니다.
16 이 두 함수의 주요 차이점을 이해하고 적절한 상황에서 사용하는 것이 Unreal Engine 개발에서 중요합니다.
```

아무튼 CreateDefaultSubobject가 안되어서 NewObject로 해결함

NewObject로 생성후에는 RegisterComponent를 하여서 등록해줘야 Scene 컴포넌트가 정상 작동하였음.

```
1 // InParent는 USpringArmComponent
2 void UMiniViewComponent::SetComponent(USceneComponent* InParent)
3 {
4     sceneCapture = NewObject<USceneCaptureComponent2D>(InParent, USceneCaptureComponent2D::StaticClass());
5     sceneCapture->RegisterComponent();
6
7     if (sceneCapture)
8         sceneCapture->AttachToComponent(InParent, FAttachmentTransformRules::KeepRelativeTransform);
9
10    sceneCapture->TextureTarget = renderTarget;
11 }
12
```

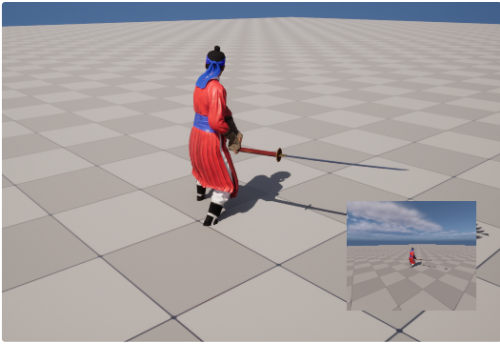
CreateDefaultSubobject가 안되는 이유가 위젯 구조 때문일 것 같긴함..

hud → miniview -> secne이라서 그런가...

hud의 생성을(CreateWidget) AUSPlayer::BeginPlay()에서 해서 그런가 하고 생성자로 이동 시켜봤지만 그래도 크래시남..

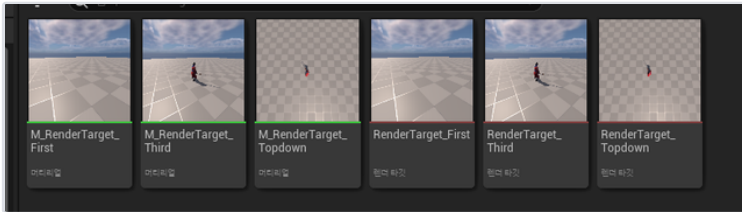
아무튼 만들고보니..... 이득우 교수님의 카메라 교체로는 다른 뷰를 보여줄 수 가 없음

스프링 암에 USceneCaptureComponent2D를 붙여서 시점 이동을 하면 거기밖에 안 보여짐



해결책으로 스프링암을 여러개로 만들어서 각 시점마다 만들어서 USceneCaptureComponent2D생성..

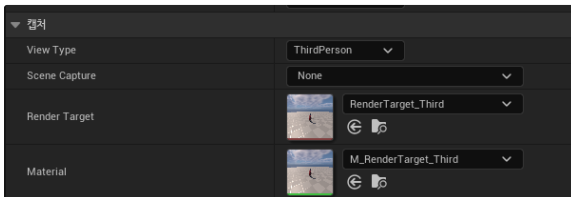
코드로 생성하고 싶었는데... 잘안됨.. 에디터로는 잘 만들어지는데.....



코드로 못 만들어서 어쩔 수 없이...에디터에 다 지정 해줬음

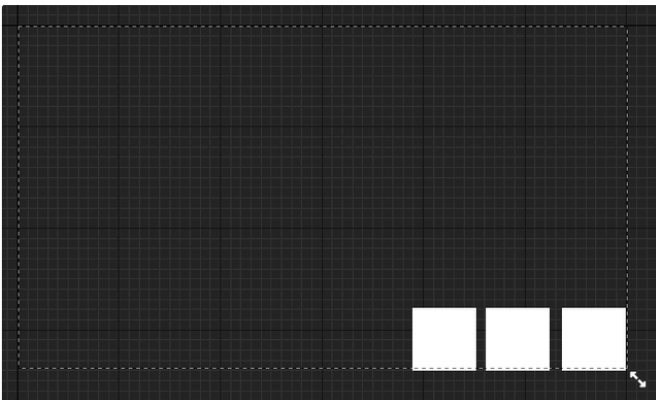
무엇보다 생성을 해도 이게 맞는지 알 수가 없는데 어떻게 확인하지..

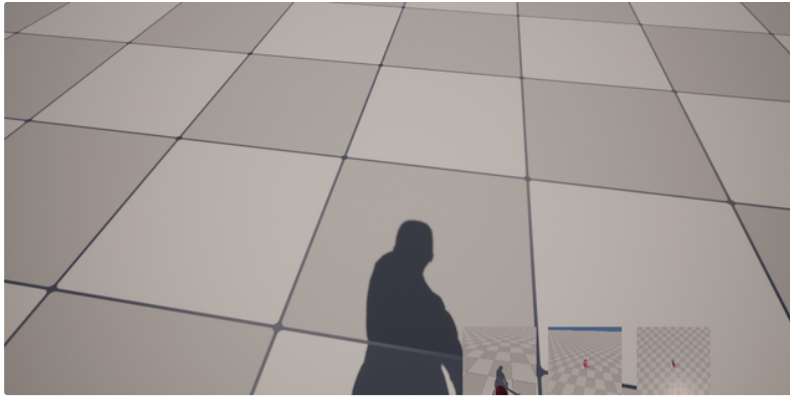
이건 조건문으로 안 만들고 뵈 수 있어서 좋긴함..



그치만 이게 한단계 밖에 안보여지는데 클래스로 묶은 정보는 알 수 없는데 방법이 없나...

그리고 이를 분명 끝으로 만들었는데 비율문제인지 저렇게 보이는데.. UI쪽 한번 더 만들 때 정리해야 할 것 같음





결과물

