

Impossible Bosses

...

한웅희

목차

1. 개발환경 및 게임소개

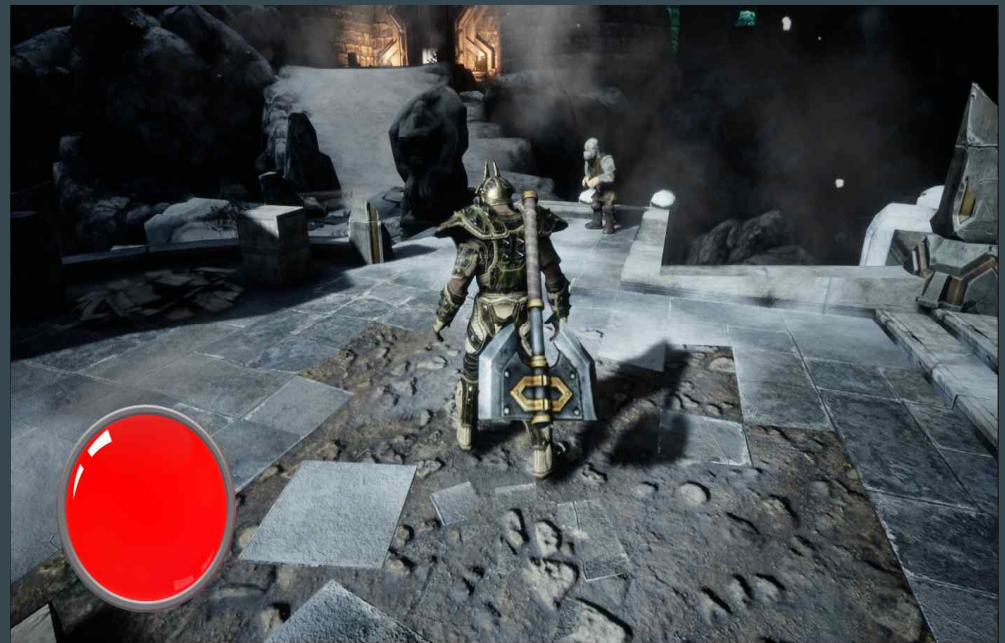
2. 주요 프로그래밍 이슈

1. 블루프린트와 C++의 조화
2. 스켈레톤 소켓을 이용한 장비 장착/해제
3. 게임 인스턴스를 통한 배경음/효과음 제어
4. 비헤이비어 트리를 이용한 보스 AI 구현

3. 시연영상

1. 개발환경 및 게임소개

게임 엔진	Unreal Engine (4.27.2)
IDE	Visual Studio 2019
주요 에셋들	언리얼 마켓플레이스 믹사모
게임명	Impossible Bosses
장르	3인칭 액션 RPG
타겟플랫폼	PC
플레이 방법	이동 : W,A,S,D 구르기 : Space 상호작용 : E 공격 및 시점조작 : Mouse 달리기 및 무기장착해제 : Shift



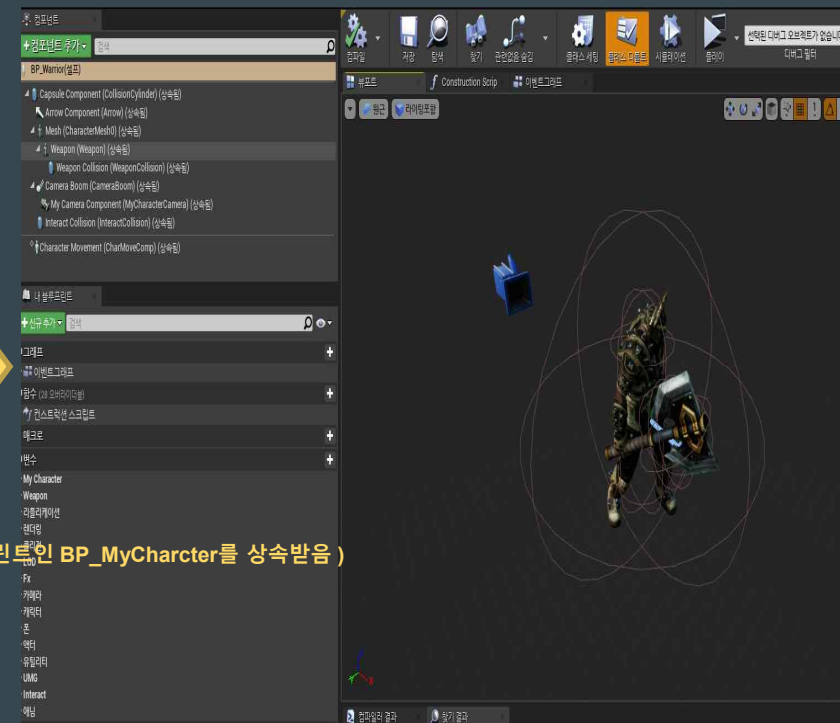
2-1 블루프린트와 C++의 조화

```
13 #include "MyCharacter.generated.h"
14
15 DECLARE_DELEGATE(testtest);
16
17 UCLASS()
18 class UNREAL3DPF_2_API MyCharacter : public ACharacter
19 {
20     GENERATED_BODY()
21
22     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Camera", meta = (AllowPrivateAccess = "true"))
23     class USpringArmComponent* CameraBoom;
24
25     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Camera", meta = (AllowPrivateAccess = "true"))
26     UCameraComponent* MyCameraComponent;
27
28 public:
29     // Sets default values for this character's properties
30     MyCharacter();
31
32 protected:
33     // Called when the game starts or when spawned
34     virtual void BeginPlay() override;
35
36 public:
37     // Called every frame
38     virtual void Tick(float DeltaTime) override;
39
40     // Called to bind functionality to input
41     virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;
42
43     UPROPERTY(VisibleDefaultsOnly, Category = "Mesh")
44     USkeletalMeshComponent* MyMesh;
45
46 protected:
47     UFUNCTION()
48     void MoveForward(float AxisValue);
49
50     UFUNCTION()
51     void MoveRight(float AxisValue);
52
53     UFUNCTION()
54     void Rolling();
55
56     UFUNCTION()
57     void Run();
58
59     UFUNCTION()
60     void RunEnd();
61
62     UFUNCTION()
```

C++클래스를
상속받은 블루
프린트 생성

(정확한 Mycharacter C++ 클래스를 상속받은 블루프린트인 BP_MyCharcter를 상속받음)

[MyCharacter.h (C++ 클래스)]



[BP_Warrior (블루프린트 클래스)]

2-1 블루프린트와 C++의 조화

```
void AMyCharacter::MyCharacterDeath()
{
    // 죽는 모션 재생
    AnimInstance->Montage_Play(deathMontage);

    // 합대를 통한 Delay 구현
    FTimerHandle testTimerHandle;
    float Time = 2.0f; // Time초 뒤
    GetWorld()->GetTimerManager().SetTimer(testTimerHandle, [&]()
    {
        // 코드 구현
        UMyGameInstance* MyGameInstance = Cast<UMyGameInstance>(GetGameInstance());

        if (MyGameInstance)
        {
            UE_LOG(LogTemp, Warning, TEXT("MyGameInstance != nullptr"));

            if (MyGameInstance->WidgetGameover != nullptr)
            {
                UE_LOG(LogTemp, Warning, TEXT("MyGameInstance->WidgetGameover != nullptr"));

                // 위젯을 만들고 화면에 보여줄
                CreateWidget(GetWorld(), MyGameInstance->WidgetGameover)->AddToViewport();

                // 효과음 재생
                MyGameInstance->PlayBGM(MyGameInstance->GameOvercue);

                // UI 입력모드
                UGameplayStatics::GetPlayerController(GetWorld(), 0)->SetInputMode(FInputModeUIOnly());

                // 마우스 커서 보이게
                UGameplayStatics::GetPlayerController(GetWorld(), 0)->bShowMouseCursor = true;

                // 게임 일시정지
                UGameplayStatics::GetPlayerController(GetWorld(), 0)->SetPause(true);
            }
        }

        // TimerHandle 초기화
        GetWorld()->GetTimerManager().ClearTimer(testTimerHandle);
    }, Time, false); // 반복하려면 false를 true로 변경
}
```

[MyCharacter.cpp 의 MyCharacterDeath 함수 부분]

MyCharacter를 상속받은 마법사, 전사는 둘 다
MyCharacterDeath 함수를 통해 죽는 모션 재생가능

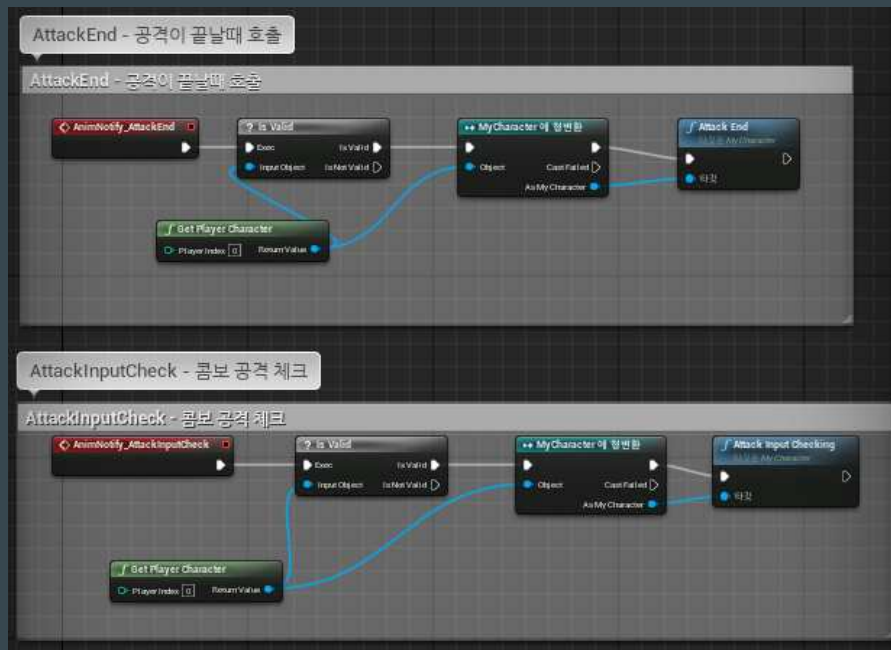


[BP_Magician]



[BP_Warrior]

2-1 블루프린트와 C++의 조화



[애니메이션 노티파이 부분과 같은 GUI단에서의 수정이 편한 건 블루프린트]

C++	<ul style="list-style-type: none"> • 블루프린트 로직보다 훨씬 빠르다 (연산성능 좋음) • 블루프린트 자체도 C++기반이고 C++에선 블루프린트보다 많은 엔진 함수기능 사용가능
블루프린트	<ul style="list-style-type: none"> • 빠르고 쉽게 생성/삭제가 가능 (컴파일 속도 빠름) • 프로그래머가 아닌 디자이너와 아티스트도 직관적으로 편집 가능

```

void AMyCharacter::Attack()
{
    if (bArmsWeapon)
    {
        UE_LOG(LogTemp, Warning, TEXT("Attack()"));
        bMBDown = true;

        // (안전상) 애니메이션스와 어택몰타우가 둘중 하나라도 nullptr이면 리턴
        if (!AnimInstance || !AttackMontage)
        {
            UE_LOG(LogTemp, Warning, TEXT("nullptr"));
            return;
        }

        // 공격중
        bAttacking = true;

        const char* comboList[] = { "Combo01", "Combo02", "Combo03" };
        UE_LOG(LogTemp, Warning, TEXT("ComboCount : %d"), ComboCount);

        // 왼쪽 마우스 버튼을 누른경우
        // 애니메이션을 실행중이지 않았을때
        if (!AnimInstance->Montage_IsPlaying(attackMontage))
        {
            AnimInstance->Montage_Play(attackMontage);
        }

        // 애니메이션이 실행중일 때
        else if (AnimInstance->Montage_IsPlaying(attackMontage))
        {
            // 현재 실행중인 애니메이션이 끝나고 다음 애니메이션 진행
            AnimInstance->Montage_SetNextSection(FName(comboList[ComboCount - 1]), FName(comboList[ComboCount]), attackMontage);
        }
    }
}

void AMyCharacter::AttackInputChecking()
{
    UE_LOG(LogTemp, Warning, TEXT("AttackInputChecking()"));

    if (ComboCount >= 2)
    {
        ComboCount = 0;
    }

    if (bComboAttack == true)
    {
        ComboCount += 1;
    }
}
    
```

[함수의 기능과 관련된 성능과 관련된 부분은 C++]

2-2 스켈레톤 소켓을 이용한 장비 장착/해제



[BP_Warrior 스켈레톤의 UnArmSocket 소켓]

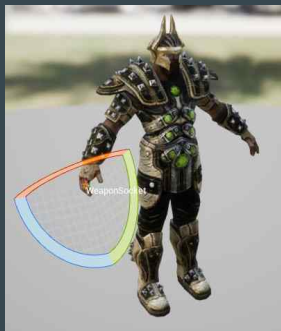


[BP_Warrior에 장착될 Weapon 메쉬]

무기 장착해제



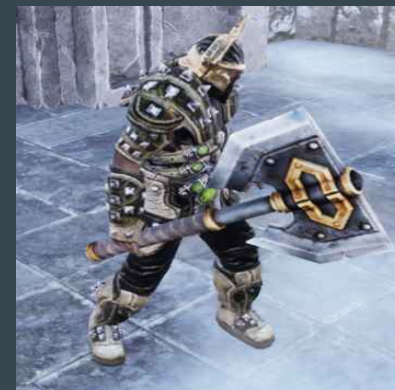
[UnArm 상태]



[BP_Warrior 스켈레톤의 WeaponSocket 소켓]

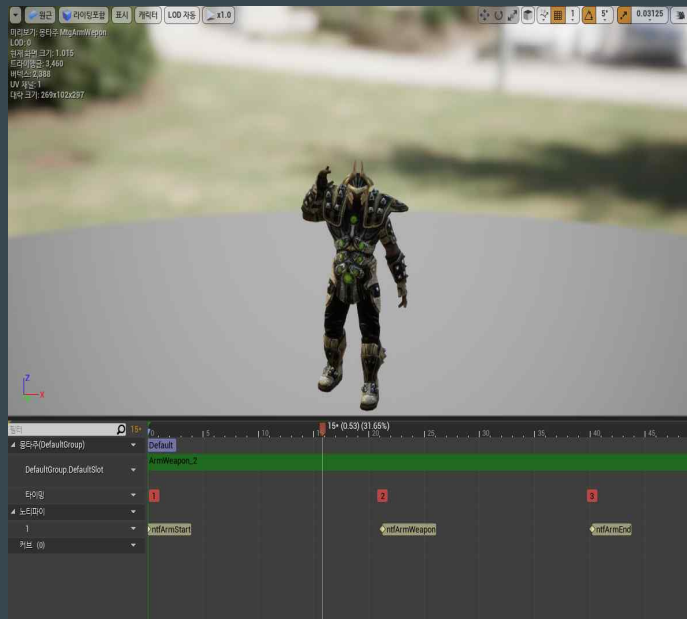


무기 장착

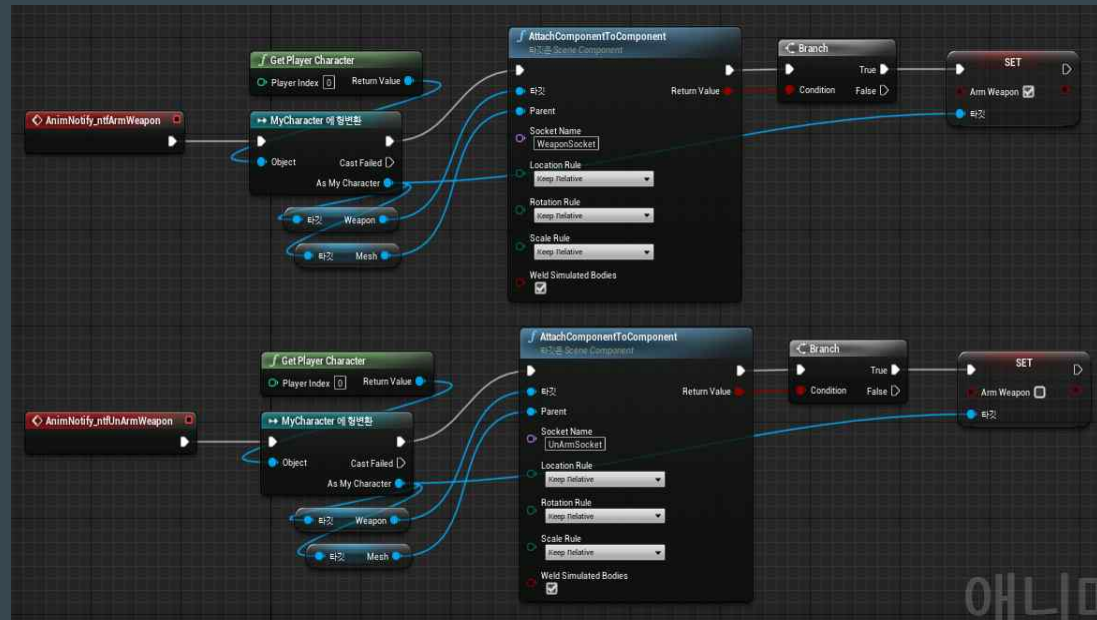


[Arm 상태]

2-2 스켈레톤 소켓을 이용한 장비 장착/해제

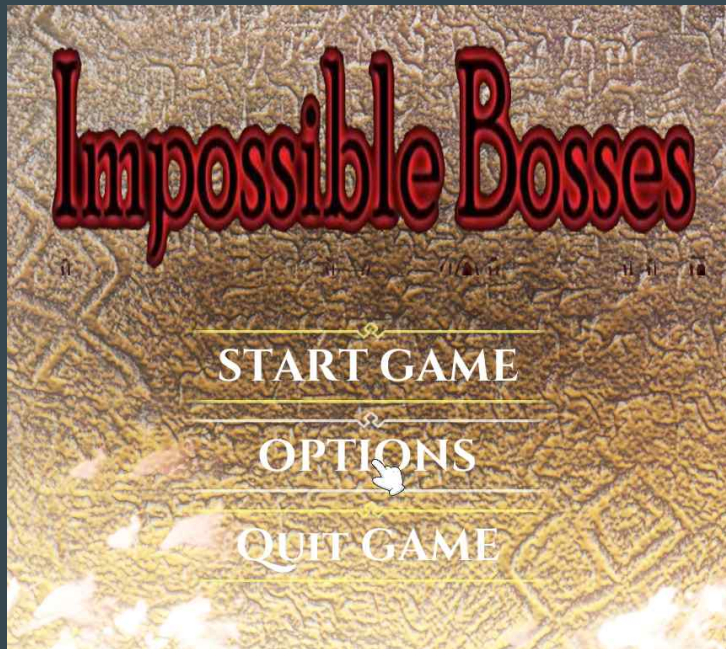


[MtgArmWeapon 애니메이션 몽타주]

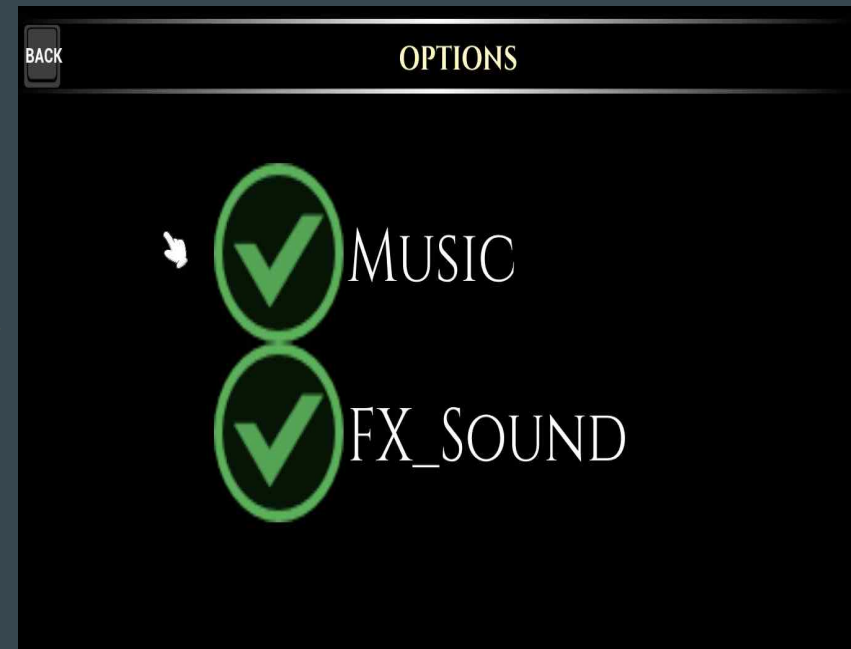


[AnimNotify를 이용해 특정 시점에 Weapon 메시를 특정 소켓에 붙인다]

2-3 게임 인스턴스를 통한 배경음/효과음 제어

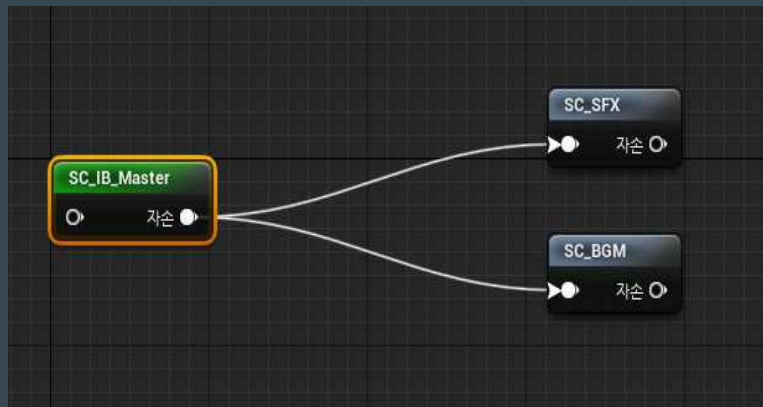


[BP_WidgetMainMenu 화면]



[BP_WidgetOptionsMenu 화면]

2-3 게임 인스턴스를 통한 배경음/효과음 제어



[사운드클래스는 크게 SC_SFX(효과음), SC_BGM(배경음)으로 나뉘짐]

BGM_KingKnight (사운드 큐)
경로: /Game/Assets/Sounds/MySound
쿠크 파일경로 길이: 152 / 260
Duration: 2.423.362061
Max Distance: 2.097,152.000000
Class: /Game/Assets/Sounds/MySound/SC_BGM.SC_BGM
Volume Multiplier: 0.200000
Pitch Multiplier: 1.000000
Total Samples: 0.000000

[게임의 배경음악]

Boss_Attack1 (사운드 큐)
경로: /Game/Assets/Sounds/MySound
쿠크 파일경로 길이: 150 / 260
Duration: 2.700000
Max Distance: 2.097,152.000000
Class: /Game/Assets/Sounds/MySound/SC_SFX.SC_SFX
Volume Multiplier: 0.450000
Pitch Multiplier: 1.000000
Total Samples: 0.000000

[보스 공격 효과음]

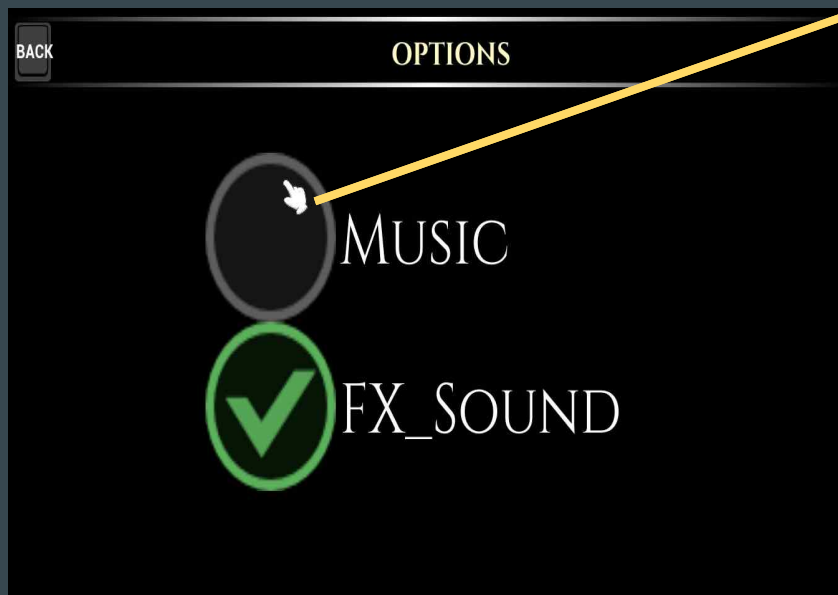
```

7  #include "Sound/SoundClass.h"
8
9  UAudioComponent* UMyGameInstance::PlayBGM(USoundBase* soundcue)
10 {
11     // 사운드 큐 생성
12     return UGameplayStatics::SpawnSound2D(this, soundcue, 1.0f, 1.0f, 0.0f, nullptr, true, false);
13 }
14
15 void UMyGameInstance::SetOption(FStruct_option curOption)
16 {
17     UE_LOG(LogTemp, Warning, TEXT("UMyGameInstance::SetOption"));
18
19     // 배경음악이 체크되어있으면
20     if (curOption.bMusic)
21     {
22         USoundClass* tSC = LoadObject<USoundClass>(nullptr, TEXT("/Game/Assets/Sounds/MySound/SC_BGM.SC_BGM"), NULL, LOAD_None, NULL);
23
24         if (tSC != nullptr)
25         {
26             // 배경음악 볼륨을 켜
27             UE_LOG(LogTemp, Warning, TEXT("tSC->Properties.Volume = 1.0f"));
28             tSC->Properties.Volume = 1.0f;
29             bBGM = true;
30         }
31     }
32
33     // 배경음악 체크 해제하면
34     else if (!curOption.bMusic)
35     {
36         USoundClass* tSC = LoadObject<USoundClass>(nullptr, TEXT("/Game/Assets/Sounds/MySound/SC_BGM.SC_BGM"), NULL, LOAD_None, NULL);
37
38         // 배경음악 볼륨을 끄
39         // 볼륨을 0으로 하면 완전하게 잠
40         if (tSC != nullptr)
41         {
42             UE_LOG(LogTemp, Warning, TEXT("tSC->Properties.Volume = 0.001f"));
43             tSC->Properties.Volume = 0.001f;
44             bBGM = false;
45         }
46     }
47
48     // 효과음이 체크되어있으면
49     if (curOption.bSound)
50     {
51         USoundClass* tSC = LoadObject<USoundClass>(nullptr, TEXT("/Game/Assets/Sounds/MySound/SC_SFX.SC_SFX"), NULL, LOAD_None, NULL);
52
53         // 효과음 켜
54         if (tSC != nullptr)
55         {
56             tSC->Properties.Volume = 1.0f;
57         }
58     }
59
60     // 효과음 체크 해제하면
61     else if (!curOption.bSound)
62     {
63         USoundClass* tSC = LoadObject<USoundClass>(nullptr, TEXT("/Game/Assets/Sounds/MySound/SC_SFX.SC_SFX"), NULL, LOAD_None, NULL);
64
65         // 효과음 끄
66         if (tSC != nullptr)
67         {
68             tSC->Properties.Volume = 0.001f;
69         }
70     }
71 }

```

[MyGameInstance.cpp]

2-3 게임 인스턴스를 통한 배경음/효과음 제어

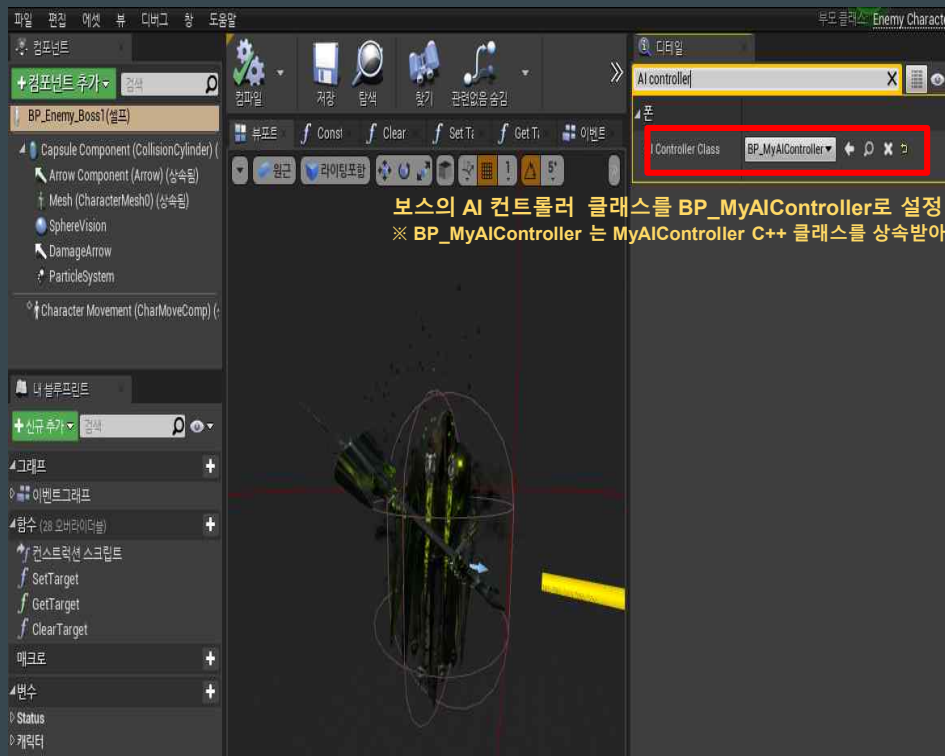


1. MUSIC이나 FX_SOUND 체크박스 클릭시
2. 텔레게이트된 함수 SetupOptionToInstance 호출
3. 게임인스턴스의 SetOption 함수로 특정 사운드클래스 볼륨조정

```
28 // 체크박스(BGM, SOUND) 누를시
29 CBMusic->onCheckStateChanged.AddDynamic(this, &UWidgetOptionsMenu::SetupOptionToInstance);
30 CBSound->onCheckStateChanged.AddDynamic(this, &UWidgetOptionsMenu::SetupOptionToInstance);
31
32
33 if (MyGameInstance->bBGM)
34 {
35     CBMusic->SetCheckedState(ECheckBoxState::Checked);
36 }
37
38 else if (!MyGameInstance->bBGM)
39 {
40     CBMusic->SetCheckedState(ECheckBoxState::Unchecked);
41 }
42
43 if (MyGameInstance->bSound)
44 {
45     CBSound->SetCheckedState(ECheckBoxState::Checked);
46 }
47
48 else if (!MyGameInstance->bSound)
49 {
50     CBSound->SetCheckedState(ECheckBoxState::Unchecked);
51 }
52
53 UE_LOG(LogTemp, Warning, TEXT("MyGameInstance->bBGM : %d "), MyGameInstance->bBGM);
54
55
56 void UWidgetOptionsMenu::SetupOptionToInstance(bool bIsChecked)
57 {
58     // 현재 체크박스 상태 저장
59     FStruct_Option curOption;
60     curOption.bMusic = CBMusic->IsChecked();
61     curOption.bSound = CBSound->IsChecked();
62
63     // 현재 체크박스 상태 게임인스턴스로
64     UMyGameInstance* MyGameInstance = Cast<UMyGameInstance>(GetGameInstance());
65     MyGameInstance->SetOption(curOption);
66 }
67
68 void UWidgetOptionsMenu::OnClickBtnBack()
69 {
70     // 사용자 정의 게임인스턴스를 가져옴
71     UMyGameInstance* MyGameInstance = Cast<UMyGameInstance>(GetGameInstance());
72
73     // 옵션메뉴 위젯애니메이션 역재생
74     PlayAnimation(FadeAnimation, 0, 1, EUMSSequencePlayMode::Reverse, 1, false);
75 }
```

[WidgetOptionsMenu.cpp]

2-4 비헤이비어 트리를 이용한 보스 AI 구현



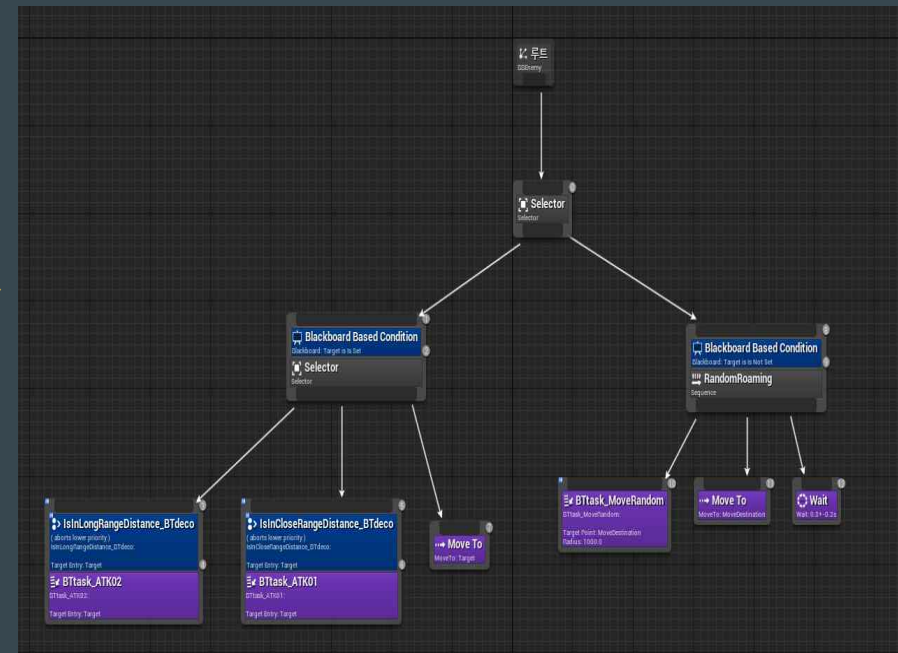
```
28 void AMyAIController::OnPossess(APawn* InPawn)
29 {
30     Super::OnPossess(InPawn);
31
32     // AI 실행
33     RunAI();
34 }
35
36 void AMyAIController::RunAI()
37 {
38     if (UseBlackboard(BBAsset, Blackboard))
39     {
40         // 비헤이비어 트리 시작
41         RunBehaviorTree(BTAsset);
42     }
43 }
44
45 void AMyAIController::StopAI()
46 {
47     UE_LOG(LogTemp, Log, TEXT("StopAI()"));
48
49     BehaviorTreeComponent = Cast<UBehaviorTreeComponent>(GetBrainComponent());
50
51     if (nullptr == BehaviorTreeComponent)
52     {
53         UE_LOG(LogTemp, Log, TEXT("nullptr == BehaviorTreeComponent"));
54         return;
55     }
56
57     // 비헤이비어 트리 멈춤
58     UE_LOG(LogTemp, Log, TEXT("nullptr != BehaviorTreeComponent"));
59     BehaviorTreeComponent->StopTree(EBTStopMode::Safe);
60 }
```

[MyAIController.cpp]

2-4 비헤이비어 트리를 이용한 보스 AI 구현

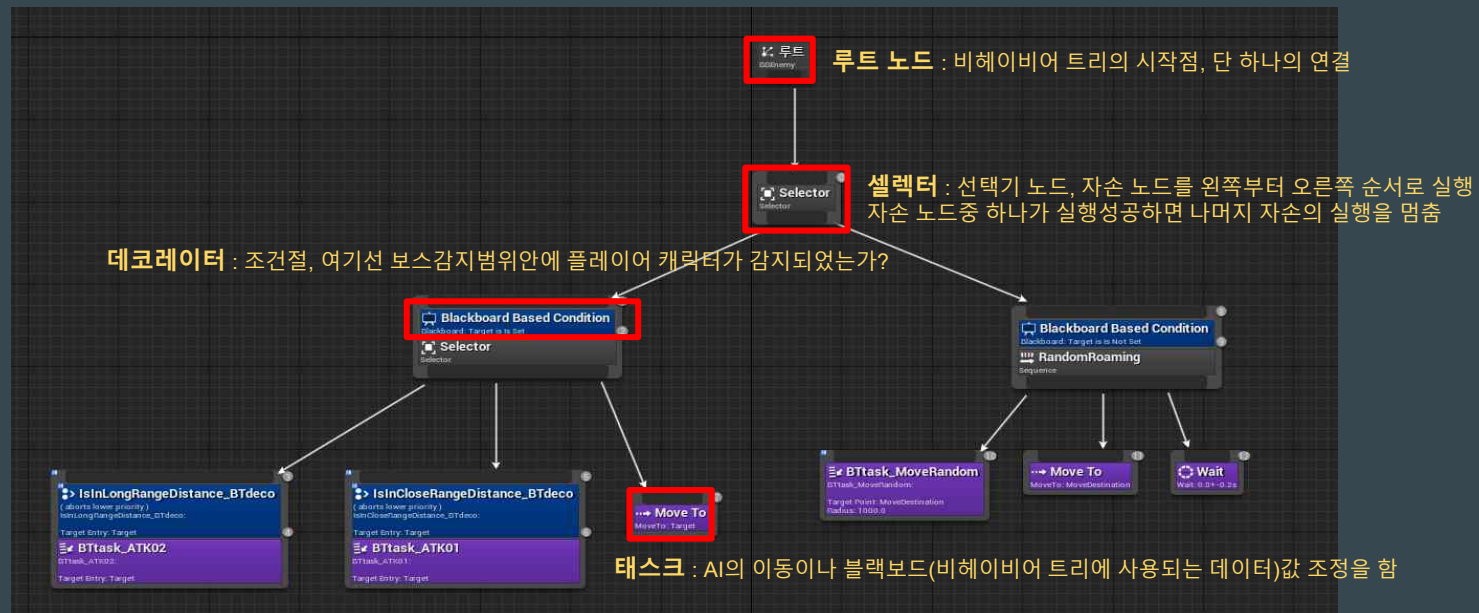
```
28 void AMyAIController::OnPossess(APawn* InPawn)
29 {
30     Super::OnPossess(InPawn);
31
32     // AI 실행 맵에 보스 스폰시 실행되는 OnPossess 함수를 통해
33     RunAI();
34     RunAI 함수를 실행해 비헤이비어 트리를 통한 보스 AI 기능 시작
35 }
36
37 void AMyAIController::RunAI()
38 {
39     if (UseBlackboard(BBAsset, Blackboard))
40     {
41         // 비헤이비어 트리 시작
42         RunBehaviorTree(BTAsset);
43     }
44 }
45
46 void AMyAIController::StopAI()
47 {
48     UE_LOG(LogTemp, Log, TEXT("StopAI()"));
49     BehaviorTreeComponent = Cast<UBehaviorTreeComponent>(GetBrainComponent());
50
51     if (nullptr == BehaviorTreeComponent)
52     {
53         UE_LOG(LogTemp, Log, TEXT("nullptr == BehaviorTreeComponent"));
54         return;
55     }
56
57     // 비헤이비어 트리 멈춤
58     UE_LOG(LogTemp, Log, TEXT("nullptr != BehaviorTreeComponent"));
59     BehaviorTreeComponent->StopTree(EBTStopMode::Safe);
60 }
```

[MyAIController.cpp]



[BTEny(비헤이비어 트리)]

2-4 비헤이비어 트리를 이용한 보스 AI 구현



1. 보스 감지범위안에 플레이어가 감지되면
 - 1-1. (플레이어가 먼 거리에 있다면) 보스의 점프공격
 - 1-2. (플레이어가 가까운 거리에 있다면) 보스의 기본공격

2. 보스 감지범위안에 플레이어가 없다면
 - 2-1. 감지범위 내 랜덤 이동하며 순찰

3. 시연 영상

<https://youtu.be/naik87LgldY>

유튜브링크

감사합니다