Unreal second :

assignment 5

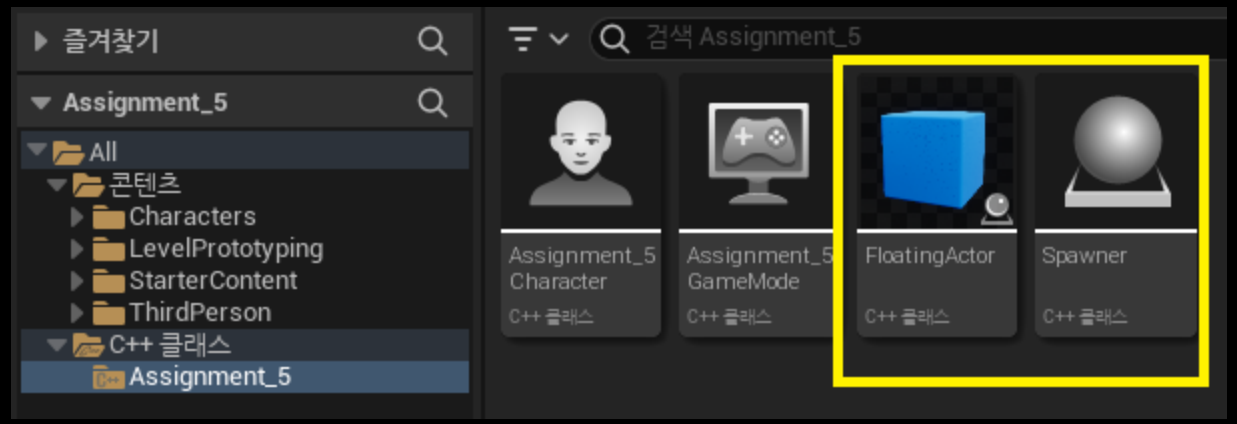
17조 김영빈

목차

개요

- Unreal 에서 c++ class 를 이용해 액터를 생성하고 프로그래밍 하는 과제이다.
- C++ class 파일을 컴파일하고 의도에 맞게 동작시켜보자.
- Begin start(), Tick() 기능을 이해한다.

액터 컴파일



- FloatingActor :

  Actor class 를 상속했다. 실질적으로 움직이고, 메시지를 출력하고, 이벤트가 발생하는 액터이다.

- Spawner :

  게임 시작시 FloatingActor 를 스폰하는 액터이다.

액터 스폰

```cpp
void SpawnFloatingActor()
{
    FVector SpawnLocation = FVector(0.0f, 0.0f, 0.0f);
    FRotator SpawnRotation = FRotator(0.0f, 0.0f, 0.0f);
    FActorSpawnParameters SpawnParams;
    SpawnParams.SpawnCollisionHandlingOverride = ESpawnActorCollisionHandlingMethod::AlwaysSpawn;

    AFloatingActor* ActorToSpawn = GetWorld()->SpawnActor<AFloatingActor>(AFloatingActor::StaticClass(), SpawnLocation, SpawnRotation, SpawnParams);

    if (ActorToSpawn)
    {
        UE_LOG(LogTemp, Log, TEXT("Spawned at %s"), *SpawnLocation.ToString());
    }
    else
    {
        UE_LOG(LogTemp, Error, TEXT("Fail to spawn"));
    }
}
```

- Spawner class 에서 FloatingActor 를 스폰하는 메소드.
- 스폰 위치, 회전, 속성을 정의하고 레벨에 스폰되며 Actor 의 정보를 포인터로 가리키게 된다.
- 스폰 성공시 초기 좌표를 출력한다.

액터 속성부여

```cpp
// Sets default values
AFloatingActor::AFloatingActor()
{
    // Set this actor to call Tick() every frame.  You can turn this off to improve performance if you don't need it.
    PrimaryActorTick.bCanEverTick = true;
    idx = 0;
    //input mesh in constructor
    MeshComponent = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("MeshComponent"));
    RootComponent = MeshComponent;
    static ConstructorHelpers::FObjectFinder<UStaticMesh> MeshAsset(TEXT("StaticMesh'/Game/LevelPrototyping/Meshes/SM_ChamferCube.SM_ChamferCube'"));

    if (MeshAsset.Succeeded())
    {
        MeshComponent->SetStaticMesh(MeshAsset.Object);
        UE_LOG(LogTemp, Warning, TEXT("succeed to mesh"));
    }
    else
    {
        UE_LOG(LogTemp, Error, TEXT("Fail to mesh"));
    }
}
```

- FloatingActor class 의 기본 생성자에서 멤버변수를 초기화하고 메쉬를 부여한다.
- 메쉬의 경로를 찾아 부여한다.
- 경로상에 존재여부 메시지를 출력한다.

액터 이동

```cpp
virtual void Move()
{

    FVector coordinate = GetActorLocation();

    while (idx < 10)
    {
        coordinate.X += Step();
        coordinate.Y += Step();
        SetActorLocation(coordinate);
        coordinates.Add(coordinate);
        UE_LOG(LogTemp, Warning, TEXT("Actor Location : x:%f, y:%f"), coordinates[idx].X, coordinates[idx].Y);
        if (idx <= 0)
        {
            UE_LOG(LogTemp, Warning, TEXT("Move distance : %f"), Distance(coordinates[idx], coordinates[idx]));
        }
        else
        {
            UE_LOG(LogTemp, Warning, TEXT("Move distance : %f"), Distance(coordinates[idx-1], coordinates[idx]));
        }
        if (Step())
        {
            OnEvent.ExecuteIfBound();
        }
        idx++;
    }
}
```

- FloatingActor class 의 Move 메소드
- GetActorLocation() 으로 현재 액터 월드 좌표를 받는다.
- Step() 메소드는 0 또는 1 을 반환한다.
- 반환받은 액터 월드 좌표에 Step() 에서 랜덤으로 반환된 값을 더한다.
- 새로운 좌표를 멤버변수 vector 에 저장한다.
- Distance() 메소드를 사용해이전 액터 월드 좌표와 현재 좌표간 거리를 계산한다.
- Step() 의 값에 따라 이벤트의 발생 유무를 따진다.

액터 이동

```
virtual float Distance(FVector before, FVector after)
{
    float dx = before.X - after.X;
    float dy = before.Y - after.Y;
    return FMath::Sqrt(dx * dx + dy * dy);
}
```

- 두 월드 좌표간 거리를 계산한다.
- 두개의 FVector 를 매개변수로 받고 피타고라스 정리에 의거해 계산하고 반환한다.

메시지 출력

```cpp
virtual void Move()
{

    FVector coordinate = GetActorLocation();

    while (idx < 10)
    {
        coordinate.X += Step();
        coordinate.Y += Step();
        SetActorLocation(coordinate);
        coordinates.Add(coordinate);
        UE_LOG(LogTemp, Warning, TEXT("Actor Location : x:%f, y:%f"), coordinates[idx].X, coordinates[idx].Y);
        if (idx <= 0)
        {
            UE_LOG(LogTemp, Warning, TEXT("Move distance : %f"), Distance(coordinates[idx], coordinates[idx]));
        }
        else
        {
            UE_LOG(LogTemp, Warning, TEXT("Move distance : %f"), Distance(coordinates[idx-1], coordinates[idx]));
        }
        if (Step())
        {
            OnEvent.ExecuteIfBound();
        }
        idx++;
    }
}
```

- UE_LOG() 메소드를 사용한다.

- 두번째 매개변수로 Log 전달시 흰색 글씨로 출력한다.
- 두번째 매개변수로 Warning 전달시 노란글씨로 출력한다.
- 두번째 매개변수로 Error 전달시 붉은글씨로 출력한다.

이벤트

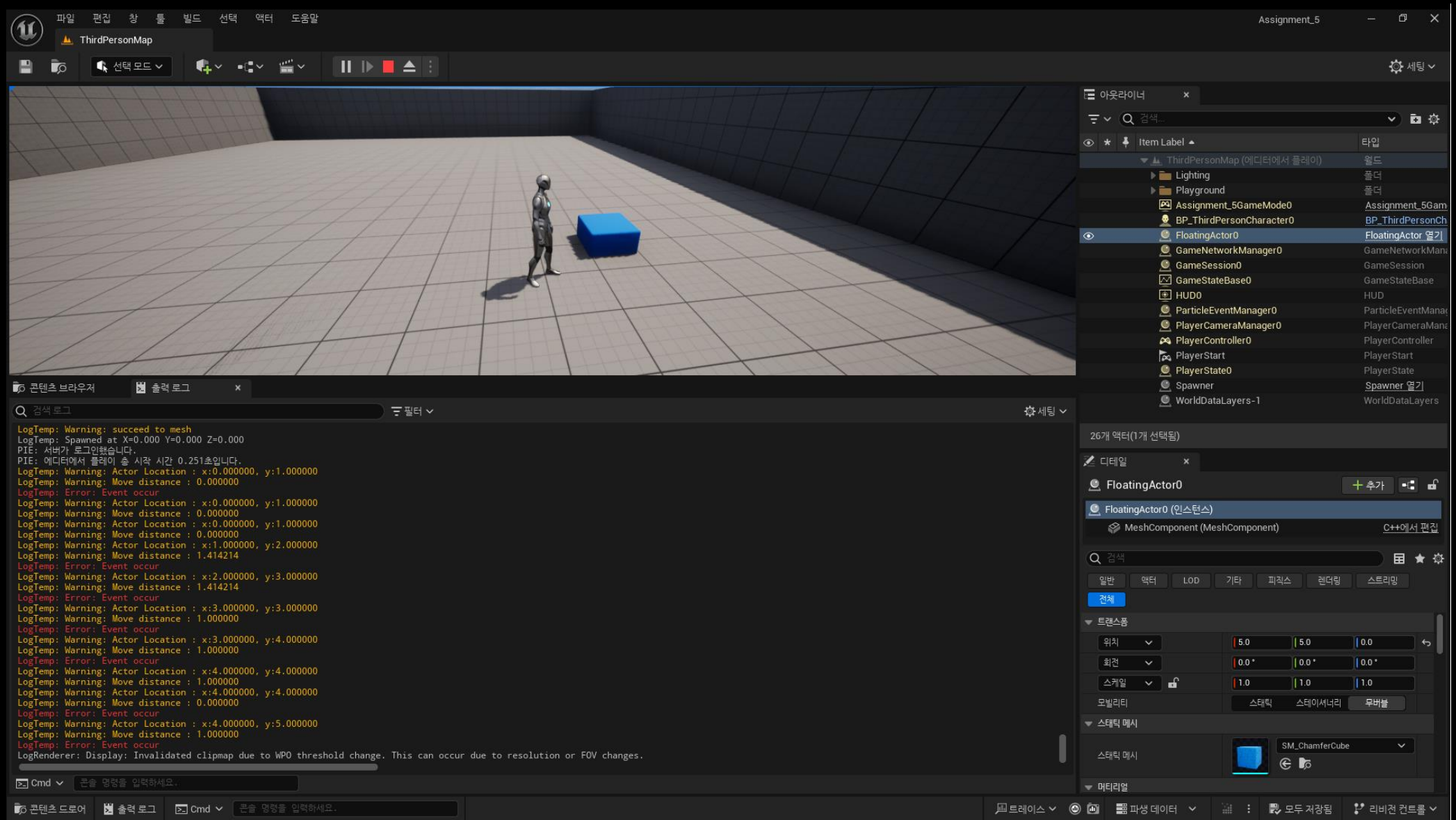
```
DECLARE_DELEGATE(FCustomEvent);

void EventHandler()
{
    UE_LOG(LogTemp, Error, TEXT("Event occur"));
}

// Called when the game starts or when spawned
void AFloatingActor::BeginPlay()
{
    Super::BeginPlay();
    OnEvent.BindUObject(this, &AFloatingActor::EventHandler);
    //Move();
}
```

- Event 처리를 위해 Delegate 를 선언했다.
- 이벤트가 발생했을때 호출된다.
- BeginPlay() 에서 이벤트를 바인드한다.
- 특정 조건에서 EventHandler() 를 호출한다.

결과

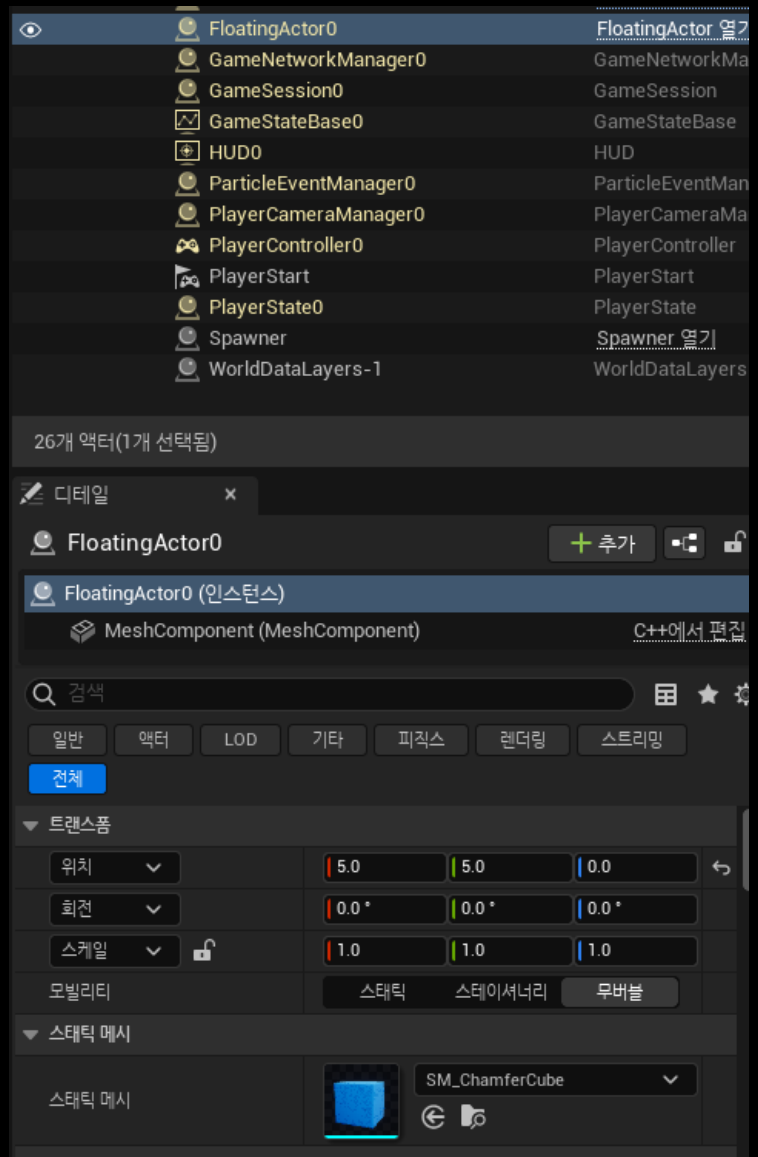

파일 편집 창 툴 빌드 선택 액터 도움말
Assignment_5
ThirdPersonMap
선택 모드
세팅
아웃라이너
Item Label
타입
ThirdPersonMap (에디터에서 플레이)
Lighting
Playground
Assignment_5GameMode0
BP_ThirdPersonCharacter0
FloatingActor0
GameNetworkManager0
GameSession0
GameStateBase0
HUD0
ParticleEventManager0
PlayerCameraManager0
PlayerController0
PlayerStart
PlayerState0
Spawner
WorldDataLayers-1
26개 액터(1개 선택됨)
콘텐츠 브라우저
출력 로그
필터
LogTemp: Warning: succeed to mesh
LogTemp: Spawned at X=0.000 Y=0.000 Z=0.000
PIE: 서버가 로그인했습니다.
PIE: 에디터에서 플레이 중 시작 시간 0.251초입니다.
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Warning: Actor Location : x:1.000000, y:2.000000
LogTemp: Warning: Move distance : 1.414214
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:2.000000, y:3.000000
LogTemp: Warning: Move distance : 1.414214
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:3.000000, y:3.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:3.000000, y:4.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:4.000000, y:4.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Warning: Actor Location : x:4.000000, y:4.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:4.000000, y:5.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogRenderer: Display: Invalidated clipmap due to WPO threshold change. This can occur due to resolution or FOV changes.
Cmd
콘솔 명령을 입력하세요.
디테일
FloatingActor0
추가
FloatingActor0 (인스턴스)
MeshComponent (MeshComponent)
C++에서 편집
검색
일반 액터 LOD 기타 피직스 렌더링 스트리밍
전체
트랜스폼
위치
회전
스케일
모빌리티
스태틱 메시
SM_ChamferCube
머티리얼
콘텐츠 드로어
출력 로그
트레이스
파생 데이터
모두 저장됨
리비전 컨트롤

결과

```
LogTemp: Warning: succeed to mesh
LogTemp: Spawned at X=0.000 Y=0.000 Z=0.000
PIE: 서버가 로그인했습니다.
PIE: 에디터에서 플레이 중 시작 시간 0.251초입니다.
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Warning: Actor Location : x:0.000000, y:1.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Warning: Actor Location : x:1.000000, y:2.000000
LogTemp: Warning: Move distance : 1.414214
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:2.000000, y:3.000000
LogTemp: Warning: Move distance : 1.414214
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:3.000000, y:3.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:3.000000, y:4.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:4.000000, y:4.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Warning: Actor Location : x:4.000000, y:4.000000
LogTemp: Warning: Move distance : 0.000000
LogTemp: Error: Event occur
LogTemp: Warning: Actor Location : x:4.000000, y:5.000000
LogTemp: Warning: Move distance : 1.000000
LogTemp: Error: Event occur
LogRenderer: Display: Invalidated clipmap due to WPO thresh
```

- FloatingActor 에 메쉬를 부여 성공하고 메시지를 출력한다.
- Spawner 가 FloatingActor 를 스폰하고 메시지를 출력한다.
- FloatingActor 의 Tick 에서 매 프레임마다 Move 를 실행하고 액터가 이동하며 각종 메시지를 출력한다.

결과



- FloatingActor 가 스폰되고, 실제 월드 좌표에서 이동했다.
- 생성자에서 경로상의 메쉬를 찾아 부여했다.