

C++ : Timer - Lamda 함수구조

```
FTimerHandle DelayTimerHandle;
//2초 Delay
UPROPERTY(EditAnywhere)
float DelayTime = 2;
//Looping 여부
UPROPERTY(EditAnywhere)
bool bIsLoop = false;
```

```
//Lambda 함수 구조 [REFERENCE 또는 CAPTURE]() -> Return 타입 {구현부}
GetWorld()->GetTimerManager().SetTimer(DelayTimerHandle, FTimerDelegate::CreateLambda(
    [this]() -> void
    {
        spawnStar();
    }
), DelayTime, bIsLoop);
```

```
FTimerHandle DelayTimerHandle;
//2초 Delay
UPROPERTY(EditAnywhere)
float DelayTime = 2;
//Looping 여부
UPROPERTY(EditAnywhere)
bool bIsLoop = false;
```

```
//Lambda 함수 구조 REFERENCE 또는 CAPTURE -> Return 타입 {구현부}
GetWorld()->GetTimerManager().SetTimer(DelayTimerHandle,
FTimerDelegate::CreateLambda(
[
this]() -> void
{
spawnStar();
```

```
}  
, DelayTime, blsLoop);
```

타이머 두개 동시에 할 때 변수 이름 다르게 해야됨

```
FTimerHandle TimerHandle1;  
FTimerHandle  
TimerHandle2;  
  
// 첫 번째 타이머 설정  
GetWorldTimerManager().SetTimer(  
TimerHandle1, this, &AKartGameState::Function1, 1.0f, true, 2.0f);  
  
// 두 번째 타이머 설정  
GetWorldTimerManager().SetTimer(  
TimerHandle2, this, &AKartGameState::Function2, 1.0f, true, 3.0f);
```