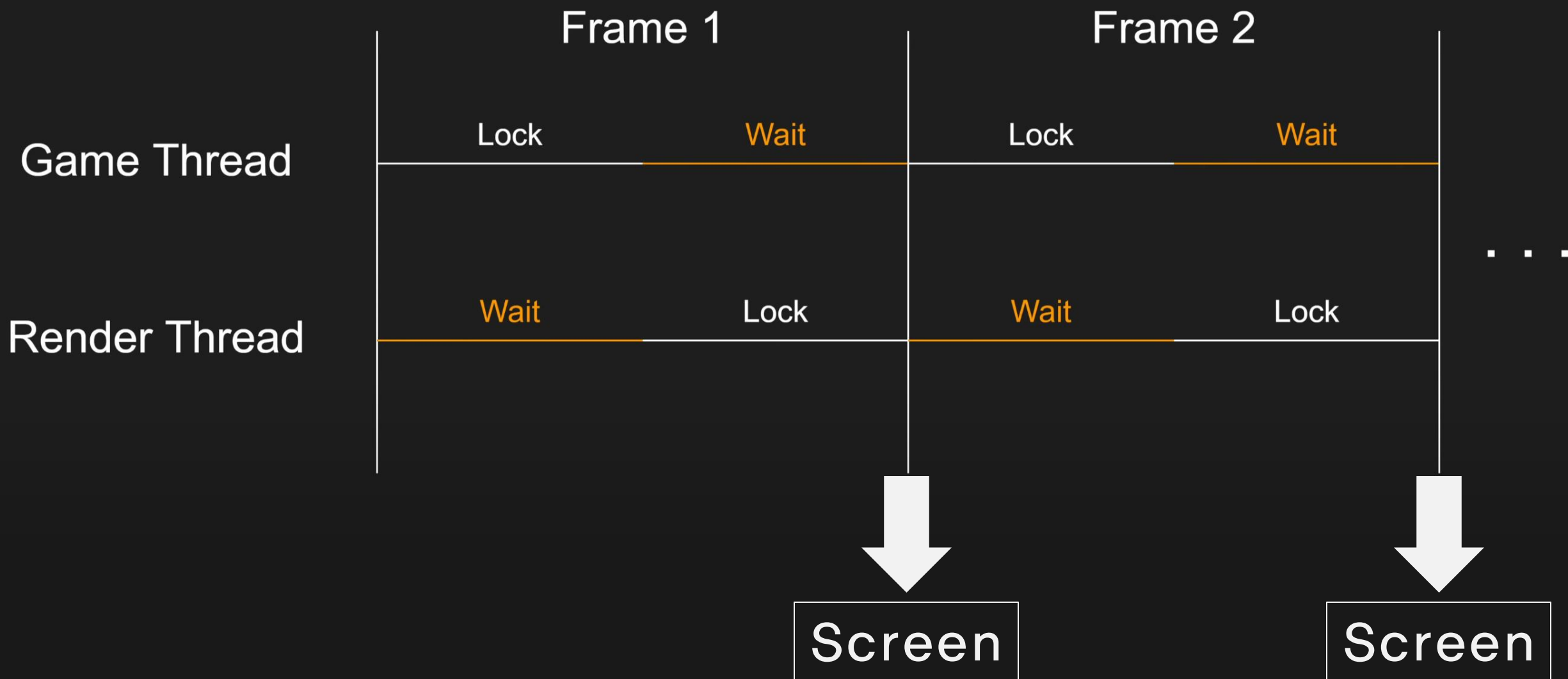


언리얼 엔진 멀티스레드 정리

이재형

게임 스레드와 렌더 스레드

이렇게 동작하면 너무 느려...

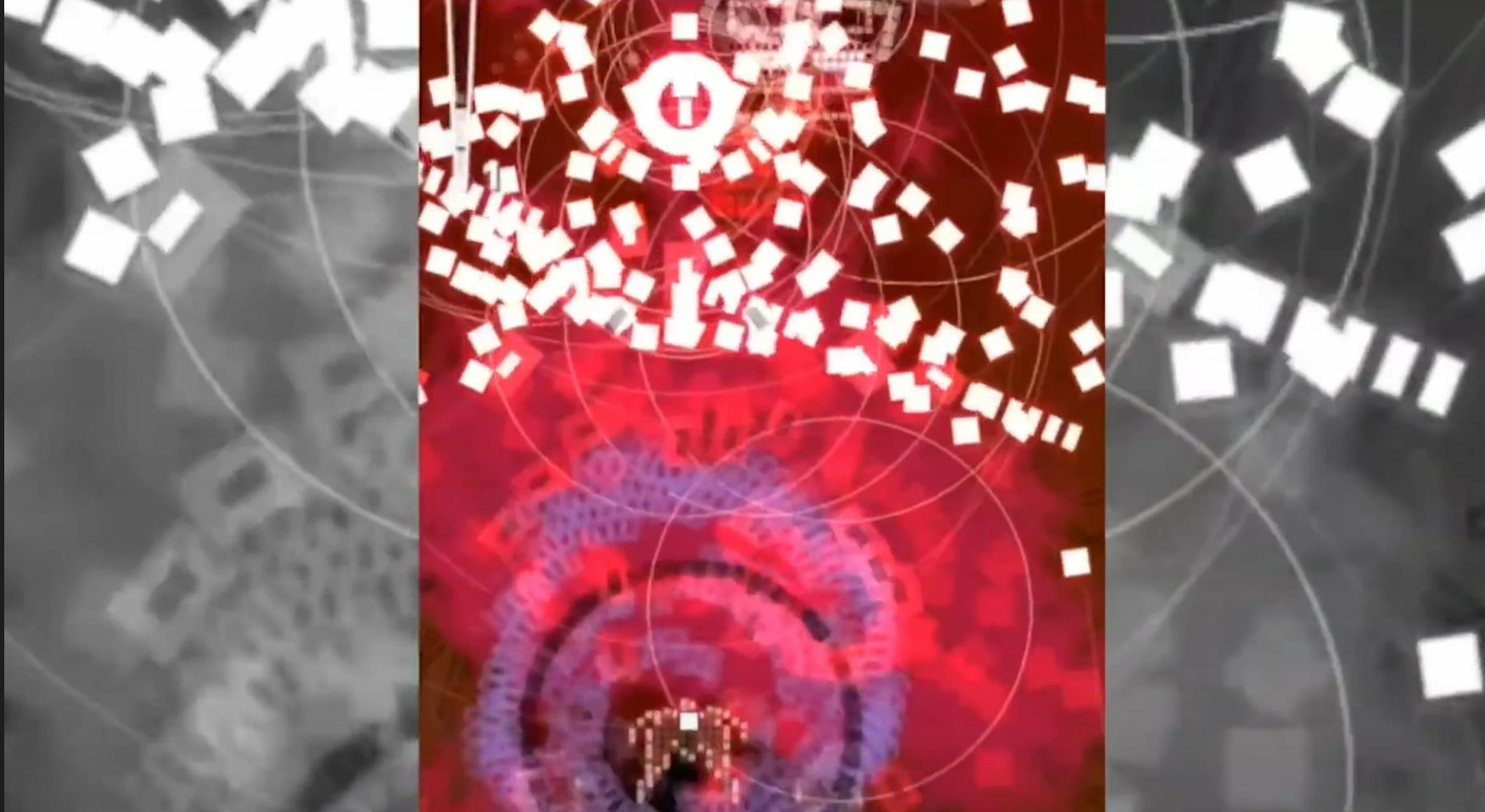


데이터를 복사해서 따로 연산해보자



할 일

게임 스레드 > 렌더 스레드



Game Thread

Frame 1

Frame 2

Frame 3

...

Render Thread

Frame 1

Wait

Frame 2

Wait

개꿀~

Screen

Screen



할 일

게임 스레드 < 렌더 스레드

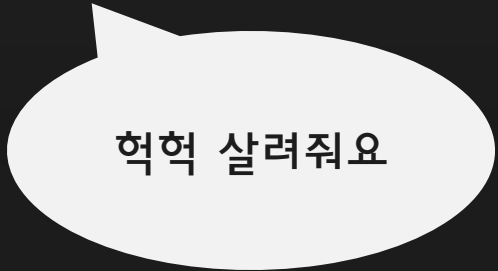


Game Thread



...

Render Thread



Screen

Screen

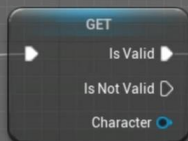
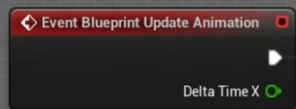
Screen

3 Frame Skip!

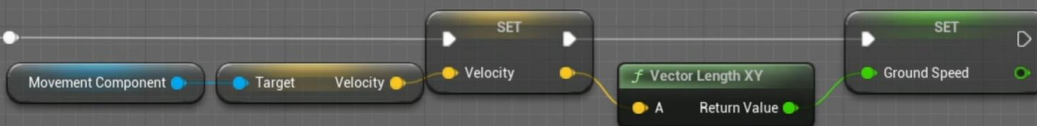
애니메이션 스레드

기존의 AnimBP

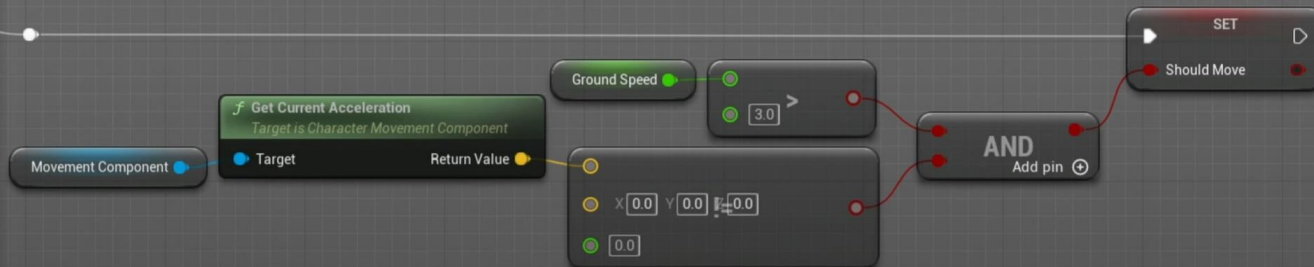
Only update values if the character is valid
(character exists in the world and the reference is set).



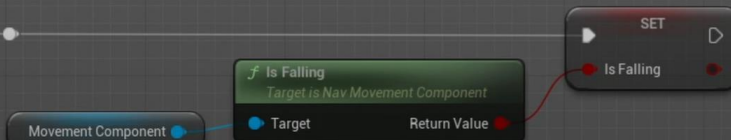
Set velocity and ground speed from the movement components velocity. Ground speed is calculated from only the X and Y axis of the velocity, so moving up or down does not affect it.



Set Should Move to true only if ground speed is above a small threshold (to prevent incredibly small velocities from triggering animations) and if there is currently acceleration (input) applied.



Set Is Falling from the movement components falling state.

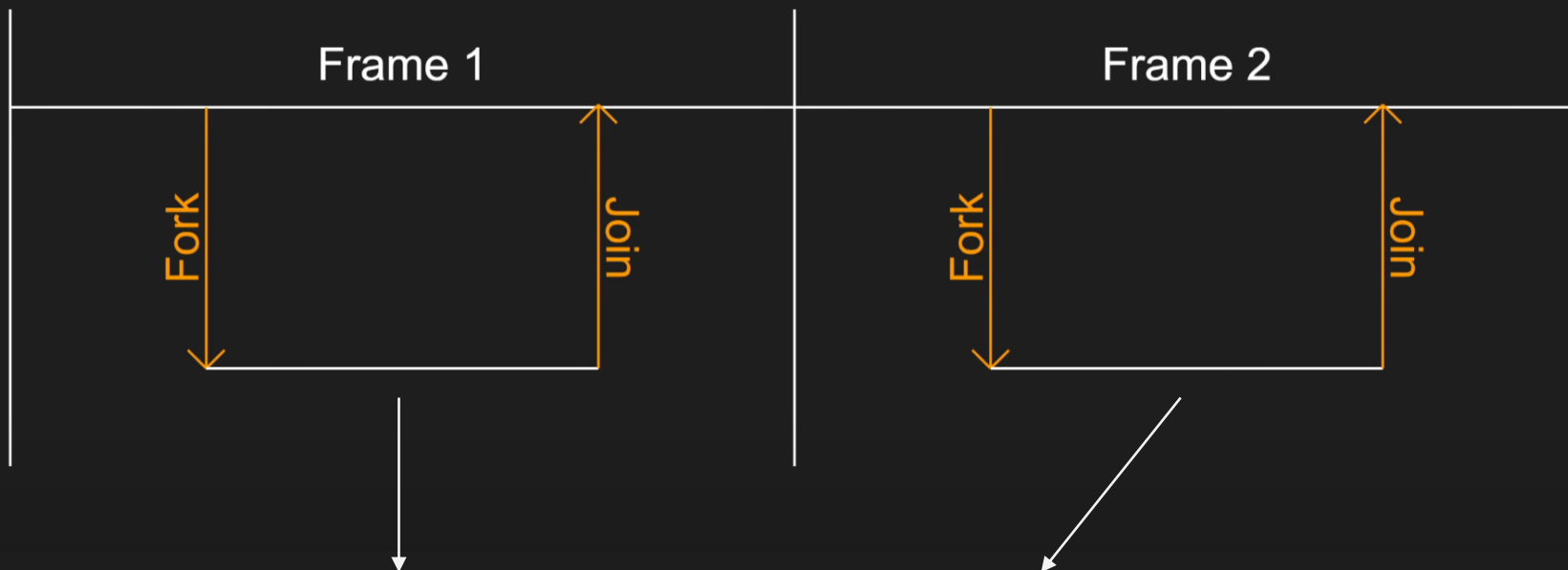


언제 끝나...



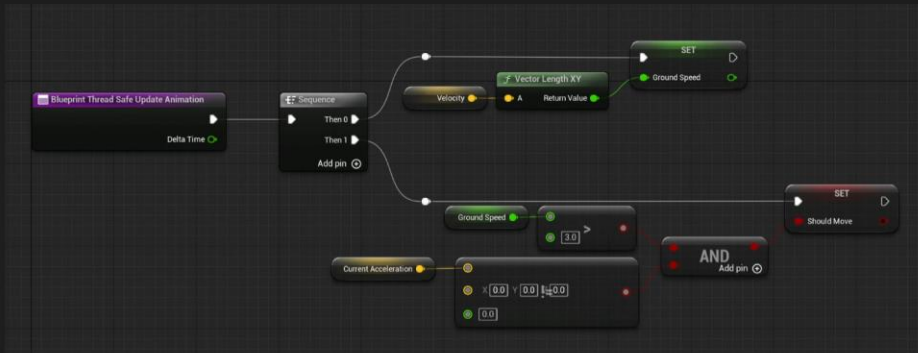
Game Thread

Animation Thread



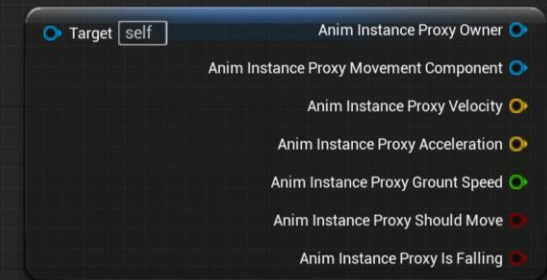
애니메이션 스레드가 일을 다 할 때 까지
게임 스레드가 기다린다

Game – Render 처럼 기다리지 말고 복사하자!



```
8 void FSampleTPAnimInstanceProxy::Update(float delta_seconds)
9 {
0     Super::Update(delta_seconds);
1
2     SCOPE_CYCLE_COUNTER(STAT_Update);
3
4     if (m_owner)
5     {
6         m_ground_speed = UKismetMathLibrary::VSizeXY(m_velocity);
7         m_should_move = m_ground_speed > 3.0f && !m_acceleration.Equals(FVector::ZeroVector);
8
9         // copy values into anim blueprint
0         m_owning_anim_instance->m_velocity = m_velocity;
1         m_owning_anim_instance->m_acceleration = m_acceleration;
2         m_owning_anim_instance->m_ground_speed = m_ground_speed;
3         m_owning_anim_instance->m_should_move = m_should_move;
4         m_owning_anim_instance->m_is_falling = m_is_falling;
```

Using Anim Instance Proxy



Using Seperate variables and copying data from anim instance proxy



혹은 Thread Safe한 프로퍼티 액세스

Only update values if the character is valid (character exists in the world and the reference is set).

Set velocity and ground speed from the movement components velocity. Ground speed is calculated from only the X and Y axis of the velocity, so moving up or down does not affect it.

Set Should Move to true only if ground speed is above a small threshold (to prevent incredibly small velocities from triggering animations) and if there is currently acceleration (input) applied.

Set Is Falling from the movement components falling state.

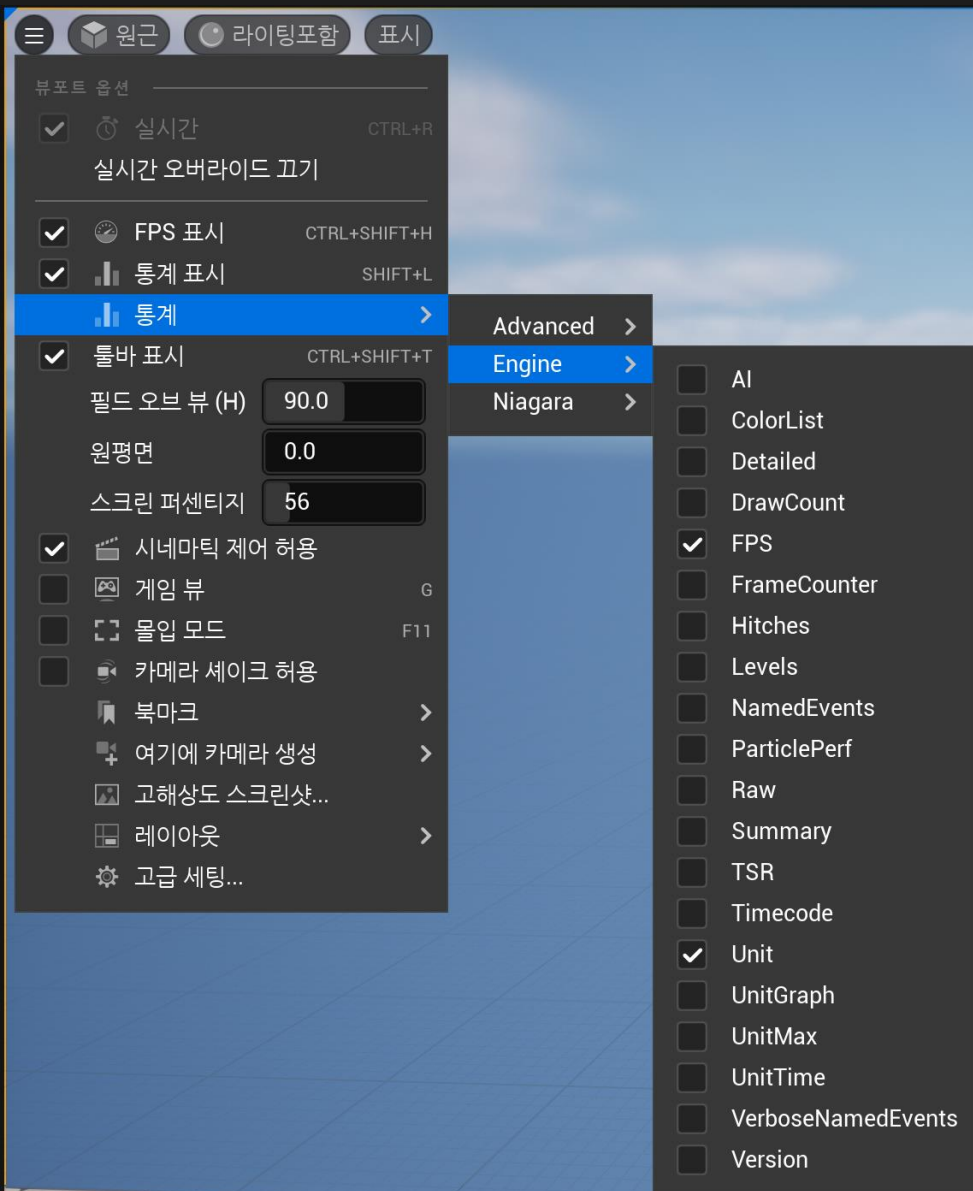
Character ☒ Thread Safe

이 블루프린트에 대한 모든 액션 ☒ 컨텍스트에 따라 ▶

✕ 프로퍼티

Variables

☒ 프로퍼티 액세스



우리의 주요
최적화 타겟

게임 스레드

렌더 스레드

Unreal Insight

멀티 스레드 동작 프로파일링 용이



참고 문서

[1. 멀티스레드 프로그래밍 소개] 07. Case Study(어디서 멀티스레드 프로그래밍을 하는가?)/ Heterogeneous , Homogeneous

애니메이션 최적화

[UE5] Understanding Render Thread and Animation Thread in Unreal Engine

언리얼 인사이트 개요

언리얼 인사이트와 친해지기 위한 준비 운동