需求：

希望做到的是，记录下每一帧，包括发生了什么事件，碰撞等等，将一些程序的数据进行记录（具体有什么数据，需要分析，是否能够记录），运行时的数据。快进，慢放功能。记录存成资产，必须的吧。

进阶一些：希望可以，比如暂停在某一帧之后，记录的事件是，剑攻击到了某个敌人，在暂停之后，调整剑的碰撞体大小，攻击到了多个敌人，那么继续播放之后，是要改变未来，并且将数据覆盖（或者，另存吧）。

## 排期：

1共12天

理解需求并根据需求进行UE重播系统的学习，向之前做过的程序进行学习 2d

根据需求与学习成果测试UE4的重播系统 1d

根据学习结果评估是否需要扩展功能或是修改现有代码，并列出文档 1d

进行录像功能的具体制作 8d

以两天为一个迭代，输出阶段成品 2d

查资料，然后想办法解决镜头错位问题，再看看能否让镜头从一个好的位置开始。

现在播动画，一闪一闪的，就行走动画能稍微好一点的记录，攻击等动画完全播不出来，想办法解决下。

特效也并没有一起被记录，解决特效没有一起播放，是不是蒙太奇没有播放？

梳理需要记录的数据信息例如按键信息，碰撞信息，伤害信息，技能信息等

测试看是不是能，正常记录需要记录的数据信息，如果不能的话，看下如何才能将其保存下来。

如果需要添加显示Replay中的播放UI，需要做哪些事情？

测试暂停后，如果动态修改Collision的属性等，是否能够动态更改“未来”？如果能，是否会将原始replay记录覆盖。

如果不能动态修改，尝试查阅资料进行修改，如果不会覆盖，尝试是否需要重新保存。

1、查阅资料了解Replay时的镜头方案 1d

2、解决重播时镜头一开始的错位问题 2d

3、查阅资料，查找动画一闪一闪无法正常播放以及特效没有显示的原因，尝试记录除了行走外的一个简单动作 2d

4、解决除了动画一闪一闪的问题，正常显示特效，正常播放replay。 2d

5、与策划等沟通，梳理需要记录的数据信息例如按键信息，碰撞信息，伤害信息，技能信息等。0.5d

6、测试是否能正常记录需要的数据，如果不能，研究如何才能将其保存。2d

7、沟通Replay功能的UI显示需求，添加播放，暂停，进度条等播放显示UI 1.5d

8、测试暂停时，是否能动态修改碰撞体等属性以影响后续replay，如果不能动态修改，查阅资料评估是否能进行扩展 1d

（此项为基于上述功能的延伸，需要额外的时间）

9、扩展Replay功能支持暂停途中修改属性并影响后续。2d

10、测试修改属性是否会将原始Repaly记录覆盖，如果会将原始Replay记录覆盖，沟通需求看是否需要还原功能。（若需要还原功能，则此项需要额外时间）2d

先去看Replay System的东西，

输出ReplaySystem的记录原理文档。

Replay一个攻击动画，

Replay含特效的动画，

1、查阅资料了解Replay时的镜头方案 1d

2、解决重播时镜头错位问题 2d

3、研究ReplaySystem的record记录方式，并输出文档 3d

4、根据研究结果，评估但丁的重播系统是采用ReplaySystem记录方式还是使用记录Input的方式。1d

5、根据评估结果，设计Input重播系统或是设计代码修改以满足ReplaySystem记录条件。

1d

6、重播系统制作或是修改代码。3d

7、Replay一个基本攻击动画。1d

8、Replay一个带有特效的动画。1d

9、Replay动画并保证碰撞正常记录1d

10、将状态机数据记录 1d

1、沟通重播系统的设计需求以及任务拆分。1d

2、研究ReplaySystem的record记录方式，并输出文档 3d

3、根据研究结果，评估但丁的重播系统是采用ReplaySystem记录方式还是使用记录Input的方式。1d

4、根据评估结果，设计Input重播系统或是设计代码修改以满足ReplaySystem记录条件。

1d

5、重播系统制作或是修改代码。3d

6、Replay一个基本攻击动画。1d

7、Replay一个带有特效的动画。1d

8、Replay动画并保证碰撞正常记录1d

10、将状态机数据记录 1d

1、查阅资料了解Replay时的镜头方案 1d

2、解决重播时镜头错位问题 2d

初步测试问题：

1：视角问题，一开始的视角总是很奇怪，当然这可能是因为，默认使用了CameraPawn作为相机，而在Replay的时候，并没有给这个ACameraPawn设置Controller

2：好像，并没有记录下人物的攻击动作，从显示上看，记录了行走的Animation，和Position

## ReplaySystem记录：

在重播系统中的DemoNetDriver和流送器，很重要，需要在

[/Script/Engine.GameEngine]

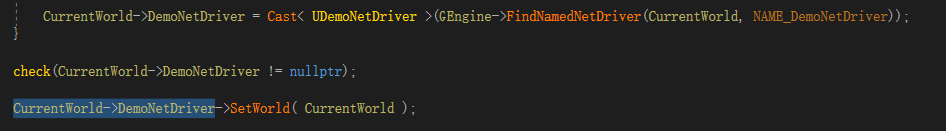
+NetDriverDefinitions=(DefName="DemoNetDriver",DriverClassName="/Script/Engine.DemoNetDriver",DriverClassNameFallback="/Script/Engine.DemoNetDriver")

中开启

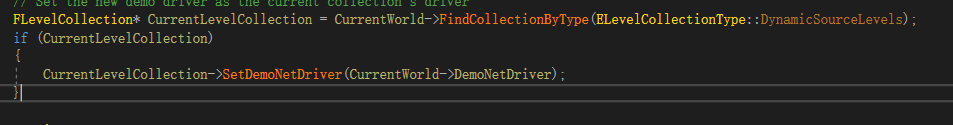
是因为创建了Actor的原因吗，所以tick和状态机都是在跑的

### StartRecord

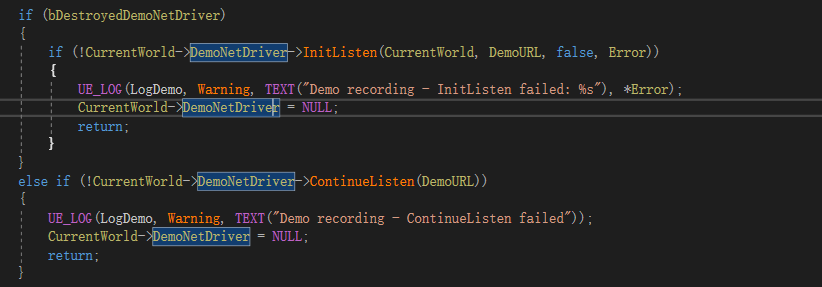
开始记录的时候会重新开启几个变量，



会重新在World上创建一个DemoNetDriver



会在当前world中找一个FLevelCollection



开始录制的步骤直到InitListon 就结束了。

**InitListon:**

先在单机本地模拟一个网络连接，然后，找到所有当前World的Controller，

ReplayStreamer->StartStreaming(Params, FStartStreamingCallback::CreateUObject(this, &UDemoNetDriver::ReplayStreamingReady));

然后开启Streaming流送，并且设置一个回调！

又多了个名词 ：

FLocalFileNetworkReplayStreamingFactory

并没有进行序列化呀，那么是怎么存储的呢？

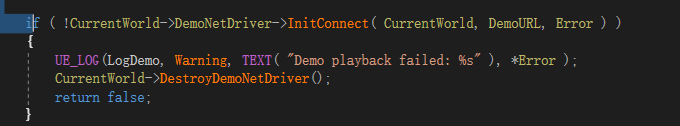
UDemoNetDriver

FURL

从PlayReplay入手：

是否有重新RunMap？ 是否有初始化，正确的character数据？

其他代码都一样，只有这一行是不一样的。



RePlay的代码都差不多，都是在最后调用StartStreaming

#### 1）、 确实重新进行了初始化

确实看到AbilityComponent进行了初始化，但是状态机是不会变化的，那就是Input状态没有记下来，UCharacterInputBufferComp中，StrategyInputActionDelegate委托也是空的。

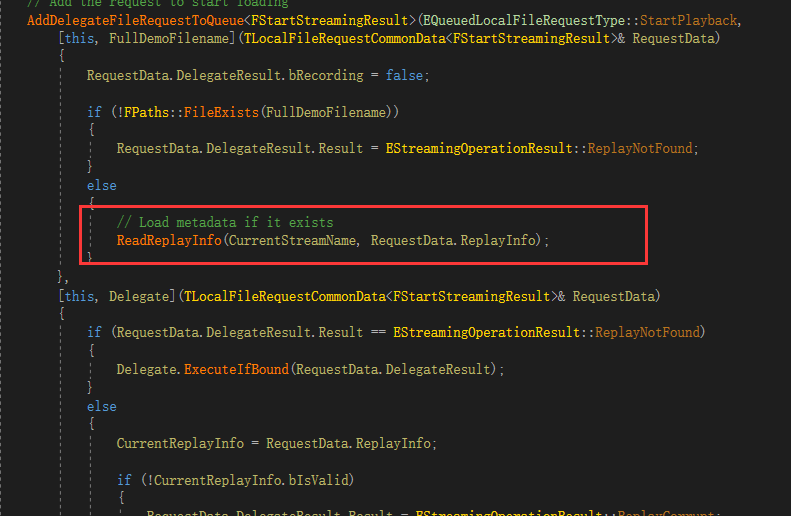
那么如果，在replay的模式下，不去Update状态机，是不是就能，正常显示动画播放了呢？！不行，测试失败了，虽然状态机没有跑，但是也没有正常播放Animation或是蒙太奇。

ReplayStreamer->StartStreaming(Params, FStartStreamingCallback::CreateUObject(this, &UDemoNetDriver::ReplayStreamingReady));

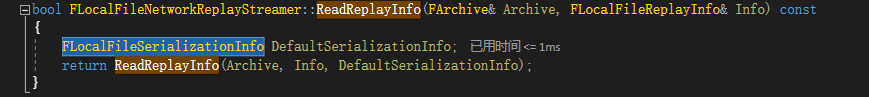
从这个入手，看看是怎么播的，并且每一帧都在干啥

如果是本地Replay，则是调用

FLocalFileNetworkReplayStreamer::StartStreaming

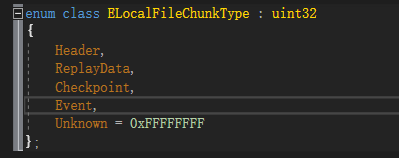
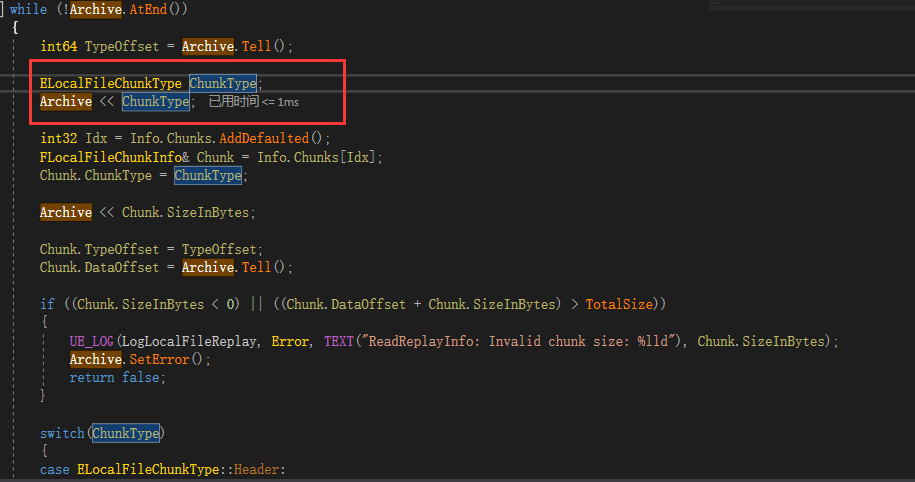


在其中终于能看到，对录屏文件的处理了。



终于要开始，读点东西了。

这里面，有个ReplayData！



通过，ReadReplayInfo函数，将数据存储在FLocalFileReplayInfo 里面，

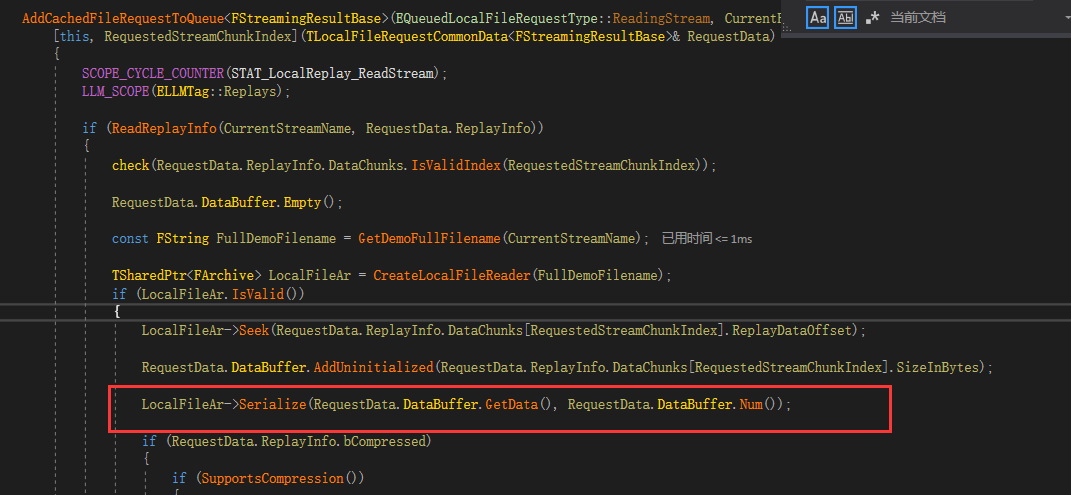
嗯？点一下F5发现，好像点击的时候加载一遍，到了这个界面，又会加载一遍。



我在

AddCachedFileRequestToQueue

中看到Serialize了，感觉，他乡遇故知。

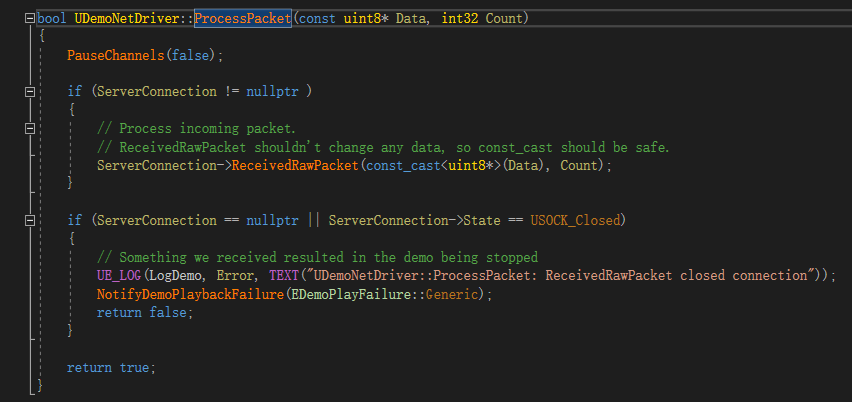


这个DataBuffer应该是一个TArray<int32>类型的 ，这个LocalFileAr是一个

FArchiveFileReaderGeneric ，调用他的FArchiveFileReaderGeneric::Serialize

嗯？好像是 隔一段时间就会读取一次数据？

嗯，正式创建场景，是在这里



再上一级是UDemoNetDriver::TickDispatch

DemoNetConnection->ReceivedRawPacket

用的一个类叫做DataChannel ，

bool UChannel::ReceivedSequencedBunch( FInBunch& Bunch )

{

SCOPED\_NAMED\_EVENT(UChannel\_ReceivedSequencedBunch, FColor::Green);

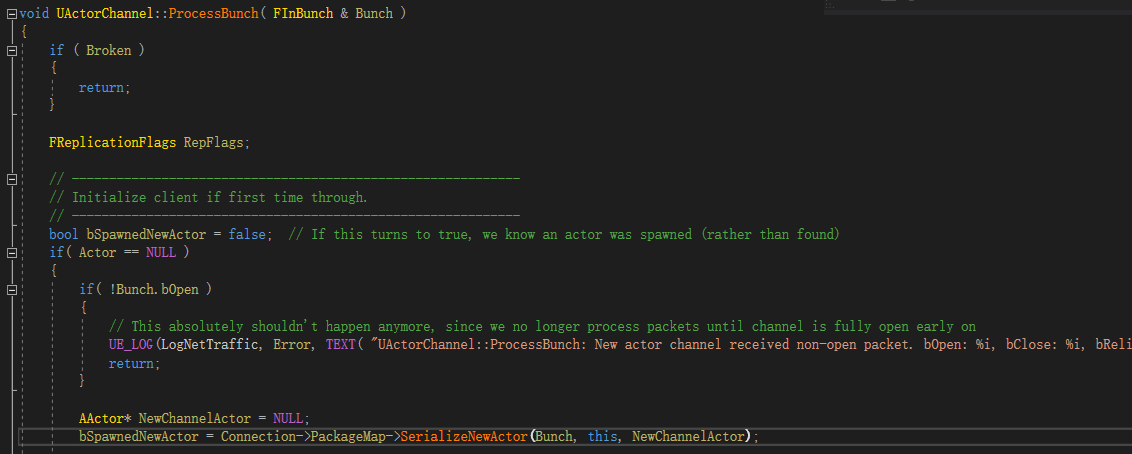
// Handle a regular bunch.

if ( !Closing )

{

ReceivedBunch( Bunch );

}



就开始正式生成记录的Actor了。

Bunch , Packet等，都是与服务器通信的网络相关，所以存储的时候，也是以网络通讯的方式！

FObjectReplicator   
属性同步的执行器，每个Actorchannel对应一个FObjectReplicator，每一个FObjectReplicator对应一个对象实例。设置ActorChannel通道的时候会创建出来。

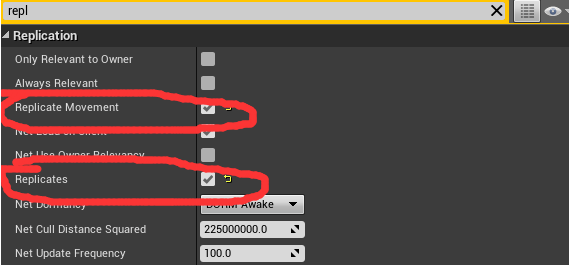
UActorChannel: 在录制视频的时候，会创建一个，每个Actor应该对应一个UActorChannel进行同步。每一个Actor都会创建一个Channel，不仅仅是玩家，包括摄像机，游戏模式，GameState，PlayerController，WorldSetting等等，都会创建。

先会SetChannelActor() ，并初始化一些数据，并且创建ActorReplicator

#### 5.29

需要在蓝图中，将Replicated勾起来，场景中的静态物体需要在Actor 中把Replicate Movement勾选。





隔一段时间就会调用一次FLocalFileNetworkReplayStreamer::FlushStream

结束录制时也会调用FlushStream。

O ~在GameMode里有个这个



UPROPERTY(replicated) 为什么不能用

  
同步数据的话，通过HasAuthority ，去调用服务器的方法。

如果是网络游戏直接使用这样的方式来同步的话，Replay会更加简单。

ReplicatedUsing 一般出现在函数的声明中,表示当该变量在服务器上变化时,会调用事先绑定好的回调函数(函数名一般以OnRep开头),需要注意的是服务端不会自动调用绑定好的该回调函数,此时需要通过代码手动调用.

UPROPERTY(ReplicatedUsing = OnRep\_AttachChildren, Transient)

TArray<USceneComponent\*> AttachChildren;

### Transient

该属性为临时属性；不应被保存，在载入时会被填零。

UPROPERTY(ReplicatedUsing = OnRep\_AttachChildren, Transient)

## 好正式开始测试记录：

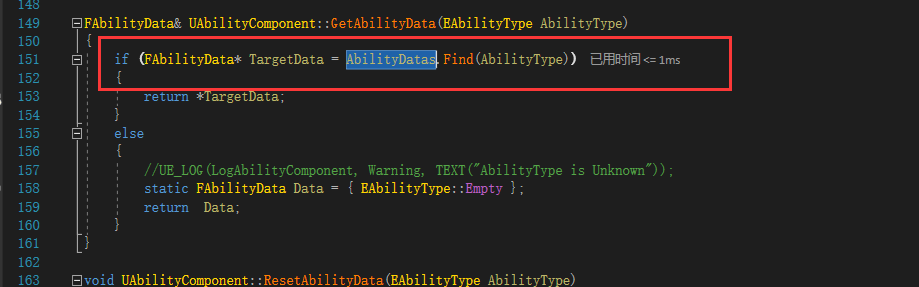
1,如果记录输入的话，可能就不使用ReplaySystem了

思路：如果，我直接把AnimationInstance记录下来呢~是否能够正常播放，还是要屏蔽状态机？ 只记录，失败了，看下如果屏蔽状态机呢 失败了 不对！我刚才忘记写进去了！

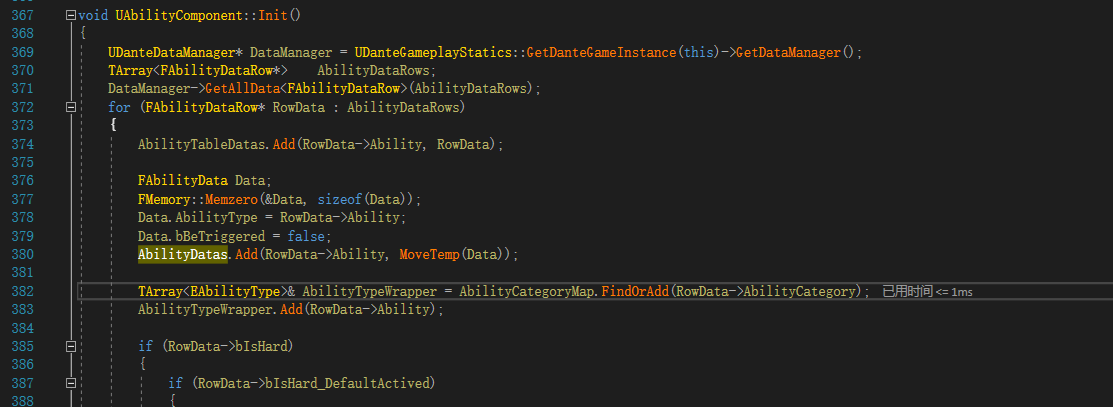
想也是，肯定不可能去赋值动画呀。。所以就还是看看，能不能把状态机的状态赋值出来。

所以应该，从SWITCH\_STATE\_AND\_RETURN 这里入手，看如何将状态机的状态记录下来？

记录状态机的StateID入手！！！ 但是思考一个问题，会不会和，本身记录的Actor移动起冲突呢



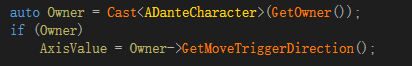
Replay时，获取的AbilityDatas不同！！！！！，TargetData的Data，会比正常载入的少一项！

关注一下

## 现在的问题是：

如果采用状态记录，则必须要勾选同步位置，否则的话，无法正确播放朝向等数据，但是记录位置，势必造成，唯一的闪烁，因为位置控制会由状态机以及rootmotion来进行。

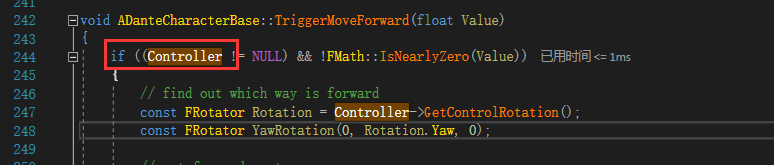
如果采用输入记录，则就是需要记录的按键比较多，是否需要记录抬起状态？在DanteCharacterBase中需要记录镜头、人物移动等情况，在CharacterInputBufferComp中记录按键状况。 那么考虑一种情况，会不会！由于同一帧有两个按键按下，造成按键无法被记录到？还有，在DanteCharacter的时候，是会不断地呢去UpdateInputKey（EinputKeyMove的）但是这个按键其实不需要记录的，因为他是获取



这个，所以，还是只需要在CharacterBase中去记录即可

Actor的bReplicateMovement一直是true，那么我手动设置一下！但是设置完，就完全不动了。。

还有一个问题，此时是没有controller的



1. 状态同步的问题：

闪回，服务端同步时间与rootMotion的对应问题

一开始的时候，会变成空中动作，但这个可能是代码的问题。

记录，并没有把每一个状态切换完美的记下来。

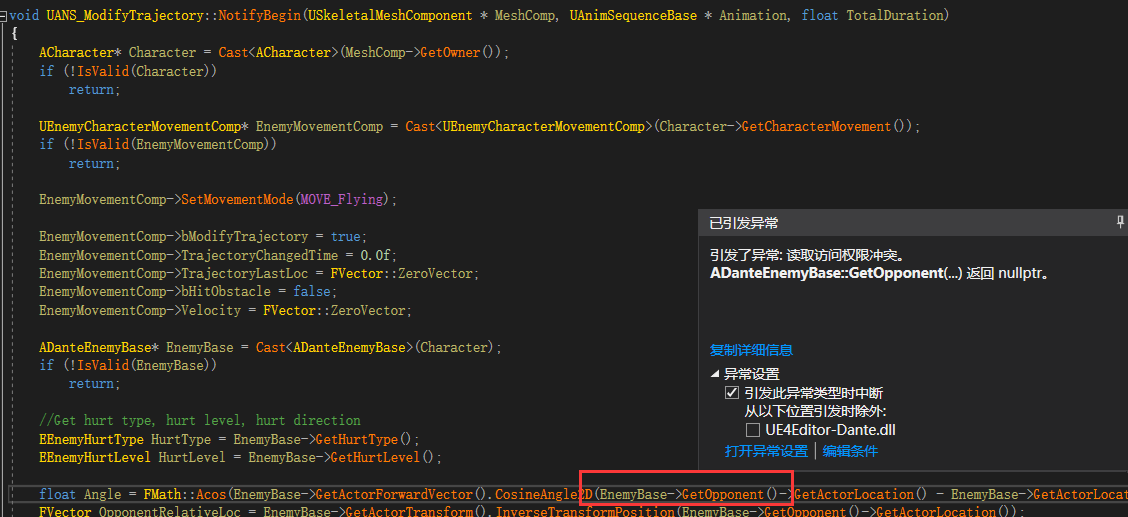
在英勇带领下。。。。。。解决了。。。。。。。。。

## 修改设计：

下落状态是否没有更新，所以会造成，在空中出拳的情况。

报错问题：

1、拿枪的Murdock敌人被攻击的动画回调中找不到对手



右键技能使用报错问题

MoveMent的状态不对，是因为没有执行Pawn的Restart，这个函数是有PlayerContoller调用的。

那么我直接在BeginPlay的时候加一个Restart呢，可以解决初始化问题

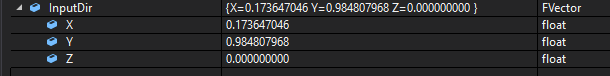
试一下，切换地图的问题。

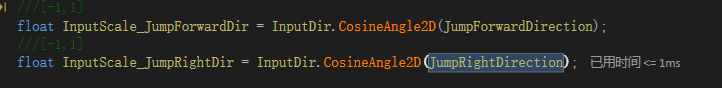
跳跃问题



正常来看，这两个是有值的







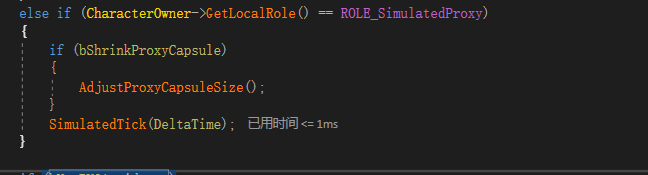




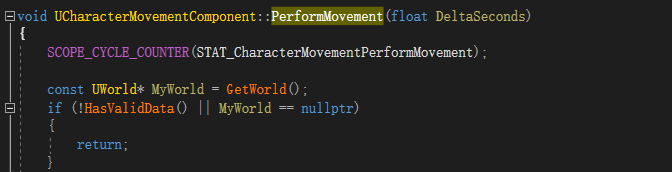
为什么重播的时候，移动输入无效呢？！

因为在UCharacterMovementComponent::TickComponent 中，直接就判断了，是本地模拟，不会走更新输入的流程！正常是走：PerformMovement 最终是调了这个更新应该是





直接走到这里来了！

如果是输入最终调用这一个.

也会去计算rootMotion等

!!添加了Controller之后，跳跃落地之后不会位移了！！

Replay可以切换地图的哦

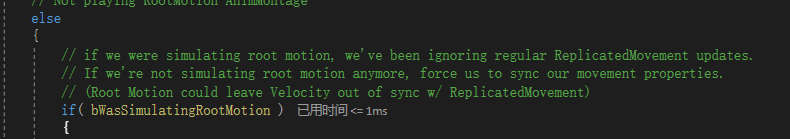
会不会就是FMoveTriggerData 没有记录呢！

几个影响移动的函数，尝试断点记录下作用：

UpdatePlayerMoveInput 暂时不知何时调用

UpdateActorRotation 只有在锁定的时候调用。

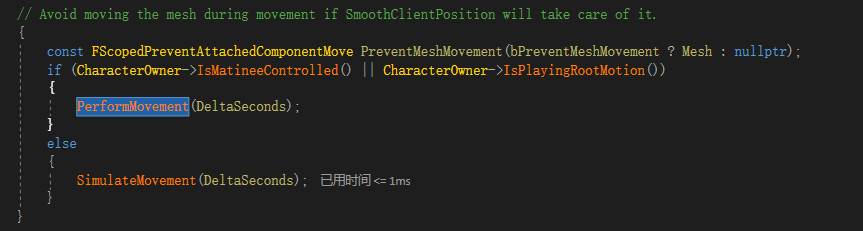
了解下同步的时候的移动更新：



如果我们在模拟根运动，我们忽略了常规的复制运动更新。

//如果我们不再模拟根运动，强制同步运动属性。

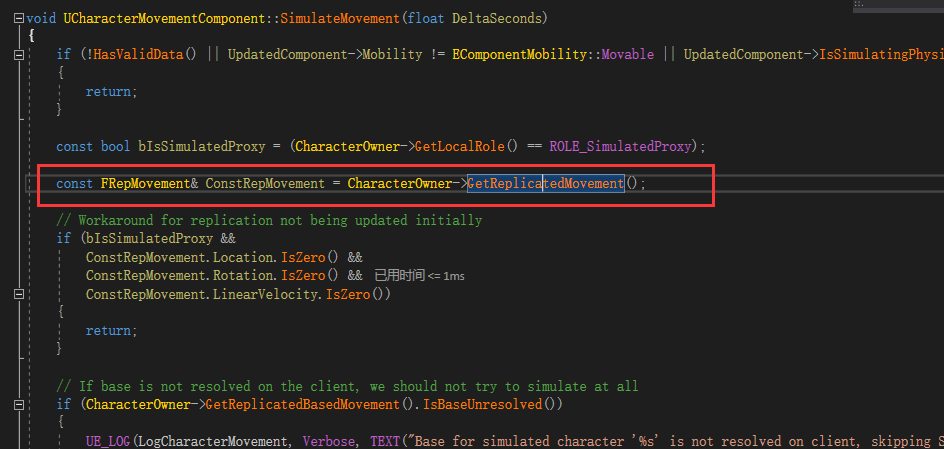
//(根运动可能会使速度与复制运动不同步)



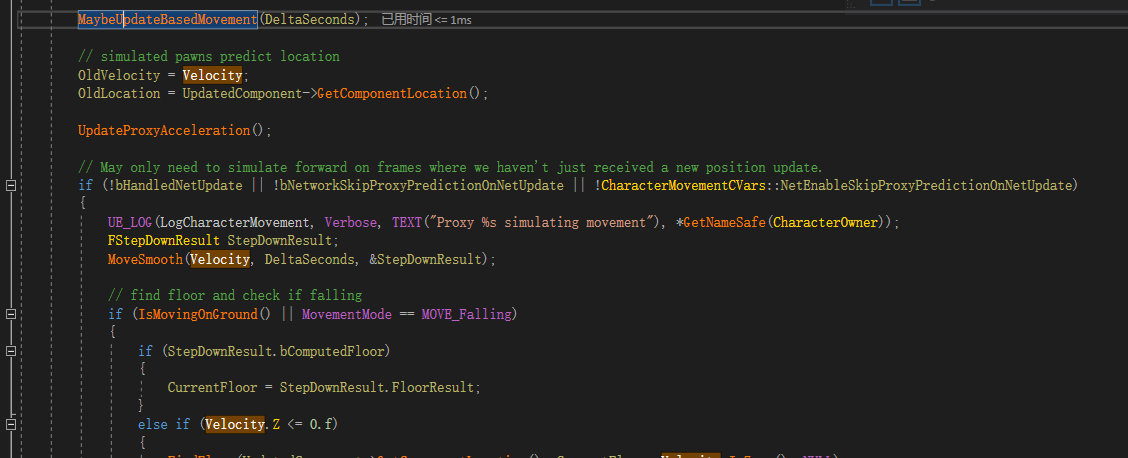
如果在运动中发现进行了RootMotion或是MatineeControlled被Matinee控制的话，更新PerformMovement，就使用输入的更新了！**所以，跳跃这样的位置更新OK，是不是就是因为，处于playRootMotion呢？**

如果是普通状态，会调用到

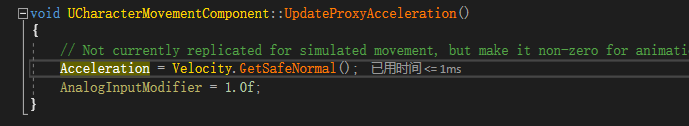
SimulateMovement() 模拟移动



会在里面去获取ReplicatedMovement复制了的运动，就是在此判断，是否需要更新移动等的吧！



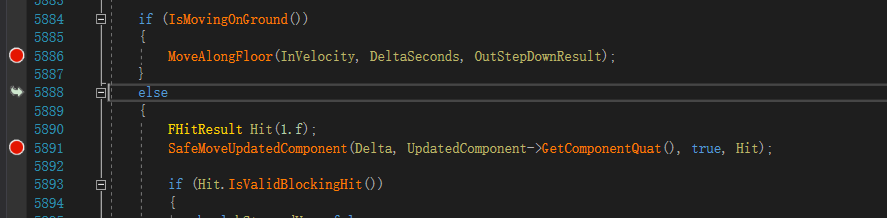
再下面一点，就是确确实实用到velocity了！

也更新了加速度，Acceleration，但是其实从复制的角度来说，也不太需要加速度？所以直接将加速度取的是Velocity的向量的方向（规范化向量）

再往下，就是这个函数了！应该正常的运行复制，就是用这个了

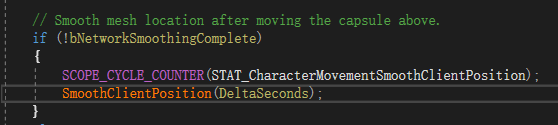
MoveSmooth(Velocity, DeltaSeconds, &StepDownResult);

在该函数内，



去调用地面或是空中移动。

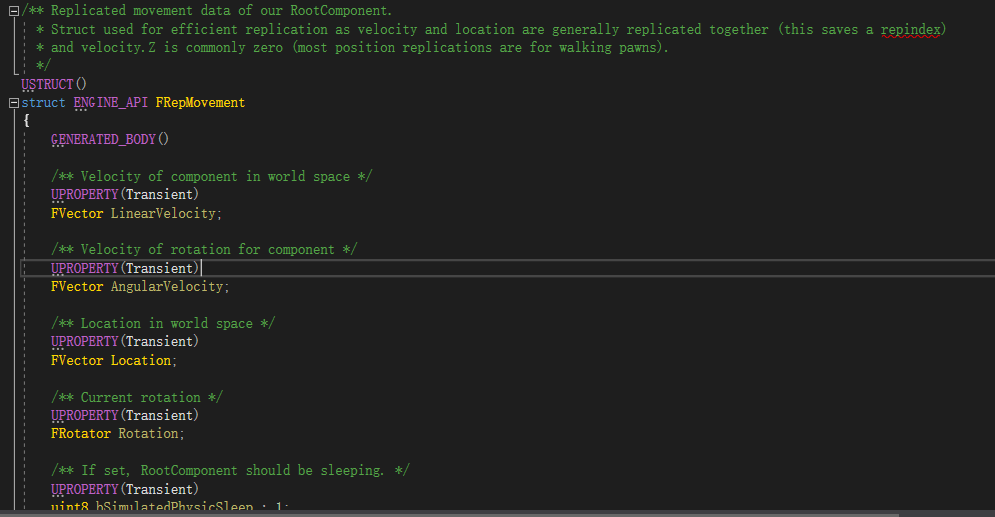
Z在移动函数调用完之后，



用SmoothClientPosition来平滑Mesh的位置了。

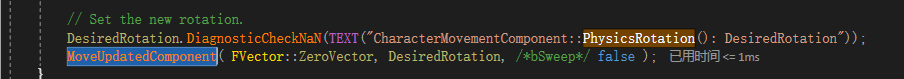
感觉，与平滑函数有关，如果会闪动的话。

FRepMovement



各种记录数据都在里面了~ 其中就有LinearVelocity复制的速度了

对于旋转角度，如果是正常的移动，最终是会到这里来的，PerformMovement，如果要记录，是否能将这个记录？



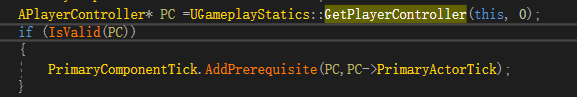
跟踪代码发现，在运行的时候，一直都是走的PerformMovement呀！那么为啥呢

不是旋转问题，仔细看的话能发现，应该是加速度太快了

现在问题：取消防御没有加，与敌人攻击的时候，会出现位置闪烁，猜测是获取playercontroller的问题。

: 使用到GetPlayerController的部分，都先打上断点，看看Replay时影响不，

最重要我觉得是InputBufferComp！因为获取的PC，



不是玩家真正的PC

还有GetPlayerCharacter 把这两个先改了尝试下

#include "Dante/Common/Actor/ActorUtility.h"

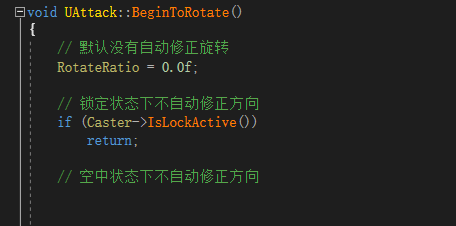
这个MatchComboKeys 在INputBufferComp中的，是否需要记录呢

UCharacterCrowComponent

Enemy的hitOtherBody 是在蓝图中写的！！ 不过反正也每调用

假设1 、冲量，是记录下来了的。

假设2，就是通过不断的MoveAlongCurve来处理的位置呢



再尝试下，如果是replay 不自动修正方向, 还有UpdateLocation如果是replay，也不修正方向。