nDisplay Rendering Pipeline 기능 추가

**1. nDisplay Rendering Pipeline에 NVIDIA DLSS 기능 추가.**

**(1) 개요**

NVIDIA DLSS(Deep Learning Super Sampling) 기능은 Real-time Ray Tracing 렌더링 속도를 높이기 위해 해상도가 낮은 프레임 버퍼에 Ray Tracing으로 렌더링하고 RTX GPU의 전용 Tensor 코어 프로세서를 이용하여 원래 해상도로 Upscale 하는 인공지능 기반 샘플링 기능이다.

DLSS UE4 Plugin을 설치하면 일반적인 플레이에서는 정상 작동을 하나, nDisplay로 작동시 DLSS 기능이 작동하지 않고 있다.(UE4 4.26.1에서)

그래서 DLSS 기능을 nDisplay에서도 사용해 볼 수 있도록 하기 위해, nDisplay Rendering 관련 코드에 DLSS 기능을 활성화하는 작업을 진행 한다.

**(2) NVIDIA DLSS 활성화**

DLSS Plugin을 프로젝트에서 사용 가능하도록 등록한다.

프로젝트의 DefaultEngine.ini 파일에 DLSS 환경설정 항목을 추가 한다.

[/Script/DLSS.DLSSSettings]

bEnableDLSSInEditorViewports=True

bEnableDLSSInPlayInEditorViewports=True

bEnableNonProductionBinariesInShipping=True

NVIDIANGXApplicationId=0

NVIDIANGXApplicationId는 NDIVIA DLSS Developer Network에서 발급 받은 Application ID를 입력해 준다.

nDisplay에서 DLSS 활성화를 위해서는 추가적인 환경설정 항목을 하나 더 추가 한다.

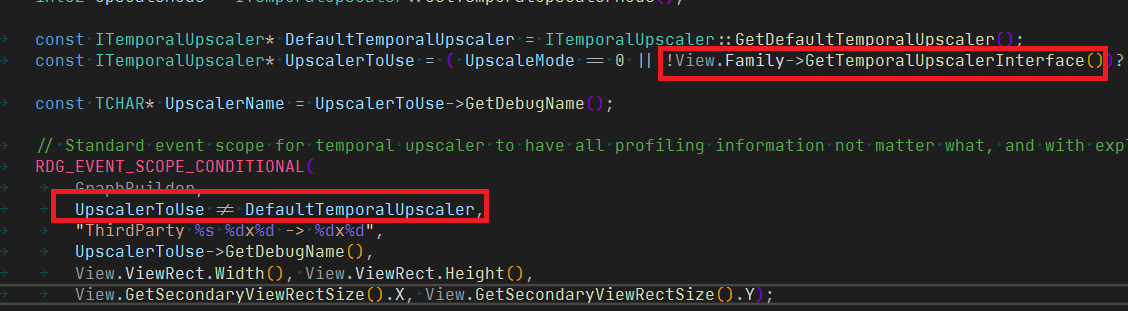
[/Script/Engine.RendererSettings]

r.TemporalAA.Upsampling=True

nDisplay 관련 코드를 더 조사해 봐야 알겠지만, nDisplay에서는 r.TemporalAA.Upsampling이 기본값으로 비활성화 되어져 있다.

**(3) NVIDIA DLSS 활성화를 위한 엔진 코드 수정.**

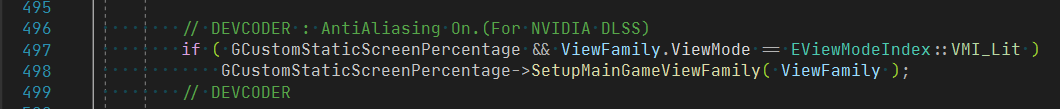
(2)번까지 설정을 하면 일반적인 UE4 실행시에는 DLSS가 작동 하지만, nDisplay 실행시에는 작동하지 않는다. 그 이유는 PostProcessin.cpp 파일 AddPostProcessingPass() 함수 내 511 라인부터 523라인까지의 코드인 다음 코드에 이유가 있다.



첫번째 빨간박스의 내용인 GetTemporalUpscalerInterface()가 nullptr 값을 반환하여 두번째 빨간박스의 내용인 Condition 항목이 false가 되어 Third Party Upscaler가 등록되지 않아서 이다.

View.Family->GetTemporalUpscalerInterface()에 유효한 인스턴스가 존재해야 하는데, ViewFamily에 TemporalUpscaler를 설정 하는 코드가 nDisplay 관련 코드에 존재 하지 않는다.

그래서 nDisplay시 사용되는 UDisplayClusterViewportClient::Draw() 함수 내 497 라인에 다음 코드를 추가 하여 ViewFamily에 DLSS Upscaler 인스턴스가 설정 되도록 해 준다.



**2. nDisplay Rendering Pipeline에 Motion Blur 기능 추가.**

**(1) 개요**

nDisplay는 한 화면을 여러 화면에서 나누어서 렌더링하여 각 화면에 출력해 주는 기능인데, 이렇게 여러화면에 나누어 렌더링 하다 보니 화면간 화면 갱신 주기가 맞지 않아 Screen Tearing 현상이 발생하게 된다.

특히 Screen Space 기반 Post-Process 결과물은 더욱 더 Screen Tearing 현상을 두더러지게 나타나게 하는 요인으로 작용하게 되어 nDisplay 사용시 사용을 권장하지 않고 있다.

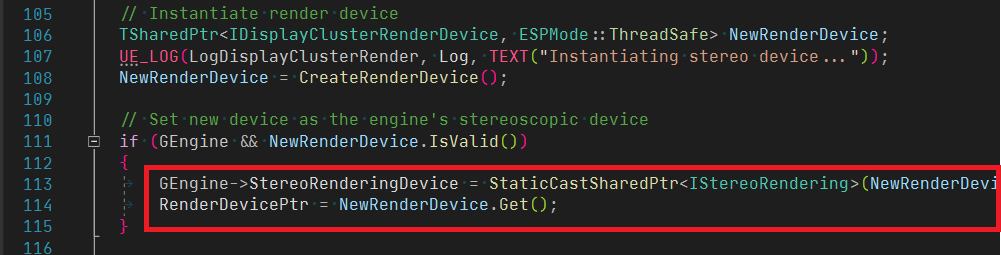
하지만, 여러 화면에 나뉘어 렌더링 하지 않고, 하나의 화면에서 In Camera VFX 기능만 이용하고 이때 Screen Space 기반 Post-Process 효과 중 하나인 Motion Blur을 사용하고자 할 때 Motion Blur 기능이 작동 하지 않는다.(UE4 4.26.1에서)

그래서 Screen Tearing 현상이 잘 발생하지 않는 구성에서 In Camera VFX 기능만 이용하는 프로젝트에서 Motion Blur 기능을 사용하고자 할 때 Motion Blur 기능이 작동 하도록 하기 위한 기능을 UE4 엔진 코드에 추가 하는 방법을 설명 한다.

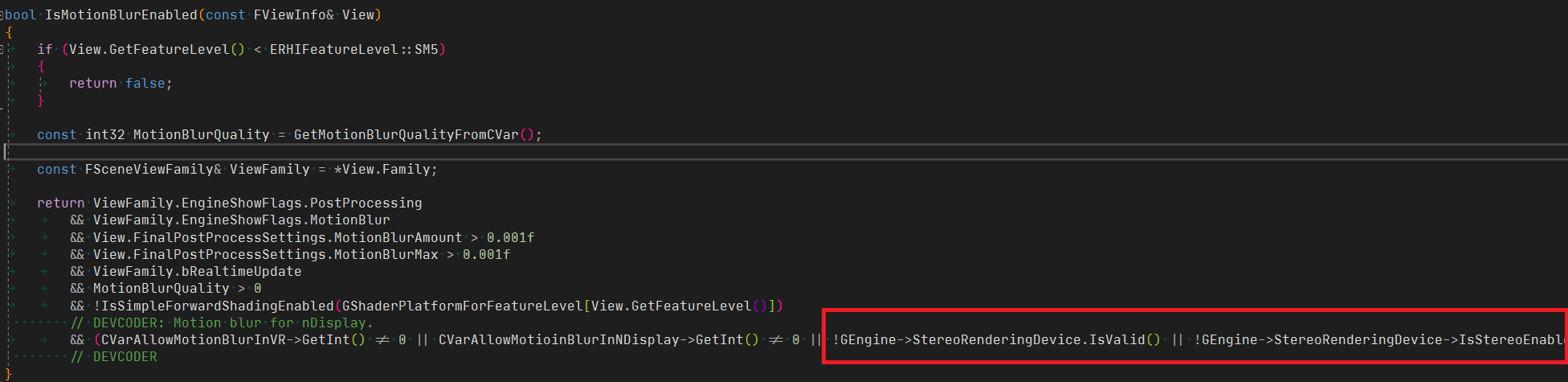
**(2) Motion Blur 기능이 활성화 되도록 엔진 코드 수정**

nDisplay에서 Motion Blur가 작동하지 않는 원인은 nDisplay는 UE4의 StreoRenderingDevice를 이용하여 Unreal 렌더링 시스템과 연동 되고 있다는 것이다.

다음 코드가 UE4의 StreoRenderingDevice에 DisplayClusterRenderDevice를 설정하는 코드이다.

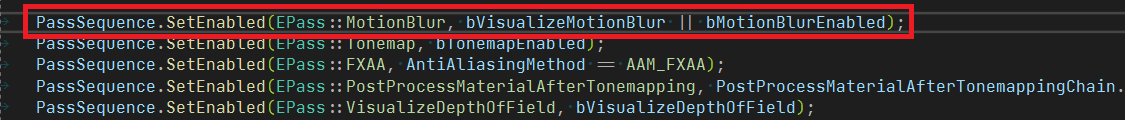


UE4 PostProcess 단계에서는 StreoRenderingDevice 참조 포인터가 유효하면 PostProcess 단계에 Motion Blur Pass를 추가 하지 않는다.



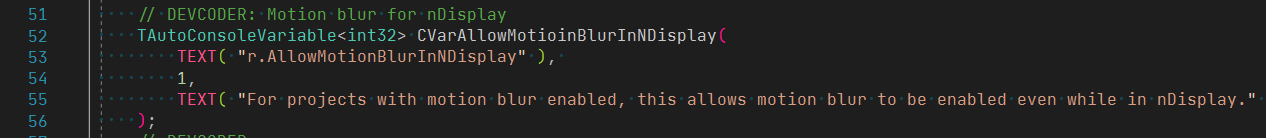
빨간색 박스 영역이 GEngine->StereoRenderingDevice 유효성 검사 및 Stereo Rendering 활성화 여부를 체크하는 부분이다.

위 코드가 true이면 PostProcess에 Motion Blur Pass가 추가가 되나, 그렇지 않다면 Motion Blur Pass가 추가 되지 않느다.

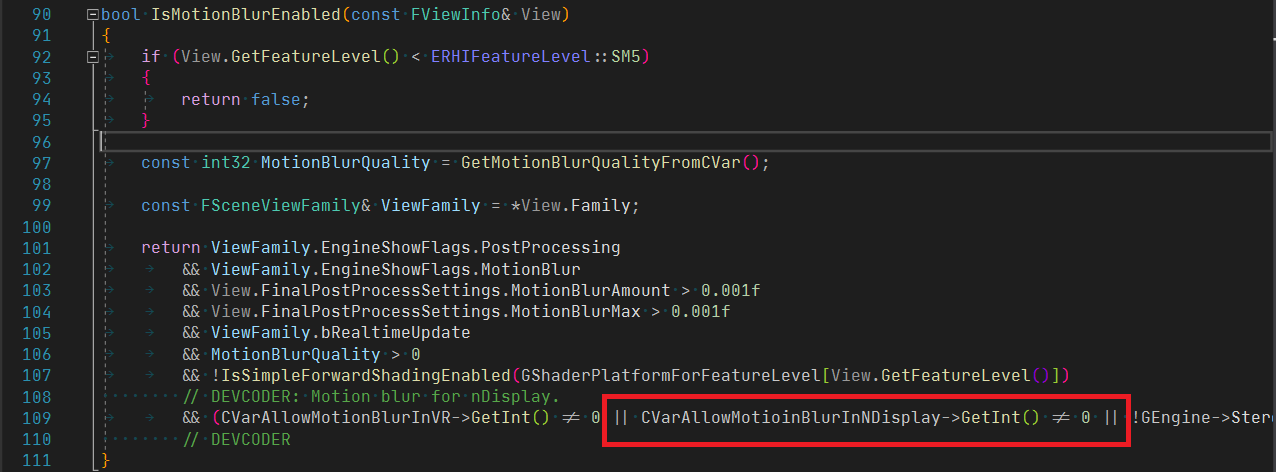


그래서 nDisplay 실행시 Motion Blur Pass를 PostProcess 단계에 추가 하기 위해 UE4 엔진 코드에 몇가지 코드를 추가 하였다.

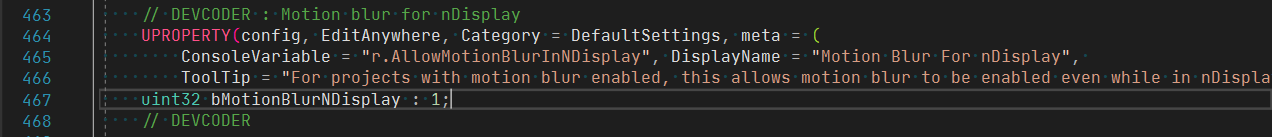
우선 nDisplay 실행 중 Motion Blur 사용유무를 조정 할 수 있는 옵션 설정 항목을 추가 하였다.



PostProcessMotionBlur.cpp 파일의 51 라인에 TAutoConsoleVariable<int32> 항목을 추가 하고, 이 항목 값을 통해 Motion Blur Pass를 추가 할지 결정 되도록 PostProcessMotionBlur.cpp 파일의 90 라인 bool IsMotionBlurEnabled() 함수에 TAutoConsoleVariable<int32> 변수를 추가 했다.



그리고, CVarAllowMotionBlurInNDisplay 항목의 값 설정을 프로젝트 설정에서 설정 할 수 있도록 하는 코드를 RenderSettings.h 파일의 463 라인에 추가 하였다.



RenderSettings.h에 코드를 추가하게 되면 프로젝트 설정에서 다음과 같이 나타나게 된다.

