□ 개요

○ 애셋

. 커브 데이터 : **TimeOfDayAsset**, **TimeOfDayPreset**, **TimeOfDayWeatherAsset**, **TimeOfDayWeatherPreset**

○ 시간 계산 & 라이팅/웨더 애셋 선택

. **TimeOfDayTime** : 시간 계산(로컬, 서버), TimeZone 에 따른 지리/물리 기반 Sun&Moon 계산.

. **TimeOfDayEnvironmentSource** : 라이팅 애셋, 웨더 애셋을 실시간 선택 및 세팅.

○ 그래픽 요소

. **TimeOfDaySky** : 스카이 박스. 해, 달, 별, 하늘색 등을 표현.

. **TimeOfDayVolumetricCloud** : 3D 클라우드.

. **TimeOfDayRainSnow** : 나이아가라를 활용한 비, 눈 표현.

. **TimeOfDayThunderLightning** : 천둥, 번개.

. **TimeOfDayBackLight** : 로컬 캐릭터 하이라이트 연출.

. **TimeOfDayWind** : 바람 제어.

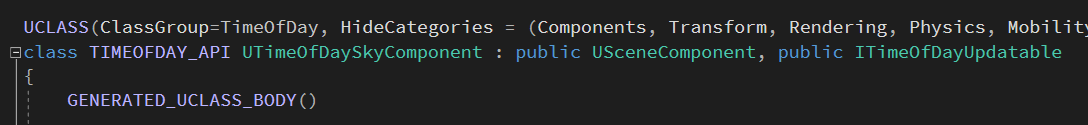
○ 커브 데이터 보간 및 적용

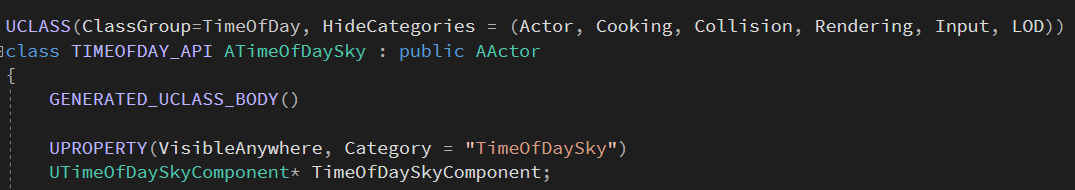
. **TimeOfDaySequencer** : 라이팅/웨더 애셋을 이용 시간별로 프로퍼티 값들을 보간하여 적용.

○ 위 기능들을 통합 관리 : **TimeOfDayWorld**.

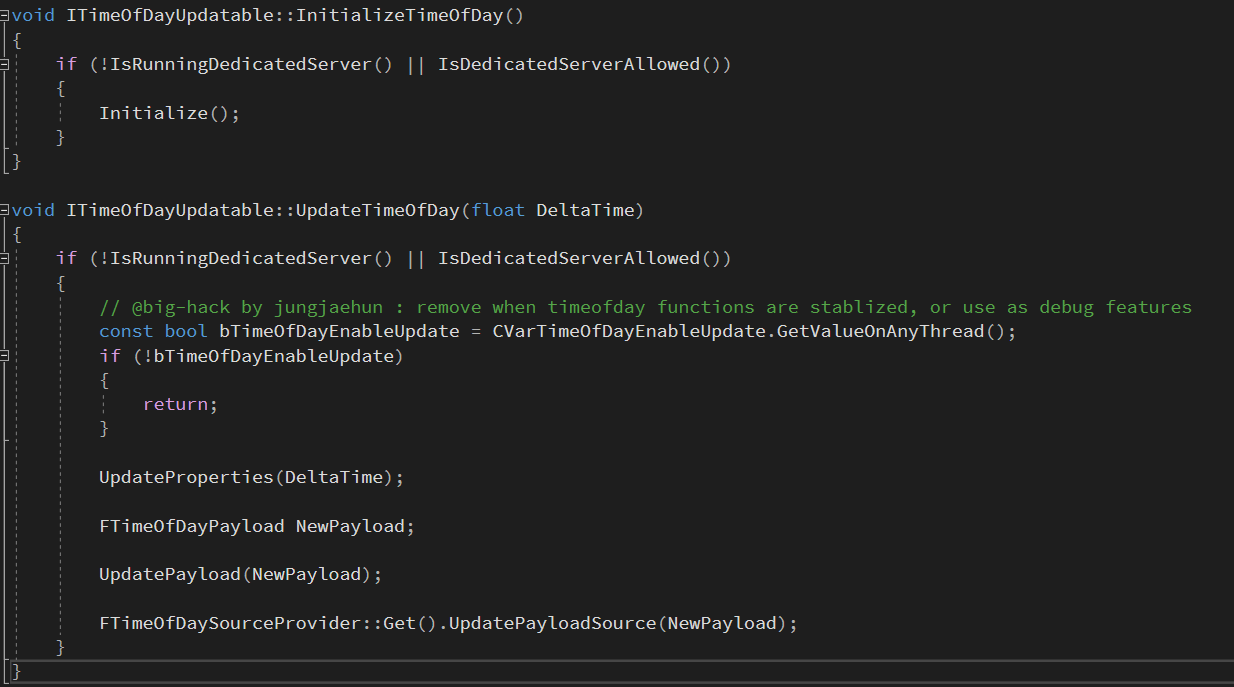
□ 모듈화

○ 컴포넌트 기반으로 구현하고 액터를 제공.



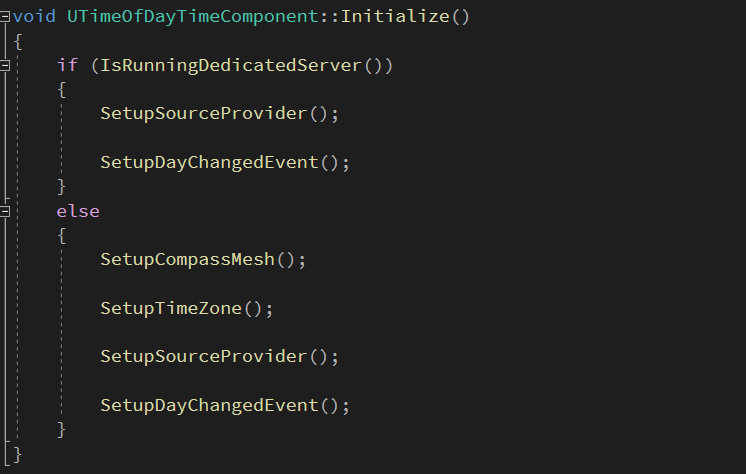


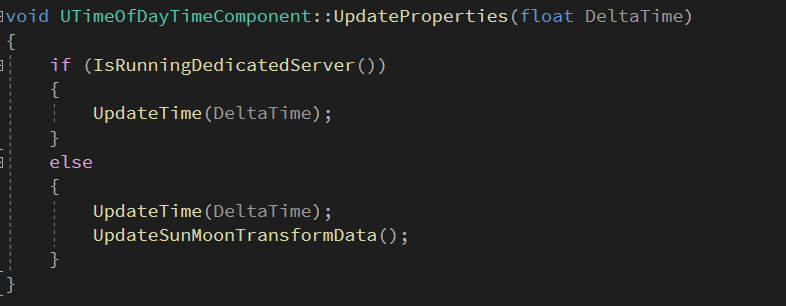
○ 컴포넌트는 ITimeOfDayUpdatable 을 상속받아 인터페이스 규칙에 따라 작동.

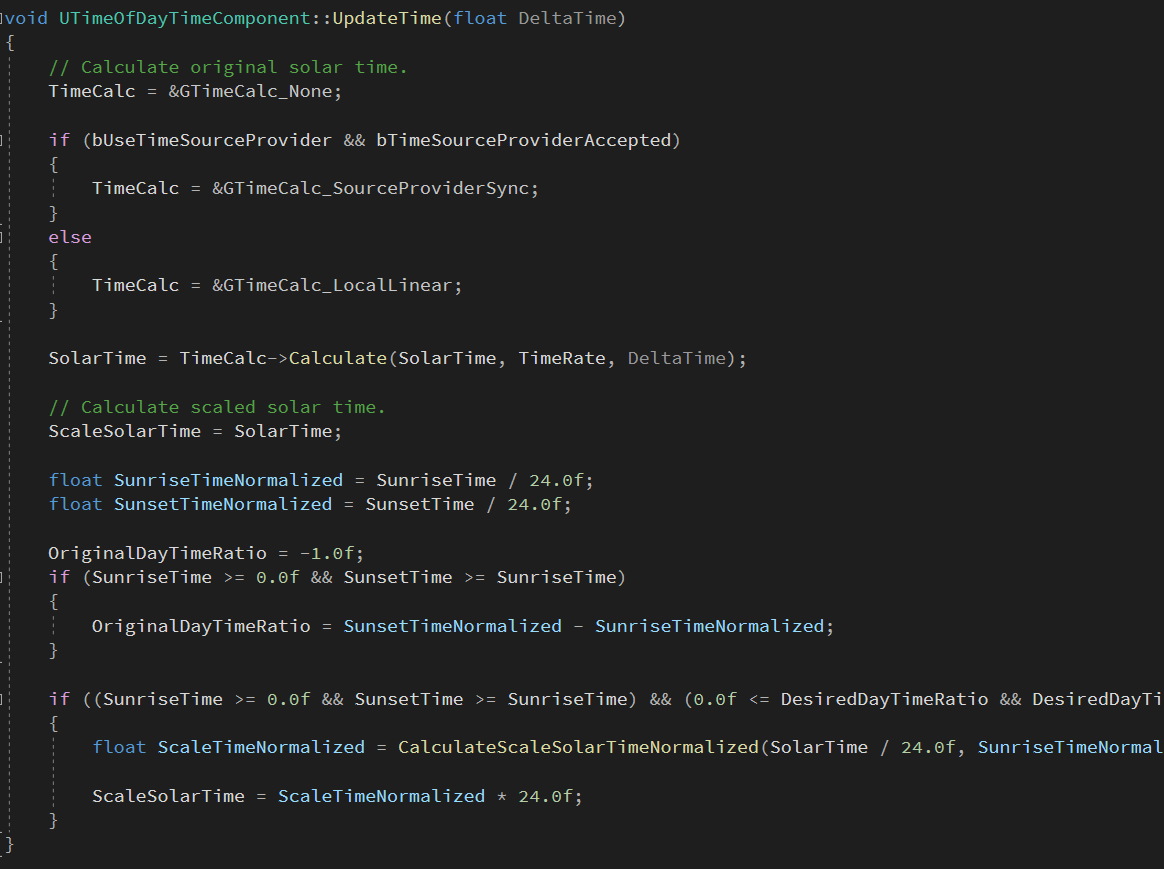


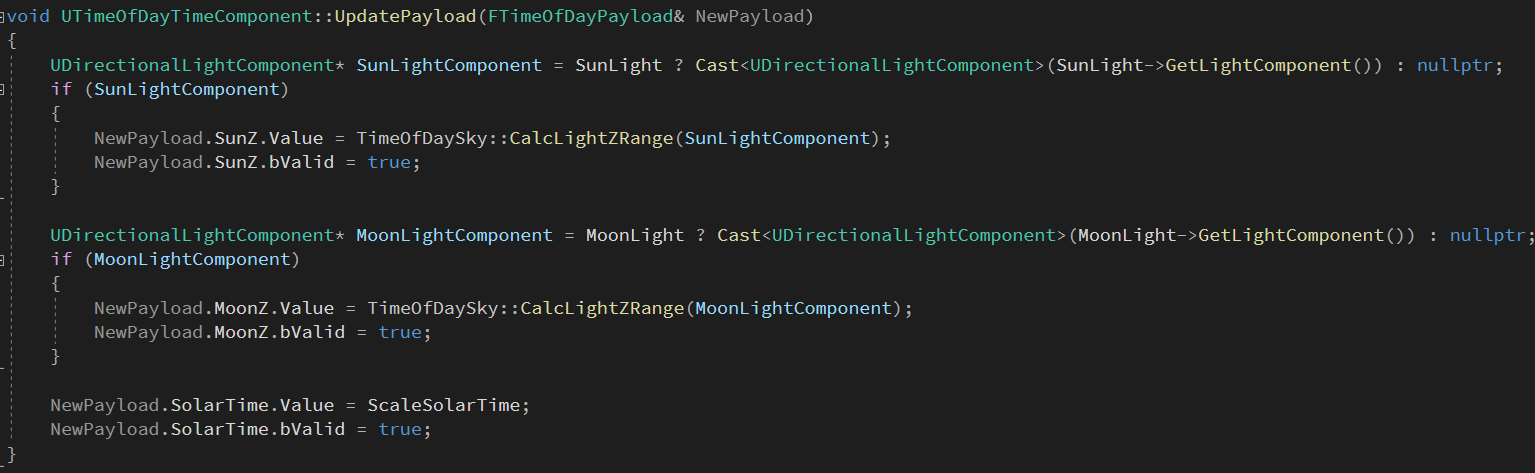
. 서버 허용 유무에 따라 Enable/Disable 되고, **‘초기화 -> 프로퍼티 업데이트 -> 페이로드 업데이트 -> 페이로드 전달’** 의 과정을 거침.

○ TimeOfDayTimeComponent 는 Initialize() 에서 해/달 위치 표현을 위한 컴파스 메시 설정, 물리 시간 존을 위한 설정, 하루가 바뀌었을 때 이벤트를 보내기 위한 콜백 설정 등을 수행하고, 매 프레임 UpdateProperties() 에서 시간을 계산하고 해와 달의 트랜스폼 정보를 갱신.



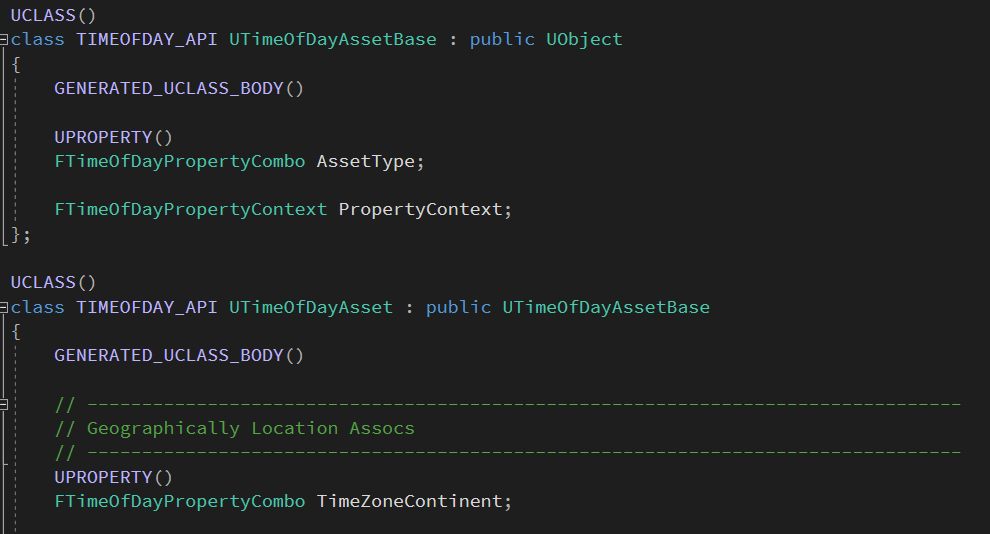




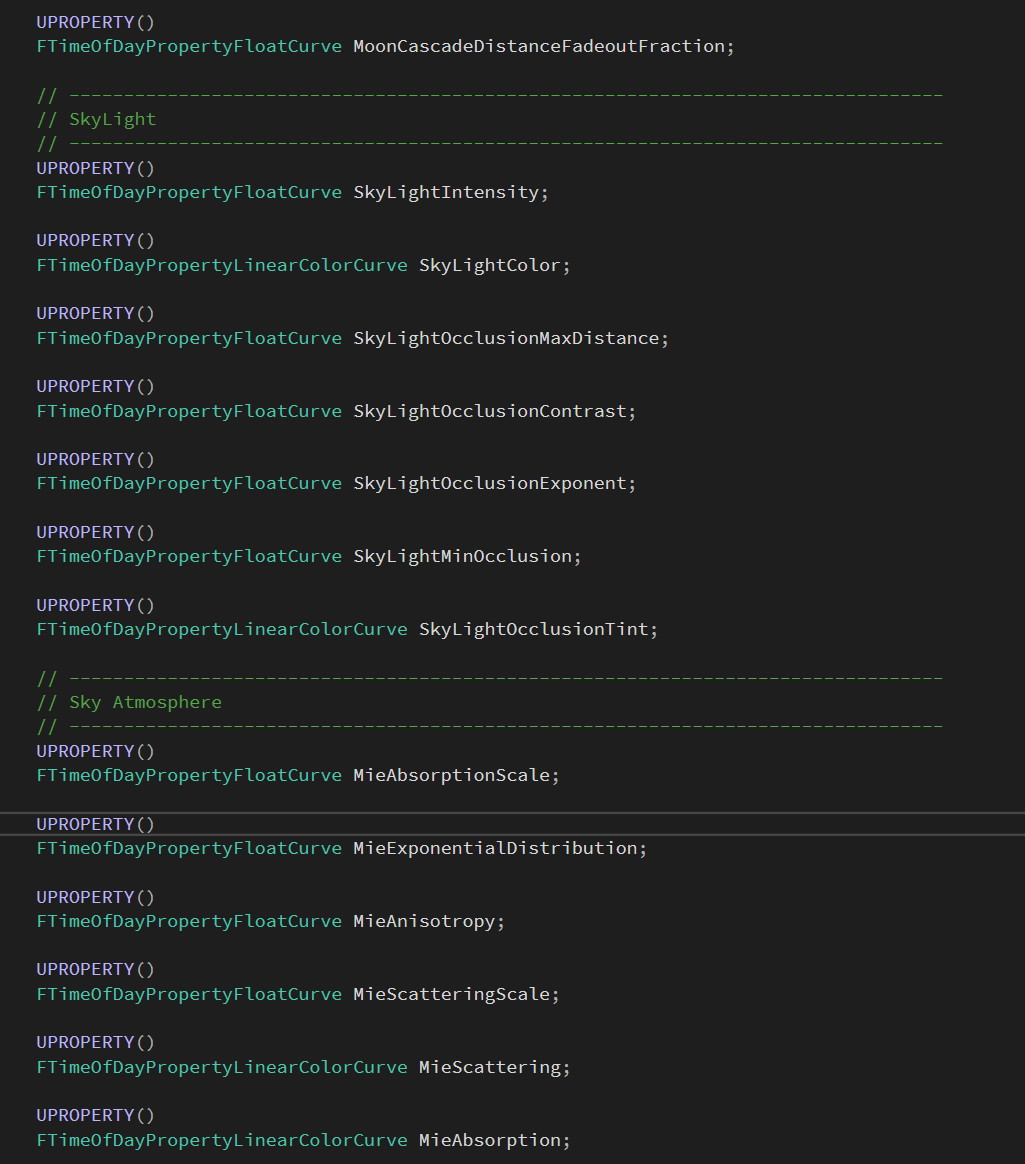


□ 라이팅 애셋

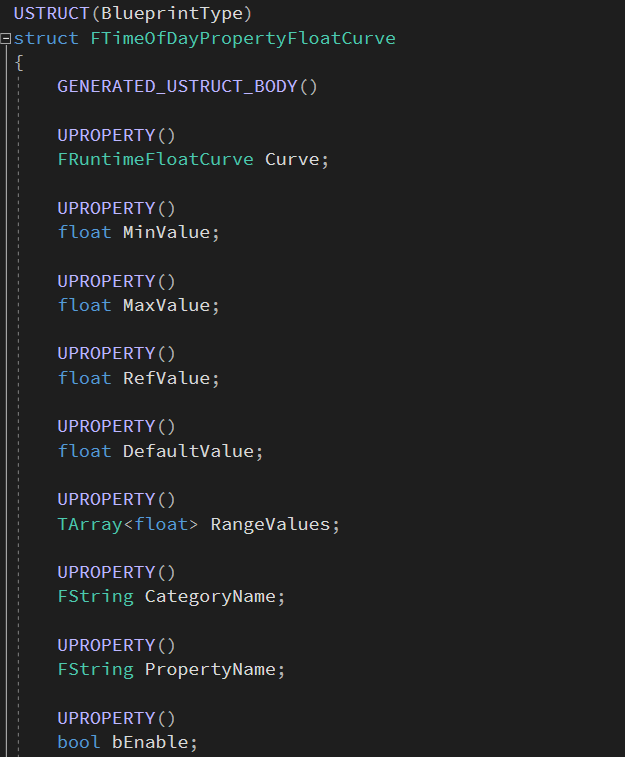
○ UTimeOfDayAsset : 하루의 시간 동안 변하는 값을 저장하기 위한 애셋.



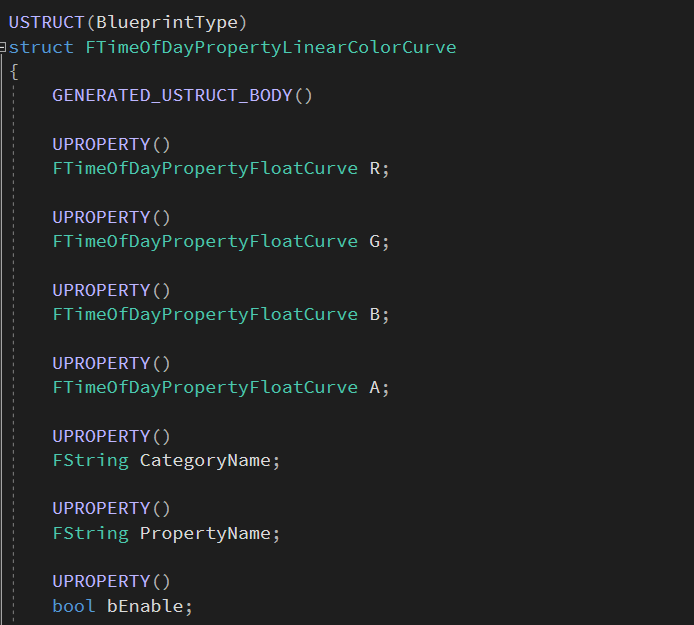
○ 애셋 데이터.



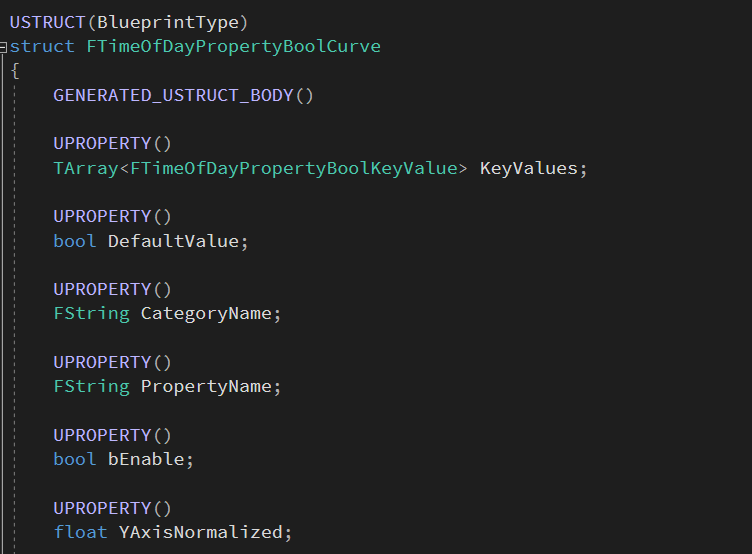
○ FTimeOfDayPropertyFloatCurve : float



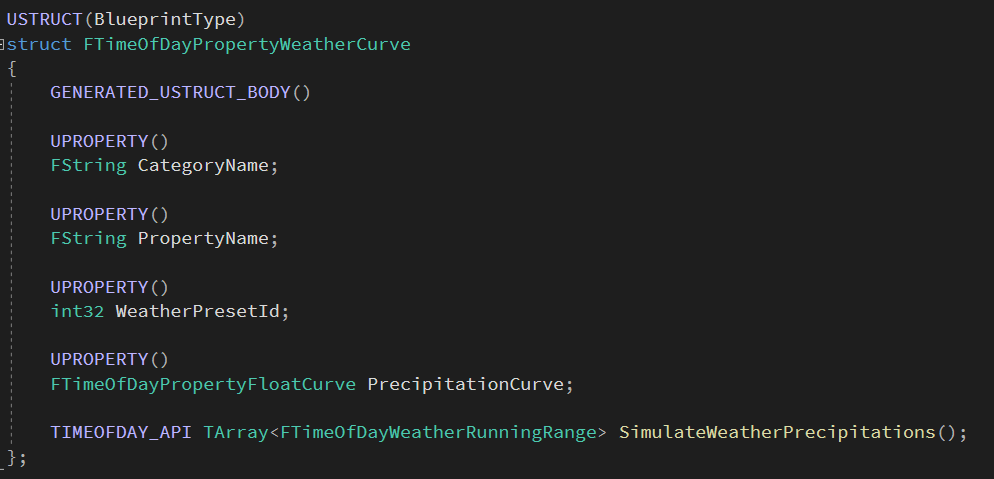
○ FTimeOfDayPropertyLinearColorCurve : Color



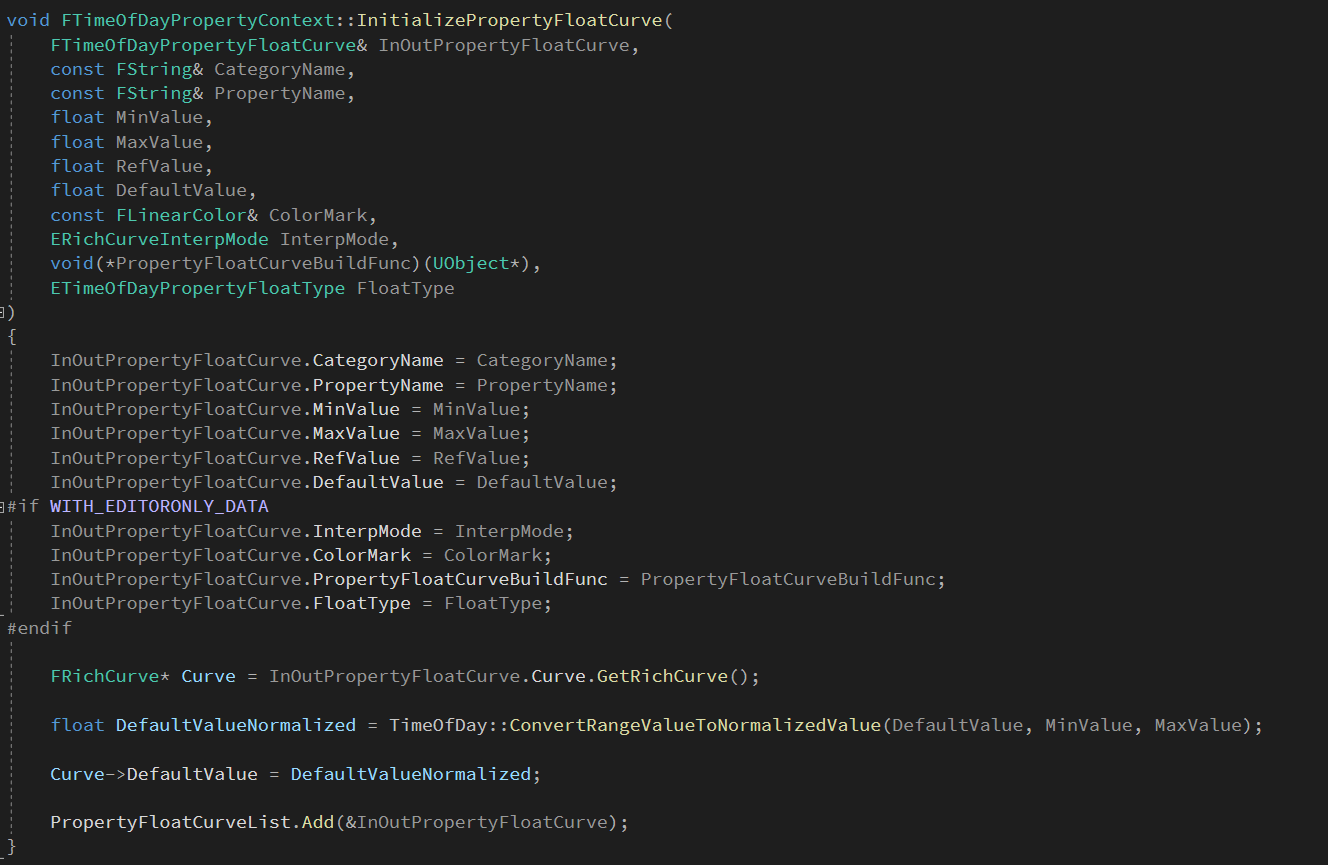
○ FTimeOfDayPropertyBoolCurve : 0, 1.



○ FTimeOfDayPropertyWeatherCurve : 격자 모양의 커브 사용(ERichCurveInterpMode::RCIM\_Constant).



○ PropertyContext : FTimeOfDayPropertyXXXCurve 들을 초기화, 저장, 관리.



□ 애셋 커브 값 적용

○ 애셋의 커브 데이터 적용.



○ 스카이라이트의 예.

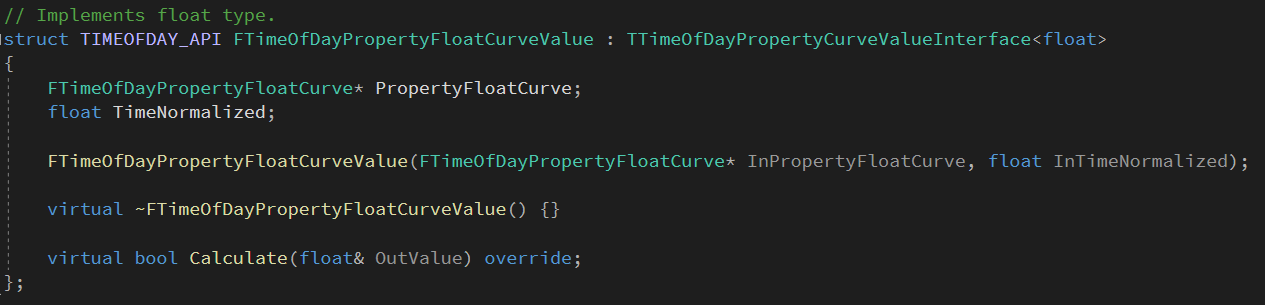


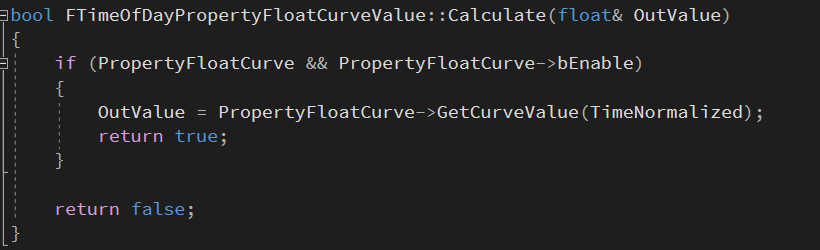
□ INTERP\_CURVE 매크로

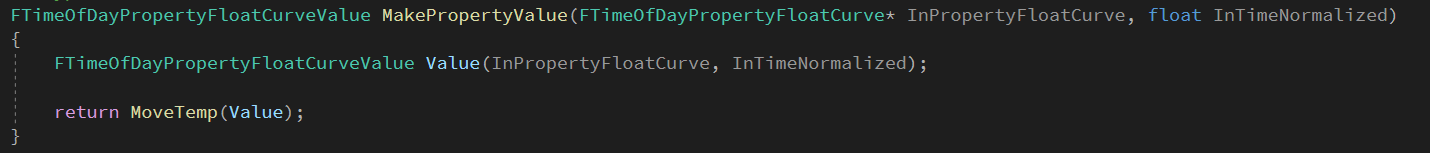
○ INTERP\_CURVE 매크로는 다양한 보간 방법을 처리하기 위해, Wrapper 인터페이스 객체를 사용.



○ 단일 Value 객체. 특정 시간대의 값을 커브 데이터에서 가져옴.

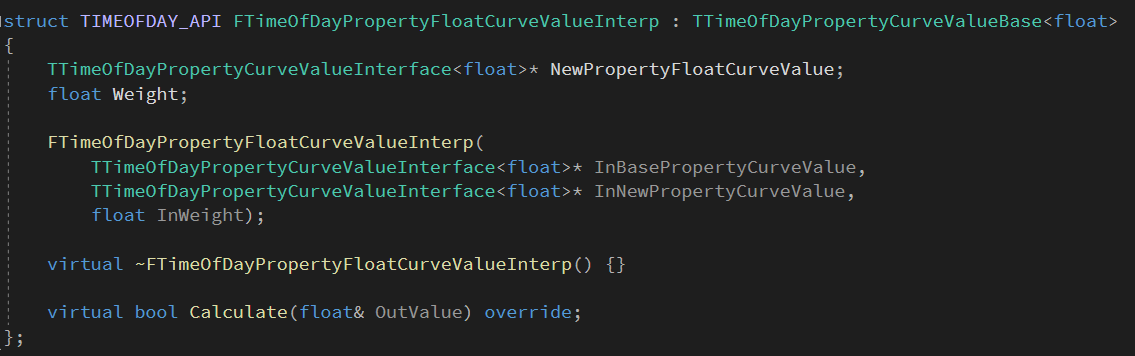


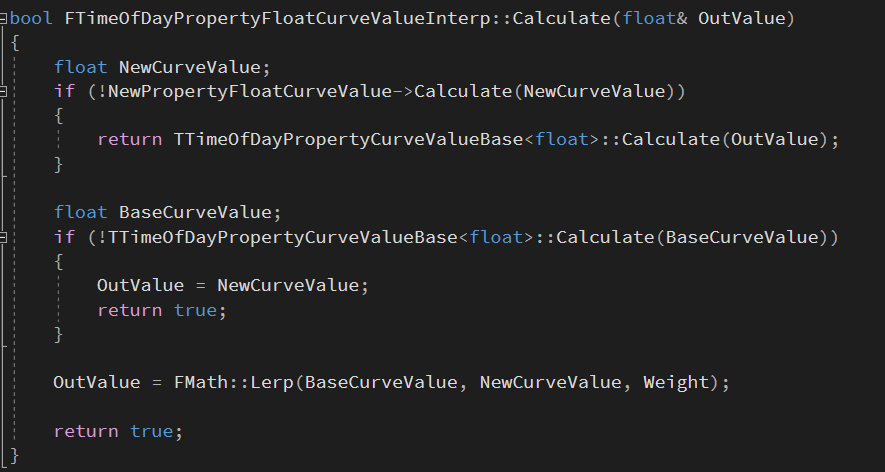


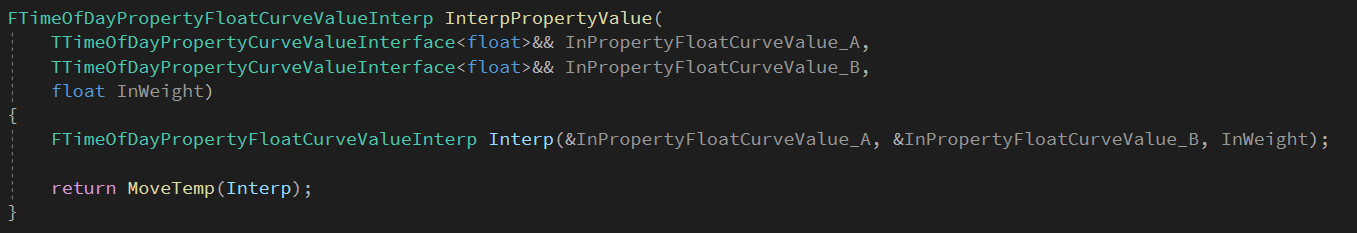


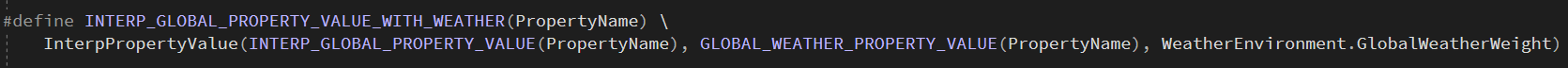


○ 두 Value 객체의 값을 단순 선형 보간

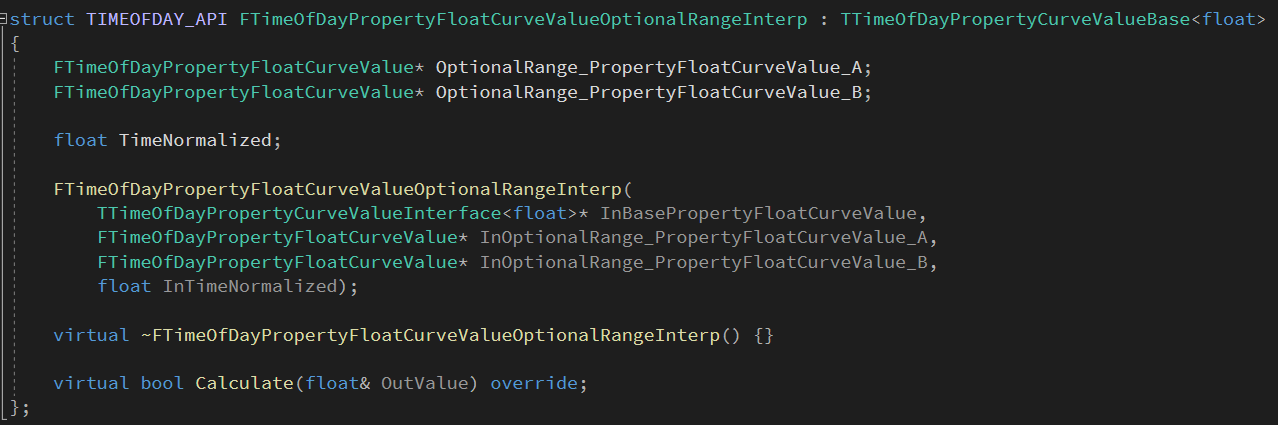


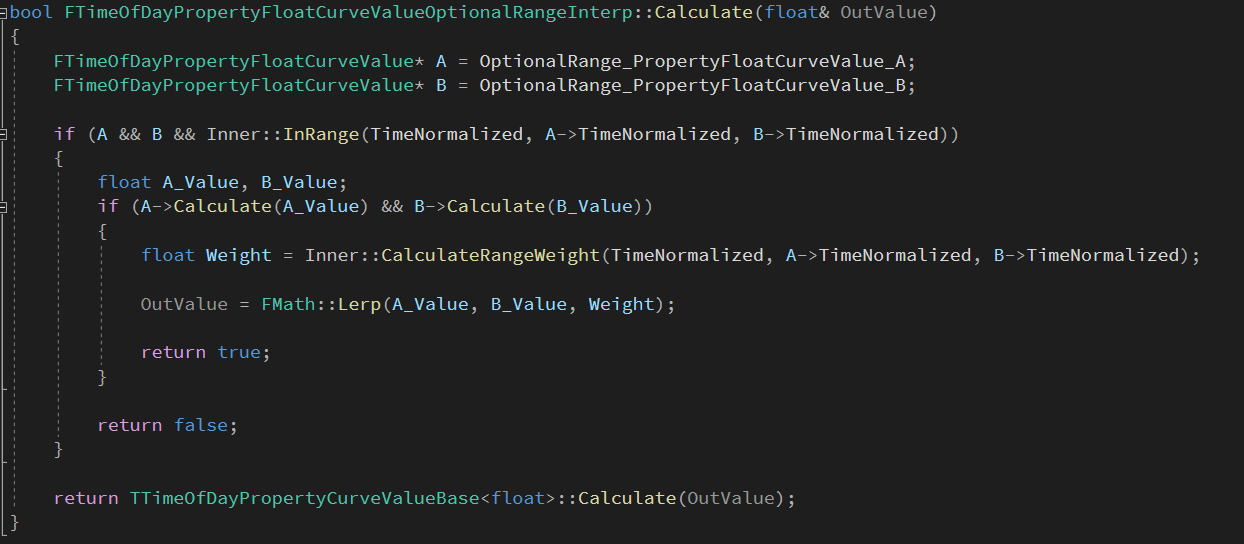


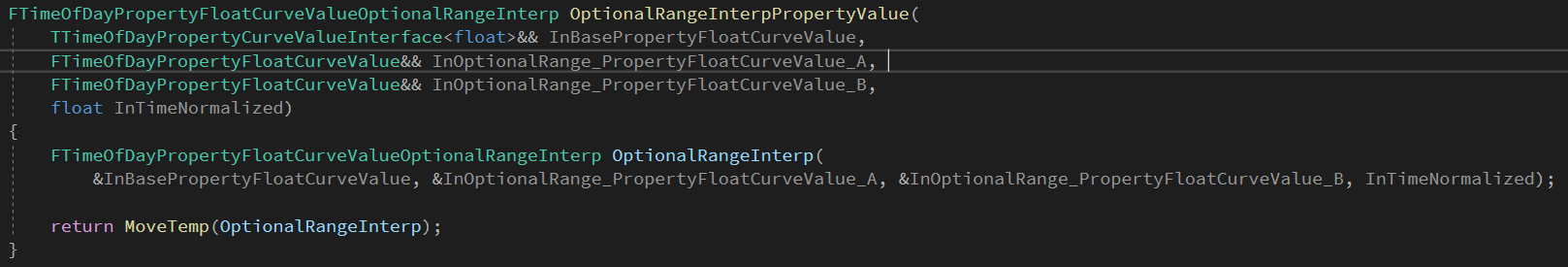




○ 특정 시간대만 단순 선형 보간.







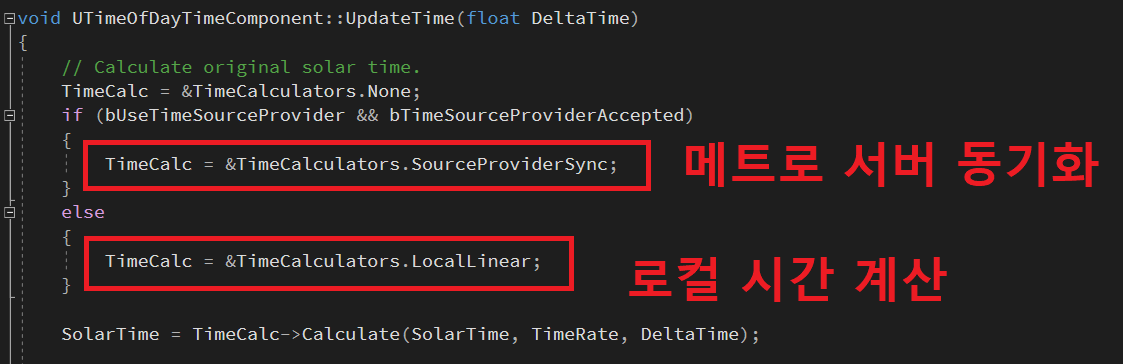


○ 기타 보간 : LinearColor, bool 타입에 대한 구현도 위와 같이 처리되어 있음.

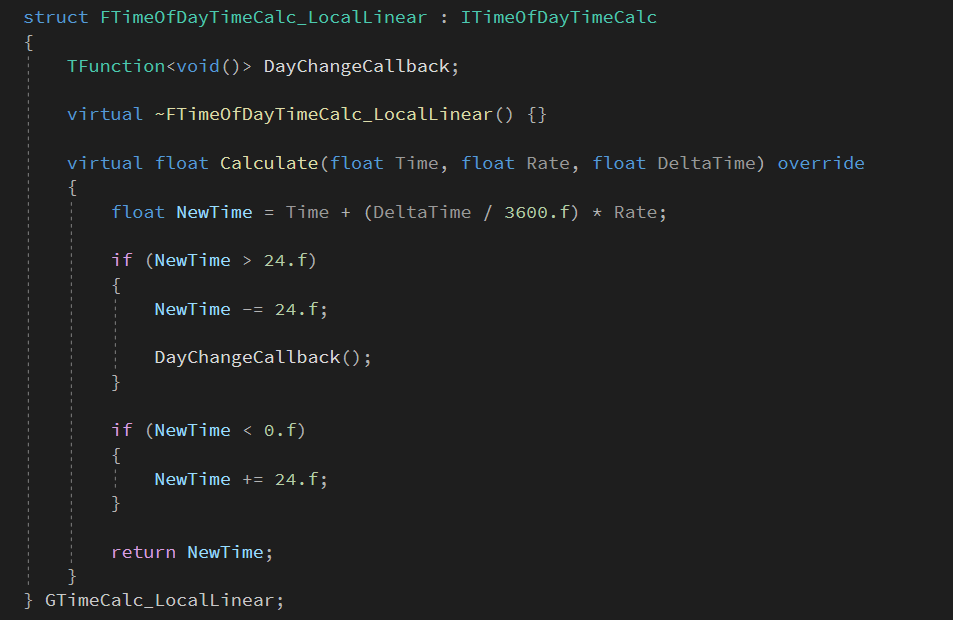
□ Time 업데이트

○ 에디터 : 로컬 시간 자체 계산.

○ 패키징 게임 : 메트로 서버 시간 동기화.



○ 로컬 시간 계산 : 단순 Linear 업데이트



○ 메트로 서버 동기화 : Source Provider 를 경유하는 서버 시간으로 동기화

