0.1.2 2023/12/12

- Ctrl키 누르고 링크 누르면 이동

[카메라](#카메라)

[쉐이더](#쉐이더)

[게임 오브젝트 만들기](#게임오브젝트만들기)

[.spt파일](#spt파일만들기)

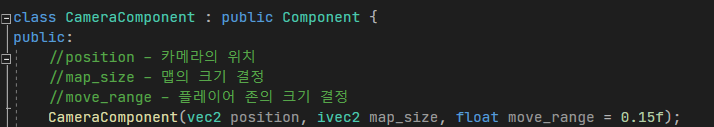
[폰트](#폰트)

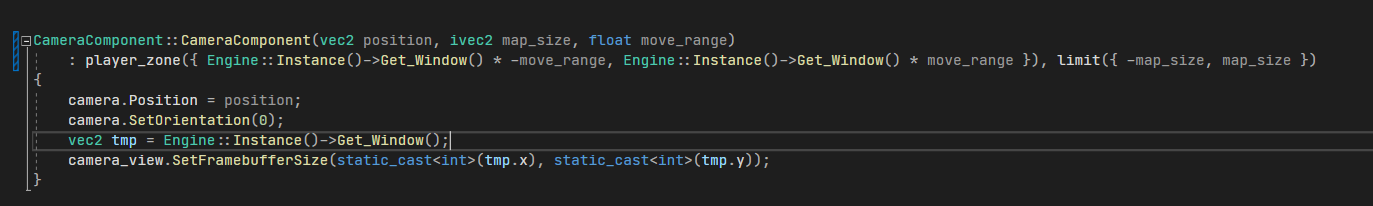
[백그라운드](#백그라운드)

[충돌](#Collosion)

[스킬](#Skill)

카메라





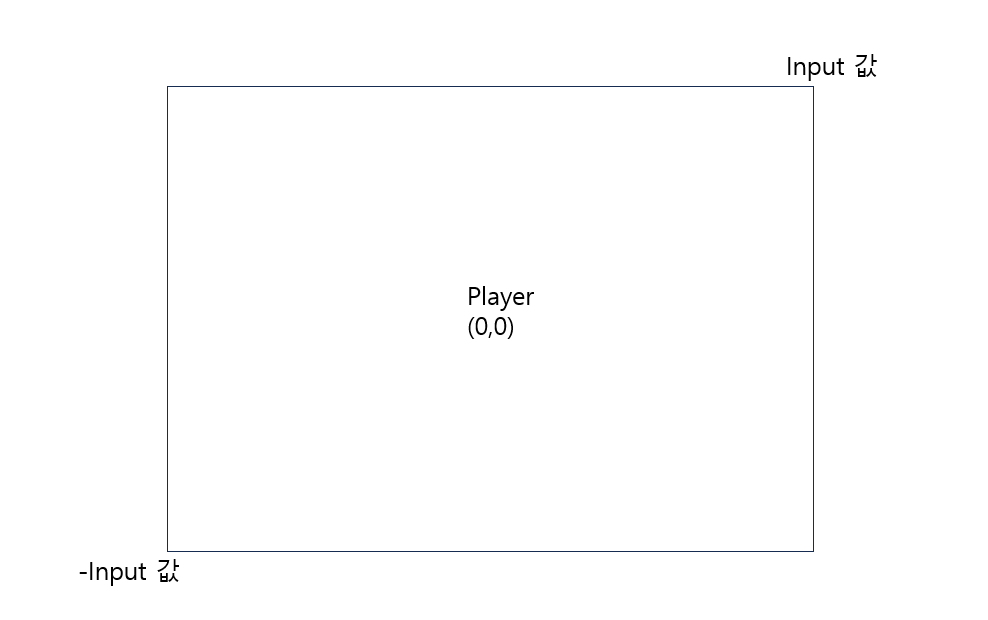
카메라 생성자

Position – 카메라의 처음 시작 위치

그냥 기본적으로 0,0에서 시작하게 하는게 그냥 무난함

Map\_size – 이 카메라가 쓰이는 state의 맵 크기

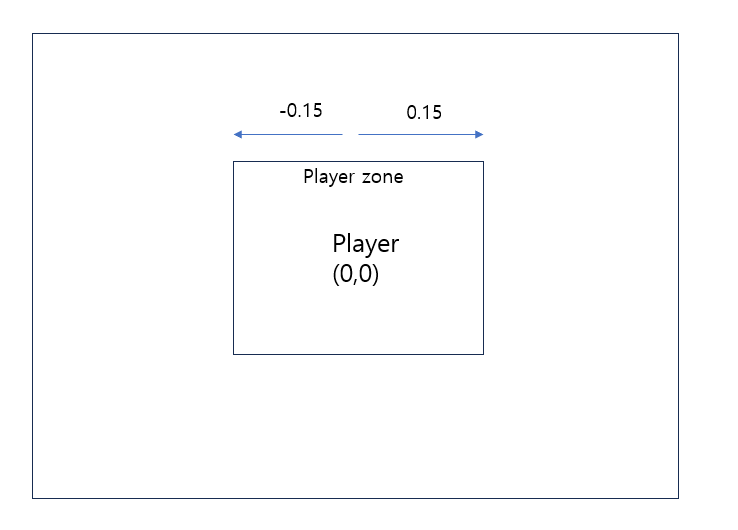
이 때 ivec를 하나 받는데 이 값이 이런식으로 쓰임



Book1.cpp 를 보면 인풋으로 window size/2 를 넣는데, (640, 360)이니까 카메라의 최대 갈 수 있는 범위가 (+-640, +-360)이 됨. 근데 이게 카메라의 범위임. 플레이어가 갈 수 있는 범위랑은 다른 개념

Move\_range – 플레이어 존 설정

이건 기본값으로 0.15가 설정되어 있는데



이런 식으로 쓰임 저 안에서는 움직여도 카메라가 딱히 따라오지 않는다는 거. 대충 알거임.

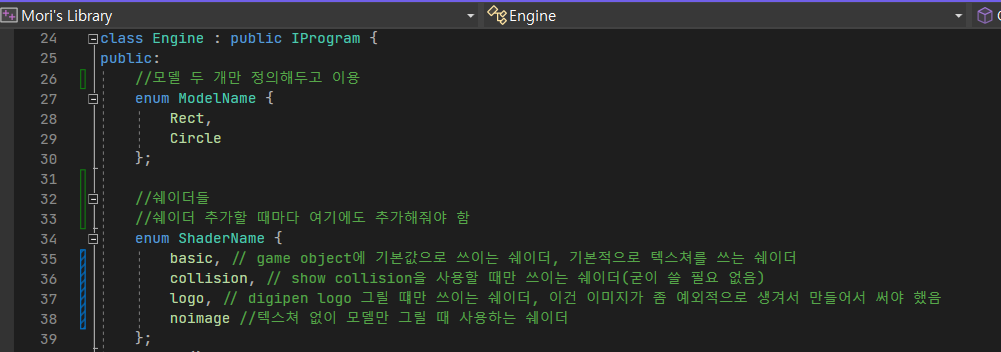
물론 카메라가 맵의 끝까지 가서 더 가지 못하는 상황이라면 플레이어가 플레이어존을 벗어나도 카메라는 못 따라옴.

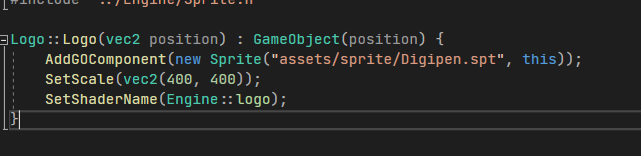
어쩃든 저 범위를 늘리고 싶으면 인수로 값을 넣으면 된다. 기본값으로는 0.15가 들어가 있는거고

쉐이더

오브젝트마다 쉐이더는 똑같이 생겼는데 많아서 통합. engine.h 에서 쉐이더 관리하는 걸로 바뀜

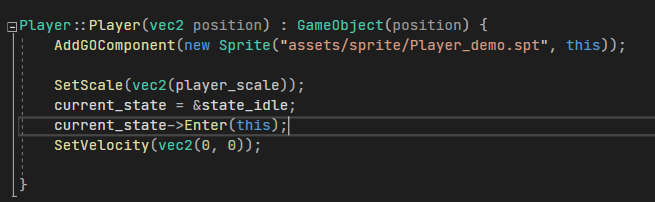
밑의 사진 주석 한 번 읽어보셈.



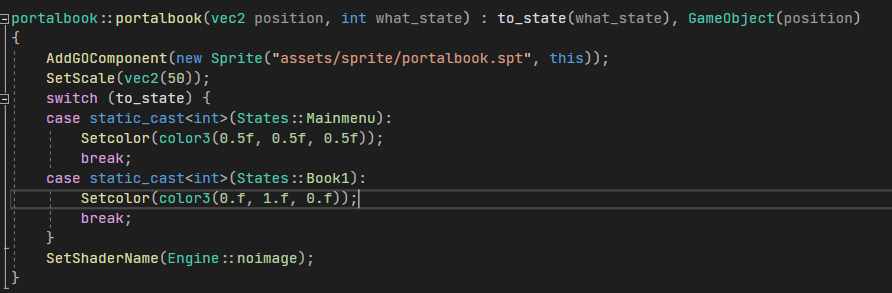


이렇게 쉐이더를 선택적으로 사용할 수 있게 됨. Sprite 생성할 때도 쉐이더 안 넣어도 됨.

그리고 기본값으로 basic이 되어 있음. 그래서 그냥 일반적인 오브젝트들은 쉐이더 지정을 따로 안 해줘도 됨. 아래 사진처럼



따로 안 해줘도 그림 잘 나옴.

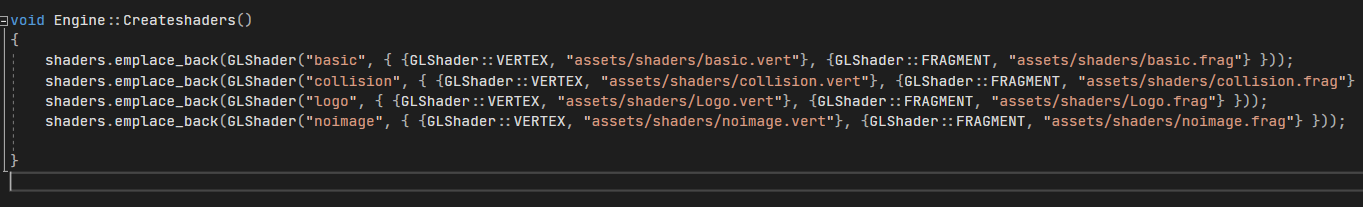


이거처럼 텍스쳐 안 쓰는거는 따로 쉐이더 지정을 해줘야 함

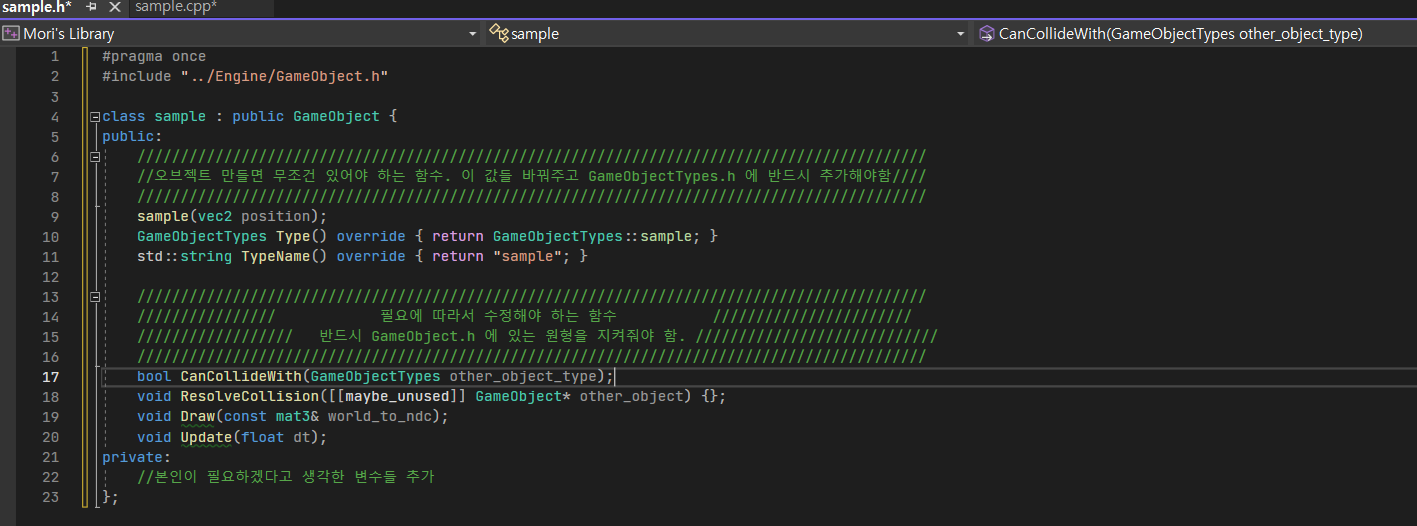
만약 쉐이더를 따로 만들어서 쓸 일이 생긴다면

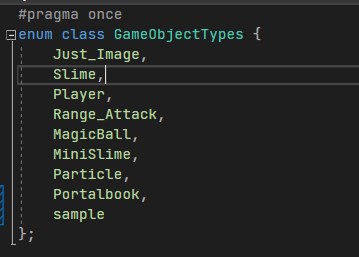
1. 첫 사진에 있는 enum에다가 쉐이더 추가하고

2. engine.cpp 제일 아래에 있는 이 함수에다가 추가해줘야 함.



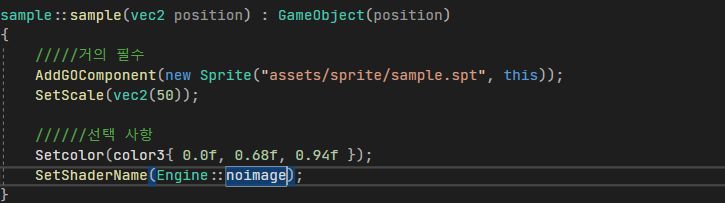
게임 오브젝트 만들기





게임 오브젝트 추가하면 GameObjectTypes.h에 가서 오브젝트 추가해줘야 함

생성자에서 반드시 해줘야 하는 코드

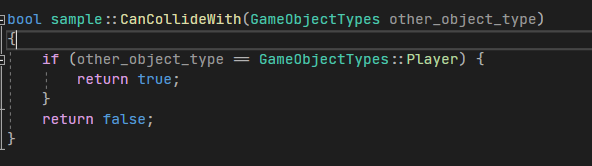


\* 우리 엔진은 .spt파일로 충돌도 만들고 draw할 수 있는 텍스쳐나 모델들도 불러오기 떄문에 반드시 sprite를 추가해줘야 한다. (물론 스프라이트 추가 안 해도 충돌은 넣을 수 있다.)

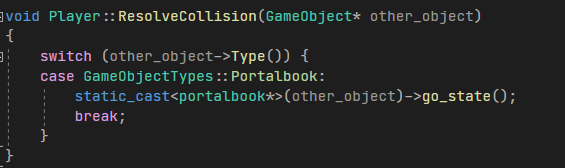
\* 그리고 이게 기본적으로 scale값을 1로 해놔서 scale을 키워줘야 한다.

\* 만약 필요하다면 Set\_Shader()함수로 다른 쉐이더를 지정해주면 된다.

\* 텍스쳐가 없을 때, 컬러도 Setcolor() 함수로 바꿔줄 수 있다.

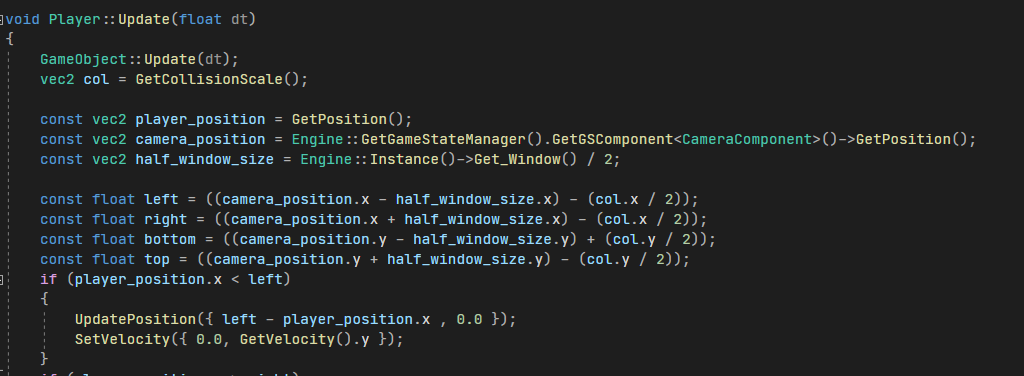


위의 코드처럼 쓰면 뭐랑 충돌판정할지 정할 수 있다.



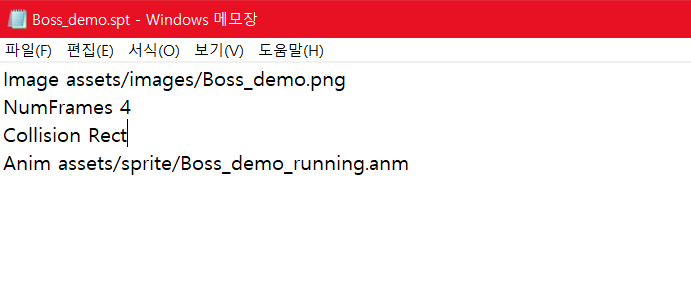
충돌이 일어났을 때, 어떤 일이 일어나게 할건지 정의하는 함수

예시는 플레이어랑 포탈북이랑 부딪히면 포탈북의 go\_state함수가 발동되어 게임 스테이트가 바뀌게 된다.



이렇게 따로 update를 정의할 때에는 첫줄에 보이는 GameObject::Update(dt)코드를 추가해주어야 스프라이트들이나 다른 컴포넌트들이 업데이트 되기 때문에 저걸 꼭 추가해줘야 한다.

.spt파일



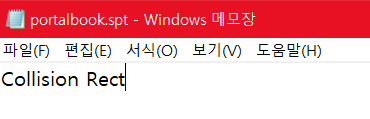
우선 Image 저걸로 어떤 이미지 불러올지 정하고

만약 스프라이트가 4개의 텍스쳐로 이루어져 있으면 NumFrames 값을 그에 맞게 설정

충돌판정은 Collision 키워드 뒤에 Rect or Circle을 적어준다

애니매이션은 저렇게 넣어두면 되고

만약 Image가 없는 그냥 네모만 그리고 싶을 경우

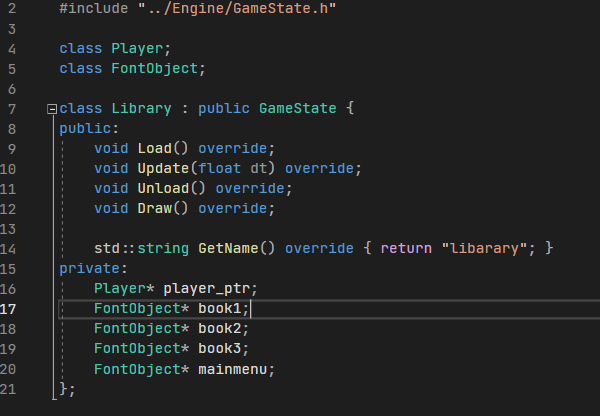


이렇게만 넣어줘도 된다.

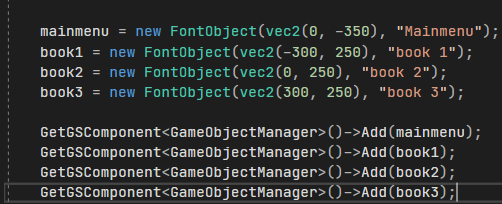
자동으로 Numframe 은 1이고 animation은 None.anm이 추가되고 Image 도 nullptr로 지정된다.

충돌도 굳이 안 넣어도 된다면 안 넣어도 된다.

폰트

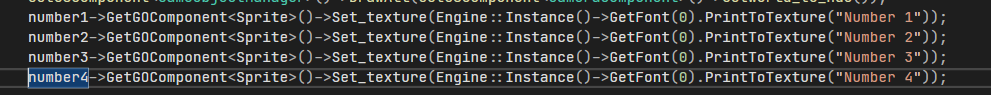


폰트를 띄울려면 위의 사진처럼 각 스테이트에다가 변수를 만들고



이런식으로 생성할 떄, 넣고 싶은 텍스트를 넣는다

만약 실행중에 저 텍스트를 바꾸게 하고 싶으면

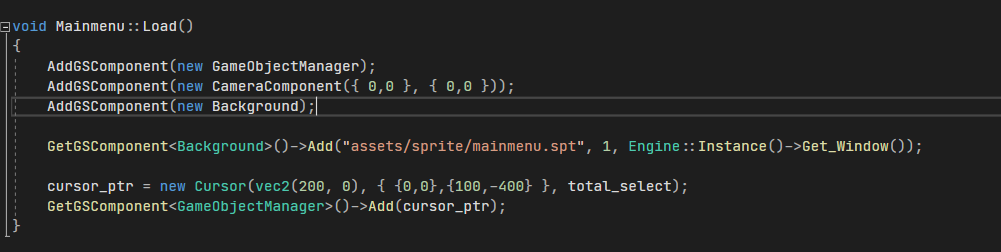


이런식으로 함수를 사용하고 넣고 싶은 텍스트를 넣으면 텍스트가 바뀐다.

물론 저기 들어있는 값에 std::string 값을 넣을 수 있다.

\* 한 번만 실행되도 바뀐다. -> draw나 update 같은 곳에 안 넣어도 됨. 저 함수 실행되는게 좀 길어서 왠만하면 별로 안 쓰는게 좋다.

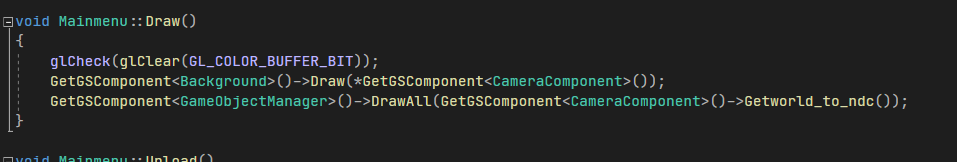
백그라운드



위의 사진처럼 게임스테이트 컴포넌트로 추가하고

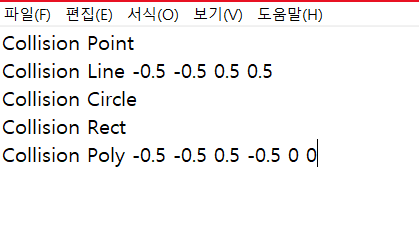
Add()함수를 통해서 (spt파일, 스피드(230엔진 생각하면 됨), scale) 이렇게 3개의 값 받는다.

참고로



draw()함수에 저렇게 따로 background 만의 draw함수를 써줘야 한다.

Collosion



.spt 파일에 충돌 넣을 때, 5가지 넣을 수 있게 됨.

근데 line 이랑 poly는 직접 가로 세로 -0.5 ~ 0.5 사이의 값 안에서 자신이 지정해줘야 함

저 예시를 보면 라인은 아래 처럼 나오고

(0.5, 0.5)

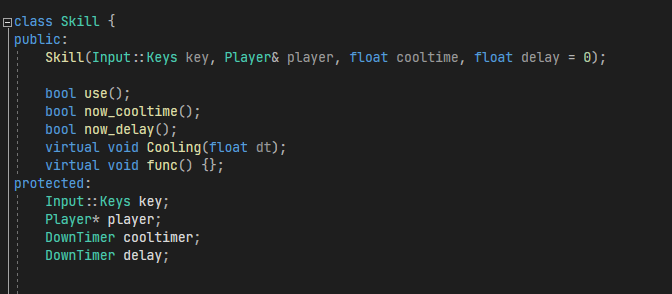
(-0.5, -0.5)

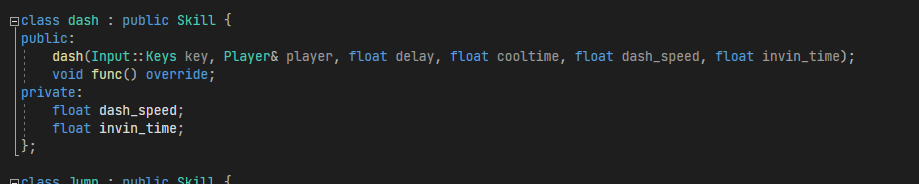
poly는

이런 식으로 히트박스가 생김

\* 폴리 히트박스 좌표 넣어줄때, 반드시 시계방향이나 반시계 방향으로 순서대로 넣어줘야 함.

Skill





새로운 스킬 만들 때

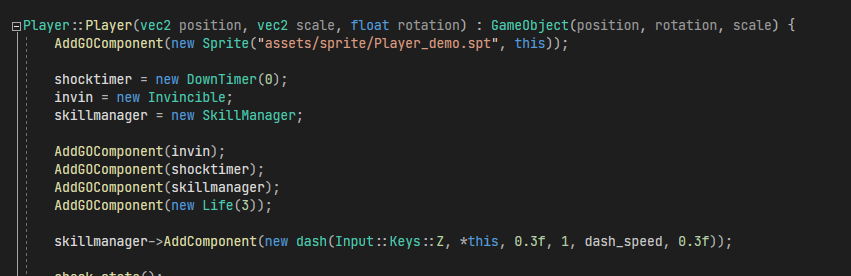
제일 기본적으로 넣어줘야 하는 정보가

1. Key : 무슨 키 누르면 실행할건지

2. Player : 스킬 쓰는 주체는 뭐 플레이어밖에 없으니 플레이어 포인터 받아와주고

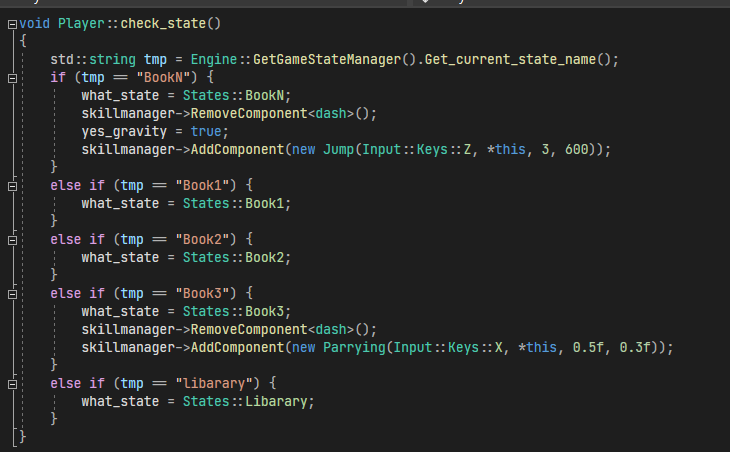
3. Cooltime : 스킬 쿨타임

4. Delay(선택사항) : 이거는 스킬 시전시간 예를들어 패링스킬을 실행하고 0.5초간 패링 상태에 들어가있는 거



위의 예시처럼 skillmanager에다가 스킬을 넣어서 쓰면 된다.

대쉬는 다 쓰기 때문에 플레이어 생성자에다가 넣어둠.



하지만 book3 에서는 대쉬를 안 쓰기 때문에 위의 예시처럼 대쉬를 제거하고 패링을 대신 넣으주면 됨. 만약 위의 같은 상황이 필요하면 이런식으로 하면됨.