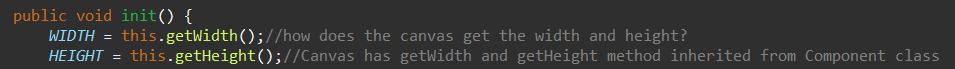
Window class

JFrame이 생성되는 곳.

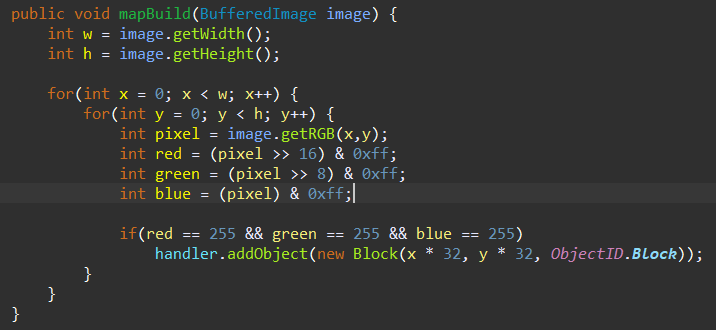
게임인스턴스가 여기서 start된다.

컨스트럭터만 존재한다. (고정관념을 버려라)

Game class – 컨스트럭터가 없다. (고정관념을 버려라)



mapBuild method

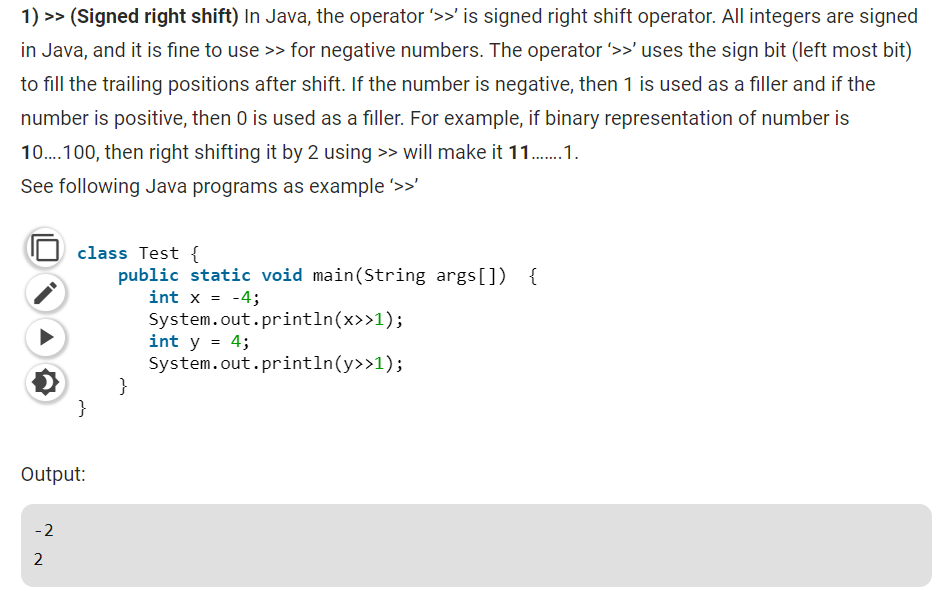


\*pass 된 이미지는 level.png로서 파일의 왼쪽 윗부분에 맵을 미리 그려놓았다. 각각의 하얀색 픽셀은 맵의 블럭을 의미한다.

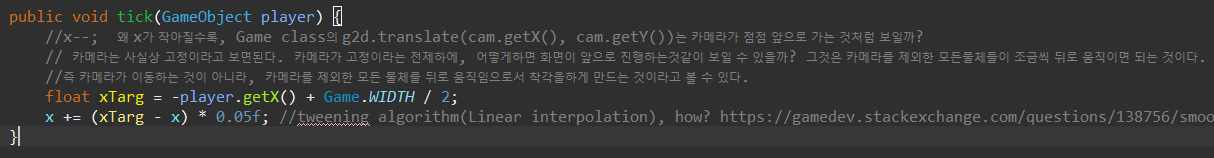
Bitwise operator

C나 C++ 에서는 negative number에 bitwise operator를 사용하는 것을 권하지 않는다. Compiler에 따라 응답하는 방법이 다르다.

밑에 설명에서는 자바에서는 negative number에도 bitwise operator가 사용가능하다고 한다. 숫자가 양수냐 음수냐에 따라서 filler가 (bitwise operator가 적용된 후 채워지는 수) 1이냐 0이냐가 정해진다는 말 같다. 확인해보기.



Camera class



0.05f가 1.0f라는 숫자가 주어지면, x = xTarg와 다름없다.

우선 위의 두 라인보다 x = -player.getX() + Game.WIDTH / 2;를 이해하자. 이 코드보다 업그레이드 된 것이 위의 두 라인이기 때문이다. 이 코드를 돌릴 경우, 카메라는 캐릭터를 화면에 중간에 (FIX)한채로 두려고 할 것이다.

x += (xTarg - x) \* 0.05f;는 무엇을 하는가?

\*위에서 말했든이 xTarg를 x로 바꾸고, x += (xTarg – x) \*0.05f를 지우면, 카메라는 고정된채로 캐릭터를 중간에두고 따라간다. 즉 xTarg는 최종적으로 캐릭터를 중간에 배치시키는 카메라의 x좌표값이라고 할 수 있다.

\* x += (xTarg - x) \* 0.05f; 에서 x +=를 통해서 우리는 이 코드가 x를 마치 x += velX; 처럼 점진적으로 카메라의 x좌표의 움직임을 변화시키려고 한다는 것을 알 수있다. 위에서 말했듯이, xTarg는 최종적으로 캐릭터를 중간에 배치시키는 카메라의 좌표이고 이것은 우리의 최종 Target X좌표라고 할 수 있다. 우리가 타겟으로 정한 x좌표, 즉 xTarg, 와 현재의 카메라 좌표의 차에 0.05f를 곱한 것을 현재의 카메라의 x좌표에 더함으로 카메라의 tick()이 업데이트 될 때 마다, xTarg – x는 점차 0에 수렴할 것이고 (0.05배 씩), 결국 우리의 x는 xTarg에 다다를 것이다. 이후에는 xTarg – x가 0이 되기때문에 0.05f를 곱해도 x의 변화는 없어진다.이렇게, 우리는 카메라가 캐릭터를 고정된 것이 아니라, 점진적으로 쫓아 갈 수 있게 만 들 수 있는 것이다.

\*2번째 설명의 증거로서, \* 0.05f 배제시키면, x += (xTarg - x)만이 남게된다. 즉 현재의 카메라의 x좌표에 타겟 x와 현재 카메라의 x의 차이를 한번에 더해버리는 것이다. 그러면 역시 고정된 카메라가 따라가는 것처럼 보이게된다. 0.05f라는 것은 점진적으로 따라가게 해주는 rate인셈

Handler/ Camera / Player의 관계

Handler class는 게임 오브젝트를 총괄관리한다. (업데이트 렌더링)



\*Why should we pass cam as an argument? Because it is not a GameObject.

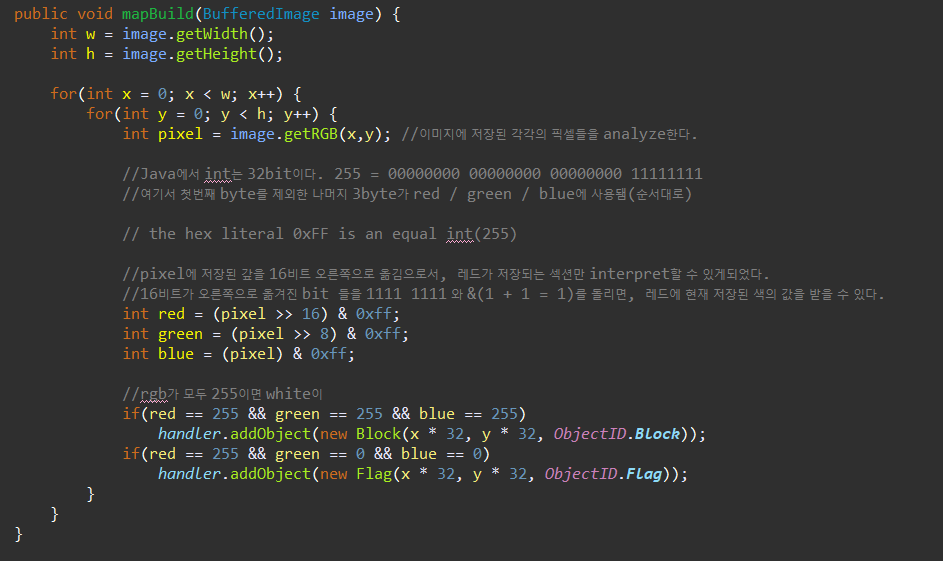
\*In Handler’s tick method, we constistently check if the object is a Player’s instance. However, if the camera is the one that controls the game’s view in user’s perspective, why should we still update or use the tick method of the Player’s instance? Because the instance also needs to move in the game. If we only update the cam then the cam will be stuck there forever because the player instance never moves.

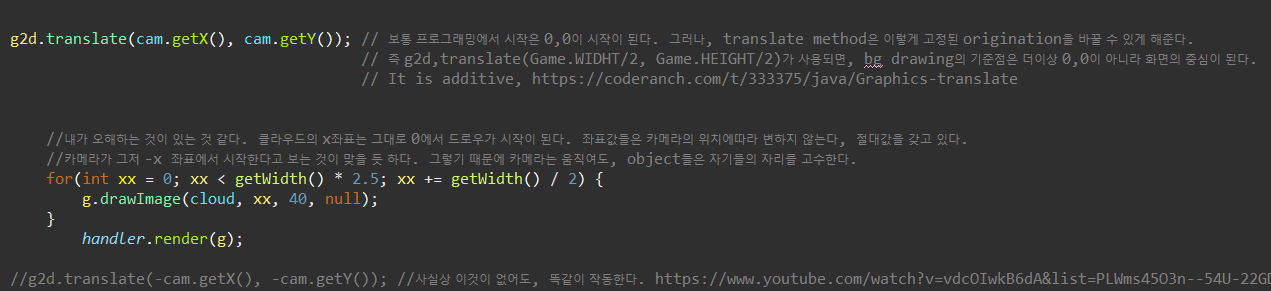
Texture class

텍스쳐 클래스는 게임내의 모든 이미지를 로드하고 준비해놓는다.

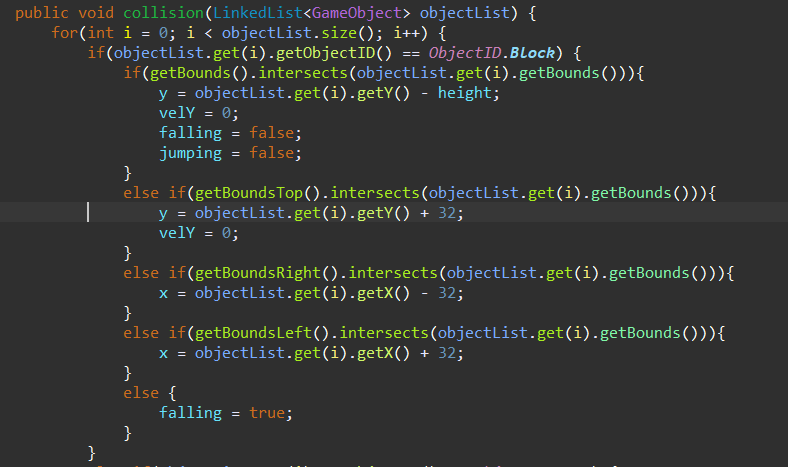
중요한 코드

Game class

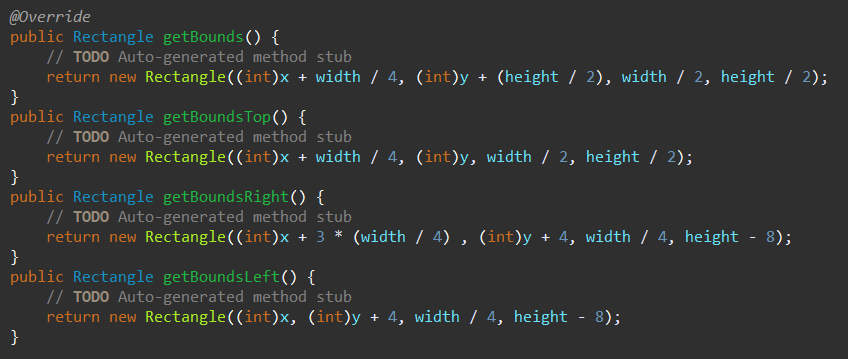




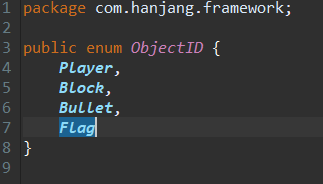
Player class



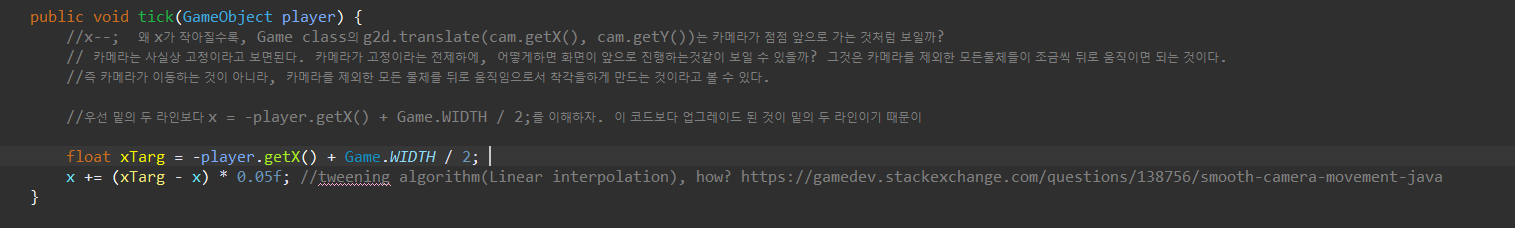




Enum class



Camera class



Animation class

