**Initialized the buildLevel variable to the value of the local cBuild variable value and also called the GameState.setHeight method from the createLevel method that is located in the Level class.**

**package** sonar.gamestates.states.levels;

**import** sonar.Screen;

**import** sonar.gamestates.states.levels.stages.entities.animations.mobs.MobManager;

**public** **class** Level

{

//Level retrieves all the tiles that build our level.

**private** **static** LM *lm*;

**private** **static** LevelBuilder *buildLevel*;

**private** **static** MobManager *mmanager*;

**final** **static** **void** createLevel(**final** LevelBuilder cBuild)

{

*buildLevel* = cBuild;

GameState.setHeight(*buildLevel*.getHeight());

}

**protected** Level(LevelBuilder buildLevel, LM lm)

{

**this**.*buildLevel* = buildLevel;

**this**.*lm* = lm;

lm.*getStage*().*setHeight*(buildLevel.getHeight());

lm.*getStage*().*setWidth*(buildLevel.getWidth());

lm.*getStage*().*setTiles*(buildLevel.getTiles());

}

**void** update()

{

**if**(*mmanager* == **null**) *mmanager* = **new** MobManager(*lm*.*getStage*().*getSmanage*(), "StarterLevel1");

**else**

{

**if**(*mmanager*.starterMob.getLm() == **null**) *mmanager*.starterMob.setLm(*lm*);

*mmanager*.starterMob.update();

}

}

**void** render(**int** xScroll, **int** yScroll, Screen screen)

{

screen.*setOffset*(xScroll, yScroll);

*lm*.*getStage*().drawTiles(xScroll, yScroll, screen);

**if**(*mmanager* != **null**) *mmanager*.starterMob.render(screen);

}

LM getLm(){**return** *lm*;}

LevelBuilder getBuildLevel(){**return** *buildLevel*;}

**public** MobManager getMmanager(){**return** *mmanager*;}

}