**Changed from the GameState.setHeight method call to the lm.getStage.setHeight method call in the createLevel method and also changed from the GameState.setWidth method call to the lm.getStage.setWidth method call in the createLevel method that is located in the Level class.**

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

**import** sonar.GameState;

**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*;

**protected** Level(){} //Prevents multiple levels from being created

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

{

*buildLevel* = cBuild;

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

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

GameState.setTiles(*buildLevel*.getTiles());

}

**void** update()

{

**if**(*mmanager* == **null**) *mmanager* = **new** MobManager(GameState.*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);

}

**final** **static** **void** setLm(**final** LM cLM){*lm* = cLM;}

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

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

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

}