**Removed the this parameter from the StarterStage2 and StarterStage3 method calls in the loadLevel method since they are no longer needed in the LM class.**

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

**import** sonar.GameState;

**import** sonar.Screen;

**import** sonar.gamestates.states.levels.stages.StarterStage1;

**import** sonar.gamestates.states.levels.stages.StarterStage2;

**import** sonar.gamestates.states.levels.stages.StarterStage3;

**public** **class** LM

{

//LevelManager allows us to switch between different levels.

**private** **static** Level *currentLevel*;

**private** **static** GameState *stage*;

**final** **static** LM createLM(GameState cStage)

{

*stage* = cStage;

LM lm = **new** LM();

**if**(*stage*.getBuildState().getIdentity().equals("Starter")){*setLevel*(LevelHolder.***starterStage1***);}

**return** lm;

}

**private** **final** **static** **void** loadLevel(**int** level)

{

*currentLevel* = **null**;

**if**(level == LevelHolder.***starterStage1***) *currentLevel* = **new** StarterStage1(**new** DynamicLevelBuilder("/textures/stages/starter/Starter1.png"));

**if**(level == LevelHolder.***starterStage2***) *currentLevel* = **new** StarterStage2(**new** DynamicLevelBuilder("/textures/stages/starter/Starter2.png"));

**if**(level == LevelHolder.***starterStage3***) *currentLevel* = **new** StarterStage3(**new** DynamicLevelBuilder("/textures/stages/starter/Starter3.png"));

}

**final** **static** **void** setLevel(**int** level){*loadLevel*(level);}

**public** **void** update()

{

*currentLevel*.update();

}

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

{

**int** xLocation = xScroll;///-screen.getWidth();

**int** yLocation = yScroll; //-screen.getHeight();

*currentLevel*.render(xLocation, yLocation, screen);

}

**public** **final** **static** GameState getStage(){**return** *stage*;}

**public** **final** **static** Level getCurrentLevel(){**return** *currentLevel*;}

}