**Added image and pixels array variables to the render method of LevelManager.**

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

**import** java.awt.Graphics;

**import** java.awt.image.BufferedImage;

**import** sonar.gamestates.GSM;

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

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

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

**import** sonar.gamestates.states.levels.stages.entities.SpriteManager;

**import** sonar.gamestates.states.levels.stages.entities.animations.tiles.TileManager;

**public** **class** LM

{

**private** Level currentLevel;

**private** SpriteManager smanage;

**private** TileManager tmanage;

**private** Screen screen;

**public** LM(String stageType, GSM gsm)

{

smanage = **new** SpriteManager();

tmanage = **new** TileManager(smanage);

screen = **new** Screen(gsm.getGameWindow()[0], gsm.getGameWindow()[1]);

**if**(stageType.equals("Starter")){setLevel(LevelHolder.***starterStage1***);}

}

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

{

currentLevel = **null**;

**if**(level == LevelHolder.***starterStage1***) currentLevel = **new** StarterStage1(**new** DynamicLevelBuilder("Starter"), **this**);

**if**(level == LevelHolder.***starterStage2***) currentLevel = **new** StarterStage2(**new** DynamicLevelBuilder("Starter"), **this**);

**if**(level == LevelHolder.***starterStage3***) currentLevel = **new** StarterStage3(**new** DynamicLevelBuilder("Starter"), **this**);

}

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

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

{

currentLevel.update();

}

**public** **void** render(Graphics g)

{

currentLevel.render();

BufferedImage image = **new** BufferedImage(screen.getWidth(), screen.getHeight(), BufferedImage.***TYPE\_INT\_RGB***);

**int**[] pixels = ((DataBufferInt) image.getRaster().getDataBuffer()).getData();

**for**(**int** i = 0; i < pixels.length; i++)

{

pixels[i] = screen.getPixels()[i];

}

}

SpriteManager getSmanage(){**return** smanage;}

TileManager getTmanager(){**return** tmanage;}

}