**Set width and height methods for StaticLevelBuilder to return the appropriate levelWindow element for each.**

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

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

**import** java.io.IOException;

**import** java.util.Random;

**import** javax.imageio.ImageIO;

**public** **interface** LevelBuilder

{

**int** getWidth();

**int** getHeight();

**int**[] getTiles();

}

**class** StaticLevelBuilder **implements** LevelBuilder

{

**private** **int**[] levelWindow, tiles;

**private** Random random;

StaticLevelBuilder(**int** width, **int** height)

{

levelWindow = **new** **int**[2];

levelWindow[0] = width;

levelWindow[1] = height;

tiles = **new** **int**[width \* height];

createLevel(width, height);

}

**private** **void** createLevel(**int** width, **int** height)

{

**for**(**int** y = 0; y < height; y++)

{

**for**(**int** x = 0; x < width; x++){tiles[x + y \* width] = random.nextInt(4);}

}

}

**public** **int** getWidth(){**return** levelWindow[0];}

**public** **int** getHeight(){**return** levelWindow[1];}

**public** **int**[] getTiles(){**return** **null**;}

}

**class** DynamicLevelBuilder **implements** LevelBuilder

{

**private** **int**[] levelWindow, tiles;

DynamicLevelBuilder(String path)

{

levelWindow = **new** **int**[2];

createLevel(path);

}

**private** **void** createLevel(String path)

{

**try**

{

BufferedImage image = ImageIO.*read*(DynamicLevelBuilder.**class**.getResource(path));

levelWindow[0] = image.getWidth();

levelWindow[1] = image.getHeight();

tiles = **new** **int**[levelWindow[0] \* levelWindow[1]];

image.getRGB(0, 0, levelWindow[0], levelWindow[1], tiles, 0, levelWindow[0]);

}

**catch** (IOException e) {e.printStackTrace();}

}

}