**Added two for loops to the ColourSpriteBuilder class in loadColour method**

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

**public** **interface** SpriteBuilder

{

**int** getWidth();

**int** getHeight();

**int**[] getPixels();

}

**class** HUDSpriteBuilder **implements** SpriteBuilder

{

**private** SpriteSize size;

HUDSpriteBuilder(SpriteSize size, **int** colour)

{

**this**.size = size;

loadHUD(size, colour);

}

**private** **void** loadHUD(SpriteSize size, **int** colour)

{

**for**(**int** y = 0; y < size.getHeight(); y++)

{

**for**(**int** x = 0; x < size.getWidth(); x++)

{

size.getPixels()[x + y \* size.getWidth()] = colour;

}

}

}

**public** **int** getWidth(){**return** size.getWidth();}

**public** **int** getHeight(){**return** size.getHeight();}

**public** **int**[] getPixels(){**return** size.getPixels();}

}

**class** ColourSpriteBuilder

{

**private** SpriteSize size;

ColourSpriteBuilder(SpriteSize size, SpriteLocation location)

{

**this**.size = size;

loadColour(size, location);

}

**private** **void** loadColour(SpriteSize size, SpriteLocation location)

{

**for**(**int** y = 0; y < size.getHeight(); y++)

{

**for**(**int** x = 0; x < size.getWidth(); x++)

{

}

}

}

}

**class** SpriteSize

{

**private** **int** width, height;

**private** **int**[] pixels;

SpriteSize(**int** width, **int** height)

{

**this**.width = width;

**this**.height = height;

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

}

**int** getWidth(){**return** width;}

**int** getHeight(){**return** height;}

**int**[] getPixels(){**return** pixels;}

}