**Removed getWidth and getSprite methods from the VoidTileBuilder class.**

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

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

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

**import** sonar.gamestates.states.levels.stages.entities.animations.AnimationType;

**import** sonar.gamestates.states.levels.stages.entities.animations.DynamicAnimation;

**import** sonar.gamestates.states.levels.stages.entities.animations.StaticAnimation;

**class** TileManager

{

**public** Tile voidTile, grass;

TileManager(SpriteManager manage)

{

voidTile = **new** Tile(**new** VoidTileBuilder(**new** StaticAnimation(manage.voidSprite)));

grass = **new** Tile(**new** GrassTileBuilder(**new** StaticAnimation(manage.grass)));

}

}

**abstract** **class** TileBuilder

{

**private** AnimationType buildAnimation;

TileBuilder(AnimationType buildAnimation){**this**.buildAnimation = buildAnimation;}

**void** update()

{

**if**(buildAnimation.animType().equals("Dynamic"))

{

DynamicAnimation anim = (DynamicAnimation) buildAnimation;

anim.update();

buildAnimation = anim;

}

}

Sprite getSprite(){**return** buildAnimation.getSprite();}

**int** getWidth(){**return** buildAnimation.getSprite().getWidth();}

**int** getHeight(){**return** buildAnimation.getSprite().getHeight();}

**boolean** solid(){**return** **false**;}

**boolean** light(){**return** **false**;}

**abstract** String attribute();

}

**class** VoidTileBuilder **extends** TileBuilder

{

**private** AnimationType buildAnimation;

VoidTileBuilder(AnimationType buildAnimation)

{

**super**(buildAnimation);

}

**public** **int** getHeight(){**return** buildAnimation.getSprite().getHeight();}

**public** **boolean** solid(){**return** **false**;}

**public** **boolean** light(){**return** **false**;}

**public** String attribute(){**return** "";}

}

**class** GrassTileBuilder **implements** TileBuilder

{

**private** AnimationType buildAnimation;

GrassTileBuilder(AnimationType buildAnimation){**this**.buildAnimation = buildAnimation;}

**public** Sprite getSprite(){**return** buildAnimation.getSprite();}

**public** **int** getWidth(){**return** buildAnimation.getSprite().getWidth();}

**public** **int** getHeight(){**return** buildAnimation.getSprite().getHeight();}

**public** **boolean** solid(){**return** **true**;}

**public** **boolean** light(){**return** **false**;}

**public** String attribute(){**return** "";}

}