import java.awt.Color;

public class Tile {

public static final int WIDTH = 100;

public static final int HEIGHT = 100;

public static final int SLIDE\_SPEED=30;

public static final int ARC\_WIDTH = 15;

public static final int ARC\_HEIGHT = 15;

private int value;

private BufferedImage tilelmage;

private Color background;

private Color text;

private Font font;

private Point slideTo;

private int x;

private int y;

private boolean beginningAnimation = true;

private double scaleFirst = 0.1;

private boolean canCombine = true;

private boolean combineAnimation = false;

private double scaleCombine = 1.2;

private BufferedImage combinelmage;

private boolean canCombine = true;

public Tile(int value, int x, int y){

this.value = value;

this.x = x;

this.y = y;

slideTo = new Point(x, y);

tilelmage = new BufferedImage(WIDTH, HEIGHT, BufferedImage.TYPE\_INT\_ARGB);

beginningImage = new BufferedImage(WIDTH, HEIGHT, BufferedImage.TYPE\_INT\_ARGB); combinelmage = new BufferedImage(WIDTH\*2, HEIGHT\*2, BufferedImage.TYPE\_INT\_ARGB);

drawImage();

}

﻿

private void drawImage(){

Graphics2D g = (Graphics2D)tilelmage.getGraphics();

if (value == 2){

}

background = new Color (0xe9e9e9);

text = new Color (0x000000);

else if (value == 4){

background = new Color (Oxe6daab);

text = new Color (0x000000);

}

else if (value = 8){

background = new Color (0xf79d3d);

text = new Color (0xffffff);

}

else if (value == 16){

background = new Color(0xf28007);

text = new Color (0xffffff);

}

else if (value == 32){

background = new Color(0xf55e3b);

text = new Color(0xffffff);

}

else if (value == 64){

background = new Color (0xff0000);

text = new Color (0xffffff);

}

else if (value == 128){

background = new Color (0xe9de84);

text = new Color (0xffffff);

}

else if (value == 256){

background = new Color (0xf6e873);

text = new Color(0xffffff);

}

else if (value == 512){

background = new Color(0xf5e455);

text = new Color (0xffffff);

}

else if (value == 1024){

background = new Color (0xf7e12c);

text = new Color(0xffffff);

}

else if (value == 2048){

background = new Color(0xffe400);

text = new Color (0xffffff);

}

else{

background = Color.black;

text = Color.white;

}

g.setColor(new Color(0, 0, 0, 0));

g.fillRect(0, 0, WIDTH, HEIGHT);

g.setColor(background);

g.fillRoundRect(0, 0, WIDTH, HEIGHT, ARC\_WIDTH, ARC\_HEIGHT);

g.setColor(text);

if(value <= 64){

fontGame.main.deriveFont(36f);

}

else{

font = Game.main;

}

g.setFont(font);

int drawX = WIDTH / 2 - DrawUtils.getMessageWidth("" + value, font, g) / 2;

int drawY = HEIGHT / 2 - DrawUtils.getMessageWidth("" + value, font, g) / 2;

g.drawString("" + value, drawX, drawY);

g.dispose();

}

public void update(){

if (beginningAnimation) {

AffineTransform transform = new AffineTransform();

transform.translate (WIDTH / 2 - scaleFirst\* WIDTH / 2, HEIGHT/2 - scaleFirst\* HEIGHT / 2); transform.scale(scaleFirst, scaleFirst);

Graphics2D g2d = (Graphics2D) beginningImage.getGraphics();

g2d.setRenderingHint (RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BICUBIC);

g2d.setColor(new Color(0, 0, 0, 0));

g2d.fillRect(0, 0, WIDTH, HEIGHT);

g2d.drawImage(tileImage, transform, null);

scaleFirst += 0.1;

g2d.dispose();

if(scaleFirst >= 1)

beginningAnimation = false;

}

else if(combineAnimation){

AffineTransform transform = new AffineTransform();

transform.translate (WIDTH / 2 - scaleCombine\* WIDTH / 2, HEIGHT/2 - scaleCombine\* HEIGHT / 2); transform.scale(scaleCombine, scaleCombine);

Graphics2D g2d = (Graphics2D) beginningImage.getGraphics();

g2d.setRenderingHint (RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BICUBIC);

g2d.setColor(new Color(0, 0, 0, 0));

g2d.fillRect(0, 0, WIDTH, HEIGHT);

g2d.drawImage(tileImage, transform, null);

scaleCombine -= 0.1;

g2d.dispose();

if(scaleFirst <= 1)

combineAnimation = false;

}

public void render (Graphics2D g) {

if (beginningAnimation){

g.drawImage(beginningImage, x, y, null);

}

else if(combineAnimation){

g.drawImage(combinelmage, (int)(x+ WIDTH / 2 scaleCombine WIDTH / 2),

(int)(y+HEIGHT/2-scale Combine HEIGHT / 2), null);

}

else{

g.drawImage(tilelmage, x, y, null);

}

public int getValue() {

return value;

}

public int getX() {

return x;

}

public void setX(int x) { this.x = x;

}

public int getY() {

return y;

}

public void setY(int y) {

}

this.y = y;

public boolean isCombineAnimation() {

return combineAnimation;

}

public void setCombineAnimation(boolean combineAnimation) {

this.combineAnimation = combineAnimation;

If(combineAnimation) scaleCombine = 1.3;}