

**University of Dhaka**

**Department of Computer Science and Engineering**

**CSE – 2112**

**Project: Don’t Touch the White Tile**

Submitted to:

Dr. Md. Mustafizur Rahman Shiplu Hawladar

Associate Proffesor Lecturer

Submitted by:

Mehreen Rahman (Roll-61)

Zadidul Hasan (Roll-11)

Date of Submission: 25. 05. 16

Contents Page

**1.** Code 03

**2.** UML 10

**Code:**

**//Tile.java**

**package** dontTouchTheWhiteTile;

**public** **class** Tile

{

**public** **int** x, y;

**public** **int** animateY; //when it's hit, we will know where the next one's going to be

//but won't affect y, just render it's position

**public** **boolean** black;

**public** Tile(**int** x, **int** y, **boolean** black)

{

**this**.x = x;

**this**.y =y;

**this**.black = black;

}

**public** **boolean** pointinTile(**int** x, **int** y)

{

**int** width = DontTouchTheWhiteTile.***tile\_width***;

**int** height = DontTouchTheWhiteTile.***tile\_height***;

**return** x > **this**.x \* width && x < **this**.x \* width + width && y > **this**.y \* height && y < **this**.y \* height + height;

//bottomleft corner of tile is (this.x \* width , this.y \* width)

}

}

**//Renderer.java**

**package** dontTouchTheWhiteTile;

**import** java.awt.Graphics;

**import** javax.swing.JPanel;

@SuppressWarnings("serial")

**public** **class** Renderer **extends** JPanel

{

@Override

**protected** **void** paintComponent(Graphics g) {

**super**.paintComponent(g);

**if**(DontTouchTheWhiteTile.*dttwt* != **null**)

{

DontTouchTheWhiteTile.*dttwt*.render(g);

}

}

}

}

**//DontTouchTheWhiteTile.java**

**package** dontTouchTheWhiteTile;

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** java.util.ArrayList;

**import** java.util.Random;

**import** javax.swing.JFrame;

**import** javax.swing.Timer;

**public** **class** DontTouchTheWhiteTile **implements** ActionListener, MouseListener //access to ActionListener

{

**public** **static** **final** **int** ***tile\_width*** = 200, ***tile\_height***=250; //want to access from Tile.java

**public** **static** DontTouchTheWhiteTile *dttwt*;

**public** ArrayList<Tile> tiles;

**public** Renderer renderer;

**public** Random random;

**public** **int** score, milSecDelay;

**public** **boolean** delay;

**public** **boolean** gameOver;

**public** DontTouchTheWhiteTile()

{

JFrame frame = **new** JFrame("Don't Touch The White Tile!");

Timer timer = **new** Timer(20, **this**);

//tiles = new ArrayList<Tile>();

renderer = **new** Renderer();

random = **new** Random();

frame.setSize(***tile\_width***\*3, ***tile\_height***\*3);

frame.add(renderer);

frame.setVisible(**true**);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.addMouseListener(**this**);

frame.setResizable(**false**);

start();

timer.start();

}

**public** **void** start()

{

score = 0; //

gameOver = **false**; // new game after game over

tiles = **new** ArrayList<Tile>(); //

**for**(**int** x =0; x<3; x++)

{

**for**( **int** y=0; y<4; y++)

{

**boolean** canBeBlack = **true**;

**for**( Tile tile: tiles)

{

**if**( tile.y == y && tile.black)

{

canBeBlack = **false**;

}

}

**if**(!canBeBlack)

{

tiles.add(**new** Tile(x, y, **false**));

}

**else**

{

tiles.add(**new** Tile(x, y, random.nextInt(2)==0 || x == 2)); //if it's in a random out of 3, and equals to 0, it can be black

// if it's less 1 it can be black

// there will always be a black one no matter what, that's why x==2

}

}

}

}

@Override

**public** **void** actionPerformed(ActionEvent e) {

renderer.repaint();

**for**(**int** i=0; i< tiles.size(); i++)

{

Tile tile = tiles.get(i);

**if**(tile.animateY < 0)

{

tile.animateY += ***tile\_height***/5 ;

}

}

milSecDelay++;

}

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

{

g.setColor(Color.***WHITE***);

g.fillRect(0, 0, ***tile\_width***\*3, ***tile\_height***\*3);

g.setFont(**new** Font("Arial", 1, 75));

**if**(!gameOver)

{

**for**(Tile tile: tiles)

{

g.setColor(tile.black? Color.***BLACK*** : Color.***WHITE***);

g.fillRect(tile.x \* ***tile\_width***, tile.y \* ***tile\_height*** - tile.animateY, ***tile\_width***, ***tile\_height***);

g.setColor(tile.black? Color.***WHITE*** : Color.***BLACK*** );

g.drawRect(tile.x \* ***tile\_width***, tile.y \* ***tile\_height*** - tile.animateY, ***tile\_width***, ***tile\_height***);

}

g.setColor(Color.***RED***);

g.drawString(String.*valueOf*(score), 10, 70);

}

**else**

{

g.setColor(Color.***BLACK***);

g.drawString("Game Over!", 50, ***tile\_height***/2+20);

g.drawString("Score: "+String.*valueOf*(score), 50, ***tile\_height***+100);

g.drawString("Click again", 50, ***tile\_height***+280);

g.drawString("to restart.", 50, ***tile\_height***+370);

}

}

**public** **static** **void** main(String[] args)

{

*dttwt* = **new** DontTouchTheWhiteTile();

}

@Override

**public** **void** mouseClicked(MouseEvent e)

{

}

@Override

**public** **void** mouseEntered(MouseEvent e) {

}

@Override

**public** **void** mouseExited(MouseEvent e) {

}

@Override

**public** **void** mousePressed(MouseEvent e) //when the mouse is clicked, it's going to check whether the tile is black/white

//add to the score accordingly

{

**boolean** clicked = **false**;

**if**(!gameOver)

{

**for**(**int** i=0; i<tiles.size(); i++)

{

Tile tile = tiles.get(i);

**if**(tile.pointinTile(e.getX(), e.getY()) && !clicked)

{

**if**(e.getY() > ***tile\_height***\*2 )

{

**if**(tile.black)

{

**for**( **int** j=0; j<tiles.size(); j++)

{

**if**(tiles.get(j).y == 2)

{

tiles.remove(j);

}

tiles.get(j).y++; //going to move them all up

tiles.get(j).animateY -= ***tile\_height*** ;

}

score += Math.*max*(100 - milSecDelay, 10);

//System.out.println(Math.max(100 - milSecDelay, 10));

System.***out***.println("You've scored " + Math.*max*(100 - milSecDelay, 10)+ " points!" );

milSecDelay=0;

**boolean** canBeBlack = **true**;

**for**(**int** x =0; x<3; x++)

{

**boolean** black = random.nextInt(3) == 0 || x==2;

Tile newTile = **null**;

**if**(canBeBlack && black)

{

newTile= **new** Tile(x, 0, **true**);

canBeBlack = **false**;

}

**else**

{

newTile= **new** Tile(x, 0, **false**);

}

newTile.animateY -= ***tile\_height***;

tiles.add(newTile);

}

}

**else**

{

gameOver = **true**;

}

clicked = **true**;

}

**else**

{

gameOver = **true**;

}

}

}

}

**else**

{

start();

}

}

@Override

**public** **void** mouseReleased(MouseEvent e)

{

}

}

UML:

