package game;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.Toolkit;

import java.util.ArrayList;

import java.io.\*;

//Nikhil Shende, Jamie Caille, Daniel Papadopoulos

//June 5, 2017

//The map containing all extra components not part of the basic map

//it contains the fortress pieces, rocks, and grass

public class ExtraMap {

//creating an arrayList of all the extra components of the map

ArrayList<ExtraStuff> extraMap = new ArrayList<ExtraStuff>();

//defining all the images that will be drawn

private Image rock1;

private Image rock2;

private Image rock3;

private Image grass1;

private Image grass2;

private Image grass3;

private Image cFortress;

private Image vFortress;

private Image hFortress;

private Image ULFortress;

private Image DLFortress;

private Image URFortress;

private Image DRFortress;

private int tileSize;

public ExtraMap() {

tileSize = DS.tileSize;

//loading all the images into their variable

rock1 = loadImage("/GameFiles/Tiles/tile\_49.png");

rock2 = loadImage("/GameFiles/Tiles/tile\_50.png");

rock3 = loadImage("/GameFiles/Tiles/tile\_52.png");

grass1 = loadImage("/GameFiles/Tiles/tile\_70.png") ;

grass2 = loadImage("/GameFiles/Tiles/tile\_71.png");

grass3 = loadImage("/GameFiles/Tiles/tile\_72.png");

cFortress = loadImage("/GameFiles/Tiles/tile\_13.png");

vFortress = loadImage("/GameFiles/Tiles/tile\_15.png");

hFortress = loadImage("/GameFiles/Tiles/tile\_16.png");

ULFortress = loadImage("/GameFiles/Tiles/tile\_77.png");

DLFortress = loadImage("/GameFiles/Tiles/tile\_93.png");

URFortress = loadImage("/GameFiles/Tiles/tile\_78.png");

DRFortress = loadImage("/GameFiles/Tiles/tile\_94.png");

readFile("/GameFiles/ExtraMap.txt");

}

//creating a method to load an image

public Image loadImage(String src) {

Image im = null;

try {

//loading the image

im = Toolkit.getDefaultToolkit().getImage(getClass().getResource(src));

} catch (Exception e) {

System.out.println("Error: " + e);

}

return im;

}

//method used to read a data file for all extra map components

public void readFile(String filePath){

int numLines = 0;

//counting the number of lines

try{

//creating an input stream

InputStream in = getClass().getResourceAsStream(filePath);

BufferedReader br = new BufferedReader(new InputStreamReader(in));

while (br.readLine() != null){

numLines++;

}

br.close();

}

catch(IOException e){

System.out.println("Error: " + e);

}

try{

//reading the file and storing the data in the arrayList

InputStream in = getClass().getResourceAsStream(filePath);

BufferedReader br = new BufferedReader(new InputStreamReader(in));

for(int i = 0; i < numLines/5; i++){

//reading the map location

int mX = Integer.parseInt(br.readLine());

int mY = Integer.parseInt(br.readLine());

int sX = Integer.parseInt(br.readLine());

int sY = Integer.parseInt(br.readLine());

//reading the reference number

int r = Integer.parseInt(br.readLine());

//adding a new extra map component

extraMap.add(new ExtraStuff(mX,mY,sX,sY,r));

}

}

catch(IOException e){

System.out.println("Error: " + e);

}

}

public void drawExtraStuff(Graphics2D g, int mX,int mY){

//drawing all extra objects

for(int i = 0; i < extraMap.size(); i++){

//drawing an object if map section is correct

if(extraMap.get(i).getMapSecX() == mX && extraMap.get(i).getMapSecY() == mY){

//drawing something if the reference number matches

if (extraMap.get(i).getRef() == 0) {

g.drawImage(rock1, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 1) {

g.drawImage(rock2, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 2) {

g.drawImage(rock3, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 3) {

g.drawImage(grass1, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 4) {

g.drawImage(grass2, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 5) {

g.drawImage(grass3, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 6) {

g.drawImage(cFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 7) {

g.drawImage(vFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 8) {

g.drawImage(hFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 9) {

g.drawImage(ULFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 10) {

g.drawImage(DLFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 11) {

g.drawImage(URFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

} else if (extraMap.get(i).getRef() == 12) {

g.drawImage(DRFortress, extraMap.get(i).getSquareX() \* tileSize, extraMap.get(i).getSquareY() \* tileSize + DS.gap, tileSize, tileSize, null);

}

}

}

}

}