/\*

Assignment 7

Name: Sarah Depew

E-mail: sdepew@brynmawr.edu

Course: CS 110 - Section 01

Submitted: April 29, 2016

This code visualizes a data set that I found on the internet of patient reaction times with differing sleep deprivation. The information and reaction times are taken over the course of ten days. In total, the file has 181 lines of code: 10 reaction times for each of 10 respective days of sleep deprivation for each of the 18 patients. I custom designed a class that had a constructor that stores two arrays of the patient's information patient's specific number. Then, I wrote methods to fill the arrays and graph the values.

This code has tiered interactivity. If the user moves the mouse on the screen a function will use the mouseX and mouseY values and call a function that displays instructions on the screen on how to use this visualization and clears any previous interactions. The instructions outline each of the interactivity options for this graph. If the user presses uppercase letters from A-R, the individual patient's graph is displayed. If lowercase letters a-r are pressed, the graph for that patient is colored blue out of all the graphs. Finally, if the user presses letters s or S, the visualization is reset.

\*/

String[] data; //array of type string of the data

Subject[] patients; //array of patients of type subject

void setup() {

size(800, 500); //size does adjust, but a size of 800, 500 looks best

background(255); //white background

data = loadStrings("sleepStudy.csv"); //loads the data file into strings

patients = new Subject[18]; //new array of the Subject class for each of the 18 patients in the study

//fill object array with all data

for (int i=1; i<data.length; i+=10) {

String[] patientNums = data[i].split(","); //loads each of the patient numbers into their own array

patients[i/10] = new Subject(int(patientNums[3]));

for (int j=0; j<10; j++) {

String[] dayAndTime = data[i+j].split(","); //splits the correct line in the data array

patients[i/10].addDay(int(dayAndTime[2]), j); //adds the day to the array

patients[i/10].addRxnTime(float(dayAndTime[1]), j); //adds the time to the array

} //done filling the patients array with each patient's data and information

}

initalGraphs(); //the inital background displayed

textAlign(CENTER);

text("All Patients' Average Reaction Times (ms) vs. Days of Lost Sleep", width/2, height/25); //title at the head of the screen

graphBackground(); //the graph's x and y axis with notation

}

void draw() { //makes keyPressed() and mouseMoved work

}

void keyPressed() {

textAlign(LEFT);

if (key>='A' && key<='R') { //shows the individual graph when letters A-R are pressed

background(255);

patients[key-'A'].graph(color(0), true);

stroke(0);

if (width<height||width==height) {

textSize(width/30);

} else {

textSize(height/30);

}

textAlign(CENTER);

text("Patient " + str(key-'A'+1) + " Average Reaction Time (ms) vs. Days of Lost Sleep", width/2, height/25);

graphBackground();

}

if (key=='s'||key=='S') { //resets the graphs if S or s is pressed

initalGraphs();

textAlign(CENTER);

if (width<height||width==height) {

textSize(width/30);

} else {

textSize(height/30);

}

text("All Patients' Average Reaction Times (ms) vs. Days of Lost Sleep", width/2, height/25);

graphBackground();

}

if (key>='a' && key<='r') { //picks the graphs out of the bunch when lowercase letters a-r are pressed

initalGraphs();

patients[key-'a'].graphColor();

stroke(0);

if (width<height) {

textSize(width/30);

} else {

textSize(height/30);

}

textAlign(CENTER);

fill(0);

text("Patient " + str(key-'a'+1) + " Average Reaction Time (ms) vs. Days of Lost Sleep", width/2, height/25);

graphBackground();

}

if (key==’x’) {//saves the visualization

save("Visualization 1.jpg");

}

}

void mouseMoved() { //shows and instruction box when the mouse is moved and resets the background

instructions(mouseX, mouseY);

}

void graphBackground() { //the drawn coordinate system on the screen

float shift = width/10;

float inc = width/11;

int textL; //the location of my signature

if (width<height) {

textSize(width/50);

textL = width/50;

} else {

textSize(height/50);

textL = height/50;

}

for (int i=0; i<10; i++) { //for loop to draw the graph lines and number them

println(height-((45\*height/50)+(5\*height/50))/10\*i);

line(0, 45\*height/50, width, 45\*height/50); //x-axis

line(i\*inc+shift, 46\*height/50, i\*inc+shift, 44\*height/50); //hatch marks on x-axis

fill(0);

textAlign(CENTER);

text(str(i), i\*inc+shift, 47.5\*height/50); //number lables at x-axis

line(3\*width/50, 3\*height/50, 3\*width/50, height); //y-axis

line(2.5\*width/50, (height-((45\*height/50)+(5\*height/50))/10\*i), 3.5\*width/50, (height-((45\*height/50)+(5\*height/50))/10\*i));

if (i==1||i==5||i==9) {

pushMatrix(); //the title at the y-axis

translate(3\*height/50, (height-((45\*height/50)+(5\*height/50))/10\*i));

//rotate(3\*PI/2);

text(str(163.25+33.25\*i), 0, 0); //number lables at y-axis

popMatrix();

}

}

pushMatrix(); //the title at the y-axis

translate(1\*width/50, height/2);

rotate(3\*PI/2);

text("Reaction Time (ms)", 0, 0);

popMatrix();

text("Sleep Deprivation (Days)", width/2, 49\*height/50); //the title at the bottom of the x-axis

text("Sarah Depew CS110-01 Spring 2016", textL\*10, textL);

}

void initalGraphs() { //inital background graphs 1-18

background(255);

for (int j=0; j<patients.length; j++) {

patients[j].graph(color(0), false);

}

if (width<height) {

textSize(width/30);

} else {

textSize(height/30);

}

}

void instructions(int x, int y) {//function that causes the instruction box to display on the screen

initalGraphs();

graphBackground();

if (width<height) {

textSize(width/30);

} else {

textSize(height/30);

}

textAlign(CENTER);

text("All Patients' Average Reaction Times (ms) vs. Days of Lost Sleep", width/2, height/25);

fill(255);

rectMode(CENTER);

if (width<height || width==height) { //draws the text at different places in the instructions box

instructionBox(x, y, width);

} else {

instructionBox(x, y, height);

}

}

void instructionBox(int x, int y, float value) { //the function that draws the instruction box

rect(x, y, value/2, value/2.5);

fill(0);

textSize(value/50);

pushMatrix();

translate(x, y-14\*value/100);

text("Instructions for Visualization:", 0, 0);

popMatrix();

pushMatrix();

translate(x, y-9\*value/100);

text("Press letters A-R for individal, labeled graphs.", 0, 0);

popMatrix();

pushMatrix();

translate(x, y-6\*value/100);

text("Press letters a-r for group, colored graphs.", 0, 0);

popMatrix();

pushMatrix();

translate(x, y-3\*value/100);

text("Press letters s or S to reset the graphs.", 0, 0);

popMatrix();

pushMatrix();

translate(x, y);

text("Moving this box on the screen resets the graphs.", 0, 0);

popMatrix();

}

class Subject { //fields for class

int number; //patient number

int[] days; //days of sleep deprivation per patient

float[] rxnTimes; //reaction times corresponding to each day

Subject(int number) { //constructor

this.number = number;

days = new int[10];

rxnTimes = new float[10];

}

void addDay(int day, int i) { //loads the days into the days array

days[i]=day;

}

void addRxnTime(float rxnTime, int i) { //loads the reaction times into the rxnTimes array

rxnTimes[i]=rxnTime;

}

void graph(color c, boolean label) { //the graph method with color and label parameters

float inc = width/11; //stretching out the data along the x-axis

float shift = width/10; //shifts the points to the right

float textShiftX = width/30; //shifts the text along the x-axis

float textShiftY = height/40; //shifts the text along the y-axis

float maxY = 466.3535; //maximum y-value found for data (used excel max function)

float minY = 194.3322; //minimum y-value found for data (used excel min function)

float ellipseSize; //the size of the graph's ellipse

if (width<height) {

ellipseSize = width/50;

} else {

ellipseSize = height/50;

}

fill(c);

for (int i=0; i<days.length; i++) { //loop that graph

float yPos = map(rxnTimes[i], maxY, minY, 5\*height/50, 45\*height/50);

ellipse(days[i]\*inc+shift, yPos, ellipseSize, ellipseSize); //height-rxnTimes[i] corrects for y-axis

if (i+1<days.length) {

float yPos2 = map(rxnTimes[i+1], maxY, minY, 5\*height/50, 45\*height/50);

line(days[i]\*inc+shift, yPos, days[i+1]\*inc+shift, yPos2);

}

if (label) {

text("("+str(days[i])+", "+str(rxnTimes[i])+")", days[i]\*inc+textShiftX, yPos-textShiftY); //writes the points in coordinate form

}

}

}

void graphColor() { //uses the graph method to create a labeled and colored graph

stroke(0, 0, 255);

graph(color(0, 0, 255), false);

}

String toString() { //redefine how the class prints the string

String msg = "The patient " + str(number);

for (int i=0; i<days.length; i++) {

msg +=" had a reaction time of " + str(rxnTimes[i]) + " of ms on day ";

msg+= str(days[i]);

msg+="\n";

}

return msg;

}

}